

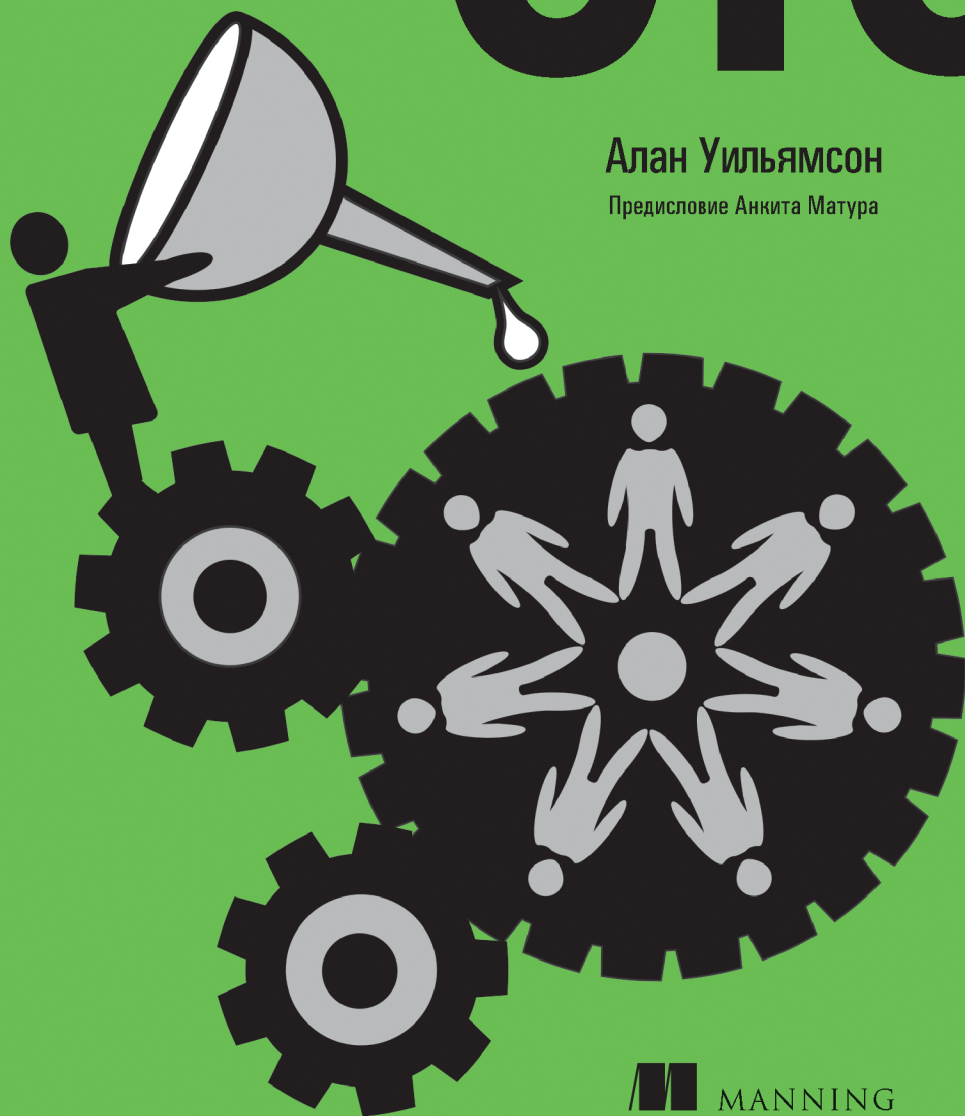
# Настоящий

Думай  
как технический  
директор

# СТО

Алан Уильямсон

Предисловие Анкита Матура



MANNING



# *Think Like a CTO*

ALAN WILLIAMSON



MANNING  
SHELTER ISLAND

# Настоящий СТО

Думай  
как технический  
директор

Алан Уильямсон



Санкт-Петербург • Москва • Минск

2024

ББК 65.290-21+ 32.973.223  
УДК 005.93:004  
У36

### **Уильямсон Алан**

У36 Настоящий СТО: думай как технический директор. — СПб.: Питер, 2024. — 352 с.: ил. — (Серия «Библиотека программиста»).

ISBN 978-5-4461-2114-4

Технический директор (СТО) находит баланс между потребностями бизнеса и постоянно развивающимся миром технологий. Развейте навыки и мышление, необходимые, чтобы взять на себя эту важную роль и стать успешным лидером. Автор приводит мнения отраслевых экспертов и опытных СТО и делится практическими стратегиями навигации в мире технологического лидерства, где ставки чрезвычайно высоки.

Книга на примерах из реальной практики показывает, как преуспеть в быстро меняющейся роли технического директора. Прочитав ее, вы научитесь создавать успешные технологические платформы и формировать эффективные команды, грамотно выбирать и внедрять программные продукты, проводить собеседования и перформанс ревью в беспристрастной манере, а также сможете по праву занять место за столом топ-менеджмента. Оцените деловые советы, идеи и истории из практики от наставника СТО Алана Уильямсона.

**16+** (В соответствии с Федеральным законом от 29 декабря 2010 г. № 436-ФЗ.)

ББК 65.290-21+ 32.973.223  
УДК 005.93:004

Права на издание получены по соглашению с Manning Publications. Все права защищены. Никакая часть данной книги не может быть воспроизведена в какой бы то ни было форме без письменного разрешения владельцев авторских прав.

Информация, содержащаяся в данной книге, получена из источников, рассматриваемых издательством как надежные. Тем не менее, имея в виду возможные человеческие или технические ошибки, издательство не может гарантировать абсолютную точность и полноту приводимых сведений и не несет ответственности за возможные ошибки, связанные с использованием книги.

В книге возможны упоминания организаций, деятельность которых запрещена на территории Российской Федерации, таких как Meta Platforms Inc., Facebook, Instagram и др.

Издательство не несет ответственности за доступность материалов, ссылки на которые вы можете найти в этой книге. На момент подготовки книги к изданию все ссылки на интернет-ресурсы были действующими.

ISBN 978-1617298851 англ.

Authorized translation of the English edition © 2023 Manning Publications.

This translation is published and sold by permission of Manning Publications, the owner of all rights to publish and sell the same.

ISBN 978-5-4461-2114-4

© Перевод на русский язык ООО «Прогресс книга», 2023

© Издание на русском языке, оформление ООО «Прогресс книга», 2023

© Серия «Библиотека программиста», 2023

# Оглавление

---

<b>Вступительное слово .....</b>	<b>15</b>
<b>Предисловие .....</b>	<b>17</b>
<b>Благодарности .....</b>	<b>19</b>
<b>О книге .....</b>	<b>21</b>
Для кого эта книга .....	22
Структура книги.....	23
Форум liveBook.....	24
<b>Об авторе .....</b>	<b>25</b>
<b>О редакторе русского издания.....</b>	<b>26</b>
От издательства .....	26
<b>Глава 1. Технический директор (СТО).....</b>	<b>27</b>
1.1. Суть работы технического директора .....	28
1.2. Различные типы СТО.....	30
1.2.1. Предстартап, номинальная должность.....	31
1.2.2. Финансируемый стартап: технологический эксперт с деньгами.....	32
1.2.3. Устоявшаяся компания: первый технический директор.....	33
1.2.4. Устоявшаяся компания с техническим директором.....	34
1.3. Определение потребности в техническом директоре.....	35
1.4. Эволюция от инженера .....	36
1.4.1. Первые 100 дней.....	38
1.5. 10 главных качеств технического директора.....	41
Итоги .....	41
Проверьте себя .....	42
<b>Глава 2. Взаимодействие с руководством и коллегами .....</b>	<b>43</b>
2.1. Партнерство с генеральным директором.....	44
2.1.1. Офис генерального директора.....	44
2.1.2. Типы генеральных директоров.....	46
2.1.3. Найти подход.....	48
2.1.4. Как наладить успешное партнерство .....	50

## 6 Оглавление

2.2. Ставка на финансового директора .....	52
2.2.1. Как контролировать свои расходы .....	53
2.2.2. Обратитесь за помощью .....	56
2.3. Работа с коллегами .....	56
2.3.1. Знайте свою компанию .....	57
2.3.2. Говорите на их языке .....	58
2.3.3. Найдите болевые точки .....	58
2.3.4. Мыслите стратегически .....	59
2.4. Принимаем дела у предыдущего технического директора .....	60
2.4.1. Цените, а не принижайте .....	60
2.4.2. Поговорите с уходящим СТО .....	61
2.4.3. Изучите команду .....	62
2.5. Представление членам правления .....	63
2.5.1. Узнайте свою аудиторию .....	64
2.5.2. Формат встречи .....	64
2.5.3. Ваша роль .....	65
2.5.4. Что можно, а чего нельзя .....	66
2.6. Коммуникация .....	67
2.6.1. Внутренняя коммуникация .....	68
2.6.2. Внешняя коммуникация .....	69
2.7. Внутренняя политика .....	70
2.7.1. Основные типы .....	70
2.7.2. Смягчение политики .....	72
2.8. Управление изменениями .....	73
2.8.1. Лягушка в кипятке .....	73
2.8.2. Комитет воодушевления .....	74
2.8.3. Долина страданий .....	75
2.8.4. Последствия .....	76
Итоги .....	76
Проверьте себя .....	77
<b>Глава 3. Долгосрочное видение .....</b>	<b>78</b>
3.1. Общее видение .....	79
3.1.1. Часть «что»/«зачем» .....	80
3.1.2. Часть «как» .....	81
3.1.3. Определите критерии успеха .....	82
3.2. Сотрудничество с клиентами .....	83
3.2.1. Знай своего клиента .....	84
3.2.2. Ваш партнер .....	85
3.3. Долгосрочное планирование .....	87
3.3.1. Время решает все .....	87
3.3.2. Обдумайте детали .....	90

3.4. Работа с постоянными изменениями .....	90
3.4.1. Паралич решений.....	91
3.4.2. Определите, на чем основывается ваше видение.....	92
3.4.3. Контролируйте состояние опор.....	93
3.4.4. В поисках простоты.....	94
3.5. Презентация для лифта.....	95
3.5.1. Подготовка речи.....	96
3.5.2. Практика и оттачивание навыка презентации.....	97
3.6. Составление бюджета .....	98
3.6.1. Что включать в бюджет.....	98
3.6.2. Формат бюджета .....	99
3.6.3. Контроль выполнения бюджета .....	100
3.6.4. Окупаемость (ROI).....	101
3.7. Краткосрочное планирование и ожидания .....	101
3.7.1. Коммуникация.....	102
3.7.2. Контроль инфополя .....	103
Итоги .....	104
Проверьте себя .....	104
<b>Глава 4. Создание команды.....</b>	<b>105</b>
4.1. Типы ресурсов .....	106
4.1.1. Штатные сотрудники .....	107
4.1.2. Сотрудники, работающие по договору.....	107
4.1.3. Привлечение подрядчиков .....	108
4.1.4. Местный аутсорсинг.....	109
4.1.5. Зарубежный аутсорсинг.....	110
4.1.6. Сравнительный обзор.....	111
4.2. Когда набирать сотрудников.....	112
4.2.1. Причины найма .....	113
4.2.2. Матрица навыков.....	117
4.2.3. Последствия отказа от найма.....	119
4.3. Поиск кандидатов .....	119
4.3.1. Реферальный поиск .....	120
4.3.2. Агентства по подбору персонала .....	120
4.3.3. Хедхантинг .....	121
4.3.4. Ярмарки вакансий для выпускников / стажировки.....	122
4.3.5. Самостоятельный поиск в интернете .....	124
4.4. Составление описания вакансии .....	125
4.4.1. Структура .....	125
4.4.2. Название должности.....	126
4.4.3. Описание работы.....	127
4.4.4. Минимальные требования к кандидату.....	129

4.4.5. Требуемые навыки .....	129
4.4.6. Информация о компании .....	130
Итоги .....	130
Проверьте себя .....	131
<b>Глава 5. Собеседования, выбор подходящего кандидата и адаптация новых сотрудников .....</b>	<b>132</b>
5.1. Собеседование .....	133
5.1.1. Правильный настрой .....	133
5.1.2. Цели собеседования .....	134
5.1.3. Телефонное (или предварительное) собеседование .....	135
5.1.4. Личное собеседование .....	136
5.1.5. Видеоинтервью .....	141
5.2. Оценка кандидатов .....	143
5.2.1. Выбор критериев оценки .....	144
5.2.2. Выставление оценок .....	145
5.3. Как сказать «нет» .....	146
5.3.1. Определенно «нет» .....	146
5.3.2. Определенно «может быть» .....	147
5.4. Адаптация .....	148
5.4.1. Вводим в курс дела .....	148
5.4.2. Ваши ожидания .....	149
5.4.3. Приступаем к работе .....	149
Итоги .....	150
Проверьте себя .....	150
<b>Глава 6. Управление командой .....</b>	<b>151</b>
6.1. Устав .....	153
6.1.1. Знания .....	154
6.1.2. Образец устава .....	154
6.2. Структура команды .....	155
6.2.1. Вокруг продукта .....	156
6.2.2. Вокруг жизненного цикла .....	157
6.2.3. Вокруг клиента/вертикали .....	157
6.2.4. Гибридный подход .....	158
6.3. Названия должностей .....	158
6.3.1. Что включать в название? .....	159
6.3.2. Ступени к совершенству .....	160
6.3.3. Частые ошибки .....	161
6.4. Взаимодействие с командой .....	162
6.4.1. Определение степени самостоятельности .....	162
6.4.2. Метрики .....	164



6.4.3. Ответственность за результат.....	165
6.5. Индивидуальные встречи.....	166
6.6. Обучение и тренинги .....	168
6.6.1. Онлайн-библиотеки/ресурсы.....	168
6.6.2. Сертификация специалистов по платформе/ПО.....	168
6.6.3. Буткемп .....	169
6.6.4. Сообщества пользователей.....	169
6.6.5. Внутреннее обучение.....	170
6.6.6. Конференции.....	170
6.6.7. Курсы при университетах/колледжах.....	170
6.7. Коммуникация .....	171
6.7.1. Электронная почта .....	171
6.7.2. Совещания.....	173
6.7.3. Обратная связь.....	175
Итоги .....	176
Проверьте себя .....	176
<b>Глава 7. Ежегодная оценка сотрудников.....</b>	<b>177</b>
7.1. Матрица компетенций.....	178
7.1.1. Структура матрицы.....	179
7.1.2. Заполнение матрицы .....	180
7.1.3. Порядок проведения ревью.....	182
7.2. Ваш подчиненный ошибается.....	183
7.2.1. Первая реакция.....	184
7.2.2. Расследование .....	184
7.2.3. Выводы .....	186
7.3. Увольнение .....	186
7.3.1. Сокращение отдела.....	187
7.3.2. Навыки не требуются .....	188
7.3.3. Плохие результаты .....	189
7.3.4. Процедура увольнения.....	191
7.3.5. Обратная связь от увольняемого сотрудника.....	192
7.4. Удаленная работа.....	192
7.4.1. Техническое обеспечение .....	193
7.4.2. Управление ожиданиями.....	194
7.4.3. Предотвращение злоупотреблений .....	194
Итоги .....	195
Проверьте себя .....	196
<b>Глава 8. Технологические решения.....</b>	<b>197</b>
8.1. Избегайте попасть в зависимость .....	199
8.2. Разрабатывать или покупать.....	200

8.2.1. Покупка .....	201
8.2.2. Разработка .....	203
8.2.3. За и против .....	204
8.2.4. Чек-лист .....	204
8.3. Облако или свое оборудование .....	205
8.3.1. Облако.....	206
8.3.2. Собственное оборудование .....	210
8.4. Аварийное восстановление .....	214
8.4.1. Допустимое время простоя.....	215
8.4.2. Полное или частичное обслуживание .....	216
8.4.3. Переход вперед или возврат назад.....	218
8.4.4. Организация реагирования на аварии.....	218
8.4.5. Облако и аварийное восстановление .....	220
8.5. Работа с данными .....	220
8.5.1. Хранение .....	221
8.5.2. Типы данных.....	223
8.6. Микросервисы или монолит .....	225
8.7. ПО с открытым исходным кодом .....	228
8.7.1. Виды лицензий .....	228
8.7.2. Рекомендации по использованию .....	229
8.7.3. Публикация ПО с открытым исходным кодом.....	230
8.7.4. Использование кода из интернета .....	231
8.8. Языки и фреймворки .....	232
8.8.1. Устаревший язык.....	233
8.8.2. Выбор языка для перехода.....	233
Итоги .....	234
Проверьте себя .....	235
<b>Глава 9. Разработка .....</b>	<b>236</b>
9.1. Планирование проекта .....	237
9.1.1. Руководитель проектов.....	238
9.1.2. Что считать проектом .....	241
9.2. Стандарты разработки .....	244
9.3. Системы контроля версий.....	245
9.4. Обеспечение качества (QA).....	247
9.4.1. Ручное тестирование .....	248
9.4.2. Автоматизированное тестирование .....	248
9.5. Непрерывная интеграция и внедрение (CI/CD).....	250
9.6. Технический долг .....	252
9.7. Релиз .....	253
9.7.1. Релиз с прерыванием сервиса.....	254
9.7.2. Сине-зеленое (blue-green) внедрение .....	255

9.8. Запросы от клиентов .....	257
Итоги .....	258
Проверьте себя .....	259
<b>Глава 10. Работа с договорами .....</b>	<b>260</b>
10.1. Договор оказания услуг .....	261
10.1.1. Техническое задание .....	262
10.1.2. Параметры оказания услуг .....	264
10.2. Оценка поставщиков .....	265
10.3. Лицензии на использование ПО .....	267
10.4. Договор на поддержку .....	268
Итоги .....	269
Проверьте себя .....	270
<b>Глава 11. Документация .....</b>	<b>271</b>
11.1. Зачем составлять документацию? .....	272
11.1.1. Целевая аудитория .....	273
11.1.2. Формат .....	274
11.1.3. Проверка .....	275
11.2. Типы документации .....	275
11.2.1. Протоколы собраний .....	276
11.2.2. Демонстрации продукта .....	276
11.2.3. Инструкции по использованию .....	277
11.2.4. Резервное копирование и восстановление .....	278
11.2.5. Процесс деплоя .....	279
11.2.6. Комментарии в исходном коде .....	279
11.2.7. Схема архитектуры .....	280
11.2.8. Диаграммы процессов .....	281
11.2.9. Схемы сети .....	282
11.2.10. Схемы баз данных .....	283
11.2.11. Соответствие нормативным требованиям .....	284
11.2.12. Учет лицензий и результатов аудитов .....	284
11.3. Документы для публикации (whitepapers) .....	285
11.4. Рекомендации .....	285
Итоги .....	286
Проверьте себя .....	286
<b>Глава 12. Безопасность .....</b>	<b>287</b>
12.1. Внесение исправлений .....	288
12.1.1. Определение наличия исправлений .....	289
12.1.2. Планирование .....	290
12.1.3. Особые случаи .....	290
12.2. Тестирование на проникновение .....	291

12.3. Социальная инженерия .....	292
12.4. Утечка данных .....	293
12.4.1. Журналы событий .....	293
12.4.2. Сообщения об ошибках .....	294
12.4.3. Экспорт данных.....	294
12.4.4. Система контроля версий .....	295
12.5. Периодическая смена паролей .....	295
12.5.1. Учет параметров доступа .....	296
12.6. Безопасная инфраструктура .....	296
12.6.1. Инвентаризация .....	296
12.6.2. Защита .....	297
12.6.3. Мониторинг .....	299
12.7. Безопасная разработка .....	299
12.7.1. Создание безопасного исходного кода.....	300
12.7.2. Защита процесса сборки .....	301
12.8. «Нас атакуют» .....	302
12.8.1. Аварийное отключение.....	303
12.8.2. Коммуникация.....	304
12.8.3. Устранение последствий нарушения безопасности.....	305
12.9. Руководитель направления информационной безопасности (CISO) .....	307
Итоги .....	308
Проверьте себя .....	308
<b>Глава 13. Поддержка и обслуживание .....</b>	<b>309</b>
13.1. Оборудование.....	310
13.2. Договор на поддержку .....	311
13.2.1. Предоставление поддержки вашим клиентам.....	312
13.3. Мониторинг .....	313
13.3.1. Внешний мониторинг .....	313
13.3.2. Внутренний мониторинг.....	315
13.4. Резервное копирование и восстановление.....	316
13.4.1. Частота/срок хранения.....	318
13.4.2. Безопасность архива.....	318
13.5. Контроль бюджета.....	319
Итоги .....	320
Проверьте себя .....	321
<b>Глава 14. Рост компании.....</b>	<b>322</b>
14.1. Инвесторы .....	323
14.2. Импровизированный питч .....	324
14.3. В ногу с инвестором .....	324
14.4. Подготовка к интеграции .....	325

14.5. Due diligence .....	326
14.5.1. Что входит в due diligence.....	328
14.5.2. Проведение due diligence .....	332
14.5.3. Представление результатов.....	334
14.5.4. Прохождение due diligence.....	335
14.6. Передача полномочий .....	337
14.7. Принятие полномочий.....	339
Итоги .....	339
Проверьте себя .....	340
<b>Глава 15. Вы, Inc. ....</b>	<b>341</b>
15.1. Оцените себя .....	342
15.2. Наставничество .....	344
15.3. Идти в ногу со временем .....	345
15.4. Отслеживание изменений.....	346
15.5. Подготовка преемника .....	347
15.6. Карьерный успех .....	348
15.7. Шаг вперед .....	349
Итоги .....	350
Проверьте себя .....	351

*Я посвящаю эту книгу моему давнему другу и наставнику Джиму Милбери (Jim Milbery), коллеге-спецу (только не давайте ему программировать) и партнеру-основателю ParkerGale Capital.*

*Джим оказал огромное влияние на мою профессиональную и личную жизнь (и даже познакомил меня с моей будущей женой).*

*Я также посвящаю эту книгу моей дорогой покойной маме, которая была с нами в начале этого пути и прочла несколько первых глав, но так и не добралась до финала.*

# *Вступительное слово*

---

Книга «Настоящий СТО: думай как технический директор» — исчерпывающее и вдохновляющее руководство о том, как стать техническим директором и преуспеть в этой роли в современном мире. Эта книга помогает начинающим техническим директорам глубже понять основы, а опытным специалистам — отточить свои навыки. Мне нравится, что в этой книге собрано все, что необходимо грамотному руководителю в сфере технологий, и что акцент в ней сделан на личность руководителя. Если вы уже работаете техническим директором, раздел «Проверьте себя» в конце каждой главы послужит вам стратегическим руководством и видением, которое вы сможете воплощать в работе.

Роль технического директора быстро меняется практически во всех отраслях и секторах бизнеса. В начале 2000-х, с наступлением эпохи интернета, основными обязанностями технического директора было управление физическими серверами, командой разработчиков ПО и несколькими поставщиками, а также надзор за всем программным обеспечением, которое использовала компания. Хотя эти обязанности еще существуют, принципы управления, руководства, создания и выполнения стратегических планов радикально изменились.

Благодаря достижениям в сфере технологий сегодня технический директор — это человек, от которого ожидают намного большего. Мало того, что технический директор несет ответственность за все, что связано с технологиями, он еще и основное лицо, принимающее решения и формирующее стратегическое видение всех основных технологических аспектов. Кроме того, поскольку технологии непрерывно меняются и развиваются, он должен постоянно держать руку на пульсе этих изменений, чтобы превосходить конкурентов и злоумышленников.

Один из моих любимых разделов в книге — глава 6 «Управление командой». На мой взгляд, вы вкладываете больше всего времени и внимания в людей, которыми вы управляете и которых ведете за собой. Без этих людей вы потеряете

неудачу. Точка. Если вы приложите усилия к созданию атмосферы сотрудничества, где люди будут чувствовать себя самостоятельными и имеющими полномочия, чтобы идти вперед, быстро учиться на ошибках и расти в безопасных условиях, вы добьетесь максимально возможного. Технологии — едва ли не самая простая часть работы технического директора. Все дело в людях, которые приведут вас либо к успеху, либо к краху.

Независимо от того, новичок вы в мире высшего менеджмента и лидерства или опытный технический руководитель, эта книга станет вашим незаменимым справочником и пособием. Алан предлагает знания и идеи для решения проблем, с которыми сталкивается каждый технический лидер. Техническому директору стартапа эта книга даст четкое представление о его ответственности по мере роста и масштабирования команды и компании. Техническому директору растущего малого и среднего бизнеса она поможет ориентироваться в сложных ситуациях и предлагать стратегические решения. Для технических руководителей крупных компаний или государственных учреждений книга станет руководством по созданию и реализации их стратегического видения.

Надеюсь, «Настоящий СТО: думай как технический директор» понравится вам так же, как и мне. Советов, представленных в этой книге, давно не хватало технологическим лидерам.

— *Анкит Матхур (Ankit Mathur),  
основатель и CTO ROUNDTRIP HEALTH,  
основатель PRESS PLAY TECHNOLOGIES*



# Предисловие

---

Технический директор (СТО, Chief Technology Officer) — одна из самых интересных и плодотворных ролей в карьере технического специалиста. Технический директор определяет направление работы компании, открывает новые возможности, создает новые линейки продуктов, улучшает рабочие процессы и оказывает огромное влияние на все отделы.

Проводя аналогию с шахматами, генерального директора (CEO) можно представить в роли короля, а финансового директора (CFO) в роли ферзя, самой сильной фигуры на доске, которая может двигаться в любом направлении на любое количество клеток. В бизнесе финансовый директор также участвует в принятии каждого решения, затрагивающего финансовые потоки компании. Что же остается техническому директору?

Мне нравится думать о СТО как о ладье — следующей по силе фигуре после ферзя, поскольку она обеспечивает прикрытие для остальных фигур на доске, расчищая им путь на расстоянии. Тем не менее большинство людей не готовы ко всей мощи и ответственности этой роли, и многие не понимают, насколько она важна, особенно в современных условиях. Технический директор не только руководит исследованиями, разработкой и внедрением технологий компании, но и отвечает за процессы, лицензирование, соблюдение требований и постоянный мониторинг безопасности сервиса. В мире, где все подключено к компьютерным сетям, устройства имеют самые разные формы и размеры, а сотрудники работают как в офисе, так и удаленно, перед техническим директором стоят задачи, намного более сложные и разнообразные, чем те, с которыми работали СТО в 80-е и 90-е годы.

Более того, наша отрасль постоянно изобретает себя заново, наши инструменты и процессы претерпевают глобальные изменения каждые пять лет — от централизованных вычислений (с использованием мейнфреймов) до настольных компьютеров и интернета. Вместе с интернетом появился браузер, превратившийся из инструмента для просмотра статического контента в полноценную мощную самостоятельную платформу. Мы наблюдали мощный рост и развитие мобильных вычислений. На наших глазах облачные технологии вышли вперед

и получили развитие до такой степени, когда достаточно кредитной карты, чтобы запустить всю инфраструктуру для отдела компании, чего наши предшественники не могли даже вообразить. Безлимитные хранилища создают совершенно новый мир возможностей для анализа данных, дополненный машинным обучением и искусственным интеллектом.

Устройства становятся все меньше по размеру и все более связанными друг с другом, а интернет вещей проникает почти во все, что включено в розетку (и порой в то, что не включено). Блокчейн продвигает открытые и прозрачные взаимодействия. Языки и фреймворки становятся мощнее и быстрее. Обычные разработчики стали настолько далеки от логики управления процессором и памятью, что не каждый из них даже осознает существование ограничений: если код выполняется медленно, иногда бывает дешевле увеличить вычислительную мощность, чем тратить время на оптимизацию алгоритмов.

Кроме того, хакеры постоянно угрожают нашим системам. Программы-вымогатели нарушают работу государственных служб, наглядно демонстрируя, как плохая архитектура может поставить страну на колени. Мы стали свидетелями этого, когда в мае 2021 года хакеры вывели из строя нефтепровод на восточном побережье США, что привело к нехватке бензина на заправках, продолжавшейся более недели. И это не единичный случай. Представьте, что в то утро вы проснулись директором, отвечающим за техническую часть этого нефтепровода, — как вы будете действовать?

Неважно, опытный ли вы технический директор, новичок, пришедший на смену, или генеральный директор, пытающийся решить, нужен ли ему технический директор (ответ на этот вопрос — да, нужен), вы вряд ли осознаете, что эта должность намного шире, чем принято считать. Независимо от размера компании — стартап ли это из двух человек или корпорация с тысячами сотрудников — зона ответственности СТО весьма широка.

Эта книга родилась из-за необходимости заполнить пробелы в подготовке как компании, так и отдельного сотрудника к роли технического директора. Я написал ее после многих лет работы в компаниях разного размера — от стартапов до малого и среднего бизнеса (чей доход обычно не превышает 100 миллионов долларов) — в качестве штатного СТО или временно исполняющего обязанности СТО для портфельных компаний как в сфере прямых инвестиций, так и вне ее.

Эти компании работали в совершенно разных сферах, практически не пересекаясь. Но везде было одно и то же требование к техническому директору: он должен придумать, организовать, создать и поддерживать техническую часть продукта компании.

Когда я начинал, я понятия не имел, чего от меня ждут. Я пообещался со своей сетью контактов, с работающими техническими директорами и подробно расспросил многие доверенные источники о том, что мне нужно делать, чтобы добиться успеха. Одно было ясно: ответственность гораздо больше, чем я себе представлял. Мне предстояло многому научиться — и сделать это быстро. Я совершил много ошибок, в основном из-за незнания того, что должен делать.

Эта книга — именно то, чего мне не хватало тогда, и то, что у вас есть сейчас.

# Благодарности

---

Движущей силой создания этой книги была любовь. Через два года работы в условиях пандемии и параллельно важным событиям в личной жизни книга была наконец готова. Моя жена Келли всегда была для меня опорой и необходимой поддержкой; она воодушевляла меня, просматривала каждую главу и давала объективные комментарии. Сказать, что без нее я бы не справился, — ничего не сказать. Огромная благодарность Кейтлин Пауэлл (Kaitlin Powell), самому вдохновляющему художнику, с которым мне посчастливилось работать, поскольку она прошла со мной все итерации создания рисунков.

Большое спасибо моим близким друзьям, Джиму Хедли (Jim Headley) и Райану Берчу (Ryan Burch), за поддержку и вдохновение по многим затронутым темам. На протяжении всей книги я упоминаю имена тех, кто поделился своим опытом и идеями, — спасибо вам, без вашей помощи все было бы иначе.

Конечно же, я благодарю своих родителей, которые потакали капризам 10-летнего мальчика и потратили месячную зарплату на ZX Spectrum, чтобы ребенок проверил, есть ли у этой «компьютерной штуки» будущее. Учитывая, что я полностью износил его резиновую клавиатуру, думаю, что инвестиции себя оправдали.

Наконец, я выражаю огромную благодарность всем и каждому в Manning, особенно Дастину Арчибальду (Dustin Archibald), который помог мне придать каждой главе законченную и осмысленную форму. За каждым великим писателем стоит еще более великий редактор. Я также благодарю Дейдру Хайэм (Deirdre Níam), редактора моего проекта; Памелу Хант (Pamela Hunt), литературного редактора; и Кэти Теннант (Katie Tennant), корректора.

Всем рецензентам — Адаму Дудчаку, Адриану Билаука, Адриану Росси, Андреасу Бродманну, Антонио Бруно, Арони Пани, Бурку Гюнешу, Чаду Миарсу, Крису Хенегану, Крису Томасу, Кристиану Уиттсу, Данило Зековичу, Десмонду

Хорсли, Дираджу Гупте, Дориану Басуяу, Флавио Диесу, Франческо Персико, Гэвину Бауманису, Гордану Бэкингеу, Хироюки Муше, Йоханнесу Лохманну, Джону Маккормаку, Джону Вуду, Юре Шикину, Кевину Пелгримсу, Леонардо Анастасии, Матею Страшеку, Мэтту Фердереру, Майку Оверу, Николя Мод-шику, Нику Римингтону, Павло Ходышу, Павелу Климчику, Рику Оллеру, Сету Коупленду, Широшике Кулатилаке, Симоне Кафьеро, Стефану Мучлеру, Тьяго Болдту Соузе, Виллему ван Кетвичу, Уильяму Руденмальму (Adam Dudczak, Adrian Bilauca, Adrian Rossi, Andreas Brodmann, Antonio Bruno, Aroni Pani, Burc Gunes, Chad Miars, Chris Heneghan, Chris Thomas, Christian Witts, Danilo Zekovic, Desmond Horsley, Dhiraj Gupta, Dorian Basuyau, Flavio Diez, Francesco Persico, Gavin Baumanis, Gordan Buckingham, Hiroyuki Musha, Johannes Lochmann, John McCormack, John Wood, Jura Shikin, Kevin Pelgrims, Leonardo Anastasia, Matej Strašek, Matt Ferderer, Mike Over, Nicolas Modrzyk, Nik Rimington, Pavlo Hodysh, Pawel Klimczyk, Rick Oller, Seth Copeland, Shiroshica Kulatilake, Simone Cafiero, Stefan Mutschler, Tiago Boldt Sousa, Willem van Ketwich, William Rudenmalm) — спасибо; ваши предложения помогли сделать эту книгу лучше.

## О книге

---

Эта книга ответит на некоторые вопросы, которые вы, возможно, задавали себе, но не осмеливались задать своему генеральному директору из боязни показаться несведущим. В конце концов, вас наняли, чтобы вы позаботились обо всем, чего не могут сделать другие, поэтому они должны быть уверены, что вы тот, кто с этим справится.

Каждая глава посвящена одной из сфер ответственности СТО. Я подробно рассмотрю их, выделяя соображения, о которых следует помнить, и стратегии управления каждой из этих сфер. Однако это не кулинарная книга технологий. Я не буду рекомендовать Java вместо C#, или Amazon вместо Azure, или облако вместо центра обработки данных.

Вместо этого я укажу, о чем следует помнить при принятии решений на уровне организации, потому что решение, которое вы примете, может иметь долгосрочные последствия даже после того, как вы покинете эту должность. Я предлагаю небольшие кейсы для иллюстрации процесса принятия решения. Каждое ваше решение будет зависеть от того, что нужно компании и какие ресурсы у вас есть. Не существует правильного или неправильного ответа — может быть «менее правильный» или «более правильный» способ сделать что-то, но решение, основанное на данных, а не на интуиции, всегда оправдано.

Эта книга даст вам представление о возможных сферах ответственности технического директора, а также необходимые навыки оценки, понимания и исполнения, чтобы вы могли показать успешный результат в контексте компании.

Роль СТО постоянно меняется, и каждый день приходится решать новые задачи. Вот почему я люблю эту работу — с самого первого дня на ней невозможно заскучать. Вы должны следить за трендами. Мы работаем в одной из самых быстроразвивающихся отраслей, где каждый день появляются новые интересные технологии. Как выбрать, чему следовать, а что игнорировать? Как проверить,

что ваш стиль не устарел? Как много необходимо знать? Насколько подробно разбираться в теме? Эти вопросы мы, технические специалисты, задаем себе постоянно. Но когда вы принимаете решения для всей компании, последствия могут быть далеко идущими.

Мы все страдаем от синдрома самозванца — ощущения, что мы на самом деле не знаем, что делаем, а кто-то другой всегда знает лучше. Оно совершенно естественно и широко распространено в сфере технологий, учитывая, как быстро все развивается. Каждый раз, читая новости о последних выдающихся инновациях, вы задаетесь вопросом, успеваете ли вы за ними или подводите свою команду или компанию. Уверяю вас, что это чувство не исчезнет и вы должны научиться жить с ним и принимать его. Я раскрою несколько приемов, которые помогут вам убедиться, что вы не отстаете и не поддерживаете технологию, которая устареет в следующем году.

Эта книга содержит множество практических примеров, иллюстрирующих затронутые темы, а также множество реальных историй о проблемах, с которыми столкнулись технические директора и которые им удалось преодолеть. В рамках своих обязанностей в MacLaurin Group, а теперь и в частной инвестиционной компании New Harbour Capital я работаю с техническими директорами из различных отраслей и в сфере прямых инвестиций, оказывая им поддержку, менторство и консультируя по конкретным проблемам, с которыми они сталкиваются.

Многих ситуаций, с которыми мне приходилось разбираться, можно было бы избежать при лучшем планировании и наличии некоторого опыта. В основе книги лежат именно эти встречи — реальные проблемы реальных СТО, — а также мой собственный опыт создания технических команд в качестве штатного технического директора.

Книга рассматривает именно те области, которые считаются компетенцией технического директора. Поскольку границы этой роли не так четко определены, как для более устоявшихся руководящих ролей, она в значительной степени станет такой, какой ее сделаете вы. СТО может сделать себя бесценным для компании, если разглядит имеющиеся у него возможности помочь ей работать как хорошо смазанный двигатель.

Итак, работаете ли вы в одиночку или руководите командой из 100 инженеров, из книги вы узнаете, что нужно, чтобы стать успешным СТО.

## ДЛЯ КОГО ЭТА КНИГА

Если вы принадлежите к одной из следующих категорий, значит, вы выбрали правильную книгу:

- Опытный технический специалист, желающий перейти на следующую ступень карьеры — к роли технического директора.
- Технический директор, впервые оказавшийся на этой должности и желающий убедиться, что он делает все от него зависящее.

- Опытный технический директор, чья компания быстро растет и у которого нет ответов на задаваемые вопросы.
- Технический директор, который пришел на смену предыдущему и от которого руководство ждет больших, революционных изменений.
- Генеральный или финансовый директор, пытающийся решить, необходим ли компании технический директор (и чем он будет заниматься), и если да, то как подобрать человека на эту должность.

## СТРУКТУРА КНИГИ

В книге 15 глав:

- В главе 1 рассматривается, кто такой технический директор и какие бывают разновидности этой должности, а также какова роль технического директора в организации.
- Глава 2 посвящена взаимодействию СТО с различными сотрудниками в организации — от генерального директора до коллег.
- Глава 3 посвящена разработке видения роли технического директора в организации.
- В главе 4 описаны некоторые рекомендации по созданию команды.
- В главе 5 показано, как подбирать будущих членов команды, проводить собеседования и принимать на работу новых сотрудников.
- Глава 6 посвящена управлению командой.
- В главе 7 объясняется процесс проведения ежегодных ревью.
- Глава 8 поможет принимать технологические решения.
- Глава 9 посвящена управлению проектами и информированию организации о ходе их выполнения.
- Глава 10 объясняет, как работать с договорами.
- Глава 11 посвящена созданию документации и работе с ней.
- В главе 12 приводятся некоторые важные соображения, касающиеся безопасности компании.
- В главе 13 рассматриваются дополнительные соображения, которые необходимо учесть в работе.
- Глава 14 посвящена управлению ростом компании.
- Глава 15 посвящена определению и анализу вашего собственного прогресса и планов на будущее.
- В зависимости от размера организации некоторые главы могут быть неактуальны для вас. Они могут подойти к другому периоду вашей карьеры или развития компании. Однако, даже если это так, не будет лишним ознакомиться с этими главами на случай организационных изменений, при которых вам придется начать управлять этими областями.

## ФОРУМ LIVEBOOK

Приобретая книгу «Настоящий СТО: думай как технический директор», вы получаете бесплатный доступ к закрытому веб-форуму издательства Manning (на английском языке), на котором можно оставлять комментарии о книге, задавать технические вопросы и получать помощь от автора и других пользователей. Чтобы получить доступ к форуму, откройте страницу <https://livebook.manning.com/book/think-like-a-cto/discussion>. Информацию о форумах Manning и правилах поведения на них см. на <https://livebook.manning.com/#!/discussion>.

В рамках своих обязательств перед читателями издательство Manning предоставляет ресурс для содержательного общения читателей и авторов. Эти обязательства не подразумевают конкретную степень участия автора, которое остается добровольным (и неоплачиваемым). Задавайте автору хорошие вопросы, чтобы он не терял интереса к происходящему! Форум и архивы обсуждений доступны на веб-сайте издательства, пока книга продолжает издаваться.



## Об авторе

---

**АЛАН УИЛЬЯМСОН** более 25 лет работает в области данных и технологий, внес свой вклад в спецификацию ключевых серверных Java API, создав использовавшийся в MySpace первый в мире движок CFML, написанный на Java. Он стал первым евангелистом Java (Java Champion) в Великобритании и опубликовал несколько книг по Java, посвященных Java Enterprise, сервлетам, JavaMail и работе с базами данных.

Он работал с частными инвестиционными компаниями более 15 лет, создавая и развивая команды, а также выполняя функции технического директора в ряде портфельных компаний. Алан являлся техническим директором и партнером MacLaurin Group и поддерживал деятельность портфельных компаний, консультируя технических директоров и системных архитекторов. Он осуществлял техническое руководство в нескольких организациях, поддерживаемых частными инвесторами. В настоящее время он является партнером группы портфельных операций в New Harbour Capital, чикагской инвестиционной компании, специализирующейся на предприятиях среднего бизнеса, и предоставляет услуги исполняющего обязанности технического директора и ментора.

Алан имеет степень в области computer science Университета Пейсли, Шотландия, со специализацией «цифровое управление».

# *О редакторе русского издания*

---

Логинов Олег Евгеньевич — разработчик, руководитель разработки и технический директор с более чем 20-летним опытом. Создавал успешные команды и проекты в крупнейших российских IT-компаниях и медиахолдингах: ВГТРК, «Газпром Медиа», VK, «Национальная Медиа Группа».

На практике знаком со всеми аспектами работы технического директора: от создания и эксплуатации технологий для высоконагруженных проектов с аудиторией в десятки миллионов пользователей до формирования команды, организации работы, планирования и бюджетирования.

## **ОТ ИЗДАТЕЛЬСТВА**

Ваши замечания, предложения, вопросы отправляйте по адресу [comp@piter.com](mailto:comp@piter.com) (издательство «Питер», компьютерная редакция).

Мы будем рады узнать ваше мнение!

На веб-сайте издательства [www.piter.com](http://www.piter.com) вы найдете подробную информацию о наших книгах.

# 1

## Технический директор (СТО)

---

### В этой главе

- ✓ Суть работы технического директора (СТО).
- ✓ Различные типы СТО.
- ✓ Эволюция от инженера до технического директора.
- ✓ Определяем, нужен ли компании технический директор.

Вероятно, раз уж вы читаете эту книгу, вы уже представляете себе, кто такой технический директор (СТО). Тем не менее, если попросить каждого читателя дать определение этой роли, мы бы получили столько же определений, сколько и читателей. И, скорее всего, каждое из этих определений будет верно. Подобное разнообразие интерпретаций делает эту роль не только сложной, но также и очень увлекательной.

Роли ближайших коллег, генерального директора (CEO) и финансового директора (CFO), более четко определены и их сферы ответственности более четко обозначены: генеральный директор указывает направление, а финансовый директор выписывает чеки! Хорошо, мы немного лукавим, но эти роли действительно общепризнаны, в отличие от неопределенной позиции технического директора.

Например, в выпадающем списке должностей в онлайн-формах позиция «технический директор» часто отсутствует. Хотя ситуация меняется — особенно в сфере страхования, где *СТО* начал все же появляться в списках, — неужели эта позиция предполагает повышенные страховые риски?

Определение СТО трансформируется и расширяется в зависимости от организации, при этом его обязанности сильно варьируются от компании к компании. На верхнем уровне технический директор отвечает за технологическое видение и его реализацию, хотя некоторые компании даже не в курсе, что технический директор отвечает за организацию работы. Некоторые крупные организации не видят ничего необычного в том, что у них есть СТО, которому никто не подчиняется. Однако в этой книге мы исходим из того, что технический директор отвечает за обе области: и видение, и руководство.

Мы рассмотрим различные типы СТО, чтобы вы определили, какой из них вам лучше всего подходит. Это поможет сформулировать и закрепить ваши карьерные цели. Кроме того, мы выделим качества, присущие успешным техническим директорам.

В оставшихся главах будут более подробно рассмотрены ключевые вопросы, решать которые в какой-то момент приходится подавляющему большинству технических директоров. Хотя не все главы могут иметь отношение к вам, но за рассматриваемый вопрос в вашей компании в любом случае отвечает кто-то из ваших коллег, поэтому никогда не помешает ознакомиться, с чем им приходится работать.

## 1.1. СУТЬ РАБОТЫ ТЕХНИЧЕСКОГО ДИРЕКТОРА

На самом верхнем уровне СТО прежде всего отвечает за техническое руководство и реализацию основного продукта компании, то есть решение задач бизнеса в интересах клиента. СТО обычно подчиняется генеральному директору, но нередко и финансовому директору, главному операционному директору (Chief Operating Officer, COO) или даже ИТ-директору (Chief Information Officer, CIO), в зависимости от размера компании и количества прямых подчиненных, которым генеральному директору комфортно управлять.

И наоборот, в компетенцию СТО обычно не входит ИТ-инфраструктура, или бэк-офис. Хотя серверы и программное обеспечение, из которых состоит ваш продукт, относятся к ведению технического директора, но настольные компьютеры, почтовые серверы, телефоны, принтеры и т. д. — нет. Обычно это зона ответственности ИТ-директора или отдельного менеджера, который чаще всего подчиняется финансовому директору. Технический директор же сфокусирован на разработке, поддержке и эксплуатации основной платформы, технологии или линейки продуктов, которые обслуживают клиентов (заказчиков), — иногда это называют фронт-офисом.

Тем не менее в небольших компаниях эта граница нередко бывает размытой, и в зависимости от своих навыков вы можете заниматься тем, что потенциально не входит в вашу зону ответственности. В стартапах и небольших компаниях СТО обычно делает все — от заказа ноутбуков и заключения договоров с интернет-провайдером до замены чернил в принтере, написания кода и выкладывания его на серверы.

С другой стороны, в устоявшемся (малом или среднем) бизнесе обычно нет формальной должности СТО. В нем может быть команда разработчиков, группа поддержки и команда системных инженеров, но никто не следит за технологической стратегией и не управляет ею как единым целым. В подобную ситуацию легко попасть, если компания долгое время развивается, не осознавая, что ее связь с данными/технологиями теснее, чем может показаться.

В этом нет ничьей вины, поскольку большинство не понимает, как воспользоваться преимуществами роли СТО, чтобы получить максимальную отдачу от своей деятельности. Опять же, эта книга поможет вам убедить руководство в необходимости введения должности технического директора.

Подобно тому, как финансовый директор организует финансы и бухгалтерию таким образом, что работа с ними становится удобной и беспроблемной, так и хороший технический директор разгружает генерального директора и совет директоров, чтобы они могли сосредоточиться на вопросе *что*, в то время как он сам занимается вопросом *как*. Хороший технический директор обеспечивает для своей компании:

- Стабильную, масштабируемую и управляемую платформу, которая обеспечит развитие компании.
- Компетентное лидерство для поддержания актуальности платформы.
- Возможность использовать потенциал платформы для исследования новых областей бизнеса.
- Операционную устойчивость для надежного и предсказуемого обслуживания клиентов.
- Гибкое использование технологий на благо компании и клиентов.
- Создание и сохранение высокоэффективной команды, а также управление ею.

Переход финансового директора на должность генерального — это обычная и естественная практика, а вот разработчики и ИТ-менеджеры часто плохо подготовлены к роли технического директора. Большинство из них считает, что это просто расширение их текущих повседневных задач. Такое отношение, хотя и часто встречающееся, ведет к провалу.

СТО — это гораздо больше, чем просто управление командой и написание кода, особенно когда компания начинает расти и, тем более, когда ее приобретает другая компания или инвестиционная группа. Как уже отмечалось, у хорошего специалиста все выглядит просто, но за этой простотой скрываются многолетний опыт и серьезные испытания.



В зону ответственности технического директора входит широкий круг вопросов, от проектирования архитектуры и дизайна продукта до приема на работу новых сотрудников, вопросов внедрения, соответствия нормативным требованиям, безопасности, отчетности/взаимодействия, стратегии/видения, а также планирования бюджета. По мере развития компании СТО может начать заниматься и другими вопросами, включая подготовку к презентациям генерального директора и/или совета директоров, работу с поставщиками, подготовку для внешних инвесторов, получение и обработку запросов на due diligence и даже оценку других команд / технологических стеков, если планируется их интеграция с потенциальным партнерством или поглощением.

Хорошего СТО отличает способность адаптироваться к потребностям компании или, по крайней мере, вовремя признать, что он больше не может выполнять свои обязанности и должен уступить место тому, кто может.

#### **ВИЦЕ-ПРЕЗИДЕНТ ПО РАЗРАБОТКЕ**

Стоит отметить, что грань между вице-президентом по разработке и техническим директором иногда довольно размыта и часто является предметом дискуссий. Разница может существовать только в названиях должностей в штатном расписании, а может быть четкое разделение, когда вице-президент по разработке подчиняется техническому директору, отвечая за обеспечение работы команды инженеров и оставляя все остальное техническому директору. Для целей этой книги многое, о чем я говорю, применимо к обоим вариантам, и я считаю, что вице-президент по разработке — это промежуточная ступень на пути к должности СТО.

## **1.2. РАЗЛИЧНЫЕ ТИПЫ СТО**

Определение СТО во многом зависит от среды, в которой он работает, поэтому рассмотрим немного подробнее распространенные типы компаний, в которых может быть позиция технического директора. Вопросы в конце каждого раздела помогут вам определить, к какому типу вы ближе всего:

- Предстартап, номинальная должность.
- Стартап, эксперт в области технологий.
- Устоявшаяся/зрелая компания:
  - Технический директор, должность учреждена впервые;
  - Технический директор, пришедший на смену прежнему СТО.

### 1.2.1. Предстартап, номинальная должность

Типичный стартап на начальном этапе может состоять всего из двух человек — визионера и технаря. Скорее всего, они оба в начале своего профессионального пути, возможно, только что закончили учебу и не имеют реального опыта работы в отрасли. Тем не менее их привлекают перспективы, которых может достичь только мечтатель с железной волей.

Здесь технарь каждый день пишет код, пытаясь не отставать от видения основателя в том, как они представляют компанию. Продукт ожидает множество крутых поворотов и редизайнов, поскольку творение начинает попадать в руки пользователей. Для солидности основатель печатает технарю визитки с названием должности «технический дирек-



тор» (в основном чтобы компенсировать низкую зарплату, если она вообще есть).

Они идут на всевозможные ухищрения, чтобы добиться результата; но, когда клиенты начинают платить, необходимо сместить фокус внимания с разработки на рост продукта. Это подразумевает вопросы поддержания работоспособности систем, защиты данных в соответствии с нормативными требованиями и роста команды — все это вдобавок к написанию кода.

Хотя такой технический директор и не находится в одной лиге с лучшими представителями отрасли (возможный претендент на такое звание — Вернер Фогельс (Werner Vogels) из Amazon), забот и тревоги у него ничуть не меньше. Он, как правило, не желает отказываться от контроля и чувствует угрозу, когда кто-то из новичков в команде кажется более опытным. Тому, кто тащил проект ночами, кровью и потом, трудно признавать, что может появиться кто-то более подходящий, чем он.

Когда к проекту подключаются крупные инвесторы, такие как венчурные фонды, — они могут попытаться привлечь к управлению специалиста с большим опытом, поскольку технология является ключевым компонентом успеха. Однако существующий технический директор вполне может справиться с новыми обязанностями, если получит небольшую помощь и наставничество. Этим техническим директором можете быть вы, если ответите «да» на один или несколько из следующих вопросов:

- Когда вы принимаете технические решения, никто их не оспаривает и не проверяет.
- Вы единственный человек, который занимается разработкой.
- Это ваша первая настоящая работа, даже если она не очень щедро оплачивается.

- В компании очень мало процессов или структур.
- Только вы знаете, как управлять системой (системами) и обслуживать ее.

### 1.2.2. Финансируемый стартап: технологический эксперт с деньгами

На следующем этапе развития стартапа, описанный в предыдущем разделе, убеждает одного или нескольких инвесторов (членов семьи, бизнес-ангелов или венчурный фонд) вложиться в развитие бизнеса. На кону серьезные деньги, а размер бюджета заставляет поверить, будто успех уже в кармане. Однако большинство не понимает, что живет в кредит, который рано или поздно придется возвращать.

Техническому директору поручено быстро сформировать команду для поддержки архитектуры, которая еще не приобрела окончательную форму. Необходимо выдавать продукт, чтобы бизнес мог двигаться вперед, но команда разработчиков еще не определилась с тем, что конкретно они делают, поэтому трудно понять, что создавать.

В конце концов продукт поставлен, и заплатившие за него клиенты ожидают определенного уровня обслуживания. СТО теперь должен сместить акцент с исследований и разработки на поддержку и развитие. Удовлетворение потребностей клиентов и масштабирование при одновременном управлении созданием следующего поколения продукта — это искусство, требующее строгой дисциплины.

Со временем вопросы экономии начинают выходить на первый план. СТО должен выбрать подходящую технологию, не слишком затратную, но при этом с возможностью масштабировать расходы по мере роста клиентской базы. В прошлом большая часть инвестиций стартапа тратилась на дорогостоящие серверы

и хостинг. Теперь же при грамотном ведении дел и стратегическом использовании облачных сервисов расходы на эксплуатацию могут составлять лишь малую часть общих инвестиций. Тем не менее неверное решение может повлечь рост затрат.

Многие технические директора в подобном положении не желают следовать принятым правилам и начинают испытывать трудности (в основном потому, что не знают правил, ведь это их первая подобная должность). Иногда в их действиях наблюдается полное отсутствие какой-либо системы, потому что полная свобода им





кажется современной и модной. Тем не менее структуры и процессы существуют не просто так: они необходимы для предоставления клиентам стабильного и предсказуемого сервиса.

С другой стороны, некоторые СТО отчаянно стремятся придерживаться однажды заданной философии и испытывают параноидальный страх принять неверное решение. В итоге они доводят себя до состояния, в котором не могут двигаться вперед, потому что цена ошибки слишком велика. Это явление — паралич принятия решений, когда страх перед неверным решением препятствует принятию вообще любого решения.

Компании этого типа уже участвуют в финансовой гонке, пытаются получить прибыль до того, как у них закончатся деньги, и превратиться из стартапа в устоявшийся, прибыльный бизнес, и это вызывает стресс у всех участников. Вы работаете в подобной компании, если ответили «да» на один или несколько из следующих вопросов:

- Компания моложе трех лет.
- Никто в вашем отделе еще не проработал больше 18 месяцев.
- Ваша платформа переживает вторую (или третью) итерацию.
- Компания тратит больше денег, чем зарабатывает.
- Вам кажется, что параметры продукта меняются каждую неделю/месяц.

### **1.2.3. Устоявшаяся компания: первый технический директор**

Успешная и зарекомендовавшая себя компания, обычно возглавляемая основателем, развивается до этапа, когда ее глава решает, что необходимо более твердо управлять платформой. В его распоряжении есть несколько разработчиков и вспомогательный персонал, возможно, подчиняющийся финансовому директору. Системы играют ключевую роль в поддержании бизнеса, но со временем они начинают устаревать, что затрудняет либо исправление ошибок, либо расширение (такие платформы, как PowerBuilder, Microsoft Access и Visual Basic, используются гораздо шире, чем многие готовы признать).

Главная проблема подобных компаний, которую они еще не осознали, заключается в том, что никто не занимается обеспечением бесперебойной работы и жизнеспособности технологической платформы в долгосрочной перспективе. Это все равно что жить в доме много лет, забывая, насколько важны такие вещи, как кондиционер и горячая вода, не понимая, что с каждым годом коммуникации постепенно ветшают и в конце концов откажут, оставив дом без самого необходимого.

Иногда компании везет — руководство понимает, что необходимо заменить критически важные системы, но просто не знает, как это сделать. Может быть, настойчивый торговый представитель убедил всех, что проблемы будут решены,

если компания перейдет на дорогое готовое решение, но цена таких вложений заставляет задуматься. Отрасль знает много примеров неудачного внедрения Salesforce/Microsoft/Oracle, случившихся по вине некомпетентных решений. Платформа ни при чем, дело в том, что в большинстве случаев достаточно не полного пакета услуг, а лишь его небольшой части. На другой стороне диапазона находится руководство, которое не понимает, что их время давно вышло и при следующей перезагрузке все сломается, и считает, что ничего не нужно менять — *работает сегодня, будет работать и завтра*.

После обсуждения ситуации, возможно, выслушав советников или инвесторов, руководство решает нанять первого технического директора. У успешного кандидата впереди много возможностей, но также ему придется доказать свою полезность, потому что некоторые до сих пор считают, что можно прекрасно обойтись без него.

Такой технический директор должен уделять больше внимания стратегии и учитывать, что внедрение изменений или новых систем может вызвать нервозность в команде. Одна из его первоочередных задач будет заключаться в замене устаревших систем без нарушения текущих операций. Ему предстоит столкнуться со множеством возражений из разряда «*У нас так не принято*», и, хотя задача может быть трудной, ее решение приведет к значительной трансформации и принесет пользу компании. Эта роль идеально подходит для технического директора с опытом или для тех, кто хочет перейти на следующий уровень профессионального развития. У таких компаний следующие особенности:

- Это устоявшаяся компания с платежеспособными клиентами и зрелой линейкой продуктов.
- Большинству платформ более 10 лет.
- Лишь немногие в курсе, как поддерживать все в работоспособном состоянии; они не знают, почему делают те или иные вещи, но знают, что если они их делают, то все работает как надо.
- Большинство людей в компании не осознают, насколько сильно они полагаются на технологии или данные.
- Отсутствуют стандартные политики или практики, в том числе реальное внимание к безопасности или соответствию требованиям.

#### 1.2.4. Устоявшаяся компания с техническим директором

Наконец, солидная компания осознает ценность технического директора. Возможно, он уже был в компании с самого начала и получил все необходимые знания в процессе работы без подсказок и указаний, что и как делать или не делать. Кажется, у него все работает, потому что компания оправдывает ожидания клиентов. Но пришло время двигаться дальше, поэтому он оставляет хорошо работающую машину.

Смена руководства может означать как глоток свежего воздуха и новые задачи, так и просто поддержание существующего порядка вещей. Частные инвестиционные компании, как правило, меняют топ-менеджеров, если считают, что у нынешних менеджеров недостаточно опыта или знаний, чтобы привести компанию к желаемой цели. Им некогда ждать и наблюдать, потому что период владения компанией обычно составляет от трех до пяти лет. Подобную компанию легко узнать:

- Позиция технического директора в ней есть не менее трех лет.
- Организован технический отдел, определены роли и иерархия.
- Установлены процедуры и стандартные методы работы, и в основном они соблюдаются.
- Существуют общие планы, стратегическое видение и планы проектов.
- Технология стабильна и понятна большинству сотрудников компании.

Конечно, существуют и смешанные типы компаний, но в каждой из них СТО в какой-то момент проходит один из описанных этапов развития карьеры.

## 1.3. ОПРЕДЕЛЕНИЕ ПОТРЕБНОСТИ В ТЕХНИЧЕСКОМ ДИРЕКТОРЕ

Руководство часто задается вопросом, нужен ли им СТО. Или — если он уже есть — нужно ли заменить его кем-то, кто обладает другим набором навыков и драйвом? Это решение может быть вызвано наблюдениями за другими похожими компаниями аналогичного размера, в которых эта позиция уже имеется. Еще одним триггером может быть стойкое убеждение, что технологии приносят меньше пользы, чем могли бы — вера в то, что все должно быть намного проще (и, возможно, дешевле), чем сейчас.

Если в компании, помимо ИТ-систем бэк-офиса (электронная почта, файлы, календари, чат и т. д.), есть технологии, с которыми взаимодействуют клиенты или которые имеют решающее значение для получения прибыли, — крайне важно, чтобы кто-то контролировал их эксплуатацию, постоянное совершенствование и развитие. Еще один простой тест: есть ли у вас специально разработанный программный продукт, без которого клиенты не смогут использовать то, за что они платят компании?

Так как же доказать руководству, что эта должность необходима? Это может быть непросто, особенно если вы сами на нее претендуете. Самая успешная аргументация всегда та, которая основывается на данных, а не на эмоциях. Это не значит, что интуиция не играет роли в принятии решений, но она никогда не должна быть единственным фактором.

Рассмотрите следующие вопросы. Если ответ на них утвердительный, значит, должность технического директора вам необходима:

- Разрабатывается ли в компании специальное ПО для клиентов или для поддержки клиентов?
- Проводилась ли значительная доработка сторонней платформы (например, Salesforce или Microsoft Dynamics) для поддержки клиентов?
- У вас отсутствуют данные о взаимодействии клиентов с вашими системами?
- В вашей отрасли существуют требования или законы, которые необходимо соблюдать (например, вы храните медицинскую или финансовую информацию о своих клиентах или для них)?
- Относятся ли в компании к вопросам безопасности как к досадной помехе и часто оставляют пароли в открытом доступе?
- Вы используете и поддерживаете свои собственные серверы, с которыми работают клиенты?
- У вас отсутствует песочница или безопасная среда для тестирования и проверки нового ПО или конфигураций?
- У вас маленькая команда начинающих или совсем неопытных технических специалистов без технического руководства?
- Слова «исправление» или «обновление» программного обеспечения звучат для вас в новинку и вы годами этим не занимались?
- У вас отсутствует документация или управление ресурсами/версиями для систем, используемых клиентами?
- У вас нет политики резервного копирования или аварийного восстановления без прерывания сервиса?
- Принципы работы системы известны лишь немногим ключевым сотрудникам, а это значит, что без них компания окажется в беде в случае критического сбоя?
- Решает ли служба поддержки проблемы, связанные с некачественным или устаревшим программным обеспечением?
- Текущие системы начинают устаревать и не в состоянии обеспечить функционал, который требуется клиентам?

Ресторан не может работать без шеф-повара, который следит за кухней и за тем, чтобы каждый выполнял свою работу, а посетители получали вкусную еду. Технический директор — это шеф-повар компании, следящий за всем, что необходимо, чтобы блюда подавались.

## 1.4. ЭВОЛЮЦИЯ ОТ ИНЖЕНЕРА

Подавляющее большинство технических директоров, с которыми мне посчастливилось работать, быть наставником или просто иметь знакомство, имели богатое техническое прошлое, что довольно важно. Многие из них обладали хорошим практическим опытом и тягой ко всему новому — это отличные качества для



надежного и грамотного старшего инженера, даже архитектора. Но делают ли они своего обладателя хорошим техническим директором?

К сожалению, не всегда. Хотя роль СТО выглядит привлекательной, она подразумевает множество обязанностей, неочевидных на первый взгляд (в этом вы убедитесь, прочитав эту книгу), и для большинства инженеров эти обязанности представляют сложность. Не каждый технический директор с легкостью несет бремя офисной работы, потому что оно не соответствует его природным инженерным склонностям. Почему так происходит?

Одна из главных причин — неготовность мыслить более масштабными и долгосрочными категориями. Масштабность — это не создание более крупных платформ или управление ими, а умение рассматривать эти системы в контексте компании и конечного пользователя. Долгосрочное мышление основано на пяти-летних блоках — естественном цикле бизнеса. Как платформа будет выглядеть через пять лет? Будет ли она удовлетворять потребности бизнеса и останется ли при этом актуальной?

На первый взгляд это кажется логичным и простым, но стоит копнуть немного глубже — и проблема становится сложной и запутанной и никогда не решается до конца, но требует гибкого и адаптируемого подхода.

Помимо социальных навыков, которые необходимо освоить, — умения общаться с коллегами, не имеющими инженерного образования, готовить бюджет для проекта, который еще непонятно что собой представляет, набирать и поддерживать команду талантливых специалистов, — при всем этом чтобы еще и клиент остался доволен, — существует мир неопределенных решений и неизвестных результатов, такой непохожий на бинарный мир, где все определено, из которого пришел новоиспеченный СТО и в котором ему комфортно.

С этим можно справиться. Самые успешные технические директора — это те, кто знает, как применить свои технические знания на пользу компании. Самый эффективный инструмент в арсенале инженера на должности СТО — это

понимание, чего ожидать: вы знаете, что будет работать, а что нет, и сколько времени что-то потребует или будет продолжаться. Этот инструмент поможет стать эффективным и результативным техническим директором.

### МУДРЫЕ СЛОВА

*Вам не скажут спасибо*

Я вспоминаю один из лучших советов, который дал мой давний наставник и друг Джим Милбери, партнер-основатель ParkerGale Capital. В то время я был архитектором, переделывал проект платформы для портфельной компании и совершил ошибку новичка, думая о сегодняшнем дне и забывая о завтрашнем. Моим оружием было то, что благодаря новой архитектуре я сэкономил компании 10 000 долларов в год. По тем временам это были большие деньги. Однако существовал риск (небольшой) сбоя компонента. Джим сказал, что экономия — это хорошо, но попробуй объясни совету директоров, что ты сэкономил компании 10 000 долларов, когда они потеряют репутацию у клиентов из-за сбоя. Это был урок, который показал мне, что у моих решений более далеко идущие последствия, чем просто наличие работающего функционала.

## 1.4.1. Первые 100 дней

Типичный вопрос, который задают на собеседованиях на руководящие должности, чтобы понять, насколько кандидат думающий и мыслит ли он стратегически: «Что вы сделаете в первые 100 дней работы?» «100 дней» — отличный инструмент для начинающих технических директоров, дающий важные идеи о том, что им делать и как справляться с новой ролью. Опытные технические директора с приходом в новую компанию тоже будут делать все это, прежде чем приступить к работе.

В следующем разделе мы узнаем, как начать приносить пользу еще до наступления 100-го дня, не принимая при этом необдуманных/реакционных решений. Как инженер, не поддавайтесь искушению выносить суждения (по крайней мере, вслух) и засучив рукава все менять. Инженеры любят брать быка за рога — они видят проблему и хотят ее решить. Боритесь с этим желанием.

### 0–7 ДНЕЙ

Первую неделю необходимо посвятить знакомству с компанией с позиции клиента. Узнайте точно, чем занимается компания, какая терминология в ней используется, какая модель ценообразования применяется и какие клиенты наиболее важны. Не погружайтесь в детали — разберитесь, как работает компания, чем занимается каждый отдел и кто ключевые лица.

### 7–30 ДНЕЙ

Следующие несколько недель уделите знакомству с техническим отделом, а также подробному изучению работы каждого отдела компании. Технический отдел

будет (или, по крайней мере, должен) работать на автопилоте, пока вы набираете скорость. Вас могут привлекать к участию во встречах, но пока просто слушайте и наблюдайте, обращая внимание на то, как другие сотрудники воспринимают технический отдел и с какими типами людей вам предстоит взаимодействовать.



Понять окружение чрезвычайно важно. Какие проблемы стоят перед компанией и отделом? С какими болевыми точками клиентов может помочь справиться ваша новая команда? Важно также познакомиться с каждым членом команды. Изучив их опыт, достижения, проблемы и цели, вы сможете лучше понять, что вам предстоит. На выстраивание доверительных отношений потребуются время, но наблюдения за тем, как люди общаются между собой, дадут вам много пищи для размышлений.

### **ЗАМЕТКИ С ПОЛЕЙ**

#### *Поиск подсказок*

Я называю эту фазу «проигрывать в течение одного часа» после фильма 1994 года «Маверик» («*Maverick*») с Мелом Гибсоном. Его герой просит разрешения сесть за покерный стол и, видя, что другие игроки немного колеблются, убеждает их, что в течение первого часа будет только проигрывать, что и делает. А когда час проходит, он начинает играть по-настоящему. Когда он в итоге выигрывает, один из мужчин обвиняет его в мошенничестве. Персонаж Гибсона объясняет: «Как вы думаете, что я делал в первый час? Я изучал вас». Технический директор — это марафонец, а не спринтер. У него есть время, чтобы освоиться.

Попутно вы узнаете рабочие процессы, разберетесь, какие технологии для каких продуктов используются и кто за что отвечает. Знакомство с планируемыми и реализуемыми проектами поможет вам понять порядок обсуждения и принятия решений и представить масштабы инициатив.

### **30–70 ДНЕЙ**

Следующий этап — более глубокое погружение, получение доступа к некоторым системам и их самостоятельное изучение. Если это собственная разработка,

посмотрите на исходный код, чтобы оценить его качество. Если компания занимается данными, изучите различные хранилища данных, чтобы увидеть, какую структуру они имеют и как организована работа с ними. Проверка серверов, то есть инфраструктуры, покажет вам состояние эксплуатации. Короче — загляните под капот и посмотрите, в каком состоянии находятся области, за которые вы отвечаете.

На этом этапе вы узнаете многое о состоянии и возможностях команды. Например, вы удивитесь, насколько часто оказывается, что для ключевой части системы нет специалистов ни внутри компании, ни у подрядчиков. Изучая каждую область, записывайте любые простые шаги или быстрые решения, которые можно принять, чтобы в корне изменить ситуацию.

Кроме того, полезно познакомиться с постоянными клиентами — это отличный способ взглянуть на положение дел со стороны. Совместный обед или краткий визит к клиенту дадут много информации, которая может быть еще не известна компании.

К концу этого этапа вы начнете составлять краткосрочные планы, а также сформируете примерное видение того, что необходимо сделать для долгосрочного успеха. Вы можете просмотреть текущие проекты и внести небольшие коррективы на основании того, что вы узнали.

### **70–100 ДНЕЙ**

На этом этапе вы приступите к непосредственному руководству простыми вещами, например к внедрению недостающих процессов или структурированию областей, которые не получали достаточного внимания. Ваша цель — определить компоненты, ключевые для поддержания работы компании, и понять, что необходимо для их укрепления, чтобы любые крупные инициативы, которые вы определяете как требующие дальнейшего изучения, не отвлекли вас от обеспечения работы бизнеса.

Например, вы можете найти компонент, на который не обращали внимания и который устарел во всех смыслах и нуждается в модернизации, потому что либо это ПО больше не поддерживается, либо трудно найти ресурсы для управления им. Хотя заняться новыми интересными функциями очень заманчиво, жизненно важно сохранить работоспособность старых скучных модулей, потому что именно они обеспечивают всем зарплату.

К этому времени вы уже хорошо будете разбираться, что есть что, несколько раз встретитесь с генеральным директором и поймете его видение компании. Вы начнете детализировать и свое видение, но сначала вам нужно успешно завершить несколько проектов.

К сотому дню вы должны иметь хорошее представление о состоянии отдела, внести несколько небольших улучшений/изменений и понимать, куда движется компания. Теперь вы можете начать управлять отделом и выдавать результаты.



## 1.5. 10 ГЛАВНЫХ КАЧЕСТВ ТЕХНИЧЕСКОГО ДИРЕКТОРА

Широко распространено заблуждение, что СТО должен иметь опыт разработки — уметь разобраться в шестеренках платформы. Это не обязательно. Хотя большинство технических директоров действительно имеют опыт разработки или исследований в сфере информационных технологий, многие из них в первую очередь менеджеры и хорошо понимают, как управлять процессами и достигать поставленных целей. Хороший технический директор:

- Ведет за собой и вдохновляет, побуждая других тянуться за собой.
- Заглядывает далеко вперед, прежде чем принимать решения.
- Общается уверенно, но не покровительственно.
- Прислушивается к генеральному директору и совету директоров, чтобы понять их желания и цели.
- Следит за наиболее актуальными для отрасли и компании технологиями.
- Не боится потерпеть неудачу и в случае неудачи не показывает ни на кого пальцем.
- Не пытается делать все сам, а делегирует полномочия команде.
- Знает: тот факт, что он начальник, не делает его самым умным в команде.
- Не боится принимать хорошие идеи и доверять им, независимо от их источника.
- Имеет видение, которое можно донести до других и реализовать.

Техническому директору нужны те же навыки, что и любому другому бизнес-лидеру, чтобы мотивировать и вдохновлять команду на максимальную отдачу. Важно, чтобы эта книга была не единственной на вашей полке — чтобы развиваться как руководитель, у вас должны быть минимум одна или две книги по общему лидерству.

На СТО лежит большая ответственность. Им приходится работать в режиме многозадачности, на уровне, намного превышающем уровень ведущего инженера или даже руководителя проекта. Лучших технических директоров отличает понимание этой разницы.

## ИТОГИ

- Технический директор — это тот, кто заботится о текущем и долгосрочном состоянии платформы.
- В компаниях, находящихся на ранних стадиях жизненного цикла, технический директор — это скорее практик, выполняющий все функции, потому что именно так можно быстро расти.

- По мере роста компании технический директор сосредотачивается на руководстве/архитектуре, чаще делегируя задачи и больше общаясь с внешними партнерами.
- В устоявшейся компании роль технического директора продолжает развиваться — он занимается укреплением технологии, чтобы обеспечить рост и масштабируемость, а также решением проблем устаревания.
- Мы составили контрольный список вопросов для компаний, в которых нет штатного технического директора, чтобы определить, не пора ли вводить эту должность.
- Первые 100 дней в должности технический директор определяет области, нуждающиеся в руководстве и вдумчивом анализе, углубляясь в поиск предстоящих задач и проводя собственный *due diligence*.
- Несмотря на то что отличному техническому директору нужны многие качества, мы определили 10 из них, которые показывают, что он по максимуму использует потенциал своей команды и создает среду, в которой работа спорится.

## **ПРОВЕРЬТЕ СЕБЯ**

Знаете ли вы...

- В чем состоит роль технического директора?
- Какова разница между техническим директором и старшим инженером?
- Каким образом техническому директору определить, как ему принести пользу или выстроить свою работу в организации?

# Взаимодействие с руководством и коллегами

---

## **В этой главе**

- ✓ Партнерство и выстраивание отношений с генеральным директором
- ✓ Эффективное взаимодействие с финансовым директором
- ✓ Работа с коллегами внутри компании
- ✓ Принимаем дела у предыдущего технического директора
- ✓ Представление членам совета директоров
- ✓ Работа с внутренней политикой
- ✓ Поиск путей внедрения изменений в компании

Вы замечали, что инженерам в целом очень комфортно общаться с инженерами-единомышленниками и они с радостью обмениваются жаргонными словечками? Действительно, «рыбак рыбака видит издалека». Подобную коммуникацию можно считать родственной — общение происходит с близкими по духу людьми, на своем языке, без всяких усилий. Если же мы обратим свой взгляд вовне, то столкнемся с чуждым нам миром, в котором люди говорят на другом языке и конкурируют между собой.

Новички, приходящие на позицию технического директора, часто не готовы к работе в этом мире. Они попадают в среду, где одна неуместная фраза может ввергнуть всю команду в панику, потому что ее истолковали совершенно превратно. Слова технического директора имеют высокий вес, и нахождение общего языка с другими людьми в компании — один из ключей к успеху.

Необходимость общаться с коллегами, начальством и правлением вызывает у технических специалистов большое беспокойство. Однако достаточно понять, что секрет коммуникации на любом уровне заключается в эмпатии — знании человека, с которым вы общаетесь, его мотивов, желаний и забот — и тогда установить диалог станет значительно легче.

В этой главе мы опишем основные заинтересованные стороны, с которыми вам потребуется налаживать контакт, и то, какими способами можно успешно взаимодействовать с каждой из них. Если вы похожи на меня, поначалу вам будет сложно и придется принуждать себя говорить чаще и авторитетнее. Это требует практики и, что немаловажно, уверенности в себе.

## 2.1. ПАРТНЕРСТВО С ГЕНЕРАЛЬНЫМ ДИРЕКТОРОМ

Вам нужно поддерживать отношения со многими людьми, но один из самых важных — ваш генеральный директор (CEO) либо другой непосредственный руководитель, если вы подчиняетесь операционному директору (COO) или директору по информационным технологиям (CIO). Без исключительного доверия и уважительного партнерства с генеральным директором вы не сможете достичь целей и реализовать стратегию компании, как бы ни старались. Это не обычная модель «руководитель — подчиненный» — здесь цена неудачи намного выше. В этом разделе описаны возможные типы гендиректоров и проблемы взаимодействия с каждым из них, а также инструменты для налаживания отношений.

### МОЙ ДИРЕКТОР НЕ ПОНИМАЕТ

Работая в сфере частных инвестиций в качестве наставника технических директоров, я часто слышу слова «Мой генеральный директор просто не понимает». В таких случаях я выступаю дипломатом-посредником в достижении взаимопонимания двух сторон, организуя их встречу и налаживая плодотворный диалог. Подобная ситуация возникает из-за неспособности донести информацию на уровне восприятия генерального директора. Мы, технари, склонны слишком быстро углубляться в подробности. Но слушателя они могут не интересовать. Все, что его волнует, это результат или последствия обсуждаемого события: как оно повлияет на клиента?

### 2.1.1. Офис генерального директора

Прежде всего, перестаньте думать о генеральном директоре как об одном человеке. Представьте его как целый офис. Все, что вы ему скажете, понадобится ему, когда какой-то из отделов обратится к нему за советом. Такой подход может показаться искусственным и надуманным, но ваша задача здесь — сформировать соответствующее мышление для взаимодействия с этим офисом. Вы здесь, чтобы

внести свой вклад в его эффективность и результат работы. Считайте, что когда вы сообщаете генеральному директору определенную информацию, то для него она представляет набор фраз, используя которые он должен иметь возможность рассказать другим о том же событии или факте, но на их языке.

### ОТСУТСТВУЮЩИЙ ПОЧТОВЫЙ ЯЩИК

Однажды у меня внезапно отказал ключевой сторонний API, что негативно сказалось на клиентах. Проблема была на стороне поставщика — они внесли изменения в записи DNS, в результате доменные имена стали вести на неверный адрес. Зная, что проблему придется объяснять многим людям с разным опытом, я сказал генеральному директору, что партнеры изменили свой почтовый ящик и случайно ошиблись цифрой. Поэтому все отправления уходили в неизвестный ящик и, пока они исправляли цифры, отправки задерживались. Так я сообщил все, что было нужно, и генеральный сразу понял проблему. Подробности работы с DNS и настроек времени кэширования записей ничего бы не дали. Затем я увидел, как та же метафора использовалась нашей командой по работе с клиентами и отделом продаж, и, чтобы узнать, как идут дела, меня просто спрашивали: «У почтового ящика уже правильный номер?»

В зависимости от размера/типа компании генеральный директор необязательно должен находиться на вершине ее организационной схемы. Возможно, он отчитывается перед советом директоров, партнерами или внешними инвесторами. Каждый из них в определенной степени заинтересован в результатах компании. Каждый из них будет запрашивать у сотрудников уровня ниже информацию об актуальном положении дел. Это экосистема, которую вы поддерживаете, ведя коммуникацию с офисом генерального директора.



Объясняя сложные вопросы, старайтесь использовать метафоры в разговоре и с теми, кто разбирается в технологиях, и с теми, кто от них далек. Иногда метафора может избавить технического специалиста от излишнего копания в деталях. Тот же, кто далек от технологий, сможет почувствовать в первую очередь удовлетворение от причастности к решению технической проблемы, а не разочарование от незнания предмета. Вы будете удивлены, узнав, как часто какая-то действительно полезная информация исходит от тех, кто, казалось бы, совершенно не связан с темой.

Мы, инженеры и «ботаники», любим детали. Мы любим окружать себя цифрами и точными фактами. Эффективнее всего мы работаем, когда нет никакой неопределенности. Люди бизнеса, напротив, живут в абстрактном мире, где при заданном наборе исходных данных результат заранее неизвестен. Когда каждый из нас использует свой привычный стиль общения, мы обвиняем бизнес в том, что он не сообщил нам достаточно подробностей для выполнения работы, а бизнес возражает, что, наоборот, это мы на самом деле не поняли сути того, о чем они просили. И каждый по-своему прав.

Чтобы добиться успеха, необходимо понять мир генерального директора, посмотреть на него как на партнера и сообщить ему информацию в том виде, который позволит ему принять эффективное решение. Далее мы рассмотрим типы генеральных директоров, с которыми вы можете встретиться.

### 2.1.2. Типы генеральных директоров

Быть генеральным директором — тяжелая работа. В условиях давления с разных сторон ему необходимо организовать людей для совместной работы над продуктом или услугой, чтобы обеспечить нужный результат для клиентов, стабильность для сотрудников и, наконец, прибыль для инвесторов.

Представьте компанию как оркестр: ваша секция струнных инструментов очень важна, но в создании гармоничного звучания участвуют многие другие секции, а генеральный директор дирижирует и координирует всю симфонию. Когда оркестр сыгран, его приятно слушать, хотя бывает и так, что генеральный директор просто размахивает руками, сбивая всех с толку, пока каждый пытается понять, что от него хотят. Давайте составим классификацию руководителей на основе их профессионального опыта.

#### **ГЕНЕРАЛЬНЫЙ ДИРЕКТОР БЕЗ ТЕХНИЧЕСКОГО ОПЫТА**

Чаще всего генеральные директора имеют традиционный управленческий или финансовый опыт, но не технический. Они имеют представление о технологии с точки зрения конечного потребителя, но не разбираются в подробностях так, как вы. Это совершенно нормально, потому что именно для этого в первую очередь вы и нужны: взять на себя все нюансы и головную боль, чтобы генеральный мог сосредоточиться на бизнесе.

Многие СТО решают игнорировать или недооценивают этот тип генеральных директоров, так как считают, что если генеральный директор не разбирается в технологии и, возможно, сделал пару ошибок, пытаясь что-то описать, он не заслуживает внимания технического директора. Но это самый распространенный тип генеральных директоров, и к нему не следует относиться легкомысленно.

**ГЕНЕРАЛЬНЫЙ ДИРЕКТОР БЕЗ ТЕХНИЧЕСКОГО ОПЫТА, ДУМАЮЩИЙ, ЧТО РАЗБИРАЕТСЯ В ТЕХНОЛОГИЯХ**

Следующий тип генеральных директоров (после тех, кто, по крайней мере, осознает, что им не хватает знаний) — это генеральный директор, который думает, что владеет технологией, и уверенно сыплет модными словечками, что для неискушенного уха звучит правдоподобно. Вы, вероятно, встречали этот тип людей, которые кивают с умным видом, вводя вас в заблуждение и заставляя думать, что они «в теме», тем самым создавая вам ложное чувство уверенности, но на самом деле не имеют представления, о чем вы говорите.

**ВЫ НЕ В ТРЕНДЕ**

Я вспоминаю случай, когда я встретился с генеральным директором одной компании, узнав, что он увлекся технологией искусственного интеллекта (ИИ) и объявил ее частью своего продукта. Я чувствовал подвох и вскоре выяснил, что он прочитал об ИИ в газете и думал, что они уже это делают, но ИИ — это не объединение двух таблиц базы данных. Он раскрутил продажи, и все были в восторге от того, что оказались на острие инноваций. По секрету я сообщил ему, что на самом деле ИИ там и не пахло. Мне было неловко, что я спустил его с небес на землю, но он был благодарен мне за то, что я осторожно направил его на истинный путь, не подняв шумихи и не заставив его выглядеть глупо в глазах своей команды.

**ГЕНЕРАЛЬНЫЙ ДИРЕКТОР С ТЕХНИЧЕСКИМ ОПЫТОМ**

Неужели! Наконец — хоть кто-то говорит на вашем языке... но постой, Белоснежка, это яблоко может быть отравлено. Этот генеральный директор понимает, что вы делаете, но, учитывая, что он больше не работает в технической сфере, он может забыть о каких-то сложностях и отбросить их как неважные или тривиальные.

Он также может ставить под сомнение каждое решение, которое вы принимаете, от архитектуры и выбора серверов для продакшена до системы деплоя и найма сотрудников. Задавать вопросы и оспаривать решения полезно, но если все, что вы делаете, подвергается сомнению, это начинает душить. Это не его вина — он смотрит с комфортной для себя позиции, которую знает как свои пять пальцев, и думает, что на самом деле помогает вам. Хороший генеральный директор такого типа поддержит вас, поднимет на новый уровень, поможет в профессиональном развитии и сделает вас лучшим техническим директором.

### 2.1.3. Найти подход

Независимо от того, к какому типу относится ваш генеральный директор, вам необходимо наладить с ним сотрудничество так, чтобы это принесло пользу и вам, и ему, и компании, и клиентам. Лучший способ сделать это — вести его за собой так же, как он ведет вас.



#### УЗНАЙТЕ СВОЕГО ГЕНЕРАЛЬНОГО ДИРЕКТОРА

Прежде всего узнайте, что за человек ваш генеральный директор. Это особенно важно, если вы пришли в новую для себя компанию:

- Узнайте его предыдущий опыт и образование (в Google, на LinkedIn и т. д.).
- Что его мотивирует, что приносит ему удовлетворение/разочарование?
- Что для него значит успех?
- Предпочитает ли он заниматься микроменеджментом, погружаться в детали или получать комплексное представление?
- Поговорите с ним.

Основную информацию можно узнать из разговора с директором и от коллег. Если вы новичок в компании, вполне уместно задать прямой вопрос: «Есть ли что-нибудь, что я должен знать о человеке X, что поможет мне в работе?» Это продемонстрирует, что вы готовы приложить усилия. Знание личности генерального директора поможет вам более эффективно общаться с ним.

#### ДЕТАЛИ ВАЖНЫ

Я вспоминаю одного генерального директора, которого заботило в первую очередь правильное написание названия компании. По его мнению, если человек не удосужился уделить достаточно внимания даже этому, то как можно доверить ему работу с клиентами? Это может показаться смешным, но в то же время это говорит о его внимании к деталям, и как только новый технический директор, которому мы помогали, понял это, он изменил свой стиль общения и стал постоянно демонстрировать, что детали важны и для него.

#### ДОВЕРЕННЫЙ КРУГ ГЕНЕРАЛЬНОГО ДИРЕКТОРА

Люди — не одинокие островки в океане. У каждого из нас есть ближайшее окружение, к которому можно обратиться за помощью, и генеральный директор не



исключение. У него в компании есть свои доверенные советники, к которым он может обратиться за оценкой какой-либо идеи. У них в свою очередь также могут быть помощники, которым они полностью доверяют выполнение различных поручений. Важно узнать, кто входит в эту сеть и насколько к их мнению прислушивается генеральный директор. Заручившись поддержкой этих людей, вам будет легче получить одобрение в будущем, поскольку на них вы сможете проверить, верно ли вы подаете ваши предложения.

### **КТО ДЕЛАЕТ ПОГОДУ**

СТО одной из компаний, с которыми работал инвестиционный фонд, испытывал сложности с утверждением своих планов. Он обратился ко мне за советом. Покопавшись, я обнаружил, что необходимо было убедить не генерального директора, а финансового, на которого генеральный, вероятно, очень сильно полагался. СТО не разобрался в тонкостях взаимоотношений в управленческой команде и не понял, что власть принадлежит финансовому директору, с которым генеральный всегда советуется. Как только фокус внимания был перенесен на финансового директора, техническому директору стало проще добиваться успеха. Позже в этой главе мы поговорим о взаимодействии с финансовым директором.

Узнайте, кто на самом деле принимает решения в компании. Создавайте союзы.

### **ЧТО ЕМУ НУЖНО?**

Иногда лучше всего задать самый простой вопрос: что ему нужно? Сядьте рядом с генеральным директором и спросите его прямо, что вы можете сделать, чтобы добиться успеха. Попросите его привести пример желаемой детализации, о каких событиях необходимо ему сообщать, что не дает ему спать по ночам и так далее.

Это не должен быть разовый разговор. Эту тему необходимо обсуждать регулярно (по крайней мере, каждый квартал), чтобы убедиться, что вы делаете то, чего от вас ожидают. С каждой такой встречей у вас будет новая пища для размышлений. Вы также будете уверены, что контролируете задачи своего отдела.

Кедрик Гриффин (Kedric Griffin), старший директор по разработке в ЕАВ, прекрасно подметил: «Доверие требует времени, уверенности в человеке и возможности положиться на него. Слишком часто я видел, как технарь старался одержать “быструю победу”, выдавая мнение за факты. Достаточно один раз попасться на этом, чтобы потерять все приобретенное доверие».

### **РЕГУЛЯРНЫЕ ИНДИВИДУАЛЬНЫЕ ВСТРЕЧИ**

Вы, скорее всего, будете встречаться с генеральным директором на общих собраниях руководства, но важно также проводить с ним индивидуальные (one-on-one) встречи. Они должны стать регулярными и служить средством информирования генерального директора о ваших глобальных планах и подходах, которые выходят

за рамки повседневной работы. Используйте эти встречи, чтобы лучше понять долгосрочную стратегию генерального директора и, если нужно, скорректировать что-то в ваших действиях, чтобы приспособиться к ней. Эти беседы могут быть чрезвычайно содержательными и поспособствуют достижению согласия между вами. Встречу можно начать с краткой информации о текущих проектах и с конкретных вопросов, по которым вам нужно узнать его мнение, а затем генеральный директор может поделиться с вами любой информацией, которая будет полезна для планирования вашей работы.

### 2.1.4. Как наладить успешное партнерство



Волшебной формулы идеальных отношений не существует. Чтобы стать хорошим партнером для генерального директора и организации, потребуется усердно потрудиться, экспериментировать и проявлять терпение. Тем не менее некоторые приемы немного облегчат вашу задачу.

#### **ОПИРАЙТЕСЬ НА ДАННЫЕ**

Эмоции, чувства и интуиция субъективны. Данные опровергнуть гораздо труднее. Любое решение, которое вы представляете генеральному директору, должно быть основано на данных. Если вам в команду нужен еще один сотрудник, покажите, насколько увеличится

производительность с его приходом. Если вам необходимо получить одобрение проекта, продемонстрируйте, как он повлияет на финансовую сторону бизнеса, например, сколько денег он позволит сэкономить на поддержке или лицензиях. Ведите разговор с опорой на данные.

#### **ПОДГОТОВЬТЕ ГЕНЕРАЛЬНОГО ДИРЕКТОРА К СЛОЖНОМУ РЕШЕНИЮ**

Рано или поздно вам придется принимать решение, которое может иметь большее влияние, чем вам хотелось бы. Вы рассмотрели множество вариантов, но идеального среди них нет. Изучив их все, изложите генеральному директору последствия каждого из возможных путей и подведите его к решению, чтобы он участвовал в его принятии вместе с вами. Он может указать новые детали, позволяющие обойти препятствие, или же согласиться с вами в том, что необходимо сделать сложный выбор. Это момент «выбора Софи», названный так в честь фильма 1982 года с Мерил Стрип, в котором героине пришлось выбирать,

какого ребенка спасти. Ответственность за подобные важные решения не должна лежать только на вас.

### ***ДУМАЙТЕ МЕДЛЕННО, ДЕЙСТВУЙТЕ БЫСТРО***

Люди по своей натуре склонны помогать и идти навстречу, что в контексте бизнеса означает соглашаться на просьбы. Это приносит нам удовлетворение. Мы быстро говорим «да», особенно в ситуациях, когда чувствуем повышенное давление коллектива, например на крупном собрании. Подавляйте это желание. Примите все запросы к сведению и скажите, что постараетесь ответить на них как можно скорее. Вам не нужно слишком долго размышлять, но вам требуется пространство, чтобы обдумать все варианты, прежде чем принять решение или дать рекомендации. Подобная практика избавит вас от перегрузки обещаниями сделать то, чего, как вы знаете, вы не сможете выполнить.

Если решение принято, действуйте быстро. Не медлите и не сомневайтесь. За время, которое вы потратили на размышления и обдумывание решения, вы должны были убедиться в его обоснованности.

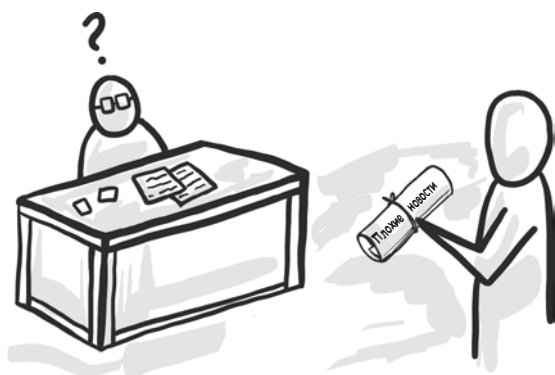
### ***ДАЙТЕ ГЕНДИРЕКТОРУ ВОЗМОЖНОСТЬ УЧАСТВОВАТЬ / ВНОСИТЬ ИЗМЕНЕНИЯ***

Никому не нравится чувствовать, что у него нет выбора. Если у предложения, с которым вы пришли к генеральному директору, есть только два варианта — да или нет, — то вы уже заранее наполовину сократили свои шансы на успех. Если вы предложите несколько вариантов, предполагающих участие и контроль со стороны генерального директора, он, скорее всего, с большей готовностью рассмотрит ваше предложение.

Частая ошибка — подача в стиле «все или ничего». Используйте поэтапный, инкрементный подход, демонстрируя промежуточную выгоду для компании. Это даст генеральному директору пространство для маневра, избавив от необходимости подписываться на слишком многое слишком быстро, еще до полного понимания всей ситуации, о которой вы говорите. Сделайте варианты выбора максимально соответствующими текущему положению вещей. Если вы представите слишком мало вариантов или они не будут относиться к делу, эффект будет противоположен желаемому.

### ***ПРОБЛЕМЫ БЫВАЮТ НЕ ОТ ПЛОХИХ НОВОСТЕЙ, А ОТ НЕПРАВИЛЬНОЙ РЕАКЦИИ НА НИХ***

Вам когда-нибудь придется сообщать плохие новости — система не работает, база данных удалена (что-о-о?) и т. д. Это будет трудный разговор, и вам будет некомфортно. Первое, что вы должны сделать, — взять на себя ответственность. Если проблема возникла под вашим руководством, ответственность не сете вы. Признайте проблему, извлеките из нее уроки и примите необходимые меры, чтобы она не повторилась. Во-вторых, сообщайте новости без эмоций



и обвинений. Хотя соблазн велик (и это случается чаще, чем следовало бы), не называйте никого из сотрудников, кто может быть виновен в проблеме. Мы подробнее поговорим об этом в одной из следующих глав: вы должны действовать как одна команда.

О каких плохих новостях следует сообщать генеральному директору? Допустим, вы еще не выяснили это у самого

директора, тогда стоит придерживаться правила, согласно которому все, что влияет на клиентов или выручку компании, должно быть немедленно доведено до его сведения. Его необходимо поставить в известность как можно скорее. Никогда не пытайтесь скрыть или замаскировать потенциальную проблему. Ваш худший кошмар — когда гендиректор приходит к вам с проблемой, хотя вы уже знаете о ней. Это означает, что он сам обнаружил ее и вы только что подорвали его доверие и расположение к вам, скрыв информацию. Хороший пример — нестабильная работа клиентского сайта. Не ждите — немедленно сообщите об этом генеральному директору (и остальным сотрудникам компании) и дайте им знать, что вы уже занимаетесь проблемой.

Поговорив об установлении партнерских отношений с генеральным директором, обратим внимание на финансового директора (CFO).

## 2.2. СТАВКА НА ФИНАНСОВОГО ДИРЕКТОРА

Вы, возможно, думаете, что управление затратами и расходами не входит в компетенцию вашего отдела — это забота финансового директора или контролера. Хотя вы можете не иметь доступа ко всей финансовой информации компании (например, сведения о зарплатах обычно конфиденциальны), полная оторванность от финансовых показателей своего отдела говорит о несерьезном отношении к вашим обязанностям и не принесет вам ничего хорошего. В этом разделе расскажем, почему вам необходимо уделять внимание финансам и как использовать финансовую информацию для принятия более обоснованных и правильных решений.

Печальная реальность такова, что ваш отдел, скорее всего, — это центр затрат: он не приносит прямого дохода. Да, безусловно, он очень важен, и без него компании, вероятно, не существовало бы. Но в данном случае это не имеет значения. С точки зрения бухгалтерии на вас тратятся деньги компании. Лучшее, на что вы можете надеяться, — что их тратится не слишком много. Поэтому для каждого доллара, который вы просите, важно показать, что этот доллар принесет компании взамен.

Звучит пугающе, но это не так сложно, как кажется, если у вас хорошие рабочие отношения с финансовым директором. В конце концов СФО несет ответственность за предоставление всех цифр не только генеральному директору и совету директоров, но и государству, потому что он ежегодно готовит отчетность. Он должен быть в состоянии защитить все цифры и обосновать все доходы и расходы компании. Понимание его целей поможет вам их учитывать и тем самым облегчить ему жизнь, потому что он будет знать, что может доверять тому, как вы управляете своим бюджетом.

### 2.2.1. Как контролировать свои расходы

Первое, что вам нужно сделать, особенно если вы приняли новый отдел или недавно присоединились к компании, — выяснить скорость «сжигания» бюджета, то есть сколько обычно отдел тратит ежемесячно и ежегодно при нормальной работе. Одного числа тут недостаточно — нужно дополнить его контекстом таким образом, чтобы безболезненно обосновать эти расходы бизнесу.



Разработка



Эксплуатация



Поддержка

Эффективная рабочая стратегия — разделить расходы на три части следующим образом:

- Разработка
- Эксплуатация
- Поддержка

В зависимости от компании также может иметь смысл разбить эти категории на линейки продуктов или группы, понятные бизнесу. Вам не нужен бухгалтер — подойдет простая электронная таблица, в которой перечислены все ваши расходы. Каждый раз, приобретая лицензию или подписку, записывайте приобретение и его ежемесячную стоимость в таблицу. Разберем каждую категорию.

#### РАЗРАБОТКА

Сюда входит все необходимое для производства того, что компания считает «продуктом», например:

- Расходы на персонал (зарплата и компьютеры).

- Лицензии на инструменты для разработки (IDE, инструменты управления проектами).
- Лицензии на ПО (базы данных, библиотеки).
- Оборудование / облако для разработки / тестирования.
- Сторонние поставщики, например подрядчики.
- Обучение (курсы, книги, конференции).

В расходы на персонал обычно включается только годовая заработная плата сотрудников, но лучше включать в бюджет 120–130 % зарплаты, чтобы учесть дополнительные расходы (например, льготы, оборудование, мебель и кофе).

#### КАК ПРОЩЕ ПЛАНИРОВАТЬ БЮДЖЕТ

Рассчитайте ежедневные расходы на каждую должность. Например: допустим, средняя зарплата веб-разработчика составляет 100 000 долларов (добавьте 20% накладных расходов — 120 000 долларов), что составляет 10 000 долларов в месяц или 500 долларов в день (при условии, что в месяце 20 рабочих дней). Если для будущего проекта потребуется 14 дней работы этого специалиста, вам не придется выяснять или угадывать стоимость — это будет стоить  $14 \times 500$  долларов. Ваш бюджет только что стал намного более обоснованным.

Зная, во сколько обходится производство продукта, вы сможете эффективнее спланировать бюджет и будущий рост. Когда ваш генеральный директор спросит, можно ли сделать что-то быстрее, вы будете понимать, во сколько эта дополнительная скорость обойдется компании. Опять же, ваши решения должны основываться на данных, а не на ощущениях (ощущения должны подтверждать то, что говорят данные).

#### ЭКСПЛУАТАЦИЯ

Сюда входит все, что связано с предоставлением услуги клиенту, в том числе:

- Расходы на персонал (настройка конфигурации клиента или ввод клиентских данных).
- Стоимость оборудования/облака.
- Лицензии на программное обеспечение (базы данных, библиотеки).
- Сторонние сервисы (домены, API).
- Аудиты безопасности.

Популярные облачные сервисы, такие как AWS и Azure, упрощают маркировку ресурсов, что позволяет разделить затраты на эксплуатацию и разработку. Точно так же лицензии на ПО, используемое для обслуживания клиентов, обычно отличаются от лицензий на разработку и могут стоить значительно дороже. Расходов на персонал здесь может не быть в зависимости от особенностей вашего проекта, но если для подключения нового клиента требуются трудозатраты, то их необходимо учитывать.

В идеале должно получиться число, зависящее от количества клиентов. Это поможет бизнесу точнее рассчитывать стоимость эксплуатации и потенциальные затраты, если они решат привлекать больше клиентов. Некоторые организации достигают такого уровня детализации, что могут точно сказать, во сколько им обходится обслуживание каждого клиента и какую часть инфраструктуры каждый клиент использует.

### ПОДДЕРЖКА

И, наконец, поддержка. Когда что-то идет не так или клиенту нужна помощь, сколько это стоит? Сколько усилий тратится на обслуживание системы, например на мониторинг файлов журналов и проверку резервных копий?

Эти расходы должны соответствовать расходам на эксплуатацию, поэтому, когда они увеличиваются, растут и расходы на поддержку. Затраты на службу поддержки необходимо по возможности рассчитывать на почасовой основе, чтобы понимать, сколько стоит обработка каждого типа обращений / инцидентов.

Поддержка многих устаревших систем стоит очень дорого из-за отсутствия в них современного функционала. Для одной системы мы обнаружили, что 60 % всех обращений в поддержку были запросами на сброс пароля, и, зная почасовую стоимость поддержки, было несложно рассчитать реальные расходы на них. При составлении бюджета само собой было принято решение о внедрении механизма сброса пароля, и оно окупилось в первый месяц работы. Бюджетирование мы рассмотрим далее в этой главе.

Службы поддержки не всегда отслеживают все данные, полезные для анализа. Очень часто из виду упускается, к какому клиенту (или линейке продуктов) относится заявка. Здесь могут выясниться очень интересные вещи, например, что какому-то клиенту требуется гораздо больше поддержки, чем остальным.

Учитывайте время и усилия, необходимые для обслуживания запросов клиентов. Эти параметры помогают понять, как улучшить ваши системы и упростить общение с финансовым директором, чтобы снизить и затраты, и количество вопросов. Пример простого учета ежемесячных расходов на эксплуатацию приведен в табл. 2.1.

**Таблица 2.1.** Примерные ежемесячные расходы

Категория	Ежемесячно
Затраты на AWS	\$15 000
DevOps (один человек)	\$2 000
Лицензии на ПО (JIRA и GitHub)	\$500
Поддержка (два человека по \$10 000 каждый)	\$20 000

### 2.2.2. Обратитесь за помощью

Область финансов обычно чужда новому техническому директору и поначалу может быть довольно пугающей. Наладьте диалог с финансовым директором и попросите его помочь вам. Возможно, вы видите не все счета, и будет правильно попросить показать их, ведь они имеют отношение к вашей команде. Только не спрашивайте, сколько получает генеральный директор!

После того как вы составите примерный список расходов отдела, просмотрите его с финансовым директором. Возможно, вы учли какие-то расходы дважды (например, лицензия Microsoft Office может быть частью общей лицензии на компанию и вам не нужно включать ее в свой бюджет) или что-то упустили.

Хороший CFO ценит, когда руководители отделов разбираются в своих расходах, потому что это облегчает его работу. Полезно периодически встречаться с ним для анализа списка расходов (например, раз в квартал), поскольку он подскажет вам точные цифры и вы оцените, насколько были к ним близки.

Секрет хороших отношений с финансовым директором — никаких сюрпризов. CFO несет ответственность за все расходы компании, а не только вашего отдела. От ваших цифр ему требуется предсказуемость и надежность, и если вы докажете, что на вас можно положиться, то при утверждении бюджета ваша жизнь станет намного проще.

Многие технические директора делают себе хуже из-за того, что не стремятся разобраться в затратах и бывают застигнуты врасплох в критический момент. При этом, если бы они обладали информацией, сложностей у них не возникло бы.

Если вы не знаете, с какой скоростью «сгорает» ваш бюджет, это сигнал для вас. Задумайтесь: вы упускаете много полезной информации.

## 2.3. РАБОТА С КОЛЛЕГАМИ

Насколько хорошо вы знаете отделы, из которых состоит компания? Поскольку технологиями пользуются повсеместно, технический директор должен быть в курсе, чем занимается каждый отдел.

Джеймс Хедли (James Headly), основатель и партнер MacLaurin Group и ментор ИТ-директоров, отмечает: «Технический директор должен хорошо разбираться в потребностях всех членов организации и тесно сотрудничать с владельцами бизнеса. Он должен не близоруко рассматривать ситуацию с точки зрения разработчика, а сотрудничать с бизнесом, продвигая всю организацию вперед».



В этом разделе рассматривается, насколько важно установить хорошие рабочие отношения с коллегами и узнать их ближе, что в конечном итоге поможет вам добиться успеха.

### 2.3.1. Знайте свою компанию

Для начала определим, что означает коллега в контексте этого раздела. Коллега (peer) — это сотрудник, находящийся на том же уровне в организации, что и технический директор. Если ваша компания не настолько формальна, то коллегами можно считать тех, кто присутствует на тех же собраниях, что и вы, и подчиняется генеральному директору. Этих людей и их отделы вы должны знать.

В общих чертах вы, вероятно, знаете, чем занимается отдел продаж, но понимаете ли вы, как он устроен, с какими проблемами он сталкивается, какие у него болевые точки и каковы его целевые показатели? Не обязательно знать каждую деталь, но вы должны хорошо представлять себе, что требуется для успешной работы отдела.

У генерального директора бывает любимый отдел или более тесные связи с каким-то из отделов. Обычно это связано с его зоной комфорта. Если генеральный директор пришел из сферы продаж, то он будет лучше понимать тех, кто занимается этой работой, и уделять им больше времени. Не обращайтесь на это внимания, потому что хороший генеральный директор одинаково ценит все отделы (в идеале, разумеется; это не всегда так, но считайте, что так оно и есть, пока не будет доказано обратное).

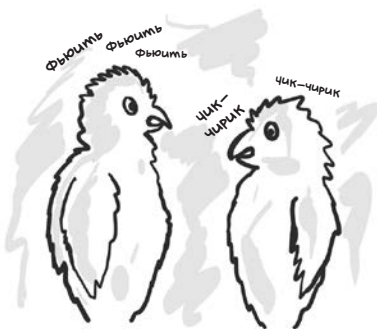
Вы руководите одним из немногих отделов (второй такой — это отдел кадров), от которого зависят другие подразделения компании. Клиент взаимодействует с технологией, за которую отвечаете именно вы, а цель каждого из отделов — так или иначе приносить пользу клиенту. Поэтому ваши успехи (и неудачи) напрямую влияют на то, что другие отделы могут сделать для клиентов.

#### НА ОБЕД

Один из советов, который я даю техническим директорам, заключается в том, чтобы обедать, хотя бы раз в месяц, с каждым из коллег. Познакомьтесь с ними лично. Вы будете удивлены, сколько всего можно узнать, когда вы вместе преломляете хлеб. Еще один интересный способ — пригласить руководителей прийти на одно из собраний вашего отдела и выступить там, чтобы дать вашим сотрудникам представление об их работе.

И чрезвычайно важно не только вам знать, как работают другие отделы, но и им тоже понимать вашу работу. Каждый человек в вашем отделе должен разделять это стремление и участвовать в установлении связей.

### 2.3.2. Говорите на их языке



Даже в небольшой компании сотрудники разных сфер бизнеса могут иметь свой лексикон для описания специфических понятий. Мы все знаем, как важно свести эти различия к минимуму, но разговорный язык имеет тенденцию развиваться. Как только кто-то один уверенно использует новое словечко на собрании, оно пополняет словарный запас сотрудников этого отдела и может просочиться за его пределы.

#### СЛОВА РАЗНЫЕ — СУТЬ ОДНА

Я помню, как в одной компании, где я работал, для обозначения клиента применялись два разных термина: идентификатор проекта (project ID) и идентификатор кампании (campaign ID). Разные отделы использовали какой-то один из этих терминов, утверждая, что ничего не знают о втором. Мне потребовалось несколько месяцев, прежде чем я понял, что оба они означают одно и то же — никакой разницы. Так что не бойтесь задавать простые вопросы.

Общий язык важен, но понимание различий может значительно облегчить взаимодействие с внешними отделами. Узнайте, какие фразы и выражения у них в ходу, и при общении используйте их язык, не навязывая свой собственный.

### 2.3.3. Найдите болевые точки

У каждого отдела есть свои внутренние рабочие процессы и бизнес-правила обслуживания клиентов. Иногда они не лучшие и не самые эффективные, но они известны и понятны сотрудникам.

По моему опыту, после того как вы разберетесь в том, как работает тот или иной отдел, вы можете предложить небольшие изменения, которые порой радикально улучшают их рабочий процесс. Это прекрасный способ навести мосты, добиться быстрых результатов и лучше понять функции отдела.

Ищите возможности помочь и поделиться знаниями, особенно в отношении процессов, в которые вовлечено большое количество участников и где велика вероятность ошибок. Если вы сможете устранить человеческий фактор, то от этого выиграют все. При этом дайте решениям развиваться постепенно; не перегружайте коллег незнакомыми им процессами и не нарушайте их работу.

**НЕЗНАЧИТЕЛЬНЫЕ ИЗМЕНЕНИЯ МОГУТ ИМЕТЬ БОЛЬШОЕ ЗНАЧЕНИЕ**

Иногда самое простое решение может оказаться самым эффективным. Одна из команд фиксировала процесс работы в электронной таблице, чтобы любой мог видеть текущее состояние каждого из клиентов. Эту таблицу рассылали по электронной почте, если кто-то вносил в нее изменения. Вы уже видите проблему — кто-то мог не заметить присланную свежую версию, внести данные в устаревшую и переслать ее.

Несмотря на соблазн кардинально изменить порядок работы (например, порекомендовать внедрить систему CRM), самым простым и эффективным краткосрочным решением стало перевести команду на Google Таблицы. Общая таблица решила самую большую проблему отдела и при этом осталась знакомой сотрудникам, поэтому их рабочий процесс не был нарушен. Не стоит недооценивать зависимость корпоративного мира от скромных электронных таблиц.

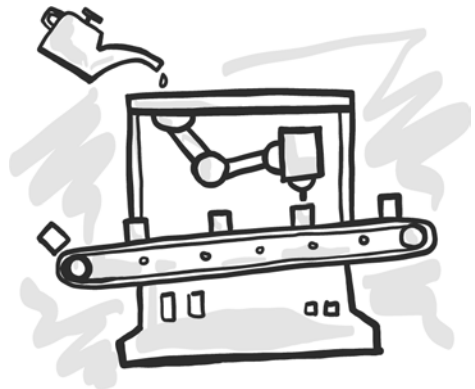
**2.3.4. Мыслите стратегически**

Узнав, как устроены все отделы, и наладив хорошие отношения с ними, начинайте совместную работу над повышением эффективности. Некоторые компании проводят регулярную реорганизацию, в ходе которой формируются новые отделы или сотрудники из одного перемещаются в другой — *обычно* генеральный директор таким образом пытается оптимизировать процессы для достижения лучшего результата. Но вы и ваши коллеги, ежедневно выполняющие работу, ради которой существует компания, лучше всех знаете, что будет полезно, а что нет.

Здесь не идет речь о том, что нужно захватывать территорию, чтобы сделать ваш отдел больше и могущественнее. Но вы можете внести небольшие рациональные изменения, чтобы улучшить рабочий процесс в целом. Вы можете обнаружить, что какие-то группы внутри одного из больших отделов будут работать эффективнее в другом месте. Подобные мини-отделы обычно с годами вырастали из одной функции и оставались на том же месте просто потому, что никто не думал их перемещать.

Обычно такой командой, постепенно образовавшейся из ниоткуда, является служба поддержки клиентов, части которой бываю разбросаны по разным отделам с разной специализацией. В результате обсуждений и переговоров часто выясняется, что эти мини-отделы можно объединить под единым руководством, чтобы лучше обслуживать клиентов.

Если ваши с коллегами предложения имеют смысл, их легко будет



продать генеральному директору, который, скорее всего, их одобрит. Хороший генеральный директор не будет мешать реорганизации, потому что он доверяет решениям своих менеджеров.

В хорошо работающем двигателе каждая деталь выполняет свою функцию, и ни одна из них не создает проблем другим. Как технологическому лидеру и стратегу в компании, для вас жизненно необходимо понимать компанию в целом и что конкретно каждый отдел делает для обслуживания клиентов. В противном случае вам будет сложнее создавать союзы и получать поддержку ваших планов.

## 2.4. ПРИНИМАЕМ ДЕЛА У ПРЕДЫДУЩЕГО ТЕХНИЧЕСКОГО ДИРЕКТОРА

Одна из самых сложных вещей в нашей работе — принимать дела у технического директора, работавшего до нас. Как говорится в старой поговорке: «Тот, кто был здесь до тебя, был идиотом, а тот, кто будет после, подумает, что идиотом был ты». Мы, технари, всегда скоры на суждения, особенно если что-то делается не так, как сделали бы мы. Встречается и обратное отношение: предыдущий технический директор был безупречен, но пришло его время расти и двигаться вперед. Не впадайте ни в ту, ни в другую крайность.

В этом разделе рассматривается, что стоит и чего не стоит делать, чтобы переход от одного руководства к другому — вашему — прошел успешно.

### 2.4.1. Цените, а не принижайте

Очевидно, но от этого не менее важно: когда вы начнете разбираться в наследстве предыдущего технического директора — никогда не относитесь пренебрежительно к его решениям и методам работы, как бы вам того ни хотелось. Не стоит становиться «директором по обвинениям», который мигом переводит стрелки, чтобы отвести критику от себя. Не беспокойтесь — все знают, что вы новичок и не несете ответственности за прошлое. Вам не нужно об этом напоминать.

Лучше отметьте достижения. В конце концов, независимо от того, что вы думаете о тех или иных решениях предыдущего СТО — он успешно управлял технологиями, и компания генерировала достаточно прибыли, чтобы предложить вам достойную зарплату. Для понимания каких-то решений вам будет не хватать исторического контекста: они были приняты в свое время и в своих условиях. Не спешите судить.

Иногда дело не в том, что предыдущий СТО не знал лучшего способа (хотя в большинстве случаев так и есть — *люди не знают того, чего не знают*). Возможно, их заставил пойти по данному пути генеральный директор, который считал, что знает лучше, а у технического директора не было навыков или данных, чтобы

убедить его в обратном. Вы никогда не узнаете истинную причину, поэтому нет смысла оглядываться назад, только вперед. Какие бы проблемы ни возникли, теперь их решать вам.

### ЗАМЕТКИ С ПОЛЕЙ

*Вы сделали... что?*

Я часто сталкивался с очень креативными решениями и, вместо того чтобы осуждать их, хвалил творчество и изобретательность их создателя, мысленно крича: «О чем ты вообще думал?» Для одной портфельной компании мы искали нового технического директора на смену тому, который работал со дня ее основания. Этот джентльмен не доверял стандартным системам контроля версий (CVS, SVN или Git). Что сделал он? Он создал и поддерживал свой собственный стандарт и заставлял команду использовать его. С одной стороны, я должен был оценить его изобретательность, но, покопавшись, я обнаружил, что это не более чем причудливое архивирование, а не контроль версий в современном представлении. Предложение перейти на Git я аргументировал возможностью использовать встроенный инструментарий в IDE, не поднимая шума и не привлекая внимания к тому, насколько слаб имеющийся «контроль версий». Таким образом все сохранили лицо. Не афишируйте свои победы и не ищите похвалы за то, что вы умнее. Просто сделайте свою работу и двигайтесь дальше.

С другой стороны, вы можете столкнуться с незнакомым вам решением и оценить его достоинства. Это прекрасная возможность поучиться, особенно если это решение работает и используется в продакшене. Вам не всегда достается в наследство беспорядок или неорганизованная команда.

Нам всем время от времени приходится переписывать что-то, потому что мы понятия не имеем, как это работает. Технический директор должен сопротивляться желанию что-то изменить только потому, что он этого не знает или привык работать по-другому. Не чините то, что не сломано.

### 2.4.2. Поговорите с уходящим СТО

Постарайтесь поговорить с уходящим техническим директором. Возможно, он увольняется или уходит на пенсию, но, если у вас есть шанс пообщаться, хватайтесь за него обеими руками. Это прекрасная возможность задать правильные вопросы, которые облегчат вам работу. Не задавайте вопросы, которые заставят прежнего СТО защищаться, потому что так вы ничему не научитесь, а только удовлетворите нездоровое любопытство или утвердите свое интеллектуальное превосходство. Вот о чем можно спросить:

- *Что не давало ему спать по ночам?* Ответ даст вам представление о текущих проблемах, доставляющих беспокойство и не имеющих немедленного решения, — обычно это вещи, о которых вы никогда бы не подумали.

- *Какое давление оказывал на него бизнес?* Например, есть ли повторяющиеся обстоятельства, о которых вы можете не знать и на которые вам нужно обратить особое внимание, или кто-то, кто будет сильно влиять на вашу работу.
- *Какой клиент самый важный?* У каждой компании есть особенный клиент, ради которого она готова на все. Зная, кто это, вы сможете не сломать сложившийся подход к работе с ним.
- *Обслуживание какого клиента требует больше всего ресурсов отдела?* А кроме того, какой клиент доставляет больше всего хлопот в виде трудозатрат или сложности его систем? Например, для него был разработан специальный функционал, который в свое время был хорошей идеей, но в дальнейшем стал источником головной боли.
- *На кого из вашей команды больше всего полагался СТО?* Вероятно, один из самых важных вопросов, он поможет вам понять, кто из сотрудников самый надежный. Когда вы будете оценивать команду, уделите особое внимание этому человеку или людям.

Важно отдать уходящему техническому директору дань уважения, которого он заслуживает (даже если он оставил после себя ужасный беспорядок, с которым вам теперь предстоит разбираться). Дайте ему понять, что вы уважаете его наследие. Если вам когда-нибудь понадобится помощь, он будет рад ответить на ваш звонок. Вот почему важно не принижать его, даже общаясь в доверенном, по вашему мнению, кругу. Вы не захотите, чтобы он узнал, что в лицо ему вы говорите одно, а команде — другое.

Да, это обычная порядочность, но слишком часто новый технический директор отсекает себе все возможности использовать исторический контекст, просто ляпнув что-то в безопасной, как ему казалось, обстановке. Не поддавайтесь этому искушению, даже если другие стремятся обвинить в проблемах вас.

### 2.4.3. Изучите команду

Технический директор хорош настолько, насколько хороша его команда, и, чтобы понять, насколько хорош был ваш предшественник, оцените команду, которую он собрал. Если технический директор работал долго и лично принимал всех на работу, у вас могут возникнуть проблемы с лояльностью. Команда может видеть в вас врага, вытеснившего их любимого лидера. Ключ к победе в этой моральной битве — не внедрять слишком много изменений за короткий срок. Продолжайте отмечать заслуги предыдущего технического директора, постепенно внося небольшие изменения. Команда будет постепенно привыкать к ним, пока не начнет удивляться, как раньше обходилась без вас. Успешные СТО способны, используя эту технику, изменить мышление всей команды и сохранить при этом уважение к предыдущему руководству.

Еще одна проблема, особенно распространенная в компаниях, возглавляемых их учредителями, где СТО работал с момента основания, заключается в том,

что члены команды, относительно слабые в техническом плане, зарабатывают намного больше, чем их коллеги в среднем, поскольку их вознаграждают за долгую работу, а не за достижения. Завышенные зарплаты рассматриваются в главе, посвященной приему на работу, поскольку это очень частая проблема. Что касается технической стороны, исходите из предположения, что сотрудники еще не полностью раскрыли свой потенциал, и разработайте программу, которая позволит им показать, на что они способны. Не списывайте их со счетов слишком быстро, дайте им возможность показать, что они достойны работать в вашей команде. Возможно, у них просто не было вдохновляющего лидера, который указал бы им путь.

В целом, приходя на место другого технического директора, не вносите слишком много изменений чересчур быстро и найдите время, чтобы изучить организацию работы и ключевых игроков и наладить контакты. Обычно процессы не требуют немедленного внимания, и у вас будет возможность оценить ситуацию и составить план будущих действий, который поможет вам оставить свой след и продвинуть компанию вперед.

## 2.5. ПРЕДСТАВЛЕНИЕ ЧЛЕНАМ ПРАВЛЕНИЯ

В зависимости от размера и этапа развития компании, в ней могут проводиться регулярные официальные заседания правления. Это не обычные собрания руководства компании или проекта, и к ним следует относиться по-особому. Это также не собрание руководителей высшего звена. В большинстве случаев на заседании правления из руководителей присутствуют только генеральный директор и финансовый директор, а остальные участники являются инвесторами компании.

В этом разделе мы рассмотрим, на что вам следует обратить внимание и как подготовиться, если вас пригласят на такое заседание. Эта перспектива может быть довольно пугающей, потому что на таких встречах принимаются долгосрочные решения с далеко идущими последствиями, и, следовательно, вам лучше не совершать ошибок.

### ОПРЕДЕЛЕНИЕ

*Заседание правления* — это официальное собрание директоров, инвесторов и приглашенных лиц, проводимое на периодической основе (обычно ежеквартально), на котором, согласно установленным процедурам, рассматриваются стратегические вопросы, основные инициативы и финансовое состояние компании. Эти заседания проходят по заранее подготовленному плану, также ведется официальный протокол, который просматривается в начале следующего заседания. Если необходимо голосование, то оно происходит именно здесь, а вес голоса каждого участника соответствует его доле в компании.

### 2.5.1. Узнайте свою аудиторию

Члены правления обычно не участвуют в повседневных делах компании, сосредотачиваясь вместо этого на долгосрочных целях и следя за тем, чтобы компания двигалась в правильном направлении. Обычно это крупные инвесторы или акционеры, представители инвестиционного



фонда (например, партнера по прямым инвестициям) и приглашенные со стороны участники, которые обладают опытом и глубокими знаниями предметной области, полезными для осуществления руководства.

Хотя эти люди могут быть знакомы друг с другом, чаще всего они собираются вместе только на таких заседаниях. Они не очень погружены в технологии (по крайней мере, не настолько, как вы), поскольку их основная задача — коммерческие вопросы. Получить информацию об этих людях несложно, потому что обычно детали их биографии описаны на странице «О нас» сайта компании или у них есть профиль в LinkedIn, поэтому вы можете найти интересующие вас сведения самостоятельно.

Кроме того, чтобы лучше понять каждого из них, поговорите с генеральным/финансовым директором. Кто из них молчалив и редко вступает в обсуждения? Кто вечно ворчит и ко всему цепляется? Что наиболее интересно каждому из них? У кого есть реальная власть принимать решения? Узнав людей, перед которыми вы будете выступать, вы сделаете свое участие продуктивнее и предотвратите ошибки новичка. Отнеситесь к подготовке к заседанию правления так же, как к собеседованию.

### 2.5.2. Формат встречи

Заседания правления официально протоколируются. Протокол этого заседания отличается от судебного, в котором фиксируется каждое слово, сказанное участниками. Вместо этого в нем перечисляются все принятые решения, кто за какой результат отвечает и что было решено по итогам голосований. В начале каждого заседания просматривается протокол предыдущего собрания. Каждое заседание имеет официальную повестку дня, порядок которой соблюдается, и в большинстве случаев она представляет собой презентацию, которая рассылается всем участникам за несколько дней до заседания.



Встреча может длиться пару часов, а иногда и несколько дней, включая совместный ужин. Если вас пригласят на нее, имейте в виду, что, скорее всего, вы не будете участвовать до конца. Вам предложат прийти в назначенное время, рассказать свою часть и покинуть заседание. Вас также могут пригласить на ужин или приветственные напитки — неформальную встречу, чтобы участники познакомились друг с другом.

Также вас могут пригласить на заседание от начала до конца. Ожидается, что в этом случае вы будете участвовать в обсуждениях. Не стоит проверять электронную почту или общаться в чате в то время, пока речь идет не про вашу зону ответственности. Вас попросили принять участие не просто так, поэтому дайте понять, что ваш вклад неоценим.

### ПРИМЕЧАНИЕ

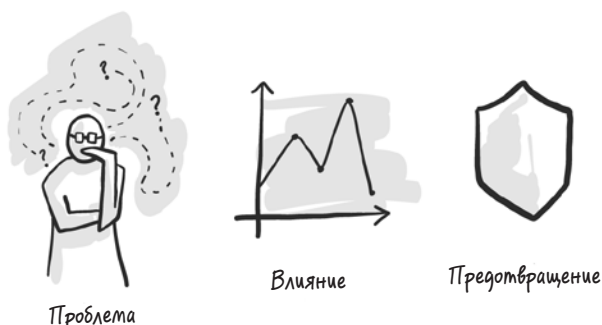
Девин Мэтьюз (Devin Mathews), партнер-основатель ParkerGale Capital, отмечает, что технический директор редко принимает активное участие в обсуждениях, но того, кто задает вопросы, оказывает поддержку и помогает коллегам, очень ценят и обычно приглашают вновь.

Генеральный директор подскажет, как себя вести, если планируется отступление от обычного формата. Справедливости ради надо сказать, что больше всего нервничает именно генеральный директор, потому что на этом заседании его оценивают те, кто может его заменить. Таким образом, все ваши действия будут отражаться на нем.

### 2.5.3. Ваша роль

Вас приглашают выступить перед правлением по определенной причине. Не нужно о ней гадать, а лучше подробно расспросить генерального директора. Технического директора приглашают нечасто (если только вы не работаете в исключительно технологической компании), а если приглашают, то по конкретному вопросу, обычно связанному с крупными техническими инициативами или изменениями в продукте. Ваша презентация должна быть верхнеуровневой, без лишних подробностей. Сосредоточьтесь на клиентах и бизнесе: все, о чем вы говорите, должно иметь отношение к ним.

Предоставьте контекст, напомните правлению, что должно было быть достигнуто к текущему моменту, и, если это не достигнуто, — объясните почему, а не просто пожимайте плечами. Вы не завоюете их сердца и уважение, если просто скажете, что все потребует больше времени и будет стоить дороже, потому что в IT всегда так бывает. Открыто и понятно объясните, почему дела идут успешно или не очень.



Вас также могут вызвать, если произошел серьезный сбой или проблема безопасности и требуются дополнительные разъяснения. Тон подобной презентации должен отличаться от обычного, потому что в этом случае вам, возможно, придется вызвать огонь на себя и принять ответственность, соответствующую вашей должности. Используйте формат PIP (problem — проблема, impact — влияние, prevention — предотвращение):

- *Проблема* — что вызвало проблему? Объясните в терминах бизнеса, понятных слушателям.
- *Влияние* — какое влияние проблема оказала на клиентов/сотрудников/партнеров?
- *Предотвращение* — какие выводы сделаны и что предпринято, чтобы проблема не повторилась?

О чем бы вас ни попросили рассказать, обсудите это подробно с генеральным директором, чтобы он знал обо всем, что вы собираетесь предложить. Он лучше всего способен оценить, подходящий ли уровень детализации вы выбрали, и ему не нужны сюрпризы, особенно на этом заседании.

## 2.5.4. Что можно, а чего нельзя

Вот несколько советов для успешного участия в заседании правления:

- Ознакомьтесь с повесткой дня и тем, что хочет сообщить генеральный директор.
- Не возражайте и не поправляйте генерального директора / финансового директора, если он говорит не о вашей сфере ответственности. Сделайте заметки, а потом задайте вопросы. Возможно, вам не хватает контекста.
- Выступайте по делу и сосредоточьтесь на главном. Найдите способы объяснить сложные инициативы языком, понятным вашим слушателям.
- Не позволяйте склонить вас к быстрым и необдуманным решениям. Примите к сведению вопросы и пообещайте дать ответ в течение нескольких дней.
- Если вы сообщаете плохие новости, никогда никого не обвиняйте, не приносите извинений и не называйте имен.

- Будьте готовы рассказать подробности и ответить на вопросы, которые могут показаться не относящимися к делу. Вы можете не знать, что еще обсуждалось.
- Не берите свой ноутбук или используйте его по минимуму, если вам нужно провести демонстрацию, и закройте его, как только закончите. Человек за ноутбуком, притворяющийся, что делает заметки, выглядит занимающимся своими делами. Используйте для записей старую добрую ручку и бумагу, чтобы другие видели, что вы вовлечены в обсуждение.
- Если вы не участвуете во всем заседании, уходите, как только закончите и ответите на вопросы. Спросите генерального директора (или председателя), можно ли вам уйти; не сидите и не дожидаетесь, пока вас попросят уйти.
- Не теряйте бдительности на неформальных встречах. Эти люди хорошо умеют располагать к себе, чтобы вы сказали им то, что не должны.

Хотя заседания правления могут проходить довольно напряженно и формально, они могут быть также интересными и информативными. Никогда не относитесь к ним как к рутине и пользуйтесь возможностью привлечь внимание очень влиятельных и важных людей.

### **ВЗГЛЯД СО СТОРОНЫ**

*Как произвести хорошее впечатление*

Как партнеру-основателю частной инвестиционной компании, Девину Мэтьюзу не привыкать к заседаниям правления. В его популярном подкасте PE Fundcast есть несколько эпизодов для генеральных, финансовых и технических директоров о том, как провести хорошую презентацию для правления. Когда я спросил его, что он посоветует техническим директорам, он ответил: «Не просто рассказывай, покажи». Слишком часто технический директор не показывает продукт, за который он несет ответственность, — это все равно что владеть шоколадной фабрикой, но никогда не пробовать, что получилось. Продемонстрируйте прогресс и позвольте продукту самому показать, что необходимо улучшить. Говорят, картинка стоит тысячи слов, а демонстрация рабочего ПО может стоить тысячи слайдов. Это также дополнительная возможность с гордостью продемонстрировать усилия своей команды просто с помощью языка тела — он полезен для тех, кому бывает сложно подобрать правильные слова.

## **2.6. КОММУНИКАЦИЯ**

Важно заранее подготовиться и подумать, каким образом лучше всего представить свой отдел внутри компании, а также внешним партнерам. Новый технический директор часто упускает это из виду и таким образом теряет прекрасную потенциальную возможность создать союзы и наладить взаимоотношения. В этом разделе рассматривается, о чем вам стоит рассказать и с чего можно начать.

## 2.6.1. Внутренняя коммуникация

Цель внутренней коммуникации не в том, чтобы показать, что вы умны, или похвастаться своими знаниями. В первую очередь она помогает другим добиться успеха и понять, чем занимается ваш отдел, чтобы к вам было проще обращаться при необходимости.

Вы можете отправить электронное письмо или сообщение в Slack/Team всем сотрудникам компании, если:

- *Вы выпускаете новый релиз своего программного продукта.* Если вы выпускаете обновление внутренней системы или сервиса для клиентов, разошлите информацию о релизе, используя термины, понятные компании, и не употребляйте сленг. Представьте тезисы, которые другим будет легко донести до клиентов.
- *Что-то пошло не так* — подобные ситуации нельзя игнорировать. Если что-то сломалось, неважно, по чьей вине, возьмите на себя информирование о ситуации, применив стратегию RIP. Люди начнут доверять вам и уважать вашу открытость и честность, а также им будет проще отвечать на вопросы со стороны.
- *В новостях опубликовали что-то, связанное с технологиями.* Тут есть нюансы. Если вышла громкая новость, скажем, о проблеме безопасности или о новой хайповой технологии, скорее всего, большинство людей не разбираются в вопросе на том же уровне, что и вы. Таким образом, если новость имеет отношение к вашей работе, найдите время, чтобы подготовить понятное объяснение, и сделайте рассылку.

### ВАЖНОСТЬ КОНТЕКСТА

#### *Проблемы безопасности*

В какой-то момент проблемы безопасности затронули таких гигантов, как BBC и CNN. Конкретных деталей было мало, но в технических новостях проскальзывали некоторые подробности, и в данном случае причина была в том, что кто-то не защитил сервис хранения файлов Amazon S3. В новостях говорили, что облако небезопасно и следует быть осторожными. Пользуясь случаем, я подготовил комментарий, пояснив, что файлы хранились за пределами защищенного хранилища и это вина не облачного провайдера, а тех, кто занимается безопасностью. Зная, что клиенты могут задать этот вопрос нашим менеджерам, я также указал процессы, которые мы внедрились, чтобы подобная ситуация никогда не произошла с нашим продуктом: мы не разрешаем создавать такие хранилища и у нас есть инструменты для проверки, кто имеет доступ к файлам. Люди запомнили информацию гораздо лучше, чем если бы я просто отправил это сообщение без контекста, потому что оно было связано с тем, что они читали и видели в новостях. Как говорится, никогда не упускайте возможность извлечь выгоду из кризиса. Если что-то у людей на слуху и вы можете помочь им понять правильный контекст — сделайте это.

Еще один возможный вариант — пригласите всех желающих к вам в отдел на посиделки в обеденный перерыв (когда все приносят с собой еду и собираются вместе) и обсудите технологические новинки. Вы никогда не знаете, кто может откликнуться на приглашение — возможно, ваш будущий сотрудник.

Что бы вы ни выбрали, старайтесь обходиться без модных словечек и технического жаргона. Если ваш отдел большой, попросите одного из членов команды сделать рассылку, особенно если это его зона ответственности. Так другие сотрудники тоже будут услышаны и вы покажете, что у вас целая команда высококвалифицированных специалистов.

### 2.6.2. Внешняя коммуникация

Налаживайте коммуникацию и за пределами компании. Смотрите шире, поскольку, обращаясь к нетехнической аудитории, вы совершенствуете навыки коммуникации. Это поможет продвинуть как ваш личный бренд, так и бренд вашей компании, а также познакомиться с новыми людьми и привлечь новые таланты в команду.

Выступления в локальных сообществах пользователей — прекрасный способ познакомиться с местным сообществом и узнать, кто есть кто. У одного из наших клиентов сотрудники плохо разбирались в SQL Server. Они отправили несколько человек на локальную встречу пользователей SQL Server. Мы посидели, послушали спикеров и познакомились с другими участниками и местными компаниями, которые в конечном итоге помогли нам. Мы смогли посмотреть на них и только после этого подумали о том, чтобы привлечь их к сотрудничеству. То же самое может работать и в обратном направлении.

Сколько раз вы находили в интернете решение конкретной проблемы, потому что кто-то рассказал о нем на Stack Overflow или другом ресурсе? А теперь спросите себя, оплатили ли вы за это? Внесли ли вы свой вклад в коллективный разум, чтобы другие могли учиться на вашем опыте? Скорее всего, большинство из тех, кто читает эти строки, взяли больше, чем вложили. Но вы не находитесь в изоляции.

Начните давать советы на Stack Overflow или заведите блог компании, чтобы делиться своими знаниями и открытиями, особенно если они касаются ПО с открытым исходным кодом. Проверьте, что вы не публикуете ничего конфиденциального. Тема ПО с открытым исходным кодом рассматривается далее



в книге, пока же имейте в виду, что это еще одна дополнительная возможность, которой можно воспользоваться в своих интересах.

Наладить внешние связи легко, и, хотя сначала они могут показаться пустой тратой времени, никогда не знаешь, чем они обернутся в будущем — возможно, привлекут новых поставщиков и подрядчиков или нового сотрудника. Если ничего не делать, то ничего и не будет. Как говорится в старой шутке: «Господи, почему я никогда не выигрываю в лотерею?» — «Так помоги мне и хотя бы купи лотерейный билет!»

## 2.7. ВНУТРЕННЯЯ ПОЛИТИКА

В какой-то момент вы столкнетесь с политикой как внутри своего отдела (над которым вы имеете больше контроля), так и на уровне компании. Скорее всего, вам много раз говорили, что компания — это одна большая команда и в ней нет политики, а затем выяснялось, что в действительности это неэффективное собрание людей с кучей внутренних проблем — просто это не называют «политикой».

В этом разделе рассматриваются некоторые часто встречающиеся сценарии, а также методы, которые помогут вам справиться со сложной ситуацией. Этот обзор далеко не исчерпывающий, и мы рекомендуем прочесть дополнительную литературу по психологии команд.

### 2.7.1. Основные типы

Рабочую политику можно определить как отсутствие данных, которые заменяются эмоциями. Мы работаем с целью сделать жизнь наших клиентов и сотрудников лучше, обеспечивая их благополучие. Кажется, если все будут грести в одном направлении, времени для конфликтов или эмоций не останется. Но мы всего лишь люди, управляемые эмоциями, и мы иногда забываем, что лучший способ обеспечить собственное благополучие, как и благополучие других — это поладить с ближними. Разберем некоторые часто встречающиеся типы эмоций или политики.

#### **ВОТЧИНА**

Вотчина — распространенная форма поведения в племени, часто встречающаяся в крупных организациях. Понятие «вотчины» было введено в начале 1800-х годов и обозначает установление контроля одного человека над определенной территорией. В бизнесе это может выражаться в том, что один или несколько человек в команде в первую очередь преследуют свои собственные цели, ставя интересы других в подобных полусвободных союзах на второй план. Это похоже

на подпольный клуб для избранных, в котором действуют неписанные правила и существование которого трудно доказать, но можно почувствовать.

Подобное поведение вредит компании в любой форме, каким бы невинным оно ни казалось. Даже если это просто хорошие друзья, присматривающие друг за другом, другие поймут, что за этим кроется нечто большее, и начнут делать выводы.

### **БУЛЛИНГ**

Трудно поверить, что проблема буллинга все еще актуальна среди взрослых профессионалов, но она чаще есть, чем ее нет, и она может проявляться во многих формах. Важно отметить, что это не то же самое, что настойчивость. Буллер часто демонстрирует непоследовательное поведение, запугивает, ведет себя иррационально и бывает груб.

#### **КРИКЛИВАЯ ЛИЧНОСТЬ**

Одна из худших ситуаций, свидетелем которых я был, возникла на видеоконференции (я был одним из участников), куда ворвался тогдашний генеральный директор компании и буквально уничтожил технического директора, ругаясь и крича о чем-то, что даже не было важным и не заслуживало таких эмоций. Генеральный директор понятия не имел, с кем шел созвон — это мог быть клиент. В данном случае инвесторы не впечатлились этой выходкой и тут же уволили генерального. Оказалось, у него это был уже не первый случай, и, к несчастью (для него), видеоконференция записывалась, поэтому оправдаться он не мог.

### **ПАССИВНАЯ АГРЕССИЯ**

Эта форма агрессии проявляется, когда люди не высказываются напрямую о проблеме. Вместо этого они демонстрируют деструктивное поведение, например тянут резину, ни с чем не соглашаются или сопротивляются решению проблемы.

Примером такого поведения может быть следующее: кто-то, кто не согласен с предпринимаемым действием, не высказывает несогласия, а, напротив, в целом выражает одобрение. Однако за пределами команды он подрывает инициативу и говорит всем, что она обречена на провал и лично он бы не принял подобного решения.

### **НАКОПИТЕЛЬ ИНФОРМАЦИИ**

Это человек, который считает, что если он поделится ценной информацией с другими, то его значимость уменьшится и его могут уволить. Этот тип людей рассматривается в следующих главах. Когда вы пытаетесь решить конкретную проблему, такие люди не помогают, а мешают.

### 2.7.2. Смягчение политики

Технические специалисты предпочитают четко структурированные процессы: сделайте это, и тогда произойдет то. Работа с эмоциями других людей не поддается логике if-then-else. Полностью рабочего способа справиться с деструктивным поведением не существует, но можно попытаться его смягчить. Если кто-то изобретет волшебную палочку, которая заставит офисную политику исчезнуть, он станет очень богатым человеком. А до тех пор рассмотрим несколько способов облегчить жизнь:

- *Распознавайте деструктивное поведение в себе.* Вы наблюдаете описанные черты и в своем поведении? Прежде чем ожидать изменений от других, начните с себя. Попросите коллегу или друга, которому вы доверяете, дать вам обратную связь.
- *Говорите об этом.* Когда кто-то демонстрирует угрожающее или пассивно-агрессивное поведение, указывайте на это. Я обнаружил, что всего одной простой фразой вроде «Попробуй еще раз, только без агрессии/сарказма?», могу разрядить самые напряженные ситуации, потому что юмор и прямота показывают, что вас не запугать.
- *Избегайте двусмысленности.* Общаясь с людьми, которые демонстрируют деструктивное поведение, старайтесь не использовать абстракции. Неопределенность дает собеседнику возможность изменить свое мнение, притворившись, что он именно это и имел в виду.
- *Уточните и подтвердите.* Когда кто-то кричит или запугивает, попросите пояснений и конкретных примеров причины недовольства. Не уступайте ни в одном вопросе, пока собеседник не разъяснит, из-за чего он злится, и пока вы не подтвердите, что причина вам понятна. Эмоции и стресс иногда могут брать верх над любым из нас, и требуется проявлять спокойствие и собранность.
- *Приветствуйте критику.* Если кто-то критикует вас или члена вашей команды, уточните подробности. Попросите привести пример возникшей проблемы. Перейдя к подробностям, вы избежите инстинктивного защитного поведения до тех пор, пока не станет ясна суть претензий.

Никто не идеален. Мы все совершаем ошибки и иногда позволяем эмоциям проникать в наши отношения с коллегами. Указанные техники служат цели разбавить эмоции щепоткой рационализма и добраться до корня проблемы. К сожалению, они не всегда работают.

Если вы оказались в ситуации, когда кажется, что ничего не работает, и вы уверены, что проблема не в вас — не бойтесь, решение есть: просто уходите. Жизнь слишком коротка, чтобы находиться в среде, в которой вы не испытываете удовлетворения, непродуктивны и не оказываете влияния. Если вы сделали все, что могли, то уход нельзя назвать провалом.



**ПРАВИЛО 24 ЧАСОВ**

Джефф Ханнер (Jeff Hanner), менеджер хранилища данных (сейчас на пенсии), прекрасно умел поддерживать конструктивные отношения так, чтобы они не ухудшались с течением времени. С момента так называемого нарушения он давал людям 24 часа, чтобы разобраться с ним; после этого об инциденте больше никогда не упоминалось. Это маленькое правило сразу же предотвратило множество недоразумений. Я сам теперь руководствуюсь им, даже если другой человек не знает об этом: если кто-то обидел меня (или, по крайней мере, я так думаю), я отпущу ситуацию, если не принял никаких шагов для ее изменения в рамках отведенного времени.

## 2.8. УПРАВЛЕНИЕ ИЗМЕНЕНИЯМИ

Искусство вносить изменения в любые организации можно выделить в отдельную дисциплину. Я часто сталкивался с инициативами, которые терпели неудачу из-за того, что слишком энергичный технический директор не сумел преподнести их должным образом и не позволил бизнесу подготовиться к изменениям. Чтобы добиться успеха, вам необходимо понять, что изменения повлияют на весь бизнес, и в первую очередь заручиться поддержкой генерального директора и всех руководителей отделов, чтобы все прошло успешно.

Соберите руководителей и расскажите им, какое влияние окажут предлагаемые изменения и каковы их потенциальные краткосрочные недостатки. Так вы настроите всех на принятие изменений, и они пройдут менее болезненно. Не обходите и не преуменьшайте возможные неудобства, которые могут при этом возникнуть. Поддержка лидеров здесь будет иметь решающее значение.

Стивен Таллами (Stephen Tallamy), технический директор EditShare, отмечает, что «каждый отдельный руководитель отвечает за свою сферу, но может легко видеть, что необходимо для достижения успеха всего бизнеса, и помогать другим в достижении этого успеха. Подобное единомыслие на уровне руководства оказывает влияние на всю компанию и действительно может уменьшить негативные последствия при внедрении значительных изменений».

Далее мы рассмотрим, о чем необходимо помнить, чтобы привести свои проекты и инициативы к успеху.

### 2.8.1. Лягушка в кипятке

Вы слышали старую байку о лягушке в кипятке? Если бросить лягушку в кастрюлю с кипящей водой, она мгновенно выпрыгнет оттуда. Но если поместить ее в кастрюлю с холодной водой и медленно нагревать, лягушка постепенно привыкнет, приспособится и вскоре будет сварена.

Если не принимать во внимание этическую сторону обращения с амфибиями, это хороший способ организовать управление изменениями. Если вы сделаете очередной релиз и переведете всех пользователей на новый продукт без обучения или предупреждения, они возмутятся и выпрыгнут из кастрюли.

Нам, технарям, это кажется нелогичным. Мы принимаем и приветствуем изменения. Наша отрасль обновляется каждые пять лет, и мы привыкли работать с этим циклом. Однако большинству (не технарям) не нравятся слишком большие перемены. Они просто хотят делать свою работу хорошо, зарабатывать деньги, обеспечивать семью и знать, что завтра они будут делать то же, что и сегодня. Любое изменение нарушит этот цикл.

Изменения необходимо внедрять постепенно, таким образом, чтобы казалось, что мы медленно повышаем температуру, и тогда большинство людей их не заметит. Бизнес не может позволить себе испытывать неудобства в обслуживании клиентов, вызванные слишком резкими изменениями. Поэтому изменений не должно быть слишком много и они не должны быть слишком быстрыми.

Гибкие методологии управления проектами (мы рассмотрим их в одной из следующих глав) продвигают принцип «делайте релизы понемногу и часто», который прекрасно подходит для медленной варки нашей лягушки. Мы даем людям время привыкнуть к новому.

### РЕАЛЬНЫЙ ПРИМЕР

#### *Microsoft Windows 10*

Если вам нужен мастер-класс по управлению изменениями и варке лягушек, обратитесь к примеру Майкрософт и ее выпуску Windows 10. Вместо масштабного перехода на новую версию ОС (3.1 → NT → 2000 → XP → Vista → 7 → 8), Windows 10, выпущенная в 2015 году, до сих пор с нами. Медленные, постепенные изменения радикально преобразили продукт за эти годы, но, поскольку ОС по-прежнему выглядит привычно, мы охотно воспринимаем эти изменения как незначительные. Майкрософт медленно варит лягушку (то есть нас) без нашего ведома. Да, на момент написания этой книги вышла Windows 11, но цикл начался снова: от малых изменений к большим.

## 2.8.2. Комитет воодушевления

К любому масштабному и долгосрочному проекту, который в перспективе повлияет на работу нескольких отделов, необходимо привлекать всех. Нельзя просто щелкнуть тумблером и переключить всю компанию на новый процесс. В каждом отделе есть свои установленные бизнес-правила и отлаженные рабочие процедуры, и все сотрудники отдела знают, как им работать. Какими бы плохими или неэффективными эти процессы ни были, они работают, потому что клиенты получают сервис.

Келли Пауэлл (Kelley Powell), генеральный директор MacLaurin Group, считает, что для каждой крупной инициативы, ведущей к большим изменениям, необходимо учреждать то, что она называет «комитетом воодушевления» (excitement committee). Это собрание представителей всего бизнеса, вовлеченных в проект с самого начала. Они станут вашими амбассадорами и защитниками, пока вы реализуете проект, и помогут двигаться в правильном направлении. Эти люди — необходимое условие успеха, отмечает Пауэлл, потому что они смогут описать своей команде грядущие перемены лучше, чем вы.

Информирование людей на самой ранней стадии помогает им подготовиться к изменениям. Не обещайте слишком многого и не убеждайте людей, что новая практика решит все их проблемы. Помимо этого, не отвергайте принятые способы работы — это вызовет обиду, потому что, каким бы уродливым ни был ребенок, для матери он всегда красив.

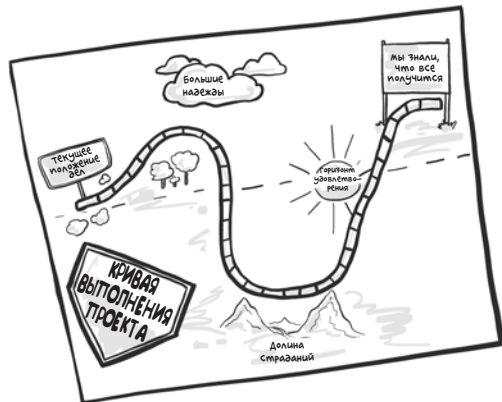
### 2.8.3. Долина страданий

В работе над любым крупным проектом наступает момент, когда сбои настолько часты, а страх настолько велик, что лидеры начинают сомневаться, стоит ли продолжать.

Вначале все были полны энтузиазма от перспектив, поэтому вы получали значительную поддержку. Затем, по ходу работы, обнаруживаются сложности, препятствующие легким победам, на которые вы рассчитывали, и требующие больше времени для проработки. По мере их возникновения и преодоления бизнес может начать терять терпение, поскольку все эти проблемы отнимают много времени.

Джеймс Милбери (James Milbery), партнер-основатель ParkerGale Capital, называет это явление «долиной страданий». Милбери утверждает, что технологические лидеры должны осознавать этот цикл и быть готовыми помочь бизнесу дойти до победы. На своем опыте он уверен, что по-настоящему успешный технический директор — это тот, кто способен оценить влияние любых изменений на бизнес и, работая с руководителями отделов, свести к минимуму негативные последствия.

Милбери также является сторонником стратегии «сжигать корабли», когда проект набирает обороты. В рамках этой стратегии разработчики с определенного момента уже не смогут вернуться к старым системам, подобно мореплавателям,



которые не смогут вернуться на родину после достижения новых земель — их капитаны в буквальном смысле сжигали корабли, чтобы заставить людей адаптироваться к новым условиям. Применительно к разработке ПО это означает, что нужно определить момент, когда старая система перестанет использоваться и будет выключена.

Поддерживать две системы, увеличивая нагрузку на саппорт и создавая дополнительные проблемы, запутывая всех, — последнее дело. В рамках своей стратегии управления изменениями вам необходимо определить критерии для «сжигания кораблей» и получить их одобрение от всех бизнес-лидеров.

### 2.8.4. Последствия

Изменения — важная и определяющая составляющая эволюции любой компании. Однако необходимо понимать, какие последствия изменения могут повлечь для бизнеса. Изменения — это больше, чем просто внедрение нового программного продукта; это вызов отлаженным бизнес-процессам, обучению сотрудников, адаптации клиентов и использованию существующих ноу-хау компании — то есть всему, что создавалось годами. Встройте процесс внесения изменений в свою стратегию и помните о лягушке — ей действительно не нравится, когда ее бросают в кипяток.



## ИТОГИ

- Установите прочные рабочие отношения с генеральным директором. Узнайте его мотивы и критерии успеха — это облегчит согласование ваших проектов и достижение результата.
- Регулярные встречи с генеральным директором помогут вам оставаться с ним на одной волне и получать мгновенную обратную связь, чтобы проводить корректировки, необходимые для достижения результата.
- Общайтесь с другими руководителями на понятном им языке и не погружайтесь в детали.
- Относитесь к генеральному директору как к партнеру, а не как к начальнику. Возьмите на себя все детали и покажите, что вы действительно вникаете в них, это значительно облегчит его работу.
- Совместно с финансовым директором ведите учет расходов и затрат вашего отдела и проектов, чтобы не было никаких сюрпризов.

- Знакомство с работой других отделов и их руководителями поможет вам составить полную картину того, как работает организация, а также лучше понять принятые в них показатели эффективности и то, какую поддержку вы можете им оказать.
- Если вы приходите на смену предыдущему техническому директору, цените его заслуги и не критикуйте его решения. Нет смысла заикливаться на прошлом: вы в ответе не за него, а за будущее.
- Если вас пригласят выступить перед советом директоров, подготовьтесь и примите активное участие в заседании. Это не рутинное собрание и не потерянное время, а признание вашей роли и возможность повлиять на стратегию компании.
- Грамотная коммуникация внутри компании и с клиентами расширит ваши перспективы, сделает вас более доступным и позволит извлечь сферу технологий от ореола таинственности.
- Научившись распознавать типичные модели внутренней политики, вы сможете предотвратить их закрепление в корпоративной культуре и работать открыто и прозрачно.
- Люди не любят перемены, но специфика вашей сферы обязывает вас находиться в авангарде непрерывного развития систем и продуктов. Постепенно вовлекая других в этот процесс изменений, вы обеспечите более высокий уровень принятия инноваций.

## **ПРОВЕРЬТЕ СЕБЯ**

Что из перечисленного ниже вы уже сделали?

- Определили тип своего генерального директора, выяснили его предпочтения и выбрали оптимальный стиль работы с ним.
- Запланировали регулярные встречи с генеральным директором.
- Установили хорошие рабочие отношения с финансовым директором и регулярно обсуждаете с ним свои актуальные бюджетные потребности.
- Наладили контакт с другими руководителями компании и поняли их цели и проблемы.
- Разобрались во внутренних процессах и политике, которые влияют на процесс принятия решений в компании.
- Подготовились к презентации на заседании правления, узнали, кто в него входит и что для них важно.
- Определили порядок своего общения с компанией в качестве технического директора.
- Получили представление о том, как лучше внедрять изменения и каким образом обосновать их необходимость.

# 3

## *Долгосрочное видение*

---

### **В этой главе**

- ✓ Важность видения
- ✓ Планирование видения вокруг потребностей бизнеса
- ✓ Определение в видении настоящей ценности для клиентов
- ✓ Планирование долгосрочных изменений, которые произведут революцию в бизнесе
- ✓ Как зафиксировать ожидания и избежать ложных обещаний
- ✓ Экономический план и видение
- ✓ Донесение видения до разных аудиторий
- ✓ Актуализация видения

Генеральный директор — не только руководитель, но и лидер компании. А лидерство — это умение смотреть вперед, предвосхищать потребности рынка и оставаться в авангарде, а не просто следовать за внешними изменениями. Ваша роль технического директора, лидера корпоративных технологий, состоит в том же самом. Недостаточно просто добиться, чтобы все работало; следует убедиться, что все будет работать достаточно долго и эффективно, независимо от того, что происходит вокруг. Компания рассчитывает, что вы обеспечите необходимый функционал и возможность развития для ее клиентов. Если вы уделите время подробной разработке и детализации видения, наметите курс и расскажете

коллегам о ваших планах — это придаст бизнесу уверенность в том, что с вами он в надежных руках.

Чтобы проиллюстрировать вышесказанное, представим вымышленную компанию ООО «HomeMax». Допустим, она существует уже более 40 лет и предоставляет услуги страхования жилья для собственников. Она использует устаревшую платформу, которая написана на языках RPG и Cobol, управляется через терминал, использует базу данных Db2 и работает на системе AS/400s. В компании никогда не было формального технического директора, вместо этого основатель сам руководил небольшой командой разработчиков. Недавно ее приобрела частная инвестиционная компания, которая хочет увеличить принадлежащую ей долю рынка и поэтому наняла первого технического директора, который должен возглавить технологическую трансформацию.

### 3.1. ОБЩЕЕ ВИДЕНИЕ

Скорее всего, у вашей компании уже есть видение или миссия. Разница между этими понятиями тонкая, но важная: видение — это цель миссии, конечный результат, в то время как миссия ориентирована на действия и описывает, зачем существует организация, какие виды услуг и продуктов она предлагает и что делает ее уникальной.

#### ЗАМЕТКИ С ПОЛЕЙ

##### *Пример видения*

Чтобы понять, что такое хорошее видение, рассмотрим некоторые крупные известные корпорации. LinkedIn ставит перед собой задачу «создать экономические возможности для каждого участника глобального рынка труда», в то время как задача Disney — «рассказывать потрясающие истории, развлекая, обучая и вдохновляя людей по всему миру», а Microsoft — «компьютер на каждом столе и в каждом доме». Общее в них то, что все эти компании используют очень простые для понимания формулировки без профессионального жаргона. Видение нашей компании HomeMax может звучать так: «Будьте спокойны за свой самый ценный актив».

У технического директора должно быть свое видение, сосредоточенное на результатах его команды, но поддерживающее общее видение и миссию компании. Общее видение описывает, над чем вы работаете, почему вы движетесь в этом направлении и что вы и бизнес сможете сделать, когда придете к цели (маленькая деталь, которую многие упускают из виду). Здорово, что вы движетесь к определенной точке, но важно помнить, что вы будете делать по прибытии.

Видение должно разрабатываться на 2-5-летнюю перспективу. Более долгосрочные цели ставить сложно, потому что бизнес может трансформироваться, изменить направление или даже быть продан прежде, чем видение будет



реализовано. Хорошее видение — это то, которое можно реализовать и воспользоваться его плодами. Вы ведь не хотите быть лидером, для которого «завтра никогда не наступит». Конечно, вам придется развивать и менять свое видение по мере того, как изменяются бизнес и технологии. Как это сделать — мы рассмотрим ниже.

Важно: если вы приходите на место технического директора, который создал работающий и хорошо отлаженный механизм, ваше видение может заключаться в том, чтобы просто продолжать двигаться и развиваться в существующем направлении. К счастью (или к сожалению, как посмотреть), наша отрасль не стоит на месте, и в ней всегда есть возможности для улуч-

шения. У вас будет много шансов добавить что-то свое.

Формулируя свое видение, учитывайте две аудитории: бизнес и свою команду. Бизнесу необходимо понимать, как ваше видение повлияет на него. Чтобы рассказать об этом, используйте общую терминологию без специальных терминов. Это часть «что мы делаем» (и «зачем») вашего видения. Вашей команде тоже необходимо все это знать, но ей нужно больше «мяса», то есть подробностей и объяснений, каким образом будет достигнута цель. Это часть «как» вашего видения.

### 3.1.1. Часть «что»/«зачем»

Бизнесу интересно, что ваше видение значит для него, то есть для остальной компании за пределами вашей команды: членов совета директоров, инвесторов и даже клиентов. Если ваше видение не обращается к каждой из этих групп и не говорит им, какую выгоду они получают, то оно является не более чем целью проекта (или даже всего лишь избавлением от технического долга). Видение может рассказывать об экономических выгодах, важных функциях платформы или новых возможностях.

#### ЗАМЕТКИ С ПОЛЕЙ

*Никакая информация не потеряется*

В одной из портфельных компаний, в которой я был техническим директором, мне нужно было сформулировать свое видение на основе наблюдений за бизнесом и его ценностями (культурой принятия решений на основе данных). Чтобы поддержать видение генерального директора, я поставил своей группе следующую цель: обеспечить бизнес всеми возможными данными для принятия обоснованных решений.



Для этого я предложил простой девиз: «Никакая информация не потеряется». Я даже попросил отдел дизайна создать огромный баннер в стиле «Назад в будущее» (*«Back to the Future»*) — он выглядел потрясающе (удобно иметь собственный отдел дизайна). Учитывая то, насколько дешевым стало хранение данных — сейчас не осталось оправданий тому, чтобы не записывать и не сохранять все, что можно. Раньше это было проблемой для компании, и теперь все, что мы делали, должно было соответствовать этому девизу и обеспечивать доступность данных для бизнеса. С каждым релизом нового функционала или новой версии платформы мы видели, как реализуется это видение и как по мере накопления все большего массива данных преумножаются знания и открываются новые возможности.

Формулировка видения не должна быть слишком сложной и подробной. Она должна быть такой, чтобы каждый слушатель мог ее усвоить и увидеть в ней пользу для себя. Для нашей компании HomeMax, основанной на устаревших технологиях и стремящейся перейти в мир веба и мобильных приложений и перестать вручную обрабатывать каждый запрос и каждое изменение, видение можно сформулировать так: «Автономная платформа, ориентированная на данные клиента».

### 3.1.2. Часть «как»

Вы и ваша команда должны будете реализовать свое видение. Для этого необходимо более подробно детализировать, что конкретно имеется в виду и что для этого нужно поменять или скорректировать. Успешное видение — это то, что каждый понимает и может внедрять в рамках своей зоны ответственности. При принятии каждого решения вами или командой спрашивайте себя — приближаетесь вы к реализации вашего видения или отдаляетесь от него. Кандидат, которого вы собираетесь принять на работу, контракт с новым поставщиком услуг, решение о выборе нового стека технологий — приближают ли они вас к цели?

Видение должно находить отклик и поддержку в вашей группе. Чтобы проверить приверженность технической команды видению, спросите: «Верите ли вы в наше видение?» Если нет, то продать это видение бизнесу будет практически невозможно. Если те, кто ежедневно работает на передовой, решает проблемы клиентов, устраняет баги и внедряет новые функции, не поддерживают ваше видение, то, скорее всего, вы сформулировали его недостаточно понятно или подробно.



**ЗАМЕТКИ С ПОЛЕЙ***Туманное видение*

Один мой знакомый СТО гордился своим видением: «Мы интегрируем машинное обучение в нашу аналитику». Это, конечно, прекрасная идея, но после расспросов выяснилось, что в ней не хватало деталей. Он не думал о том, что это видение значит для бизнеса и какую выгоду оно несет клиентам. Его собственная команда не получила ответ на вопрос «как» (вопрос «зачем» тоже не поднимался), поэтому гонялась за тенью, пытаясь решить проблему, которой не существовало. Все дело было в том, что СТО ухватился за модную тему машинного обучения, потому что это звучало круто. Он знал, что у них есть большое количество данных, и рассчитывал, что это поможет его карьере. Но никто не подумал, нужно это кому-то или нет. Новые технологии хороши, только если им находят верное применение, а не для всего подряд (даже не буду рассказывать, от какого количества идей на основе блокчейна я не оставил и камня на камне).

**3.1.3. Определите критерии успеха**

Ясно и четко сформулированное видение дает вам возможность двигаться вперед и, что важно, измерять ваш прогресс. Технический директор без видения — это капитан без цели; вы должны понимать, где находится ваш пункт назначения. Определив свое видение, сформулируйте более мелкие цели как его часть. Достижение каждой из них — это маленькая победа, которую можно отметить вместе с командой.

Хотя у многих технических директоров нет видения, те, у кого оно есть, немало теряются, когда их спрашивают о том, как будет выглядеть успешный результат. Они отвечают обтекаемо и общими фразами. Полезно сформулировать свои критерии успеха, чтобы, когда вы его достигнете, он не вызвал споров.

Рассмотрим несколько примеров структурированного видения, причем некоторые из этих формулировок подходят нашей компании HomeMax. Они соответствуют глобальной идее, состоящей в том, чтобы предоставить клиентам больше контроля.

**Таблица 3.1.** Примеры видения

Видение	Что	Как	Успех
Отказаться от оборудования и полностью перейти в облако	Избавиться от накладных расходов на обслуживание аппаратных ресурсов и перейти на модель оплаты по факту потребления, которая ускорит подключение новых клиентов	Прекратить покупать оборудование, начать изучать облачные альтернативы и определить области, в которых мы можем добиться первых успехов, не подвергая риску бизнес	Когда последний сервер будет выключен и исчезнет зависимость от офиса/дата-центра

Видение	Что	Как	Успех
Переход к архитектуре на основе API	Сделать наши данные/сервисы доступными посредством безопасной и надежной технологии, что позволит клиентам напрямую интегрироваться с нами без необходимости пересылать файлы	Определить, в каких точках клиентам требуется отправлять и получать данные, сделать то же самое для внутреннего обмена данными	Для взаимодействия с сервисом клиенту не требуется отправлять файлы; он самостоятельно выполняет операции с помощью API
Заменить устаревшую систему	Перевести систему на современную поддерживаемую архитектуру, которая позволит гораздо быстрее реагировать на запросы клиентов	Описать все бизнес-правила и зоны ответственности системы, а затем начать разработку одной или нескольких новых систем, которые должны внедряться поэтапно	Когда устаревшая система будет выключена и перестанет обслуживать клиентов
Предоставить клиенту современный интерфейс на основе веб- и мобильных технологий	Обеспечить клиентам возможность осуществлять операции самостоятельно, а не через колл-центр	Использовать современный фреймворк для веб-разработки, а также безопасное API на серверной стороне	Резкое сокращение количества входящих звонков

Оставшаяся часть главы посвящена планированию и тому, как оно помогает поддерживать общее видение.

## 3.2. СОТРУДНИЧЕСТВО С КЛИЕНТАМИ

Начиная работать над видением, в первую очередь вы должны помнить о клиенте. Для начала необходимо определить, кто ваш клиент. Клиент — это человек или группа, которые потребляют ваш основной продукт. В крупных корпорациях это может быть не основной клиент компании, а внутренняя группа. Тем не менее, независимо от того, кем он является, он — причина вашего появления здесь.

Понимание клиента, его мотивов, желаний и целей поможет вам создавать лучший продукт. Это азы, изучаемые в бизнес-школах. Несмотря на это, многие считают, что такими вещами должны заниматься отделы продаж и маркетинга и эти понятия должны применяться на уровне всей компании. Многие не видят, что все то же самое работает и на уровне отдела, особенно такого, как команда СТО, отвечающая за технологию, с которой будет взаимодействовать клиент.

### 3.2.1. Знай своего клиента

В зависимости от компании вы можете не общаться напрямую с клиентами, а вести взаимодействие через менеджера по работе с клиентами или по продажам. Возможно, это и к лучшему, особенно для нового СТО, который не имеет большого опыта в переговорах с клиентом. Последнее, чего хочет генеральный директор, — потерять крупного клиента из-за того, что чересчур бойкий СТО пообещал слишком много или проговорился о том, о чем не следовало.

Лучший совет, когда дело доходит до переговоров с клиентами, — не брать инициативу на себя, а просто слушать. Не предлагайте решений, ничего не обещайте, не давайте гарантий; просто слушайте. Вам необходимо узнать, как клиент взаимодействует с вашими системами, и выявить болевые точки. Вот какие вопросы стоит задать для начала:

- Какая система/сервис является основной точкой взаимодействия с вашей компанией?
- Если происходит обмен данными — как вы их передаете?
- Что происходит с данными после того, как вы их получили?
- Как обычно используется сервис?
- Что необходимо вашей команде для успешной работы?
- Что в ваших процессах вы бы хотели улучшить или изменить в ближайшем будущем?

Ответы на эти вопросы помогут понять, как вас видят клиенты и что из того, что интуитивно понятно вам и вашей команде, может оказаться трудным для других.

Вот как собственник жилья мог бы ответить на приведенные выше вопросы применительно к нашей компании HomeMax:

- Я могу взаимодействовать только с горячей линией колл-центра.
- Если мне необходимо внести изменения или обновить данные, я должен озвучить их по телефону.
- Я понятия не имею, получил ли оператор мои данные, пока не получу подтверждение в установленные сроки.
- Я использую сервис только в случае каких-то проблем, когда я создаю заявку. При этом статус заявки я могу узнать только по телефону.
- Нужно, чтобы оператор отвечал своевременно.
- Я хочу иметь возможность самостоятельно вносить изменения, прикладывать фотографии к заявке и проверять ее статус без необходимости звонить по телефону.

Очевидно, что конечный потребитель хотел бы по возможности взаимодействовать с сервисом самостоятельно, в своем темпе и в удобное время. Это типичные требования, но главный вывод, который здесь должен сделать СТО, — сейчас необязательно предлагать новые продукты или функции. Для начала нужно оптимизировать взаимодействие с компанией по телефону.

### ЗАМЕТКИ С ПОЛЕЙ

*Решайте проблему, а не боритесь с симптомами*

Мы работали с компанией, в которой были отлично налажены отношения с клиентами — показатель удержания более 95%. И менеджер по работе с клиентами, и сам клиент были довольны. Однако внутри компаний-клиентов все было не так радужно — слишком большие накладные расходы на управление пользователями. У них была постоянная ротация временных работников, которые работали с клиентским порталом, и очень много времени уходило на создание и удаление учетных записей пользователей. В какой-то момент в компанию пришел новый технический директор. Услышав о жалобах, он не стал ничего обещать, однако провел опрос всех клиентов о наличии проблем. Вскоре он включил в свое видение необходимость обеспечить единый вход в систему, чтобы клиенты не испытывали затруднений при управлении пользователями.

Вдобавок, знание того, куда движется клиент, будет чрезвычайно полезно при подготовке видения и планов на будущее. К примеру, вы выяснили, что большинство ваших клиентов работают с продуктами Microsoft и планируют перейти в облачную службу Microsoft Azure. Если вы еще не выбрали поставщика облачных услуг, имеет смысл следовать за клиентами, чтобы избежать возможной несовместимости. Но вы не узнаете планов клиента, если не поговорите и не познакомитесь с ним.

### 3.2.2. Ваш партнер

В случае особенно важных клиентов имеет смысл познакомиться с вашим партнером — техническим директором или вице-президентом по разработке. Это позволит установить канал для решения любых проблем до того, как они станут серьезными, а также поможет вам понять мысли и планы клиента. С помощью этих отношений вам будет проще всего убедиться, что ваше видение и планы соответствуют планам клиента.

Эти особые отношения могут оказаться особенно полезными, если возникло недопонимание. Сколько раз случалось так, что вам говорили, что клиент запросил функцию X, а вы понятия не имели об этом запросе? Вы знаете, что он прошел через все инстанции и был проверен отделом продуктового развития, и все же что-то вас смущает. К кому вы можете пойти с этим?



Помните детскую игру «испорченный телефон»? Сидя в кругу, вы что-то шепчете на ухо человеку слева от вас, а он должен передать эти слова сидящему слева от себя и так далее, пока они не вернуться к вам. В итоге в правом ухе вы слышите совсем не то, что вы изначально сказали. Ваши слова передавались от человека к человеку, получатели интерпретировали их и передавали то, что, как им казалось, они услышали, своими словами. Бизнес — это всего лишь игра в испорченный телефон.

Часто именно так и происходит с запросами клиентов, и поэтому ваше знание клиента и ваши особые отношения с ним могут помочь внести ясность, об отсутствии которой вас предупреждала интуиция. Иногда нужно задать клиенту простой вопрос: какую проблему вы пытаетесь решить?

Клиенты — это не мифические существа, которые всегда правы и которым нужно подчиняться. Они, как и вы, занимаются бизнесом и стараются удовлетворять потребности своих собственных клиентов. Их проблемы могут стать и вашими.

Подобные ситуации обычно возникают, когда бизнес клиента растет и одновременно растут требования к нему, например необходимо обеспечить соответствие нормативным требованиям или получить сертификацию по достижении определенного уровня либо чтобы выйти на новый рынок. Став партнером своего клиента, вы сможете ему не натворить бед — а это, в свою очередь, принесет пользу и вам.

#### ЗАМЕТКИ С ПОЛЕЙ

Я работал с компанией, в которой данными с одним из клиентов в основном обменивались по электронной почте, ежедневно отправляя файл. Основная проблема заключалась в том, что в этом файле было много конфиденциальных данных, а механизм передачи по электронной почте по своей природе небезопасен. Клиент был непреклонен, настаивая на электронных письмах. Технический директор же был новичком и не считал себя вправе отказать клиенту. Узнав об этом — как только я смог успокоиться и вернуть свой пульс в норму — я научил технического директора, как построить разговор с клиентом, чтобы понять, почему им нужно получать эти данные именно по электронной почте. Оказалось, что для них это просто был удобный способ сохранять информацию в свою внутреннюю сеть. Поскольку каждое утро клиент все равно заходил на их портал, технический директор добавил туда функцию загрузки файла, чтобы вместо почты клиент получал файл там. В итоге все остались довольны, потому что теперь данные хранились и передавались защищенным способом и при этом никто не потерял лица.

Некоторые клиенты будут более технологически продвинуты, чем вы; некоторые наоборот. Старайтесь походить на тех, кто впереди, и учиться на их видении. Найдите способы помочь тем, кто позади вас, избежать утечки данных.

Ваше видение должно включать или, по крайней мере, учитывать использование ваших сервисов на более низких уровнях. Таким образом вы будете

соответствовать актуальным требованиям и избавиться от потенциальной головной боли.

### 3.3. ДОЛГОСРОЧНОЕ ПЛАНИРОВАНИЕ

Как я уже говорил, если ваше видение можно реализовать за несколько месяцев, то это не видение, а проект. Видение — это комплексное начинание, на реализацию которого может уйти много лет. Видение, как правило, носит революционный характер, но может быть и эволюционным, если компания проходит процесс трансформации.

Однако, хотя подобные примеры и встречаются, планы редко бывают настолько масштабными, чтобы достижение даже первых результатов занимало годы. Если вы обнаружите, что оказались в такой ситуации, скорее всего, вы что-то упустили; вернитесь назад и попробуйте поставить меньшие цели, даже если для этого вам придется несколько изменить направление движения.

В следующем разделе рассказывается о том, что нужно, чтобы превратить видение в план действий, приемлемый для бизнеса.



#### 3.3.1. Время решает все

Масштабное видение требует значительных усилий для планирования и реализации. В основе хорошего видения лежит необходимость, а не действие. Подходящие тому примеры — проекты серьезной модернизации и обновления устаревших систем (например, в нашей компании HomeMax). Они могут занимать несколько лет, в течение которых также продолжает использоваться и поддерживаться существующая система. Принимая такой радикальный, хотя и потенциально выгодный проект, вы должны очень хорошо понимать, как



будет реагировать бизнес и насколько он готов поддержать вас как финансово, так и испытывая неудобства.

Если вы не сможете обеспечить эту поддержку — вы потерпите неудачу, не успев написать ни одной строчки кода и не запустив ни один сервис. Вы должны думать как бизнес: в чем необходимо быть уверенным, прежде чем приступать к работе?

Первое, что нужно учитывать, — график работы компании. У вас сезонный бизнес? Компании, в которых можно вносить изменения в сервисы в любой период года, встречаются редко. Например, ретейлеры (как онлайн-, так и офлайн-продавцы), скорее всего, не будут вносить серьезные изменения или делать релизы в сезон высоких продаж, например во время распродаж после Дня благодарения или в рождественский сезон, а компании, работающие в сфере образования, не захотят внедрять изменения в начале учебного года или во время сессий. Определите эти периоды и спланируйте свою работу в соответствии с ними. В организациях, в которых есть подобная сезонная нагрузка, порой приходится прекращать разработку новых продуктов на 3-4 месяца в год и перераспределять ресурсы на поддержку систем.

Не стоит недооценивать время, которое требуется для переключения контекста при переходе с проекта на проект, особенно если команда не работала над проектом несколько месяцев — с ноября по январь, учитывая выходные.

### ЗАМЕТКИ С ПОЛЕЙ

#### *Двигайтесь шаг за шагом*

Одна частная инвестиционная компания привлекла меня, чтобы я оценил план ее технического директора, который смущал своей амбициозностью. План был хорошим: это был проект модернизации, направленный на снижение зависимости компании от центров обработки данных и переход в облако. Проблема заключалась в том, что СТО делал сразу большой скачок от традиционного подхода с использованием серверов к облачной архитектуре — слишком много изменений, связанных со значительным риском. Я предложил добавить в план менее крупные шаги, чтобы разогреть бизнес и добиться нескольких побед, тем самым укрепив его уверенность и веру в это видение. Например, выделить самые старые серверы и заменить их на облако, вместо того чтобы покупать новое оборудование. Да, это не полностью облачное решение, но, уменьшая зависимость от аппаратного обеспечения, мы двигались к финальной цели, демонстрируя бизнесу преимущества на этом пути и знакомя его с новой сферой виртуальной инфраструктуры и тем самым давая ему время на адаптацию, чтобы продолжать двигаться вперед.

Второе — это насколько терпеливо компания готова дожидаться реализации вашего видения. Запомните: кто бы что ни говорил в начале, у них никогда не будет такого же терпения, как у вас. Вы, скорее всего, слышали вопрос «Как можно съесть слона?», подразумевающий, что проект настолько велик, что совершенно непонятно, как к нему приступить. Тем не менее ответ прост: «Разрезав его на бифштексы», то есть нужно разделить проблему на мелкие части.



То же и с видением; разделите его на кусочки размером с укус, которыми бизнес будет лакомиться и чувствовать прогресс. Определите, какие области можно разбить на части и предоставить бизнесу. Это должно быть то, что принесет ему реальную выгоду немедленно.

Если бизнес будет видеть пользу от вашего видения уже на ранней стадии и регулярно, его желание поддерживать и финансировать вас не уменьшится. А если (когда?) вы обнаружите, что начали буксовать, ваши прошлые достижения способствуют тому, что бизнес представит вам больший кредит доверия, чем мог бы.

Хуже всего, если сроки реализации проекта затянутся, а дата релиза будет отодвигаться все дальше. Ваше лидерство будет поставлено под сомнение, и бизнес может потерять к вам доверие и сократить финансирование. Да, это наихудший сценарий, но не такой уж редкий, даже несмотря на благие намерения.

### СИМ НАРЕКАЮ ТЕБЯ...

Я сбился со счета, сколько раз я слушал взволнованные рассказы технических директоров об их видении, подробно описывающие все его преимущества и возможные проблемы. Тем не менее, когда я спрашивал их, как они его называли, я получал в ответ недоумевающие взгляды. Назвали?

Мы называем вещи, чтобы придать им индивидуальность, наделить эмоциями. Мы привязываемся к вещам, которые имеют название, и поэтому желаем им успеха. Вот почему я дал своим трем сыновьям имена, а не просто назвал их Ребенок № 1, Ребенок № 2 и Ребенок № 3, хотя, честно говоря, так было бы намного проще. Хорошее название проекта вдохнет жизнь в ваше видение и позволит каждому вложить в него эмоции.

Не переусердствуйте с названием, потому что оно используется только внутри компании, но иногда интересные названия становятся известны за ее пределами, так что обдумайте его хорошенько. Мне кажется хорошей идеей придумать его всем вместе, чтобы каждый чувствовал свою причастность. Можно выбрать тему из «Властина колец» или классических «Звездных войн» (крупные проекты в области хранения данных часто называют «Звездой смерти»).

Хорошее название, например «Проект Гэндальф» (так назывался проект трансформации устаревшей системы, состоявший из двух этапов: «Гэндальф Серый», а затем «Гэндальф Белый») или «Позывной Борг» (для описания создания большого хранилища данных), формирует ассоциации с определенным сюжетом, чтобы помочь участникам проекта представить общую цель и сосредоточиться на ней. Одним из лучших названий, которое я встречал, было «Платформа Гамп». Так называлась система для управления устройствами Интернета вещей (IoT), которая, среди прочего, должна была работать непрерывно (вспомните, что Форрест Гамп должен был бежать без остановки).<sup>1</sup>

<sup>1</sup> Отсылка к фразе «Беги, Форрест, беги!» (англ. Run, Forrest! Run!) из фильма «Форрест Гамп» («*Forrest Gump*») режиссера Роберта Земекиса (1994). — *Примеч. пер.*

### 3.3.2. Обдумайте детали

Составляя план своего видения, ответьте на следующие вопросы:

- *Какую глобальную проблему вы хотите решить?* Формулирование проблем поможет вам аргументированно ответить на вопрос «зачем», когда бизнес задаст вам его.
- *Какие преимущества оно создает для компании и/или клиента?* Выделите преимущества для бизнеса и клиентов, которые они получают благодаря реализации вашего видения. Это может быть большая гибкость при удовлетворении потребностей клиента или новые возможности.
- *Зависит ли оно от времени реализации?* Если к реализации вашего видения можно приступать в любое время, отлично. Но если какие-то периоды предпочтительнее, чем другие, отметьте это и используйте как возможность для более тщательного планирования.
- *Каковы критерии успеха?* Соответствуют ли они критериям, определенным бизнесом? Важно определить показатели успеха, чтобы понимать, что вы достигли цели. Подумайте, как бизнес воспримет вашу инициативу и как он будет оценивать ее успех. Помните: то, что является проблемой для вас, не обязательно выглядит как проблема для бизнеса.
- *Кто вам требуется для достижения успеха?* Есть ли сотрудники, от которых зависит успех вашего видения, насколько эти люди важны для текущих повседневных задач и как компенсировать их отвлечение от этих задач?
- *Является ли проект по своей сути проектом модернизации/трансформации?* Вы хотите модернизировать устаревшую систему? Хотя это позволит решить многие ваши проблемы, клиенты и бизнес могут быть вполне довольны текущим положением дел, так что примените творческий подход, чтобы аргументировать, почему систему необходимо изменить.
- *Как новое поколение повлияет на непрерывное ведение бизнеса?* Крупные инициативы всегда нарушают привычный порядок. К каким последствиям как для вашего отдела, так и для всего бизнеса приведет реализация вашего видения? Даже если вы планируете привлечь дополнительные ресурсы, кто-то должен ими управлять, и они отвлекут вас от поддержания работы существующих систем.

Подготовившись и обдумав, какое влияние окажет ваше видение, вы покажете бизнесу, когда он начнет предъявлять вам возражения, что учли все его опасения. Ничего страшного, если у вас не будет ответов на какие-то вопросы. Лучше честно признаться, что в каких-то областях вам нужно больше информации, чем сочинять ответы на ходу, поскольку в будущем вы все равно столкнетесь с этими вопросами.

### 3.4. РАБОТА С ПОСТОЯННЫМИ ИЗМЕНЕНИЯМИ

Мы работаем в одной из самых динамичных и изменяющихся отраслей. То, что в прошлом году считалось передовым, в этом уже считается мейнстримом или

же исчезает и будет забыто. Мы находимся в постоянно меняющейся среде, которая иногда пугает и заставляет думать, что вы играете в догонялки. Это нормально, и это нужно принять.

Умение принимать правильные решения в отношении отдельных технологий само по себе является искусством — мы рассмотрим его далее в этой главе. Но в рамках этой темы мы разберем, как учитывать изменяющуюся среду при долгосрочном планировании. То, что считается рискованным в начале многолетнего проекта, может стать нормой через два-три года. Худший возможный результат — выпустить продукт, который устарел еще на стадии разработки.



### 3.4.1. Паралич решений

Мучительно раздумывать, верен ли данный путь — значит терять время и игнорировать возможность изменить направление. В сфере технологий любое решение лучше, чем его отсутствие. То, что правильно в данный момент, исходя из имеющихся данных и условий, и есть правильный путь.

#### ЗАМЕТКИ С ПОЛЕЙ

##### *Комплекс вины первопроходца*

Никому не хочется принять неверное решение, поэтому, когда СТО выбирает определенный путь, он иногда испытывает то, что я называю «комплексом вины первопроходца» — по аналогии с сомнениями покупателя, который боится, что мог сделать неправильный выбор. Мы все когда-нибудь сталкиваемся с этим чувством и задаемся вопросом, принял бы другой человек такое же решение. Уверенность, что успех всего проекта зависит от этого единственного решения и, если оно неверное, то все обречено на провал, родственна синдрому самозванца (о нем поговорим позже). Не волнуйтесь — не бывает решений, которые нельзя отменить или, по крайней мере, скорректировать.

Открою вам маленький секрет успеха долгосрочных, больших проектов: все меняется, и чаще, чем мы ожидаем, по мере того как мы учимся и узнаем больше. Совершенно нормально менять и переосмысливать какие-то исходные предположения. Главное — найти систему или подход, которые позволят вносить изменения, не начиная при этом все заново. В этом разделе предлагается схема, которая позволит вам вносить изменения, не подвергая угрозе срыва весь проект.

### 3.4.2. Определите, на чем основывается ваше видение

Все мы знаем, как важен хороший фундамент, чтобы дом стоял прочно. В сфере технологий тоже необходимо иметь прочную основу для строительства успеха нашей компании. Если наш фундамент шатается под воздействием внешних сил — что бы мы на нем ни строили, он всегда будет слабым местом.

Вам нужен хороший фундамент — опоры, если хотите, — который будет поддерживать ваш проект. Представляя фундамент состоящим из опор, легко видеть, что любую из них можно заменить, пока другие продолжают стоять и поддерживать ваше видение.



Сядьте поудобнее и подумайте, какие у вас опоры. Это самые крупные компоненты, без которых реализация вашего видения невозможна. Рассмотрим несколько примеров.

Предположим, в рамках своего видения вы планируете перейти к архитектуре на основе API, чтобы отделы компании и клиенты могли получать нужные им данные и использовать сервисы. Опорой здесь будет технология предоставления безопасного и масштабируемого API. Допустим, вы остановились на технологии HTTP REST, а для передачи токена защиты выбрали формат JWT. Это не ограничивает дальнейший выбор языка или фреймворка, но предоставляет конкретную методологию, которая позволяет сфокусироваться на деталях реализации.

Другой пример: вы основываете свое видение на том, что сбор данных и управление ими представляет особую важность и необходимо обеспечить хранение больших объемов данных без потери производительности. Ваша опора здесь — хранение и управление данными. Не заморачивайтесь — используйте озеро данных (модный термин для обозначения хранения большого объема данных независимо от их формата) для хранения, а затем решите, как и где обрабатывать эти данные. Например, для начала прекрасно подойдет AWS S3.

В идеале такими областями, или опорами, является не бизнес-логика, которую вы должны реализовать, а то, что можно отдать на аутсорс (или использовать готовый фреймворк или библиотеку). Так вы избавитесь от головной боли и необходимости выделять ресурсы на поддержание их работы. Эти сервисы можно рассматривать как отдельные продукты, на которые опирается ваше видение.

### 3.4.3. Контролируйте состояние опор

Определив опоры, не забывайте следить за ними. Вам необходимо видеть, как они развиваются и внедряются, и понимать, какое место они занимают в вашем видении, чтобы их развитие не приводило к проблемам. Если вы правильно определили опоры — выбрали то, что не является частью вашего бизнеса, но от чего он зависит — значит, в какой-то момент кто-то попытается превратить их в товар.

#### ЗАМЕТКИ С ПОЛЕЙ

Облачный подход, или, другими словами, коммодитизация (commoditization), — это этап развития процесса или технологии, на котором они становятся достаточно значимыми, инновационными и конкурентоспособными, чтобы превратиться в общедоступный продукт. Два хороших примера — веб- и почтовые серверы. Также в последнее время создавать и поддерживать собственный кластер серверов баз данных стало дороже, чем заказать БД-как-услугу (database-as-a-service) у одного из крупных облачных провайдеров, то есть базы данных тоже перемещаются в облако. Раньше у вас была целая команда администраторов баз данных, которая занималась резервным копированием, оптимизацией и аварийным восстановлением, а сейчас все это можно сделать одним щелчком мыши в браузере. Некоторые считают, что при этом утрачивается контроль, но на самом деле это означает, что теперь есть целый ряд технологий, которыми больше не нужно управлять.

Поговорите с любым (традиционным) архитектором, и он скажет, что хороший фундамент — тот, который доступен. Если вы обнаружите, что одна из ваших опор претерпевает значительные изменения, выделите отдельную рабочую группу, чтобы выяснить, подходящее ли сейчас время вносить эти изменения. Рабочие группы создают на короткий срок и включают в них несколько человек, которые подробно изучают определенную узкую область, чтобы вы могли принять более обоснованное решение. Если система спроектирована грамотно, внедрение изменений должно пройти относительно безболезненно и без больших усилий (в последующих главах я расскажу, как защитить проект с помощью архитектуры, устойчивой к изменениям).

При разработке долгосрочных инициатив обычно принято делать несколько шагов назад, чтобы разогнаться для прыжка. Тем не менее так стоит делать, только если речь идет о больших и фундаментальных частях проекта. Замена фреймворка или языка программирования происходит не так, как замена лежащих в основе проекта технологий, о которых говорилось выше. Она несет больше рисков и может поставить под угрозу успех проекта, поскольку целые большие части придется разрабатывать заново. Подобные решения также чрезвычайно важны на этапе планирования проекта, но они принимаются на более высоком уровне. О том, как их принимать в своей команде, рассказывается в следующей главе.

### 3.4.4. В поисках простоты

Надеюсь, что одной из целей своего видения вы выбрали снижение общей сложности и накладных расходов. Необходимо изучить и использовать все возможности, приближающие вас к этой цели. Никуда не годится, если в результате изменений потребуется выделять больше ресурсов на поддержание работоспособности системы, не имеющей отношения к основному бизнесу.

#### ЗАМЕТКИ С ПОЛЕЙ

##### *Невозвратные затраты*

Любой хороший игрок в покер скажет: если вы видите, что ваша рука не выигрывает, — выходите из игры, и неважно, сколько денег вы поставили. В бизнесе это называется *невозвратными затратами*. Да, вы вложили много ресурсов и поэтому хотите довести дело до конца, даже если понимаете, что шансы на успех невелики. Я видел такое много раз, обычно все заканчивается переходом либо на AWS, либо на Azure. Многие команды разрабатывали сложные и трудоемкие системы, но потом появлялся AWS с готовым сервисом — и всю работу приходилось выкидывать. Возникает вопрос: следует ли команде в такой ситуации продолжить разработку или же изменить план? Вашей первой реакцией должно быть следование первоначальному плану, тем временем присмотритесь к сервису и убедитесь, что в нем есть все, что вам нужно. После этого вы сможете перейти на готовый сервис, таким образом уменьшив операционную нагрузку, и затем перейти к следующим большим задачам. Чем меньше технологий вам нужно обслуживать — тем проще жизнь.

Во времена интереса к большим данным популярным инструментом был Apache Hadoop, распределенная система для управления большими наборами данных на нескольких серверах. Это был отличный инструмент, но настоящий кошмар для управления и обслуживания. Тем не менее только с его помощью можно было управлять наборами данных, которые не помещались на одном жестком диске. Навыки работы с Hadoop были настолько востребованы, что их обладатели просили очень много денег. Но эта специализация исчезла в одночасье, когда поставщики облачных услуг стали предлагать «большие-данные-как-сервис» (например, сервис Amazon S3). Хотя это был огромный удар для тех, кто специализировался на работе с Apache Hadoop, компании, которые стремились избавиться от накладных расходов на Hadoop, получили большие преимущества.

Рано или поздно все меняется. На момент написания этой книги отрасль переживала очередное кардинальное изменение в области управления серверами — даже эта базовая сфера перестала быть необходимой: «бессерверные вычисления» подразумевают отсутствие необходимости управлять какими-либо серверами вообще. Многие внутренние проекты теряют актуальность по мере того, как бессерверная архитектура получает все более широкое признание.

Сильное видение приветствует изменения и воспринимает их как выгоду в плане общего снижения сложности.

### 3.5. ПРЕЗЕНТАЦИЯ ДЛЯ ЛИФТА

Что вы должны быть способны делать, не задумываясь, — так это объяснить свое видение любому, кто о нем спросит, кратким и понятным языком. Представьте сотрудников с разным набором навыков и опытом в вашей компании. Подумайте, как объяснить им ваше видение так, чтобы они впечатлились и поддержали вас в его реализации, ведь, скорее всего, рано или поздно их оно тоже затронет. Тем не менее перевод с языка гиков на общепотребительный для кого-то может стать очень трудной задачей. Однажды освоив это умение, вы сможете использовать его снова и снова во всех аспектах роли технологического лидера (особенно при попытках объяснить своим родителям, чем вы занимаетесь; «он работает с компьютерами» — так моя мама говорит своим друзьям).



#### ИСТОРИЯ

##### Презентация для лифта

Единого мнения о происхождении этого термина нет. Некоторые утверждают, что он возник в 1990-е годы; другие — что в 1970-е, а одна история относится к 1850-м годам, когда инженер Отис разработал лифт собственной конструкции. Лично мне наиболее правдоподобной кажется версия с журналистами *Vanity Fair*, которым никак не удавалось обсудить свои идеи статей с главным редактором. Чтобы решить эту проблему, они подкарауливали ее в лифте и там рассказывали ей свои соображения.

В бизнесе существует понятие *презентация, или питч, для лифта (elevator pitch)*. Это краткое, примерно на 30 секунд, объяснение чего-либо так, чтобы любой слушатель понял суть. Оно часто используется в стартапах для оттачивания презентации идеи инвесторам, чтобы убедить их вложить в нее средства.

В этом разделе обсуждается, как составить хорошую презентацию для лифта, чтобы она не казалась притянутой за уши или неестественной. Хотя мы называем это «презентация», или «питч», по сути это ваш план, потому что, скорее



всего, к тому времени, когда вам придется объяснять его другим, его уже одобрит генеральный директор или совет директоров, и вы будете уже заниматься его планированием и реализацией.

### 3.5.1. Подготовка речи

Когда вы спрашиваете нового технического директора, каково его видение, часто в ответ он бросает на вас испуганный взгляд, а затем начинает извергать фонтан красноречия и сыпать технологическим жаргоном. Попробуем что-нибудь с этим сделать.

Проблема здесь не в содержании, а в форме — как представить свое видение естественным, нескучным и, главное, привлекающим внимание способом. Следуя нижеприведенному плану, вы упорядочите свое видение и заинтересуете даже самого искушенного в технологиях собеседника:

1. *Приведите до пяти преимуществ вашего видения для сотрудников или клиентов.* Это выгоды, которые бизнес или клиент получают после реализации вашего видения, например больше аналитических данных для клиента, чтобы проводить сравнительную оценку показателей по годам.
2. *Приведите до пяти особенностей вашего видения, важных для внутреннего пользования.* Составьте список особенностей, ориентированных на технологии и важных для компании или даже вашего отдела; например, сокращение накладных расходов и одновременное расширение возможностей управления данными за счет переноса базы данных в облако.
3. *Отложите этот список хотя бы на одну ночь* — это важно; на обдумывание списка требуется время, поэтому дайте мозгу возможность обработать его, пока вы спите.
4. *У вас есть 10 пунктов, теперь уберите все повторяющиеся или подразумеваемые, чтобы их осталось шесть.* Если вы выполнили шаг 3, это будет легко, потому что список будет выглядеть совершенно иначе, когда вы посмотрите на него свежим взглядом. Найдите, в каких пунктах на самом деле говорится об одном и том же разными словами или подразумевается одно и то же (например, переход в облако подразумевает снижение зависимости от собственного оборудования).
5. *Отложите получившийся список из шести пунктов хотя бы на час.* Последняя проверка позволит убедиться, что остались именно те шесть пунктов, которые нужны.
6. *Расположите шесть пунктов в порядке важности.* Теперь расположите их в порядке важности для бизнеса.
7. *Сформулируйте каждый пункт техническим и нетехническим языком.* Начните с технической версии, потому что она для вас более естественна, а затем напишите вторую формулировку, ориентированную на бизнес, в которой не будет специальных терминов и жаргона.



8. *Придумайте захватывающее начало* — найдите такую вступительную фразу, которая увлечет собеседника и заставит его слушать вас следующие 20 секунд, пока вы проходите по списку. Это может быть формулировка проблемы, которую вы хотите решить («Нам нужно подключать клиента за несколько часов, а не за месяцы, как сейчас»), или отсылка к известному понятию («Представьте, что вы — Uber в сфере страхования имущества, поскольку предоставили клиентам возможности взаимодействовать с HomeMax самостоятельно»).

Вот так просто. Финальный список вас устраивает? Ничего важного не упустили? Теперь пусть все это немного отлежится, и можно переходить к практике.

### 3.5.2. Практика и оттачивание навыка презентации

Пока вы не расскажете презентацию вслух — вы не узнаете, как она звучит. Попросите кого-нибудь, кому вы можете доверять (как людей из мира технологий, так и далеких от него), чтобы они вас послушали. Первые несколько раз у вас будет получаться плохо. Не расстраивайтесь — успех придет с практикой.

Начало речи должно быть одинаковым, но в зависимости от аудитории вы можете выбирать или техническую, или бизнес-версию каждого пункта. Сначала все, что вы говорите, будет казаться вам неестественным и надуманным. Но с каждым повторением слова будут звучать все более знакомо. Каждый раз вы будете находить немного разные способы сказать одно и то же в зависимости от того, как реагирует человек, к которому вы обращаетесь.

Чем больше вы будете практиковаться, тем меньше будете похожи на робота, читающего готовый текст. Начнет проявляться ваша индивидуальность, а ваша увлеченность проблемой станет очевидной для тех, кто вас слушает. Тренировка избавит вас от паники перед первым выступлением и позволит расслабиться, поэтому, когда вы начнете рассказывать, вы найдете способ увлечь слушателя так, чтобы он захотел узнать больше.

Если слушатель окажется заинтересован так же, как и вы, будет задавать вопросы и уточнять детали, это будет означать успех. Ваша презентация привлечет людей, и они будут поддерживать вас столько, сколько нужно для осуществления плана. Вот, например, какой может быть презентация для HomeMax:

*Позволив клиентам управлять своими данными с мобильного устройства или в браузере, мы снизим общую нагрузку на колл-центр и одновременно повысим качество данных и вовлеченность клиентов. Перейдя к масштабируемой современной архитектуре, мы откроем возможности для более тесной интеграции с партнерами, чтобы предлагать дополнительные продукты: например, годовое обслуживание дома, которое позволит своевременно устранять мелкие неисправности, пока они не превратились в большие проблемы.*

В этой очень простой презентации мы сообщили, что собираемся сократить общие эксплуатационные расходы, что мы хотим затронуть одну из самых

проблемных областей, а именно качество данных; а намекая на облако, мы показываем, что движемся в направлении новых рынков и продуктов.

## 3.6. СОСТАВЛЕНИЕ БЮДЖЕТА

Одна из самых сложных задач — это составить бюджет для реализации видения. Бюджет по своей сути — это определение суммы, в которую реализация обойдется компании. Проекты в сфере ИТ и разработки ПО печально известны перерасходом средств, но, если вы будете строго контролировать затраты, бизнес останется с вами на протяжении всего пути создания и внедрения продукта. В этом разделе рассказывается об основных составляющих бюджета и о том, как сохранять прозрачность, чтобы у вас была возможность спокойно реагировать в случае проблем.

### 3.6.1. Что включать в бюджет

Многие бюджеты страдают отсутствием подробного списка статей расходов. Интересно, что чаще всего упускаются из виду расходы на штатный персонал. Логика такова: этим людям уже платят зарплату, зачем включать их в бюджет?

Несложно понять, почему эта точка зрения ошибочна. Дело тут не в том, оплачивается уже работа этих сотрудников или нет, а в том, для чего они изначально были наняты. Если они частично или полностью будут заняты реализацией вашего видения — значит, они, вероятно, не смогут работать над другими задачами. Пострадает ли от этого какая-то область бизнеса, которая перестанет получать нужные ресурсы и поэтому будет нуждаться в дополнительной поддержке? Скорее всего. В любом случае, необходимо учесть расходы на каждого человека, работающего над проектом. Не забудьте про ежегодную индексацию зарплаты (а также другие бонусы, если вы планируете их выплачивать), потому что в большинстве компаний принято каждый год увеличивать заработную плату на несколько процентов (2–5 %, в зависимости от результатов/инфляции).

#### ЗАМЕТКИ С ПОЛЕЙ

##### *Парадокс дефляции облака*

При правильном использовании облако может экономить массу средств. В бюджет проектов, рассчитанных на несколько лет, обычно нужно закладывать инфляцию — явление в экономике, из-за которого большинство товаров и услуг дорожают с каждым годом. В случае облачных сервисов поставщики, напротив, стремятся снижать стоимость услуг. Однако опыт показывает, что экономия здесь обычно теряется по мере увеличения использования. Поскольку мы ожидаем, что облако будет дешеветь — мы используем больше ресурсов, чем нам, скорее всего, нужно, полагая, что все равно уложимся в бюджет. Я называю это парадоксом дефляции облака.

Второе распространенное упущение касается дополнительных ресурсов, необходимых для проекта. Сюда входит оборудование, лицензии на программное обеспечение, сторонние сервисы и облачные ресурсы. Недостаточно просто включить эти статьи в бюджет; они должны изменяться в соответствии с развитием проекта. Например, если вы используете облако, то затраты на него в первые несколько месяцев, скорее всего, будут отличаться от затрат в середине проекта, не говоря уже о его завершении. Тщательно продумайте, когда и как расходы могут измениться.

Вам необходимо представить бизнесу реалистичный прогноз расходов на определенный период времени. Не вздумайте подгонять цифры под те, которых, как вам кажется, ожидает генеральный директор, из-за боязни ценового шока (первой реакции при виде общей стоимости проекта). Ни к чему хорошему это не приведет.

### 3.6.2. Формат бюджета

Бюджет не должен выглядеть слишком сложно; достаточно простой электронной таблицы, где в столбцах будут месяцы, а в строках — ежемесячные расходы. Обсудите первый вариант бюджета с финансовым директором, чтобы убедиться, что вы правильно рассчитали все затраты. Он оценит каждую позицию и подтвердит или скорректирует суммы.

По завершении проверки у вас получится бюджет, максимально точно описывающий все расходы. Теперь подготовьте несколько сценариев «что, если» на случай возражений от генерального директора или совета директоров.

Бизнес, как обычно, будет тянуть одеяло на себя, желая получить все быстро, качественно и дешево. Однако в большинстве случаев одновременно можно выполнить только два из этих условий. Поэтому бизнес должен определиться, чего он хочет больше. Если ему важно, чтобы было качественно и дешево, он должен быть готов ждать. Если ему необходим быстрый и качественный результат, он должен быть готов платить больше либо за высокое качество, либо за большее количество ресурсов. Он не выберет быстро и дешево, так как понимает, что ни к чему хорошему это не приведет, но спросить вас про этот вариант тоже могут.

Модель «быстро — качественно — дешево» подходит для составления двух дополнительных вариантов бюджета, потому что позволяет поэкспериментировать с целями проекта (немного изменить требования). Возьмите первоначальный бюджет и представьте, что вам нужно реализовать все быстрее (и сохранить качество, несмотря ни на что). Что вам нужно, чтобы поставить продукт за более короткий



срок? Увеличить расходы на персонал, возможно, привлечь подрядчиков? Насколько вы действительно сможете ускориться? Вспомните слова Уоррена Баффета: «Девять женщин за месяц ребенка не родят». Некоторые процессы ускорить невозможно. Но будьте готовы ответить, когда вас спросят: если бы деньги не были проблемой, как быстро вы воплотили бы это видение? Этот вопрос часто задают наши друзья из прямых инвестиций, потому что им, как правило, не хватает времени и единственное, что они могут контролировать, — это поток денег. Еще одна область, которую можно обсудить, — это количество требуемых функций: можно ли упростить/удешевить проект, если отказаться от каких-то требований?

Рассмотрите также вариант «быстро — дешево». Речь здесь не всегда идет об экономии бюджета, проблема может быть в ограниченности внутренних ресурсов, поскольку они важны для обеспечения работы других частей бизнеса. Здесь как раз можно предложить удаление части функций, чтобы остаться в рамках бюджетной модели.

Эти варианты позволят вам определить верхние и нижние границы бюджета, а также схему расходов за год (годы). Когда генеральный или финансовый директор будут утверждать бюджет, они смогут совместить ваш проект с другими расходами и определить, когда можно будет дать вам зеленый свет.

### 3.6.3. Контроль выполнения бюджета

Само собой разумеется — но все же стоит это подчеркнуть, — что бюджет не составляется раз и навсегда. После начала работы над проектом по одному из ваших сценариев вам нужно будет актуализировать все бюджеты по крайней мере ежемесячно. Заведите привычку в конце каждого месяца проверять все расходы и счета, чтобы убедиться, что все идет по плану.

Ваш руководитель проектов (убедитесь, что в ПО для управления проектами есть возможность учитывать трудозатраты) точно скажет, кто сколько времени затратил на работу над проектом. Тратит ли кто-то больше времени, чем было запланировано? Также следите за недоиспользованием ресурсов. Вы можете думать, что вы стали тратить меньше, а на самом деле сотрудник был в отпуске или занимался чем-то другим и не выполнил запланированную работу.

Поддерживайте актуальное состояние и для альтернативных сценариев. Вполне возможно, что после запуска проекта и первых успешных результатов руководство воодушевится и генеральный директор попросит вас ускориться. Если такое случится и можно будет увеличить финансирование — вы должны быть в состоянии оценить, как дополнительные средства повлияют на результат. Если вы своевременно обновляете альтернативные бюджеты — вы сможете ответить на этот вопрос. Кроме того, по мере развития проекта запасные варианты могут оказаться более привлекательными, чем они выглядели изначально.

Не реже одного раза в квартал проверяйте текущий бюджет с финансовым директором. Он честно оценит, насколько тот реалистичен. Если вы будете предоставлять генеральному директору и правлению грамотно структурированный

и актуальный бюджет, то даже несмотря на то, что он может и будет меняться, руководство будет доверять вашим цифрам.

### 3.6.4. Окупаемость (ROI)

Нередко, особенно в случае масштабных проектов, бизнес требует оценки окупаемости (ROI, return on investment). Проще говоря, ответа на вопрос: сколько бизнес заработает на каждый потраченный доллар? Оценить это чрезвычайно сложно, но все же возможно.

Для проектов по замене или модернизации устаревших систем в этом случае важно, сколько средств они позволят сэкономить. При расчете включите сюда любые лицензии на программное обеспечение или контракты на поддержку, которые не понадобятся продлевать, экономии на закупке оборудования и, что важно, снижение затрат на управление системами. Этот случай самый простой.

Сложнее с другими большими проектами, предполагающими значительные изменения в работе бизнеса и открытие новых рынков, особенно если вы готовите бизнесу стартовую площадку для расширения или углубления продуктовой линейки. В этих случаях обратитесь к коллегам за помощью в оценке снижения затрат или увеличения доходов.

Иногда возникает коллективное внутреннее ощущение, что проект следует реализовать, в таких случаях оценка ROI не имеет большого значения. Здесь подразумевается единогласная поддержка со стороны менеджмента.

В случае же с нашей компанией HomeMax окупаемость инвестиций можно рассчитать на основе сокращения числа сотрудников колл-центра и снижения затрат на приобретение, обслуживание и размещение физического оборудования.

## 3.7. КРАТКОСРОЧНОЕ ПЛАНИРОВАНИЕ И ОЖИДАНИЯ

Начинать крупный проект всегда сложно. Все взволнованы, все ждут больших результатов, но проблема в том, что каждое большое начинание требует серьезной подготовки. По большей части она затрагивает вспомогательные системы (структуру разрабатываемого кода и организацию работы с ним, использование облачных сервисов, базовую архитектуру и т. п.), и видимых (по крайней мере, для бизнеса) результатов может не быть 2–3 месяца, поэтому бизнес начинает нервничать.

Здесь мы стараемся предотвратить момент, когда бизнес устает ждать — забыв, что проект долгосрочный и должен был занять много времени (или если началось отставание от сроков), — и начинает задаваться вопросом, стоит ли продолжать. Этот феномен, «долина страданий» (названный так Джеймсом Милбери, партнером-основателем ParkerGale Capital и описанный в главе 2), отражает отношение бизнеса к долгосрочным проектам, и если вы будете следить за своим положением на кривой, то сможете облегчить эти страдания.

В это время в бизнесе могут появиться скептики, нашептывающие генеральному директору, что они потеряли (или начинают терять) уверенность в успехе

проекта. Старая добрая система начинает вдруг казаться чрезвычайно привлекательной — старый друг лучше новых двух и все такое.

### ЗАМЕТКИ С ПОЛЕЙ

*Каждому дому нужен прочный фундамент*

Как правило, я справлялся с подобными ситуациями, проводя аналогии со строительством дома. Бизнес обычно интересуется, что будет в каждой комнате, как украсить помещения и как обустроить кухню. Я умеряю его пыл, говоря, что мы пока на этапе закладки фундамента. После того как этот этап пройден, я говорю, что мы на стадии прокладки труб и электропроводки. По мере дальнейшей работы над проектом я продолжаю сравнивать его этапы с этапами строительства: возведением стен, установкой дверей и т. п. Это создает реалистичные ожидания и помогает тем, кто не знаком с разработкой ПО, понять, что происходит.

Ваша задача здесь — не позволить бизнесу свернуть в долину страданий, а если он все же там окажется — показать ему, как выбраться оттуда. Посмотрим, как это сделать.

### 3.7.1. Коммуникация

Прежде всего, необходимо следить за ходом проекта и ничего не упускать из виду. У подавляющего большинства крупных проектов сроки превышаются — такова жизнь, — но не превышение сроков, а отсутствие фиксации их причин и должной коммуникации может привести вас и вашу команду к провалу.

Бизнес должен знать, что происходит и почему. Многие технические директора не любят сообщать плохие новости и склонны замалчивать отставание от графика. Это неправильное отношение — смотрите на бизнес как на партнеров, которые так же заинтересованы в успехе, как и вы. Затруднения обычно возникают из-за появления новых требований или необходимости отвлекаться на повседневную рутину. О таких вещах нужно обязательно сообщать, и тогда помощь может прийти оттуда, откуда вы ее совсем не ждали.

Не забывайте отмечать каждый успех, но будьте аккуратны с масштабами торжества. Обязательно отмечайте небольшие достижения, например успешные спринты, внутри команды, но слишком хвастать прогрессом перед бизнесом не стоит — просто сообщите, что все идет хорошо. Отпраздновать по-крупному можно будет, когда вы закончите что-то, от чего бизнес начнет получать выгоду. Именно тогда ваш проект начнет доказывать свою ценность (и заставит замолчать скептиков).

Кстати, о ценности для бизнеса — не забудьте заложить в плане проекта достаточно времени для обучения пользователей работе с вашим продуктом и для их поддержки. Неприятно, когда все запланированные ресурсы израсходованы уже несколько недель назад, а вам все еще нужно внедрять продукт и решать

различные проблемы. Все это следует предусмотреть и запланировать заранее. Тогда худшее, что может произойти, — это оставшееся неиспользованным запланированное время, что для бизнеса просто праздник. Таким образом, вы сдадите проект раньше срока — что намного предпочтительнее.

Разработав план коммуникации для разной целевой аудитории (например, для высшего руководства можно предусмотреть другой уровень детализации, чем для остальной компании) и определив периодичность его публикации, вы выработаете привычку своевременно сообщать как хорошие, так и плохие новости. Это может быть просто электронная рассылка с перечислением основных этапов или 1-2 слайда на регулярном собрании руководства.



### 3.7.2. Контроль инфополя

Звучит как элемент политической кампании и действительно похоже на нее в части контроля распространения информации. В отсутствие информации люди сами заполняют пробелы — когда фактами, а когда и фантазиями. Чем масштабнее ваш план, тем больше вам придется ограничивать радужные ожидания от него.

В какой-то момент ваше видение заживет собственной жизнью. Это означает, что ваш проект будут считать решением всех проблем. Что-то пошло не так, клиент недоволен, нужно увеличить продажи — ваше видение поможет. Перспектива избавиться от всех проблем очень заманчива, и вы будете удивлены, как быстро она может захватить умы. В управлении проектами это приводит к так называемому *раздутию функционала* (feature creep). Этим понятием описывается добавление в проект множества мелких функций — по отдельности они не кажутся большими, поэтому их можно впихнуть незаметно. Подобно соломинкам, которые кладут на спину верблюду, пока она не сломается, раздутие функционала может уничтожить самый лучший проект — ко времени завершения он и близко не будет соответствовать всем ожиданиям, даже если вы достигнете изначально поставленных целей.

Хотя строгое соблюдение стандартных принципов управления проектами поможет решить эту проблему, позаботьтесь, чтобы необходимая информация распространялась и за пределами вашей команды. Один из способов сделать это — при выпуске каждого обновления напоминать людям, что будет реализовано в рамках проекта, а что не будет. Нет ничего плохого в том, чтобы сразу развеять чьи-то заблуждения. Особенно важно уточнять ожидания в неофициальных кругах. Лучше сразу вызвать небольшое разочарование тем, что какой-то



дополнительной функции не будет в первом релизе, чем потом спасти проект от полного провала в попытках соответствовать всем ожиданиям, и все из-за того, что вы были слишком вежливы, чтобы указать на недопонимание.

## ИТОГИ

- Сформулировав видение, направленное на создание долгосрочной ценности, вы сможете возглавить любые изменения.
- Выделите в видении части «что» и «как» для разных целевых аудиторий.
- Хорошее видение включает показатели успеха, которые легко измерить.
- Узнайте своих клиентов и установите с ними партнерские отношения, чтобы выявить их болевые точки и тем самым подтвердить актуальность своего видения.
- Видение — это не краткосрочный проект; это совокупность проектов, реализация которых иногда занимает годы.
- Согласуйте свое видение с общим графиком компании, чтобы его было проще реализовывать.
- Учитывая, сколько времени требуется для реализации видения, крайне важно обеспечить его адаптируемость к меняющимся технологиям.
- Сотрудникам всех уровней будет легче понять, к чему вы их ведете, если вы изложите свое видение в краткой и убедительной форме — для этого необходима практика.
- Чтобы исключить проблемы с финансированием в будущем и сформировать правильные ожидания, спланируйте бюджет своего видения.
- Составьте план коммуникации, чтобы своевременно информировать всех заинтересованных лиц — это поможет предотвратить сомнения, когда начнет казаться, что путь еще слишком долг.

## ПРОВЕРЬТЕ СЕБЯ

Что из перечисленного ниже вы уже сделали?

- Создали долгосрочное видение и поняли, как его реализовать.
- Представили свое видение на разных уровнях компании.
- Подробно рассказали бизнесу о преимуществах, которые принесет ваше видение.
- Определили примерные сроки и результаты реализации вашего видения.
- Определили, какие технологические усовершенствования необходимо провести, чтобы реализовать ваше видение.
- Подготовили «презентацию для лифта» и с ее помощью представили свое видение другим.
- Можете в любой момент предоставить подробный план, а также отчет о том, как продвигается реализация вашего видения.



# 4

## Создание команды

---

### В этой главе

- ✓ Способы формирования команды
- ✓ Оценка нужных ресурсов
- ✓ Как понять, что пора привлекать новых сотрудников
- ✓ Где вести поиск кандидатов
- ✓ Как подготовить описание вакансии, которое привлечет наиболее талантливых соискателей

Как бы хорошо и как бы быстро вы ни работали — в какой-то момент вам понадобится помощь других людей. Вы можете привлечь сторонние организации или нанять собственную команду — вариантов тут множество. В современном мире над созданием и запуском вашей платформы могут работать вместе и ваши штатные сотрудники, и подрядчики, и контрактные работники, в том числе и из других стран.

Но как узнать, кто лучше подходит для работы над конкретным проектом? Это непростая задача, поскольку для ее решения необходимо учесть множество переменных, включая доступность и опыт сотрудников, бюджет, а также то, какую работу нужно выполнить.

Эта глава проливает свет на то, как организовать подбор персонала. Один совет: не относитесь к подбору сотрудников как к чему-то, чем придется заняться лишь однажды; сделайте его частью своей повседневной работы, считайте, что вы

всегда находитесь в поиске людей. Вы никогда не знаете, в какой момент кто-то соберется покинуть вашу команду или вам придется быстро масштабироваться; кадровый резерв — прекрасный инструмент, который лучше держать наготове. Самые успешные технические директора быстро находят сотрудников, потому что у них большой пул кандидатов.

Подготовив источники для поиска, составьте качественное описание вакансии, чтобы потенциальный кандидат воодушевился, прочитав его; часто это оставляют отделу кадров, который использует унылые стандартные шаблоны. Но если вакансия не заинтересует вас самих — других она не привлечет тем более. В этой главе мы расскажем, как сделать описание вашей вакансии сногсшибательным. Вы сильны настолько, насколько сильна ваша команда — поэтому позаботьтесь о ее будущем росте уже сейчас.

## 4.1. ТИПЫ РЕСУРСОВ

Набор людей в штат — не единственный способ привлечь дополнительные ресурсы. В этом разделе рассматриваются механизмы, которые успешные технические директора применяют для создания команды инженеров как для краткосрочных, так и для долгосрочных проектов. Любой финансовый директор скажет вам, что прием сотрудника в штат на полный день — это дорогостоящая затея, которая требует аргументов, подготовки и должна быть обоснована с цифрами в руках. Позже в этой главе мы рассмотрим некоторые метрики, показывающие, брать ли человека в штат. В целом доступные ресурсы можно разделить на пять типов:

- штатные сотрудники;
- сотрудники, работающие по договору подряда;
- компании-подрядчики;
- местный аутсорсинг;
- зарубежный аутсорсинг.

У каждого из них есть свои плюсы и минусы, и выбор в значительной степени зависит от бюджета и сроков подбора, а также от того, насколько широкий доступ к квалифицированным кандидатам вы имеете. Давайте более подробно рассмотрим эти типы.



### **4.1.1. Штатные сотрудники**

Штатные сотрудники — вероятно, самый распространенный тип ресурса — это член команды, который получает заработную плату в компании, стремится к достижению глобальных целей организации и должен приносить нечто большее, чем просто выполнять свою работу. Штатные сотрудники развиваются сами и помогают развивать других членов команды. Это ваш самый ценный и доверенный актив, потому что они переходят с вами от одного проекта к другому и помогают управлять другими сотрудниками. Они заинтересованы в поддержании высокого качества и в том, чтобы клиенты были довольны, а также в обеспечении прибыльности компании.

Человек в штате отлично подходит, когда вам нужен кто-то для прямого взаимодействия с клиентами или на долгосрочную перспективу, а также чтобы участвовать одновременно в нескольких проектах. Это идеальный вариант, если вам нужен сотрудник, который поможет сформировать команду или управлять ею или же определять направление развития компании. Штатные сотрудники получают фиксированную заработную плату (в Соединенных Штатах она обычно включает медицинскую страховку / пенсионные взносы) и дополнительный ежегодный бонус, привязанный к результатам работы. В зависимости от размера бизнеса у штатного сотрудника может быть возможность получить опционы или акции компании. Эти работники также защищены федеральным/региональным/местным (в зависимости от того, где находится компания) трудовым законодательством и должны проходить ежегодную оценку персонала, от результатов которой зависят возможности карьерного роста.

Вы инвестируете в этих людей и помогаете им расти в надежде, что они будут полезны компании и таким образом эти инвестиции принесут прибыль. И наоборот, увольнение штатного сотрудника требует обоснования, соблюдения установленных процессов/законов и может занять определенное время. В главе 5 также более подробно рассматривается, как проводить собеседования, как заинтересовать соискателя работой в вашей компании и как адаптировать новых сотрудников.

### **4.1.2. Сотрудники, работающие по договору**

Работник по договору — это сотрудник, работа которого оплачивается по определенной ставке (подневной или почасовой) и который привлекается для конкретной задачи либо проекта. Таких работников можно также подбирать через кадровые агентства или через специализированные организации, которые берут все расчеты с исполнителями на себя, а вы будете взаимодействовать только с компанией, выставляющей вам счета каждую неделю или месяц.

Такой договор можно расторгнуть очень быстро и по любой причине (проект завершился, или же эта специальность больше не нужна), если, конечно, не был согласован минимальный срок договора. Если агентство предлагает подписать договор на определенный период — можете торговаться. Как любой бизнес, они

стараясь удерживать клиентов как можно дольше, и у них есть свои приемы, помогающие добиваться продления договоров.

Работник по договору имеет временный статус: он покидает компанию после завершения его работы, и после этого его обычно не приглашают на собрания или мероприятия компании.

Многие думают, что все кандидаты, которых направляет агентство — перво-классные специалисты, и поэтому не проводят собеседования с ними, прежде чем подписывать договор. Это большая ошибка. С такими кандидатами тоже нужно проводить собеседование; однако социальные навыки (soft skills) для них не так важны, как профессиональные. Вы привлекаете этих людей для выполнения конкретной работы, поэтому убедитесь, что они в состоянии выдавать результат с самого первого дня. Если работник будет обладать большой степенью самостоятельности — обязательно запрашивайте и проверяйте рекомендации. Также учитывайте, что в некоторых странах законы запрещают заключать договор на долгий срок и с широким кругом обязанностей, поскольку это рассматривается как уклонение компании от оформления сотрудников в штат и невыполнение связанных с этим требований.

Такой работник идеально подойдет, если необходимо выполнить конкретную задачу, например разработать пользовательский интерфейс для мобильного приложения или перенести данные из нескольких БД в другое хранилище. Такие задачи ограничены по времени, и для них требуется конкретный набор навыков, который не нужен за пределами этой задачи.

У работников по договору есть особый подвид: работа по договору перед оформлением в штат. Это своего рода пробный период: если через какое-то время вы поймете, что человек работает хорошо и у вас есть для него постоянная должность, вы можете перевести его в штат (для этого, возможно, понадобится дополнительно заплатить агентству). Это может оказаться отличным вариантом для компаний, которые еще не определились, нужен ли им на этой позиции человек на полный день.

### 4.1.3. Привлечение подрядчиков

Раньше это было прерогативой крупного бизнеса, но цены со временем снижаются, и сейчас маленькие компании тоже могут позволить себе использование подрядчиков. Вы отдаете сторонней компании все, что связано с определенной областью: например, обслуживание рабочих мест, принтеров и локальной сети; управление базами данных; в последнее время это может быть управление облаком и обеспечение безопасности.

Подрядчик сам обеспечивает себя сотрудниками и обучает их, а также организует бесперебойное оказание услуг. Вам не придется беспокоиться о нехватке людей и связанных с этим проблемах. Контракты с подрядчиками обычно заключаются на длительное время (6–12 месяцев) и содержат соглашения об определенном уровне обслуживания, который они должны обеспечивать.

Этой схеме работы посвящена целая глава, и при правильном использовании такая схема может сэкономить компании значительные суммы, избавив ее от накладных расходов на развитие и обучение команды для создания внутренней экспертизы. Подрядчикам стоит передавать те задачи, которые вы можете просто закупать (например, электроснабжение или обеспечение работы офиса) и которые хоть и важны для бизнеса, но не требуют обязательного создания внутренней команды.

#### **4.1.4. Местный аутсорсинг**

Местный аутсорсинг — это когда вы привлекаете консалтинговую компанию, которой передаете одну или несколько частей проекта. Как правило, эта компания находится в одном городе с вами или, по крайней мере, в той же стране, что и вы. Это расширенный вариант работы по договору, отличающийся тем, что у такой компании есть собственные ресурсы, которые находятся в ее офисе, но работают большую часть времени на вас. Обычно вам назначается менеджер по работе с клиентами или руководитель проекта, который поддерживает связь с вами и вашей командой, договаривается с вами, что нужно сделать, и следит, чтобы работа выполнялась в соответствии с договоренностями.

Компания обычно располагает большим набором специалистов и таким образом может обеспечить все потребности вашего проекта. Это позволит вам получить доступ к нужным экспертам без необходимости предварительного планирования.

Такой подход идеален для автономных проектов с четко заданными целями и результатами. Если проект крупный, компания может выделить сотрудников, которые будут работать на вашей территории на протяжении всего проекта и таким образом смогут быстро реагировать на изменения в требованиях. Это обойдется дороже, но может быть выгодно, если у вас в команде нет выделенного человека для управления работой и организации взаимодействия.

Работа по такой схеме, в зависимости от сложности проекта, оплачивается либо единовременно за весь проект, либо ежемесячно. Большим преимуществом этого типа является то, что вы сможете использовать сторонние ресурсы и вся работа будет организована за вас, при этом результат будет принадлежать вам. Ну а недостаток заключается в том, что, хотя вы получите все права интеллектуальной собственности (ПИС) на готовый проект, вы не сможете ничему научиться в процессе его создания.

Поиск подходящей консалтинговой компании требует времени и усилий. Если у вас есть четкое техническое задание (не обязательно слишком подробное), лучше всего провести тендер — то есть запросить у разных компаний котировки на основе вашей документации.

Также можно поспрашивать коллег. Если это новая для вас сфера — можете мониторить посвященные поиску работы группы в соцсетях и вакансии для студентов, чтобы увидеть, какие компании их размещают. Крупные и авторитетные

компаниям имеют региональные представительства и, как правило, работают хорошо, потому что не горят желанием подпортить репутацию у себя дома, ведь слухи распространяются быстро.

При проведении тендера не соблазняйте самой низкой ценой, особенно если ценовые предложения значительно различаются. Получите как минимум три-пять заявок и сразу отбросьте те из них, цена которых отличается от остальных более чем на 20 %. Запросите несколько резюме людей, которые будут работать на вас. Скорее всего, вы увидите их в обезличенной форме, без указания персональных данных (в целях защиты сотрудников компании-исполнителя), но все же получите представление об опыте и навыках людей, которые будут работать над вашим проектом.

### 4.1.5. Зарубежный аутсорсинг

Последний тип ресурсов, на который стоит обратить внимание, — это зарубежные компании, как правило, работающие в Индии, Восточной Европе и Латинской Америке. Стоимость их рабочей силы значительно ниже по сравнению с американскими компаниями. Это может быть очень соблазнительно для организаций, стремящихся сократить расходы, но, как говорится, бесплатный сыр бывает только в мышеловке, и при выборе этого варианта следует учесть возможные подводные камни.

Первое, что нужно понять: у вас будет не только удаленная, но и, скорее всего, работающая в другом часовом поясе команда. Если разница во времени значительная — вы можете потерять целый день, если возникнет проблема или потребуется согласование, без которого нельзя двигаться дальше. В этом случае вам понадобится связующее звено между локальной и удаленной командами, и этому человеку придется жить и работать не по местному времени, а по времени другого часового пояса.

Второе — сложность в выборе подходящей компании. Вы должны доверять исполнителю и быть уверенными, что получите то, за что платите. Скорее всего, вы не поедете в компанию с проверкой, потому что это будет очень дорого и займет много времени. Вам придется полагаться на рекомендации коллег и отзывы на таких сайтах, как LinkedIn. Так что готовьтесь целовать лягушек до тех пор, пока не найдете свою принцессу.

Здесь важно не иметь завышенных ожиданий. Не ждите, что удаленная команда будет работать так же тщательно и уделять такое же внимание деталям, как ваша команда в офисе. Дело не в том, что она плохая или недобросовестная; просто ей не хватает актуального контекста, который влияет на продукт.

Начните ваше сотрудничество с небольших и измеримых задач — это поможет оценить качество и время выполнения, чтобы составить более подробный план проекта.

По моему опыту, привлечение зарубежных исполнителей бывает успешным, когда проектирование завершено и у вас уже есть готовая архитектура, а также

вся инфраструктура для автоматической проверки качества кода, полученного от удаленной команды. Еще одна область, в которой часто используются зарубежные ресурсы, — это тестирование, особенно для проектов, в которых невозможна автоматизация тестирования и требуется ручная работа.

Будьте готовы к тому, что на управление удаленной командой уйдет гораздо больше времени, чем вы ожидаете. Неудачи здесь часто происходят по причине отсутствия надлежащего надзора и управления со стороны заказчика.

### 4.1.6. Сравнительный обзор

В табл. 4.1 собраны воедино основные преимущества и недостатки каждого типа ресурсов.

**Таблица 4.1.** Сравнительный обзор ресурсов

Тип	Сильные стороны	Слабые стороны	Способ оплаты
Штатный сотрудник	<ul style="list-style-type: none"> <li>Нанимается на длительный срок</li> <li>Может взаимодействовать с клиентами</li> <li>Стремится к достижению глобальных целей компании</li> <li>Предан интересам компании</li> <li>Присутствует лично</li> </ul>	<ul style="list-style-type: none"> <li>Со временем обходится все дороже</li> <li>Сложно уволить</li> <li>Может привести к чрезмерной зависимости компании от конкретного сотрудника</li> </ul>	<ul style="list-style-type: none"> <li>Фиксированная зарплата</li> <li>Медстраховка + пенсия</li> <li>Опционы / бонусы</li> </ul>
Сотрудники, работающие по договору	<ul style="list-style-type: none"> <li>Подходит для краткосрочных проектов с конкретными результатами</li> <li>Можно быстро расторгнуть договор</li> <li>Отличный вариант, когда возникает краткосрочная потребность в специалисте с определенным набором навыков</li> <li>Простая схема оплаты</li> <li>Присутствует лично</li> </ul>	<ul style="list-style-type: none"> <li>Отсутствие лояльности и личной заинтересованности в планах компании</li> <li>Отсутствие подхода, ориентированного на достижение долгосрочных целей; сосредоточены на текущем проекте</li> </ul>	<ul style="list-style-type: none"> <li>Фиксированная почасовая/суточная ставка</li> </ul>
Привлечение подрядчиков	<ul style="list-style-type: none"> <li>Долгосрочное предоставление комплексных услуг</li> <li>Непрерывное обслуживание, не зависящее от ротации сотрудников подрядчика</li> <li>Не нужно обучать исполнителей</li> </ul>	<ul style="list-style-type: none"> <li>Не создается единая команда</li> <li>Могут делать для вас только то, что они умеют</li> <li>Заказчик не приобретает собственный опыт в данной предметной области</li> </ul>	<ul style="list-style-type: none"> <li>Фиксированная оплата по договору</li> </ul>

Тип	Сильные стороны	Слабые стороны	Способ оплаты
Местный аутсорсинг	<ul style="list-style-type: none"> <li>Подходит для долгосрочных проектов, имеющих конкретные рамки</li> <li>Привлечение целой команды, а не отдельных исполнителей</li> <li>Подходит, когда требуется целый ряд специалистов с определенным набором навыков на короткий срок</li> <li>Управление полностью на аутсорсинге</li> </ul>	<ul style="list-style-type: none"> <li>Заказчик не приобретает опыта и не обучается; получает только готовый проект</li> <li>Подходит только для больших проектов</li> <li>Требуется четкое ТЗ</li> <li>Сложно найти/проверить подходящую компанию</li> <li>Отсутствие контроля над отдельными исполнителями</li> </ul>	<ul style="list-style-type: none"> <li>Единовременно за весь проект или периодически</li> </ul>
Зарубежный аутсорсинг	<ul style="list-style-type: none"> <li>Дешево</li> <li>Разница в часовых поясах может увеличить время, когда ваши сотрудники доступны</li> <li>Заклучив договор с подходящей компанией, вы сможете при необходимости быстро увеличивать или уменьшать количество привлекаемых сотрудников.</li> </ul>	<ul style="list-style-type: none"> <li>Проблемы из-за разницы во времени</li> <li>Очень сложно найти/проверить подходящую компанию</li> <li>Сложно разрешать споры</li> <li>Культурные различия могут влиять на качество проекта.</li> </ul>	<ul style="list-style-type: none"> <li>Фиксированная цена за договор/сотрудника</li> </ul>

Мы перечислили общие черты каждого типа ресурсов. Также возможны разнообразные их комбинации как для индивидуальных исполнителей, так и для команд, как местных, так и зарубежных. Прежде чем выбрать тот или иной вариант, убедитесь, что четко понимаете, что вам нужно, и знаете, как найти партнера, который поможет вам добиться успеха. Ну и, конечно же, из любого правила можно найти исключения.

## 4.2. КОГДА НАБИРАТЬ СОТРУДНИКОВ

Подбор сотрудников — дело сложное само по себе, но иногда бывает еще сложнее понять, когда пора его начинать. В этом разделе представлены некоторые показатели и сигналы, которые помогут определить, что это время настало. Хотя в целом стоит всегда руководствоваться данными, а не ощущениями, бывают ситуации, когда вам просто не хватает информации, чтобы подтвердить то, что кажется правильным. Этот раздел покажет, как получить данные, подтверждающие правильность решения, которое подсказывает ваша интуиция (или же, наоборот, свидетельствующие, что сейчас все-таки не самое подходящее время).



Обычно сотрудников нанимают в трех случаях:

- Запуск нового проекта.
- Уход кого-то из команды.
- Ресурсы команды исчерпаны или требуется дополнительная поддержка.

Стоит отметить, что нестабильная экономическая ситуация может привести к нехватке квалифицированных кандидатов. Если вы видите, что оказались в такой ситуации (что легко понять, читая отраслевую прессу) и наткнулись на интересного кандидата, но у вас сейчас нет открытых вакансий, — спросите финансового директора, можно ли выжать что-то из бюджета, чтобы не упустить кандидата. Не спросишь — не узнаешь.

### 4.2.1. Причины найма

Рассмотрим подробнее три основные ситуации, в которых происходит рост вашей команды.

#### **ЗАПУСК НОВОГО ПРОЕКТА**

С точки зрения аргументации, подробная смета проекта — один из самых простых способов определить необходимость привлечения дополнительного персонала. Из плана проекта будет понятно не только, какие ресурсы вам потребуются, но и на какой срок, однако это не решает проблему полностью. Организации редко (что, на мой взгляд, очень правильно) нанимают кого-то в штат для решения только узкоспециализированных задач.

Вместо этого ищите человека с экспертизой в двух-трех основных областях, а также набором второстепенных навыков. Например, предположим, что кто-то отлично разбирается в архитектуре облачных решений и языке Java, а кроме этого, имеет опыт работы с базами данных, организации обмена сообщениями и обеспечении безопасности. В долгосрочной перспективе такой специалист будет очень полезен для команды, потому что не придется привлекать отдельных экспертов в каждой из этих областей.

Определите, какие навыки необходимы для вашего проекта. Выделите те, которые сочетаются друг с другом, распространены на рынке и которыми может владеть один и тот же человек. Так вы убьете сразу нескольких зайцев, и бюджет на такого сотрудника будет гораздо легче защитить перед финансовым или генеральным директором.

Вполне нормально (и желательно) для каждой компетенции включать в команду несколько человек, которые ей владеют. Так вы не только обеспечите кадровый резерв, но и стимулируете командную работу, которая поможет находить лучшие решения. Помните: хороши те идеи, которые прошли проверку критикой. Если в вашей команде нет никого, кто мог бы дать конструктивную обратную связь — это вредит проекту. Ниже я представлю матрицу компетенций — небольшую схему для учета навыков, имеющихся в команде.

Новые проекты — отличный способ расширить команду: как привлечь новые ресурсы, так и повысить или поменять местами существующих людей. Но существующие проекты тоже не должны быть обескровлены из-за перевода сотрудников на новый проект. Не стоит ожидать, что человек будет просто работать и там, и там, хотя любой финансовый или генеральный директор, скорее всего, задаст вам вопрос: «Разве те, кто у вас есть, не могут просто работать больше?» На самом деле это логичный вопрос, и вы должны быть готовы ответить на него, даже если он кажется вам совершенно неуместным. Если вы хорошо представляете себе текущую рабочую нагрузку на команду и объем помощи, который вам требуется, то, исходя из этих данных, вы поймете, нужен ли вам еще один человек. Здесь предполагается, что вы определили — команда полностью загружена, поэтому нужен именно дополнительный сотрудник. Одна из следующих глав посвящена управлению командой и в том числе измерению таких показателей.

Если вы набираете сотрудников исходя из потребностей нового проекта, учитывайте следующее:

- Требуемые знания и личные качества.
- Перспективы по завершении проекта: останется ли сотрудник востребован или в нем больше не будет необходимости?
- Может ли сотрудник принести пользу не только этому проекту, но и другим областям, за которые вы отвечаете?

### ТЕКУЧКА КАДРОВ

В команде, естественно, будут изменения, обусловленные разными жизненными обстоятельствами, такими как выход сотрудника на пенсию, переход в другую компанию или (к сожалению, это то, с чем мне лично пришлось столкнуться) смерть. Надеюсь, вам никогда не придется столкнуться с последним, но вот переход кого-то из членов команды на другую работу — это частое событие в нашей динамичной индустрии. Иногда это происходит внезапно, и у вас будет мало времени, чтобы среагировать — большинство трудовых договоров предусматривают возможность уволиться через две-четыре недели после уведомления, в зависимости от должности и местного законодательства.

Когда вы сталкиваетесь с уходом сотрудника, сделайте небольшую паузу и воспользуйтесь возможностью реструктурировать команду. Вам необходимо решить, как заполнить образовавшийся пробел, и стоит ли его заполнять. Вот о чем стоит подумать:

- *В чем вы главным образом рассчитывали на этого человека?* Назовите здесь две области. Нет необходимости в подробном описании — вы не хвалебную



речь пишете. Подумайте о двух главных областях, в которых вы полагались на этого человека, и о том, как теперь о них позаботиться.

- *Потеря навыков.* Оцените потерю опыта и компетенций. Насколько они были важны для успеха команды?
- *Какие знания о внутренних системах или продуктах для клиентов будут утрачены?* Чем дольше человек работает с вами, тем больше информации о работе ваших технологий он накапливает. Зафиксированы ли все эти знания где-либо, и если нет, то насколько вы зависите от них при решении проблем и помощи клиентам?
- *Увольняющийся сотрудник работал на полную ставку?* Если да, то присмотритесь к этой позиции внимательнее — нужна ли она в действительности, и, даже если человек выглядел полностью загруженным, выполнял ли он полезную работу?
- *Чего не хватало этому человеку, чтобы работать продуктивнее?* Иногда на работу принимают неподходящего человека, не обладающего ключевым навыком или чертой характера, наличие которых сделало бы его более успешным — например, в службу поддержки приходит сотрудник, который не любит разговаривать с клиентами.
- *Влияние на остальную команду.* Подумайте, положительно или отрицательно повлияет уход этого человека на остальную команду? Вдохновлял ли он других или же, наоборот, вызывал больше проблем, чем решал?

Хотя увольнение одного из сотрудников — это потеря, требующая вашего внимания обычно в самый неподходящий момент, не позволяйте ему заставить вас врасплох и будьте готовы рассмотреть все возможные варианты действий. Порой стоит не искать замену сразу, а немного подождать. В хорошо организованной команде уход одного человека в краткосрочной перспективе не должен оказать значительного влияния.

### **БУДЬ ГОТОВ**

*Не «если», а «когда»*

Любая команда, даже максимально эффективная и лояльная, подвержена изменениям. Как сказал Авраам Линкольн: «Это тоже пройдет». Ничто не вечно. Много раз в своей карьере я думал, что создал «команду мечты», но каждый раз возникали обстоятельства, которые я не мог контролировать, и мне приходилось создавать новую «команду мечты».

Увольнение — это возможность взвесить, что вы потеряли, и определить, как дальше развивать команду. Не стоит пытаться искать равнозначную замену. Неважно, насколько хорош (или плох) ушедший человек, вы никогда не найдете второго точно такого же, и будет недальновидным оценивать нового сотрудника, сравнивая его с предыдущим.

### **РЕСУРСЫ КОМАНДЫ ИСЧЕРПАНЫ ИЛИ ТРЕБУЕТСЯ ДОПОЛНИТЕЛЬНАЯ ПОДДЕРЖКА**

Любая команда может справиться только с определенным объемом работы; если он выше — наступает истощение, утомление и выгорание. Не допустить такой ситуации довольно сложно, потому что зачастую вы увязаете в ежедневной рутине и выполнении требований бизнеса. Вот несколько явных признаков того, что вашей команде требуется усиление:

- *Запланированные сроки проектов/спринтов превышаются чаще, чем соблюдаются.* Мы все знаем, что сроки сдвигаются, а взятые в спринты задачи возвращаются в бэклог — когда дело доходит до планирования сроков, мы неисправимые оптимисты. Проводя ретроспективный анализ, обращайте особое внимание на причины превышения сроков. Если самая частая из них — отвлечение ресурсов, то ваши проекты обречены на провал еще до запуска, поскольку вы не сможете сосредоточить ресурсы на текущей задаче.
- *Сотрудники не используют отпуска.* Своевременный отдых важен для здоровой атмосферы в любой команде. Нам всем нужно перезаряжаться и отвлекаться на вещи, не связанные с работой. Если ваши сотрудники не используют полностью свой отпуск — не думайте, что это из-за того, что они очень любят свою работу. Они просто не могут себе позволить уйти надолго, потому что слишком многое нужно сделать. Из этого вытекает следующий пункт.
- *Паника/стресс, когда ключевые сотрудники уходят в отпуск или отгул.* Возникает ли повышенный стресс у других членов команды/бизнеса, когда кто-то уходит в отпуск, особенно на неделю или больше, потому что от этого человека зависит очень многое и некоторые знания или навыки есть только у него?
- *Чрезмерная зависимость от одного человека.* Посмотрите на свою команду. На кого вы полагаетесь до такой степени, что, если он по какой-то причине уйдет, у вас будут серьезные проблемы? В каждой команде когда-нибудь случается так, что только один человек знает, как перезапустить этот сервер или ПО, или управляет какой-либо учетной записью, или имеет набор скриптов для решения проблемы. Эту ситуацию называют *рисками, связанными с зависимостью от ключевого сотрудника*.
- *Переработки по вечерам и в выходные дни.* Характер нашего бизнеса таков, что время от времени приходится работать допоздна. Для эффективного выполнения таких задач, как срочное решение проблем клиента или запуск новой версии продукта в часы минимальной нагрузки, команда инженеров должна гибко управлять своим рабочим временем. Однако, если вы регулярно просите людей задерживаться (или работать в выходные дни), имейте в виду, что вы, по сути, заставляете их работать на двух (или даже больше) работах.

Выявить этот вид проблем сложнее, чем кажется. И, что интересно, в чужих командах заметить их проще, чем в своей. Первое, что можно сделать, — попросить членов команды следить за этими признаками и не бояться сообщать вам о них. Во-вторых, супруги или члены семьи ваших сотрудников первыми заметят проявления стресса или нехватку свободного времени. Если к вам поступают жалобы или вы видите причины для беспокойства — не игнорируйте их, потому что тот, кто счастлив дома — счастлив и на работе.

Если вы видите в вашей команде перечисленные выше симптомы — ваша задача как руководителя понять, являются эти ситуации исключением или же правилом. Если второе, то это готовое обоснование, чтобы запрашивать у генерального или финансового директора дополнительные ресурсы.

### 4.2.2. Матрица навыков

Теперь вы понимаете, когда нанимать сотрудников. У вас есть данные, указывающие на необходимость расширения команды. Теперь возникает вопрос: кого нанять? Какие навыки необходимы в команде, чтобы снизить нагрузку, продолжить работу и выдавать результаты? Легко сказать: «Нам нужен еще один разработчик Java», — но, учитывая количество разных компетенций, которыми сегодня обладают высококвалифицированные инженеры, подобные скоропалительные решения могут ограничить ваши возможности.

Составьте матрицу навыков или, если хотите, ваших человеческих ресурсов, и поддерживайте ее в актуальном состоянии. Она представляет собой список основных навыков, которые требуются вашей команде для успешного выполнения проектов и достижения бизнес-целей, с оценками степени владения каждым из них для ваших текущих сотрудников. Посмотрим на пример матрицы навыков для небольшой команды.

Навык	Текущая оценка					Средняя оценка	Общая оценка	Требуемый уровень	Разница
	Кормак	Ноа	Джозеф	Диллон	Престон				
AWS	[7]	0	0	0	0	[7]	[7]	10	−3
Java-Script	2	5	5	2	1	3	15	20	−5
Java	2	0	0	8	0	2	10	5	+5
HTML/CSS	0	8	8	0	0	3	16	15	+1
MySQL	0	0	0	8	8	3	16	10	+6

В таблице перечислены пять базовых навыков, и для каждого члена команды указана оценка уровня его компетентности для каждого из навыков по шкале от 0 до 10. Это самая простая шкала, оценки в ней означают следующее:

- *от 10 до 8* — основной навык для этого сотрудника, уровень старшего специалиста/эксперта;
- *от 7 до 5* — основной навык для этого сотрудника, уровень младшего специалиста; требуется руководство или наставничество со стороны более опытного человека;
- *от 4 до 3* — второстепенный навык в дополнение к основному;
- *от 3 до 0* — уровень недостаточен для постоянной работы в этой сфере, при необходимости может быть помощником или временной заменой для основного специалиста.

Выделите уникальные навыки квадратными скобками, чтобы сразу видеть, что ими владеет только кто-то один в команде. Затем сложите оценки по каждому навыку, чтобы получить суммарное и среднее значения компетентности команды в этой области.

Значение в столбце «Требуемый уровень» субъективно и может меняться в зависимости от текущих и предстоящих проектов и задач. Этот столбец можно разделить на несколько более мелких столбцов: например, на проекты, периоды в шесть месяцев и т. д. Вам необходимо таким образом получить непредвзятое представление о слабых местах команды и о том, какими навыками должен обладать кандидат, если вы будете набирать людей.

В нашем примере сразу бросаются в глаза две проблемы: навык AWS соответствует требуемому уровню, но им владеет только один человек. Если Кормак решит уйти, у нас возникнут большие проблемы. В области JavaScript тоже есть пробел, и, учитывая, что в ней у нас имеются только начинающие разработчики (джуниоры), уместно взять в команду опытного специалиста (сеньора).

## ЗНАЙ СВОЮ КОМАНДУ

### *Аудит человеческих ресурсов*

Когда я начинаю работать с новым техническим директором или командой, первое, что я делаю, — прошу всех предоставить свои резюме для создания матрицы. В зависимости от того, насколько серьезно в компании организована оценка персонала, я также прошу результаты предыдущих оценок, чтобы понять, как сотрудники развивались со временем. Затем я провожу встречу с руководителями и прошу их подтвердить мои выводы — в результате чего обычно цифры полностью меняются, что совершенно нормально. Менять гораздо проще, чем создавать с нуля. Что неизменно — так это удивление при виде цифр, показывающих, что команда вовсе не так сильна в своей области, как она думала. Это особенно актуально для компаний, работающих в сфере прямых инвестиций: инвесторы не любят, когда бизнес зависит от отдельных сотрудников, потому что это представляет для них слишком большой риск.

Очень важно совместно с командой поддерживать эту матрицу навыков в актуальном состоянии. Она поможет определить сильные стороны и стремления сотрудников и то, кому может быть полезно пройти дополнительное обучение, чтобы повысить свой уровень до необходимого. Как обсуждается далее в главе, посвященной управлению командой, многие компании не уделяют должного внимания обучению команд новым специальностям или развитию существующих навыков, поскольку не понимают, какую огромную выгоду могут принести относительно скромные инвестиции в этой области.

### 4.2.3. Последствия отказа от найма

Хотя у вас есть все данные, указывающие на то, что вам не хватает сотрудников, это не означает, что вам одобряют новые рабочие места. Каждому отделу всегда требуется больше ресурсов, и, хотя их запросы могут быть не настолько научно обоснованными, как ваш, генеральный или финансовый директор должны все взвесить и убедиться, что бюджет достаточен, а вложения оправдают себя. Прием новых сотрудников может быть приостановлен по многим причинам, начиная от нехватки денег и заканчивая неуверенностью в будущих контрактах; кроме того, компания может находиться в процессе покупки/продажи бизнеса.

В любом случае вы обязаны проиллюстрировать последствия такого решения. К ним относятся возможный срыв сроков проектов или риски, связанные с возможным уходом ключевого сотрудника, особенно если его навыки критически необходимы.

Преподносите эти данные не как угрозу или ультиматум, а как реальность, которую нельзя игнорировать. Вы, конечно, будете делать все, что в ваших силах, чтобы добиться успеха с тем, что у вас есть, потому что вы профессионал, который не любит неудачи. Если одно из возможных последствий — задержка разработки нового функционала или срыв сроков проекта, обратитесь за помощью, чтобы сообщить об этой задержке клиенту или отделу, которые зависят от результата работы вашей команды. Это поможет показать, что у вас общая цель и что вы просите дополнительные ресурсы не просто для того, чтобы увеличить количество сотрудников у себя в подчинении.

## 4.3. ПОИСК КАНДИДАТОВ

Существуют разные способы поиска и привлечения кандидатов. В этом разделе рассматриваются общепринятые пути поиска и подробно описываются их сильные и слабые стороны, в зависимости от того, на каком этапе процесса найма вы находитесь.

Мы рассмотрим следующие способы:

- реферальный поиск;
- агентства по подбору персонала;



- хедхантинг;
- ярмарки вакансий для выпускников / стажировки;
- самостоятельный поиск в интернете.

### 4.3.1. Реферальный поиск

Если вы создали благоприятную и комфортную рабочую среду, то ваш лучший пропагандист, который поможет привлечь новых людей, — это ваша текущая команда. Попросите сотрудников узнать среди их знакомых, соответствует ли кто-то из них вашим требованиям. Обычно за каждого нанятого таким способом сотрудника выплачивается реферальный бонус (от 500 до 1000 долларов).

Не ограничивайтесь своей командой инженеров, распространите этот запрос по всей компании. Организации, задействующие связи своих сотрудников с помощью своевременной рассылки по электронной почте, всегда добивались больших успехов. Вполне возможно, что чья-то сестра, лучший друг или сын работает с человеком, который ищет новые возможности. И вы ничего не теряете, используя этот способ подбора.

Но соблюдайте осторожность и относитесь к кандидатам, привлекаемым таким образом, так же, как и ко всем остальным — никаких привилегий. Так вы избежите проблем или чувства вины, если человек не пройдет собеседование или не справится.

#### **Преимущества:**

- если повезет — можно быстро найти сотрудника;
- недорого.

#### **Недостатки:**

- нет гарантии, что рекомендуемые кандидаты подойдут;
- можно доставить неудобство рекомендателю, если действовать неграмотно.

### 4.3.2. Агентства по подбору персонала

Типичное кадровое агентство быстро предоставит вам пул резюме и возьмет за свои услуги определенный процент от годовой зарплаты кандидата после его выхода на работу. Не все агентства одинаковы, и важно изучить, как агентство находит и проверяет соискателей и специализируется ли оно в определенной области. К сожалению, многие агентства просто сопоставляют ключевые слова из резюме и вакансии, поэтому направляют заказчику как хороших, так и плохих кандидатов, если они формально соответствуют критериям.

Хорошее агентство — ваш партнер в процессе найма, вторит Лори Дженнингс (Lori Jennings), основатель Jennings ProSearch. Агентство продает и продвигает ваше предложение, отсеивает кандидатов, которые не соответствуют требованиям (как в части навыков, так и в части личных качеств), готовит кандидата к собеседованию и обеспечивает весь процесс отбора. Лучшие из агентств действительно



ведут себя как партнеры. Они даже сопровождают кандидата на собеседование, ожидая снаружи, пока интервью не закончится.

Дженнингс дает хороший совет, по каким основным показателям можно оценить работу агентства: скорость закрытия вакансии, процент закрытых вакансий и среднее время работы сотрудников. И далее она продолжает: «Регулярно ли агентство общается с людьми, занимающими должности, аналогичные вашей, и понимающими проблемы технического директора? Может ли оно проконсультировать вас по вопросам оплаты труда? Понимает ли оно, что может мотивировать кандидатов вашего уровня? Есть ли у него достаточный пул кандидатов, которые ответят на их звонок?»

### ИЗУЧЕНИЕ РАБОТЫ КАДРОВЫХ АГЕНТСТВ

#### *Тайный покупатель*

Нет ничего зазорного в том, чтобы узнать, как агентство работает с кандидатами, и лучший способ сделать это — попросить кого-то обратиться в агентство в качестве кандидата и оценить качество предлагаемых вакансий. Это трюк я проделывал несколько раз, чтобы отделить шарлатанов, просто рассылающих первые попавшиеся резюме, от тех, кто знает свое дело. Некоторые из вакансий, которые мне предлагали, были настолько далеки от моих ключевых компетенций, что я сомневался, читали ли они вообще мое резюме. Так что убедитесь, что если вы платите агентству вознаграждение в размере до 20 % от зарплаты сотрудника, то вы получите качественных кандидатов, которых вам обещают.

#### **Преимущества:**

- можно быстро получить несколько квалифицированных кандидатов;
- кандидаты уже прошли предварительный отбор.

#### **Недостатки:**

- требуются усилия, чтобы найти агентство, которое понимает ваши потребности;
- агентство редко проверяет технические навыки кандидатов;
- услуги агентства могут стоить очень дорого — от 10 до 25 % зарплаты за первый год работы сотрудника.

### 4.3.3. Хедхантинг

Хедхантинг, или охота за головами, — это привлечение кандидата, который не находится в активном поиске работы, но следит за появляющимися возможностями. Хотя некоторые агентства специализируются именно в этой области, по факту они являются все теми же кадровыми агентствами, только под другим названием. Однако вы и ваша команда по ходу своей профессиональной деятельности взаимодействуете с множеством разных людей — с некоторыми из них у вас складываются хорошие отношения, и вы уже можете предварительно оценить их навыки. Ваша сеть потенциальных кандидатов шире, чем вы себе представляете. Вот возможные направления поиска:

- Компании-поставщики услуг, с которыми вы работали, при условии, что предложение работы их сотруднику не нарушит договорные обязательства или соглашения о неразглашении коммерческой тайны.
- Проекты с открытым исходным кодом — это богатые охотничьи угодья, к тому же результаты работы специалистов находятся в публичном доступе.
- Кто может быть лучше, чем человек, написавший книгу на интересующую вас тему? Большинство авторов будут рады вашему письму и как минимум готовы пообщаться с вами. Если они сами не ищут работу — они обычно знают кого-то, кто ищет.
- Как и авторы книг, открытыми для общения могут быть авторы статей или популярных блогов по вашей теме.

#### **Преимущества:**

- кандидаты имеют необходимую квалификацию — они хорошо знают свое дело;
- если вы уже работали с этим человеком, вы знаете, что он впишется в вашу команду.

#### **Недостатки:**

- здесь велика роль случайности, поэтому этот способ не подходит в качестве стратегии;
- узкая область поиска.

### **4.3.4. Ярмарки вакансий для выпускников / стажировки**

Если структура вашей организации подходит для привлечения выпускников и может обеспечить им необходимое руководство и наставничество, то ярмарки вакансий местного университета откроют вам богатые возможности выбора кандидатов. Такие ярмарки проходят во всех университетах и колледжах в разное время года, и там всегда рады вашему участию. Это работа на будущее, потому что придется подождать, пока выпускник не закончит обучение (что логично), то есть обычно до конца календарного года или конца весны.

Этот процесс может занять до года, в зависимости от того, когда вы впервые встретитесь со студентами. За это время вы сможете построить и развить отношения с ними и наблюдать за ними в течение учебного года. (Получили ли они те оценки, на которые рассчитывали? Относится ли их выпускной проект к интересующей



вас области?) Здесь вы не ищете совершенства, так что не судите слишком строго — это не обычное собеседование. Вместо этого вы ищете неограниченный алмаз — подходящий вам тип личности и интересы. Может ли этот человек при грамотном наставнике стать первоклассным сотрудником и оказать влияние на всю индустрию?

Слишком часто организации считают выпускников просто более дешевой рабочей силой и предъявляют к ним те же требования, что и к опытным специалистам, и в итоге получают разочаровывающий результат. Но даже самую породистую скаковую лошадь нужно тренировать — она не сразу выигрывает скачки.

Вы можете познакомиться и поработать со студентами, прежде чем принять одного из них на работу — «попробуйте перед покупкой». Вот какие варианты для этого есть:

- *Корпоративное спонсорство проекта.* В большинстве случаев выпускники инженерных специальностей должны выполнить годовой проект, оценка за который составляет значительную часть их итогового балла. Профессорам сложно из года в год придумывать уникальные темы, поэтому они приветствуют участие местного профессионального сообщества. Это отличная возможность для вас делегировать студентам проект, который вам интересен, оказывая помощь и наставничество на протяжении всей работы.
- *Стажировки.* Стажировка даст молодому специалисту реальное представление о том, как работает ваша отрасль, и поможет в оставшейся учебе. Стажировка может длиться только лето или целый год, в зависимости от курса. Назначьте стажера в помощь опытному члену команды, чтобы новичок учился и участвовал в работе. В случае успеха — через год после выпуска он вернется к вам настоящей звездой.
- *Спонсорство хакатонов.* Существуют соревнования программистов, проводимые в некоторых учебных заведениях, готовящих инженеров. На них студенты работают вместе и соревнуются командами, демонстрируя умение создать проект и довести его до финального результата за короткое время. На них всегда весело и участники проявляют все свои лучшие качества, потому что написание кода проходит в игровой форме. Учебным заведениям всегда нужны спонсоры, выделяющие средства на пиццу, завтраки, кофе, напитки и прочее для таких мероприятий.

#### **Преимущества:**

- Вы довольно долго изучаете студентов, поэтому к тому времени, когда вы принимаете кого-то из них на работу, вы уже знаете, чего от них ожидать.
- Это отличный способ отблагодарить сообщество и помочь, даже если в итоге вы никого и не возьмете на работу.

#### **Недостатки:**

- Долгосрочный проект; не ждите результатов раньше, чем через полгода.
- Требуется наставничества и терпеливого обучения кандидатов.

**НАЗВАЛСЯ ГРУЗДЕМ — ПОЛЕЗАЙ В КУЗОВ***Ночь напролет*

Одно из моих самых ярких воспоминаний связано со спонсорством 24-часового хакатона в Университете содружества Вирджинии (VCU). В нем участвовали команды из 4–6 инженеров (в том числе команды из соседних учебных заведений), и в течение 24 часов им необходимо было создать проект, который произвел бы впечатление на судей. Я хотел получить максимальную отдачу от этого мероприятия и составил график, по которому все время на месте присутствовал бы кто-то из нашей команды, чтобы помочь студентам. Я хотел, чтобы студенты знали, что мы с ними. При этом одно из моих личных правил гласит: если я по какой-либо причине прошу человека задержаться допоздна, то я остаюсь вместе с ним. Эта ситуация ничем не отличалась, так что я провел все 24 часа со студентами и без сна. Это был абсолютный кайф.

**4.3.5. Самостоятельный поиск в интернете**

В конце концов, вы можете взять дело в свои руки и использовать один из множества онлайн-ресурсов для поиска подходящих кандидатов. Вот некоторые основные сервисы: LinkedIn, ZipRecruiter, Indeed и Stack Overflow. Результаты здесь будут зависеть от времени года и должности, на которую вы ищете человека. Волшебной формулы не существует; то, что сегодня работает на одном сайте, может не сработать на другом.

Здесь вы полагаетесь на то, что звезды сойдутся и именно сейчас работу ищет идеальный кандидат, который вдохновится вашей вакансией и откликнется на нее.

Будьте осторожны: вы привлечете внимание кадровых агентств, которые будут пытаться продать вам свои услуги. Учтите, что эти приставалы направят вам тех же кандидатов, которые (что намного дешевле для вас) и так нашли бы вас через сайты поиска работы.

Плата за пользование таким сайтом обычно составляет несколько сотен долларов в месяц, также вы можете заказать дополнительные услуги для продвижения вашей вакансии. В зависимости от вакансии какие-то сайты подходят больше, чем другие. Например, разработчика лучше искать на Stack Overflow, а сотрудников, которые не будут заниматься программированием, — на LinkedIn.

**Преимущества:**

- Полный контроль над описанием вакансии и тем, как она будет выглядеть.
- Очень экономно.

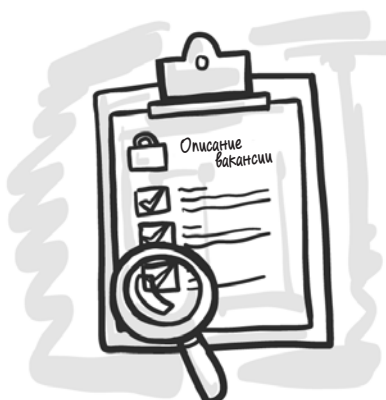
**Недостатки:**

- Не гарантирует успеха.
- Привлекает много агентств и случайных соискателей (которые откликаются на все подряд).

## 4.4. СОСТАВЛЕНИЕ ОПИСАНИЯ ВАКАНСИИ

Итак, вы готовы и получили добро на открытие новой вакансии в команде. Вам необходимо составить ее текст. Это, по сути, одностороннее описание позиции. Во всем процессе подбора сотрудников это самый важный документ. Потенциальные кандидаты, прочитав его, должны захотеть сделать следующий шаг и направить отклик. Если они этого не сделают, то дальше и обсуждать будет нечего. Все просто, не так ли?

В этом разделе мы узнаем, почему большинство технических директоров не справляются с этой важной задачей и как вы можете, используя немного ораторского и редакторского искусства, отшлифовать этот необработанный алмаз. Ваша цель — текст, который заставит соискателей немедленно устремиться к вам.



### АДАПТАЦИЯ

#### Подход Лу Адлера

Автором многих идей, которые я здесь представляю, является специалист по подбору персонала Лу Адлер. Весь его подход заключается в том, чтобы продавать возможность, а не задачи, ориентироваться на цель, а не на действие. Я немного адаптировал его метод для сферы технологий, но, если вы прочитаете его книги и посмотрите его видео, вы увидите общее. Прочтите его книгу «Hire with Your Head» (Wiley, 2007).<sup>1</sup>

Составление грамотного описания вакансии — одна из самых недооцененных частей процесса подбора персонала, и я даже не могу перечислить, сколько команд, с которыми я работал, жаловались, что им трудно найти сотрудников. Когда я просил показать их описания вакансий, причина становилась ясной как божий день — ведь пчелы летят за медом. Если вы не способны вдохновлять, то вы просто тратите деньги в погоне за несбыточной мечтой.

### 4.4.1. Структура

Если компания крупная, в ней уже может быть стандартный шаблон, который использует отдел по работе с персоналом. Мы наберемся смелости и притворимся, что его не существует. Не будем пока ограничиваться стандартами и процессами — мы от них все равно никуда не денемся.

Большинство описаний вакансий составляются по всем знакомому шаблону:

<sup>1</sup> Лу Адлер. «Нанимай с головой».

1. Название должности.
2. Описание.
3. Минимальные требования к кандидату.
4. Требуемые навыки.
5. Информация о компании.

Ничего неправильного здесь нет, все необходимое изложено. Нас, собственно, интересуют только первые четыре пункта, потому что последний обычно представляет собой стандартный рассказ о компании. Итак, всего четыре пункта, как же люди ухитряются наделать в них столько ошибок?

По мере того как мы разбираем каждый из этих пунктов, представляйте кандидата, которого вы хотите привлечь. Если идеальный кандидат похож на кого-то в вашей команде, думайте об этом человеке. Представьте его реакцию на каждое слово; понравится ли ему этот текст, или же он увидит, что вы просто набросали красивых фраз, чтобы произвести впечатление, и теряет интерес?

Вы можете заметить, что в списке отсутствует информация о заработной плате. Хотя это очень важная часть, руководство может возражать против ее указания. Некоторые считают, что это создаст проблемы с нынешними сотрудниками, которые могут решить, что им платят меньше за ту же самую работу. Не позволяйте убирать из вакансий уровень дохода; ни к чему хорошему это не приведет. Указав размер заработной платы, вы вызовете к себе доверие и отсеете тех, кого не устраивает данный уровень.

#### 4.4.2. Название должности

Заголовок — самая важная часть документа. Это первое, что прочитает ваш потенциальный сотрудник. Его можно прочитать в сообщении на телефоне, в электронной почте, твите или результатах поиска. В любом случае это первое, что он увидит, и ваш первый (и, возможно, единственный) шанс его заполучить. Совет прост: не умничайте; пишите проще. Рассмотрим несколько примеров:

- Вице-президент по разработке (VP of Engineering)
- Старший веб-разработчик (Senior Web Developer)
- Руководитель направления цифровой трансформации (Head of Digital Transformation)
- Архитектор облачных решений (Cloud Architect)
- Младший мобильный разработчик React (React Mobile Developer (Jnr))
- Руководитель проектов (Project Manager)
- Скрам-мастер (Scrum Master)

Пробежав глазами по этому списку, что вы узнали о каждой должности? Получили ли вы представление о том, чем необходимо будет заниматься, о требуемом уровне опыта? Встретились ли вам названия, которые ни о чем не говорят и являются просто набором слов?

Наниматели, как правило, мудрят с названиями. Хотя «Специалист по Связям с Обществом Охраны Окружающей Среды» звучит более пафосно, чем «уборщик мусора» — такое название не привлечет подходящих для этой работы кандидатов, потому что они просто не поймут, что конкретно вы предлагаете, и перейдут к следующей вакансии.

В списке выше «руководитель направления цифровой трансформации» звучит слишком расплывчато и может означать что угодно, в том числе не технические обязанности. «Архитектор облачных решений» не так уж плох, но лучше указать поставщика облачных услуг и требуемый уровень, поэтому «старший архитектор решений на основе AWS» скажет соискателю гораздо больше.

Придерживайтесь принятых в отрасли названий, независимо от того, как должность называется в вашей компании. Помните: вы не привлекаете людей изнутри компании — вы забрасываете сеть в большое море, и вам придется играть по принятым в нем правилам.

Добавьте в название как можно больше метаданных, но не переусердствуйте. Младший мобильный разработчик React — хороший пример грамотного обращения к целевой аудитории. Вам нужен мобильный разработчик, и, поскольку существует множество фреймворков и языков, сразу укажите требуемые навыки, чтобы людям не пришлось тратить время на дальнейшее чтение в попытках узнать, подходят ли они. Если соискатель решит читать дальше, то он уже будет знать, подходит ли он. То же самое касается последних двух формулировок. Да, «руководитель проектов» встречается часто, но, если вы уже работаете, например, со Scrum, укажите это прямо в названии; не упускайте возможности. Люди склонны выбирать то, что им знакомо, особенно когда принимают решение, способное изменить всю их жизнь.

#### 4.4.3. Описание работы

Вот мы и подошли к основному действию — описанию работы, которое даст кандидату всю необходимую информацию, чтобы понять, на верном ли он пути. Как и в названии, здесь все должно быть на своем месте. Ваша цель — заинтересовать человека первым предложением или, в крайнем случае, вторым. К тому времени, когда кандидат будет читать третье и последующие предложения, он уже определится. С этого момента вы только добавляете детали и цвета.

Первое предложение крайне важно, потому что вы не сможете контролировать, где всплывет описание вашей вакансии и по каким каналам распространится. Подумайте, где можно будет увидеть начало описания вакансии — например, в превью электронного письма в папке «Входящие», в твите или в посте в LinkedIn. Везде выводится маленькое окошко с частью текста, которое должно побудить читателя открыть полную версию. Поэтому первое предложение должно описывать все плюсы и преимущества вакансии и по-настоящему захватить читателя. Вот краткий список того, о чем можно и о чем нельзя писать в первом предложении:

- Не пишите о компании.
- Не повторяйте то, что есть в названии.
- Не тратьте место на лишние слова.
- Опишите, чем кандидат будет заниматься.

Слишком часто кучу места в описании работы занимает рассказ о компании и о том, насколько невероятную возможность она предоставляет, хотя еще даже не сказано, чем, собственно, будет заниматься кандидат. Дальше у вас будет достаточно места, чтобы описать компанию и все преимущества работы в ней.

Посмотрим на несколько хороших примеров, которые сразу переходят к делу.

- *Название:* Архитектор облачных решений.

*Первое предложение:* Возглавьте команду, которая переносит всю инфраструктуру наших интернет-проектов из традиционного центра обработки данных в облако AWS с целью обеспечения отказоустойчивости и автоматического масштабирования.

- *Название:* Младший мобильный разработчик React.

*Первое предложение:* Присоединяйтесь к команде опытных мобильных разработчиков, которые создают новую версию наших сервисов в приложениях для Android и iOS.

- *Название:* Руководитель проектов.

*Первое предложение:* Вам предстоит организовывать работу и обучение растущей команды инженеров, а также управлять новыми проектами.

Как только вы привлекли внимание читателей, подробнее опишите области, которые заинтересуют их и о которых они захотят узнать больше. Избегайте маркетинговых клише (таких как *захватывающий, интересный, динамичный* или *энергичный*) и сосредоточьтесь на словах, которые сообщают факты. Все ваши преимущества будут понятны и без красивых слов.

#### ЗАМЕТКИ С ПОЛЕЙ

##### *Мое время пришло*

Просто, но эффективно — именно так мы ищем технических директоров для своих клиентов. Самыми успешными сотрудниками, которых мы нашли, были не те, кто уже имел опыт работы на этой позиции, а те, кто чувствовал, что готов к следующему этапу карьеры и хотел проявить себя. Они долго готовились к этому и только ждали подходящего случая.

Подход Лу Адлера заключается в том, чтобы обратиться к причине, по которой человек может искать работу. Большинство меняет работу в поисках профессионального развития. Они ищут того, что будет вдохновлять их и выведет на новый уровень. Печальная правда жизни в том, что обычно, чтобы получить повышение или возможность для развития, приходится искать новую работу.



Адлер называет это «амбициозной целью» — пусть работа кажется немного сложнее, чем текущая, но не настолько сложной, чтобы отказываться от нее. Дайте кандидатам понять, что ваша вакансия — это возможность, которую они искали, чтобы наконец проявить себя.

Остальное описание должно быть адресовано непосредственно человеку, которого вы ищете. Говорите на его языке, потому что так он поймет вас и вы установите с ним связь. В описании можно указать дополнительные преимущества, такие как международные командировки, работа с клиентами или что-то еще, что, по вашему мнению, может заинтересовать кандидата.

Составляя описание, думайте о человеке, к которому вы обращаетесь. Представьте, что вы пишете ему электронное письмо или разговариваете с ним по телефону. Вот тот язык, который нужно использовать.

#### 4.4.4. Минимальные требования к кандидату

Это обычный список необходимых качеств, которыми должен обладать кандидат. Вы, как работодатель, устанавливаете фильтр для кандидатов, чтобы те, кто ему не соответствует, к вам даже не обращались. Соискатели могут пройти по этому списку и сами сойти с дистанции.

Лучше всего свести этот раздел к минимуму. Многие компании ошибочно включают в него то, что на самом деле не нужно. Сколько раз вы читали, например: «Минимальный опыт работы  $x$  лет»? Задумайтесь — вам нужен кто-то, кто сможет выполнять эту работу, и какая разница, сколько времени ему понадобится, чтобы научиться этому?

Указывайте только понятные, измеримые и бесспорные требования, которые должны выполняться. Например, требование официальных сертификатов или допусков, дающих право заниматься определенной деятельностью (например, допуск к засекреченной информации) лучше перенести в следующий раздел «Требуемые навыки».

#### 4.4.5. Требуемые навыки

Чаще всего именно этот раздел описания вакансии работодатели понимают неправильно, потому что с годами стало подразумеваться что «по большей части все это является обязательным». Большинство людей, изучив этот список, отправят отклик, даже если подходят хотя бы под 75% требований — и правильно сделают.

При составлении этого списка велик риск перестараться. Будьте внимательны. Вспомните, к кому вы обращаетесь, и спросите себя, действительно ли необходимо то, что вы указали, и не подразумевается ли это другими пунктами. Например, требование навыков работы с Microsoft Office в вакансии разработчика. Скорее всего, если человек умеет программировать, то уж с Word или Excel проблем у него не будет.

Не считайте кандидатов идиотами. Относитесь к ним как к профессионалам. Указывайте только то, что имеет отношение к делу, — это, как правило, то, о чем вы будете говорить на собеседовании. Это список навыков, которые им необходимы для успешной работы; кандидат может и не обладать всеми сразу, но быстро научиться тому, чего пока не знает.

Можно описать требования такими словами, как *демонстрирует* или *применяет на практике* — они дадут читателю представление о глубине навыков, которые вы от них требуете. Сопровождайте искушению указать требуемый опыт в годах — как уже отмечалось выше, нельзя ставить знак равенства между количеством лет опыта и уровнем знаний. Насколько долго человек работает в отрасли, можно увидеть по резюме — этого должно быть достаточно для предварительной оценки его опыта.

#### 4.4.6. Информация о компании

Наконец, мы добрались до раздела, которым может свободно распоряжаться отдел кадров / юридический отдел — ну, не совсем свободно. Немного изменений здесь не помешает. Крупные корпорации фактически превратили этот раздел в «Условия и положения», которые никто не читает. Зачем разбазаривать ценное место?

Используйте этот раздел, чтобы описать компанию, ее продукты/клиентов и рассказать, почему это хорошее место для работы. Включите сюда общепринятые пункты и перечислите имеющиеся у вас льготы, такие как отпуск, оплата пенсионных взносов<sup>1</sup> и медицинского страхования. Если в компании есть возможность работать из дома, укажите это. В этом разделе вы пишете не о самой должности, а об организации. К тому времени, когда читатель дойдет до него, он уже поймет, подходит ли он на эту позицию, и будет принимать решение, отправлять ли отклик.

### ИТОГИ

- При подборе персонала можно не ограничиваться штатными сотрудниками; при необходимости можно обращаться к подрядчикам и поставщикам услуг, чтобы закрыть потребности команды. Каждый вариант имеет свои плюсы и минусы.
- К необходимости нанимать новых сотрудников приводят различные события, которые могут произойти в бизнесе, включая увольнение, смерть сотрудника или, чаще всего, его рост внутри компании.
- Избегайте выгорания команды: окажите людям необходимую поддержку, иначе они уйдут.

---

<sup>1</sup> В России они регулируются законодательно и их можно не указывать. — *Примеч. ред.*

- Используйте разные способы поиска кандидатов.
- Составить описание вакансии — это скорее искусство, чем наука. Чтобы сделать это правильно, требуется время, усилия и множество проб и ошибок.

## **ПРОВЕРЬТЕ СЕБЯ**

Что из перечисленного ниже вы уже сделали?

- Умеете определять различные типы кадровых ресурсов и в каких ситуациях лучше подходит тот или иной тип.
- Разработали и регулярно пересматриваете метрики, указывающие на необходимость поиска новых сотрудников.
- Умеете оценить уровень технических знаний своей команды.
- Создали по крайней мере два канала для поиска новых кандидатов.
- Подготовили описания вакансии для каждой позиции в вашей команде.

# *Собеседования, выбор подходящего кандидата и адаптация новых сотрудников*

---

## **В этой главе:**

- ✓ Проведение собеседований и обработка результатов
- ✓ Каждый кандидат должен почувствовать себя особенным
- ✓ Простановка оценок для сравнения кандидатов и выбора лучших
- ✓ Как сделать предложение работы и как отказать тем, кто не подошел
- ✓ Как интегрировать новых сотрудников в команду

Приняв решение о необходимости расширения, будьте готовы заняться поиском, собеседованиями и оценкой кандидатов. После того как вы сделаете свой выбор, вам необходимо будет сообщить об этом как кандидату, которого вы выбрали, так и тем, кто не прошел отбор. На этом дело не заканчивается — выбранный сотрудник должен еще появиться в офисе и приступить к работе. Вы удивитесь, сколько кандидатов успешно преодолевают отбор и подписывают бумаги, но не приходят в свой первый день.

В этой главе я помогу вам лучше разобраться в этих темах. Не относитесь к подбору сотрудников как к чему-то, чем придется заняться лишь однажды; сделайте его частью своей повседневной работы, считайте, что вы всегда находитесь в поиске людей. Вы никогда не знаете, в какой момент кто-то соберется покинуть вашу команду или вам придется быстро масштабироваться; кадровый резерв — прекрасный инструмент, который лучше держать наготове. Самые успешные технические директора быстро находят сотрудников, потому что у них большой пул кандидатов.

Вы сильны настолько, насколько сильна ваша команда — поэтому позаботьтесь о ее будущем росте уже сейчас.

## 5.1. СОБЕСЕДОВАНИЕ

Этап собеседования — один из самых сложных в процессе найма как для интервьюера, так и для кандидата. По моему опыту наставника начинающих технических директоров могу сказать, что большинство из них уверены, что они хороши в этой области, хотя это вовсе не так. В этом разделе мы пройдем по всему процессу, в том числе обсудим, как проводить собеседование, рассмотрим схему, которая поможет вам оценивать кандидатов, и научимся замечать неявные признаки, которые часто упускают из виду. Этот разбор ни в коем случае нельзя считать исчерпывающим. Подробному рассмотрению процесса собеседования посвящены многие книги и онлайн-ресурсы.



Одно можно сказать совершенно точно: не существует универсальной техники. Вы разработаете свою собственную методику, и, скорее всего, поначалу у вас мало что будет получаться. Не расстраивайтесь — практика, анализ и честная оценка своих результатов помогут вам проводить собеседования по-настоящему профессионально.

### 5.1.1. Правильный настрой

Забудьте на время, что вы и сами нервничаете, и представьте себя на месте человека, которого вы будете собеседовать. Уж он-то, скорее всего, будет нервничать очень сильно, и ему будет сложно представить себя в наилучшем свете — что есть некий парадокс, учитывая, что он изо всех сил к этому стремится. Результат собеседования может изменить и его жизнь, и жизнь его семьи. Он не только хочет произвести впечатление на вас и вашу команду, но и старается понять, будет ли ему комфортно работать с вами. Сможете ли вы предложить ему условия, в которых он каждое утро будет с радостью спешить на работу?

Поэтому важно относиться к каждому кандидату со всем возможным уважением и вниманием — как к потенциальному клиенту. Так вы создадите

позитивную атмосферу, которая поможет кандидату немного расслабиться, и ему будет проще проявить свои сильные стороны.

Если у вас большая команда, то в дальнейшем непосредственным руководителем кандидата можете быть не вы, а один из ваших подчиненных. Как вести подбор в таком случае — решать вам, но вы уже не должны будете играть основную роль в собеседовании. Если вы возложили на кого-то ответственность за управление людьми, то доверьте ему и подобрать этих людей. В любом случае принимайте участие в процессе — присоединитесь к группе интервьюеров либо поговорите с кандидатом один на один в течение 15–30 минут после собеседования, чтобы познакомиться с ним и еще раз подчеркнуть, какая замечательная у вас команда и почему кандидат должен выбрать именно вас. Если же этот человек будет вашим непосредственным подчиненным, то вести собеседование должны вы, а еще кто-то из вашей команды или компании должен присутствовать как минимум в качестве второй пары глаз и ушей.

Собеседования запоминаются надолго, и не только из-за сильного стресса, но и потому, что это первая встреча будущего сотрудника с новой командой. Вам необходимо произвести позитивное впечатление, даже если вы решите, что этот человек вам не подходит. Чтобы подготовиться, вспомните все собеседования, которые вы сами когда-либо проходили. Даже если вы соучредитель стартапа, которым занимаетесь со школьной скамьи и у вас ни разу не было официального собеседования — скорее всего, на каком-то этапе своей жизни вы все же проходили интервью в том или ином виде. Здесь годится любой подобный опыт. Вспомните эти интервью — что вы чувствовали? Какой вопрос застал вас врасплох? На какой вопрос вам понравилось отвечать? Какой была атмосфера? Ответы на эти вопросы помогут вам лучше проводить собеседования.

Никогда не относитесь к собеседованию как к неизбежному злу и не считайте, что нужно просто перетерпеть этот час. Хотя для вас это может быть всего лишь очередной кандидат из бесконечного потока, ваш собеседник будет испытывать стресс и волноваться из-за каждой мелочи, обдумывать, что надеть, что сказать, как себя вести — все это, чтобы показать себя с лучшей стороны. Отнеситесь к нему с уважением и уделите ему достаточно времени и пространства, чтобы у кандидата остались только положительные впечатления от общения с вами, независимо от окончательного решения.

### 5.1.2. Цели собеседования

Идеальное собеседование должно быть двусторонним, когда каждая из сторон получает всю информацию для принятия решения. Однако такое происходит нечасто, поскольку кандидатам не хватает уверенности в себе, чтобы действовать на равных. Не используйте это в своих интересах, и считайте, что это норма, а не исключение.

Собеседования занимают много времени и отвлекают от всех других неотложных дел. Поэтому важно, чтобы он был максимально эффективным и вы достигали целей собеседования.

Каковы же цели? Проблема, которую вы решаете, — это необходимость расширить команду, чтобы новый сотрудник повысил ее производительность и помог развивать ваш продукт. Для этого вам нужно понять, предложите ли вы работу этому конкретному соискателю. Чтобы не ошибиться и при этом отнестись ко всем справедливо, вам следует сформулировать перечень конкретных требований, которым должны соответствовать кандидаты, и принять аргументированное решение на основе данных.

Лучше всего для этих целей создать лист оценки, в котором все интервьюеры смогут оценивать кандидатов. Это перечень компетенций, необходимых на этой должности, и вы будете оценивать их наличие у каждого кандидата по шкале от 0 до 5 (где 5 — высший балл). 0–1 означает, что у кандидата эта компетенция полностью отсутствует, а 4–5 — что уровень более чем достаточен для вашей позиции. Оцениваемые навыки должны относиться к конкретной должности и соответствовать требованиям, указанным в описании вакансии.

Разработайте отдельный лист оценки для каждой должности и до того, как проводить собеседование, убедитесь, что все интервьюеры понимают контекст и цель каждого пункта в списке. Образец листа оценки представлен в конце этого раздела.

Но нельзя отклонять или нанимать кандидатов только на основании их оценки, это подход формальный и ограниченный. Оценка должна послужить лишь основой для принятия решения о выборе кандидата и напоминать вам, какими были результаты кандидата в той или иной области. Скорее всего, за короткое время вам придется провести не одно собеседование, поэтому вам необходим механизм фиксации оценок для каждого из кандидатов.

Итак, мы поняли, что делать с информацией, полученной в ходе собеседования. Теперь перейдем к самому собеседованию. В следующих разделах мы составим список критериев и подготовим лист оценки, который интервьюеры могут заполнять, не привлекая внимания кандидата (или же после собеседования).

### **5.1.3. Телефонное (или предварительное) собеседование**

Прежде чем тратить ресурсы на личную встречу с кандидатом, стоит пообщаться с ним по телефону. Такое первичное интервью не должно быть очень долгим или подробным — не более 30 минут. Ваша цель здесь — получить общее представление о человеке и его пригодности к требуемой роли. Не стоит задавать слишком сложные технические вопросы, чтобы не отпугнуть кандидата. Поддержите его уверенность в себе и помогите ему избавиться от нервозности перед следующим этапом. А если кандидат не знает самых базовых вещей — вы можете быстро отказать ему и больше не тратить на него время.

Оцените его личностные качества — например, потратил ли он время, чтобы найти тихое место для разговора с вами, или же он находится в переполненной электричке (или, что еще хуже, в очереди на кассу в супермаркете — бывает и такое). Так можно понять уровень ответственности человека: отнесся ли он к вам так же, как вы хотите, чтобы он относился к вашим клиентам?

Насколько четко он отвечает и формулирует вопросы? Выбор слов часто может подсказать, знает ли кандидат свою предметную область или просто нахватался умных фраз. Если вы подозреваете второе, просто попросите его пояснить свой ответ, не используя модные словечки. Если человек разбирается в вопросе, он это сделает; если нет — не сможет.

Как быстро кандидат отвечает на простые вопросы? Вы смогли наладить с ним контакт? Комфортно ли было разговаривать? Ответы на эти вопросы помогут понять, стоит ли приглашать этого человека на следующий этап собеседования.

### ЗАМЕТКИ С ПОЛЕЙ

#### *Ближе к делу*

Один из кандидатов прямо во время собеседования начал искать ответы в интернете и процитировал мне Википедию. Я оценил его находчивость, но это не те навыки, которые мне были нужны. Как технический директор, вы должны уметь действовать решительно. Я вежливо прекращаю собеседование по телефону, если за 10 минут становится понятно, что человек мне не подходит, и сообщаю кандидату, что он либо не знает основ, либо, что еще хуже, даже не поинтересовался требованиями к позиции, на которую претендует. Это не грубость и не поспешность; вы уважаете свое и его время и не тратите его впустую.

## 5.1.4. Личное собеседование

После изучения резюме и проведения телефонного собеседования пришло время встретиться с кандидатом лично (или же по видеосвязи). Вам необходимо решить, как организовать этот процесс, что во многом будет зависеть от размера вашей команды и того, на какую позицию идет подбор. Основных варианта два:

- Одно собеседование с вами или командой.
- Серия собеседований с 1–2 членами команды.

Как уже говорилось, если вы руководите большой командой, кандидат может не подчиняться непосредственно вам. Вы должны решить, в какой момент присоединиться к собеседованию.

Если во время интервью вы собираетесь играть второстепенную роль, важно оставаться в этом качестве. Не поддавайтесь желанию взять инициативу на себя; позвольте вашему сотруднику управлять ходом собеседования. Или же — что даже лучше — присоединитесь в конце собеседования на 15–30 минут. К моменту встречи с вами кандидат уже успеет пообщаться с несколькими членами вашей



команды и получить их одобрение. Ваша задача теперь — оценить кандидата с точки зрения его личных качеств и отношения к работе. Также уделите внимание преимуществам работы в вашей команде и тому, каким важным шагом станет для кандидата приход к вам. Позвольте кандидату задать любые вопросы — это его единственная возможность получить ответы от технического директора, прежде чем согласиться на эту работу (если ее ему предложат).

Однако вы можете быть и основным интервьюером. Как в этом случае организовать собеседование?

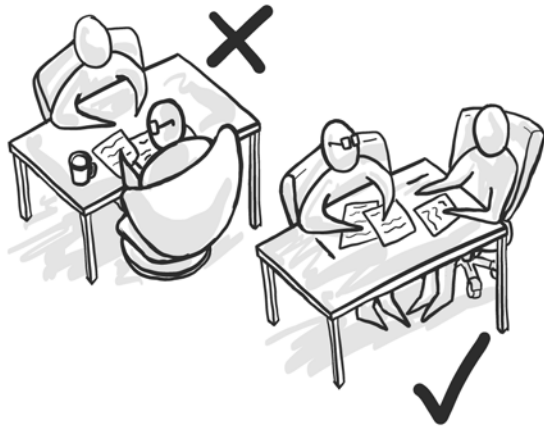
### **ПРИВЕТСТВИЕ**

Соблюдая главный принцип — относиться ко всем кандидатам как к клиентам, — встретьте его лично, проведите по офису, начните разговор. Скорее всего, говорить придется вам: кандидат будет нервничать и стесняться, кроме того, всем известно, что технари вообще не славятся мастерством непринужденной беседы. Задайте простые вопросы, например, как он добрался, легко ли нашел дорогу — ничего такого, что могло бы загнать его в тупик.

Помогите кандидату почувствовать себя комфортно и расслабиться, чтобы он начал вести себя естественно и вы могли понять, подходит ли вам этот человек. Предложите что-нибудь попить (воду, кофе), чтобы подчеркнуть ваше дружелюбное отношение. По возможности покажите офис, в котором ему предстоит работать, и понаблюдайте за реакцией. Будет он доволен или же будет с ужасом озираться в поисках ближайшего выхода, потому что увидел бесчеловечную потовыжималку с рядами ячеек для сотрудников? Не стоит недооценивать, сколько полезного вы можете узнать таким простым способом.

### **РАССАДКА**

Важный вопрос: как посадить кандидата? Прошли те дни, когда кандидаты ютились на низенькой табуретке или же на почтительном расстоянии от того, кто задает вопросы. Усадите его так, как если бы вы разговаривали с одним из членов вашей команды или с клиентом. Если вы сидите за столом — стол не должен разделять вас, потому что подсознательно это воспринимается как преграда. Не оставайтесь по ту сторону стола, когда проводите встречи. Покажите, что вы открыты и готовы к диалогу и что между вами нет препятствий.



Если в вашем офисе нет отдельного стола для совещаний, сядьте сбоку своего стола. Создайте обстановку, в которой кандидат почувствует себя комфортно, расслабится и сможет произвести наилучшее впечатление на вас и вашу команду.

Если в собеседовании участвует более одного человека, рассаживайтесь так же, как на собраниях. Если при этом кто-то будет сидеть рядом с кандидатом — не беда. Идея в том, чтобы не создавать неестественных неловких ситуаций и дать кандидату представление о том, как вы обычно работаете. Хороший кандидат будет уверен в себе и тоже будет задавать вам вопросы, чтобы определить, подходите ли вы ему.

### **СОБЕСЕДОВАНИЕ — НЕ ДОПРОС**

Ни в коем случае не превращайте собеседование в допрос. Не идите механически по заранее подготовленному списку вопросов. Это должен быть разговор, где вы слушаете ответы и направляете беседу в интересующее вас русло таким образом, чтобы кандидат не думал о том, что он отвечает на вопросы. Не старайтесь произвести впечатление — слушайте. Собеседование — это не про вас. Всегда держите в голове основные вопросы, ради ответов на которые вы, собственно, и тратите свое (и его) время на эту встречу:

- Может ли этот кандидат выполнять необходимую работу?
- Впишется ли он в команду?
- Сможете ли вы убедить его присоединиться к вашей команде?

Забудьте все модные статьи, которые вы, возможно, читали и в которых пишут о нестандартных вопросах, якобы задаваемых на собеседовании в Google и других крупных компаниях. Такие вопросы, как «Сколько люков в этом городе?» на самом деле ни для чего не нужны, разве только чтобы казаться намного умнее, чем, вероятно, есть на самом деле. Вы можете убеждать себя, что проверяете, насколько нестандартно кандидат мыслит, но на самом деле вы не получите из ответа информацию, на которую рассчитываете.

Сосредоточьтесь на том, что важно для вашей организации, задачах, с которыми кандидат, скорее всего, столкнется в работе. Лучше всего обсудить настоящие проблемы, с которыми команда сталкивалась в течение года, — такие, которые, на первый взгляд, не имели очевидного решения, но после обдумывания оказались относительно простыми. Если бы кандидат работал в то время, помог бы он команде решить их?

Как мы уже говорили, гуру подбора персонала Лу Адлер считает, что на собеседовании необходимо оценивать, как кандидат решает конкретную задачу. Задавайте вопросы, которые позволят вам определить, может ли он ее решить. Попросите его рассказать об одном из его прошлых проектов, который похож на то, чем занимаетесь вы. Это самая комфортная ситуация для кандидата, и он спокойно расскажет вам о проблемах, с которыми он сталкивался, и о решениях или обходных путях, которые он нашел.

Следуя этому же принципу, схематично опишите гипотетический проект и спросите, как кандидат будет работать над ним. Какие решения он будет использовать (если это разработчик или архитектор) и почему именно их? Обязательно поспорьте с ним, чтобы оценить, насколько он уверен в себе. Он отвечает и обосновывает свой выбор или же сдается, признавая, что не подумал об этом?

Классический вопрос о том, как кандидат справляется со стрессом или конфликтами на работе, даст много информации. Например, попросите его описать последнюю стрессовую ситуацию, когда ему пришлось работать сверхурочно. На кого возложили вину? Извлек ли он уроки из этой ситуации?

В зависимости от позиции, на которую вы ведете подбор, бывает полезно посмотреть, способен ли кандидат указать на ошибку. Вам не нужен тот, кто вечно будет вам поддакивать, как будто вы самый умный в офисе. Проверьте это качество. Не обязательно добиваться, чтобы вам возражали или говорили, что вы не правы, — можете дать кандидату возможность предложить альтернативные решения.

Здесь отлично работает сопоставление двух конкурирующих технологий или стандартов, которые знает кандидат. Например, при сравнении JSON/XML спросите: «Почему JSON сложнее и медленнее парсить, чем XML?» Если вы решите задавать вопросы с подвохом, то, пожалуй, это максимум, который можно себе позволить.

Проблема с такими вопросами в том, что они могут в конечном итоге отпугнуть кандидата и отбить у него желание работать в вашей компании. Он может решить, что вы все время так себя ведете и что он не сможет расслабиться и спокойно работать. Но к тому времени, когда вы будете беседовать с кандидатом лично, вы должны быть уже достаточно уверены в своем желании пригласить его в команду. Если это не так, вам стоило уже давно отказать ему.

Вам необходимо понять, почему кандидат меняет работу (если это не выпускник). Тут нужно выяснить две вещи:

- Как он отзывался о своем предыдущем или текущем работодателе (красный флажок: если плохо — однажды таким работодателем можете стать вы)?
- Можете ли вы предложить ему такие условия, в которых он будет доволен и эффективен?

Оставьте достаточно времени, чтобы кандидат мог задать вопросы вам. Более сильные и уверенные в себе кандидаты будут задавать вам вопросы на протяжении всего собеседования, поэтому часто в конце они говорят: «У меня больше нет вопросов». Записывайте вопросы, которые они задают, чтобы позже подумать над ними.

Как правило, большинство задает свои вопросы в конце встречи. Они хорошо усвоили, что не следует обсуждать зарплату или льготы — это главное табу собеседования. Однако, если все же эти вопросы прозвучат, отложите их обсуждение до разговора один на один после интервью.

### ТЕСТОВОЕ ЗАДАНИЕ

Если вы ищете сотрудника на техническую должность — что очень вероятно, учитывая, что вы читаете книгу для технического директора, — дайте кандидату задание, чтобы понять, как он думает и как работает в команде. Лучше всего взять реальную ситуацию из прошлого опыта, суть которой вы сможете относительно легко и быстро объяснить кандидату и для понимания которой ему не потребуется изучать ваши внутренние термины.

Не делайте задание чересчур сложным — это должно быть что-то, что кандидат сможет разобрать на доске или пояснить ход своих мыслей в поисках потенциального решения. Если нужно будет писать код — используйте псевдокод, чтобы вы обсуждали именно алгоритм.

В некоторых компаниях кандидата оставляют на час наедине с компьютером для написания фрагмента кода, который должен выполнять какую-то относительно простую задачу. На компьютере не будет никаких инструментов, и часть задания состоит в том, чтобы проверить, может ли кандидат самостоятельно установить необходимые инструменты и подготовить все для решения задачи. Но этот подход в первую очередь позволяет выявить кандидатов, которые могут самостоятельно организовать свою работу и не требуют четко описанных процессов. Подойдет ли этот метод вам — зависит от вашей корпоративной культуры. Также следите за собой, чтобы не быть высокомерными: если кандидат просит помощи, это нормально, потому что это показывает, что он знает пределы своих возможностей и обратится к коллегам, если будет нужно.

### ЗАМЕТКИ С ПОЛЕЙ

#### *Пункт оплаты*

Одно из лучших заданий, которое мне встречалось, состояло в том, что кандидату предлагалось спроектировать бэкенд для автоматического пункта оплаты проезда по автомагистрали. Задание было поистине гениальным, поскольку его предмет был хорошо знаком каждому, но в ходе работы необходимо было учитывать задержки, быстродействие, удобство для пользователя (чтобы ему не приходилось ждать даже несколько секунд), а также каким образом сообщить пользователю, что у него заканчиваются средства. До сих пор каждый раз, когда я проезжаю один из таких пунктов оплаты, я вспоминаю об этой задаче. Кстати, этим кандидатом был я, и, да — я получил эту работу.

### ПРОЩАНИЕ

По окончании собеседования вы или кто-то из вашей команды должны проводить кандидата, точно так же, как вы делаете это для клиентов компании. Лучше, если это будет кто-то, кто не участвовал в собеседовании. Тогда можно будет ненавязчиво, в рамках вежливой беседы, поинтересоваться, как все прошло. Более проницательные кандидаты знают, что собеседование продолжается до тех пор, пока они не покинут здание. Вы будете удивлены, сколько информации вы получите, просто проводив кандидата до выхода.



### ЗАМЕТКИ С ПОЛЕЙ

#### *Слева или справа?*

Раньше я часто использовал такой прием. Если, прощаясь с кандидатом на пороге своего офиса, я передавал его сотруднику, провожавшему его до выхода, и стоял при этом слева, это означало, что, скорее всего, этот кандидат нам не подходит. Если же я стоял справа, то давал понять, что кандидат мне понравился. Таким образом сопровождающий понимал, какие вопросы можно задать кандидату по пути и о чем поговорить, в том числе еще раз подчеркнуть преимущества работы у нас, если мы действительно хотели, чтобы этот кандидат почувствовал, что наши условия ему подходят.

## 5.1.5. Видеоинтервью

Если по какой-то причине человек не может приехать на собеседование в офис, можно провести интервью по видеосвязи. Это следует делать только в крайнем случае. Ничто не сравнится с личной встречей, и, хотя видеоинтервью может дать гораздо больше, чем общение по телефону, вы все равно теряете много информации. Даже когда вакансия подразумевает удаленную работу, если ваш бюджет позволяет организовать личную встречу в офисе или другом месте (хотя бы в Starbucks), обязательно сделайте это.

### ПОДГОТОВКА

Прежде всего, выберите подходящий сервис для видеоконференций. То, что хорошо для звонков друзьям и семье, может не подойти для всесторонней оценки кандидата. Классические варианты, в том числе Google Meet, Microsoft Teams, Zoom, Apple FaceTime и Skype, совместимы с разными платформами, поэтому,

если кандидат использует ПК, Mac или даже мобильный телефон, проблем быть не должно.

Коммерческие решения включают GoTo Meeting и WebEx, есть даже специально ориентированный на проведение собеседований сервис BlueJeans, все они предлагают такие функции, как чат, шеринг экрана и совместная работа, а также возможность записи всего сеанса связи. Использование этих инструментов во многом зависит от того, какое собеседование вы планируете провести. Вы можете, например, показать кандидату фрагмент кода и попросить проверить его под вашим наблюдением.

Заранее сообщите кандидату, какую технологию вы будете использовать, и попросите кого-нибудь из команды провести тестовый созвон перед собеседованием. Убедитесь, что пропускная способность канала достаточна, чтобы спокойно провести часовую встречу. Кандидат не должен нервничать еще и из-за того, что его не слышат или он не слышит вас и ваших сотрудников.

Видеособеседование — это улица с двусторонним движением; проверьте, что кандидат видит вас и остальных интервьюеров. Ничто так не обезличивает и не обесценивает общение, как односторонняя видеосвязь. Если из-за технических неполадок невозможно обеспечить стабильный видеосигнал, перенесите интервью на другое время или же попросите кандидата выключить камеру и перейти на голосовую связь. Не забывайте относиться к каждому кандидату как к клиенту, даже во время видеозвонка.

## СОБЕСЕДОВАНИЕ

Видеособеседование само по себе не отменяет наблюдений за кандидатом и оценки его поведения, как при очном интервью. Необходимо лишь учитывать ограничения, накладываемые используемой технологией. Качество видео может быть плохим (например, иногда снижается разрешение картинки) или же освещение может не соответствовать голливудским стандартам — пусть это не влияет на ваши выводы.

Тем не менее то, как кандидат ведет себя в ходе видеовстречи, многое говорит о его характере:

- Он выглядит так, будто только что проснулся, и даже не удосужился подобающе одеться?
- Нашел ли он тихое уединенное место или у него за спиной ходят люди?
- Как он сидит перед камерой? Он позаботился о том, чтобы всегда быть в кадре?
- Когда вы говорите — выглядит ли он внимательным слушателем или же в это время занимается чем-то еще, например ищет ответ в интернете (такое бывает)?

Будете ли вы учитывать эту информацию, решать вам, но эти подсказки могут сообщить вам многое.

Вы можете обратить внимание на кажущуюся напряженность кандидата. При личном общении собеседник будет (должен) в основном смотреть вам в глаза, но при этом, когда он говорит — он будет поворачиваться в ту или иную сторону, и его взгляд тоже. Во время же видеоинтервью кандидат всегда смотрит на экран, даже когда он не смотрит на вас, поэтому кажется, что он очень напряжен. Простите ему этот убийственно пристальный взгляд.

В ходе видеособеседования также используйте вашу систему оценки, ноделите на собеседование немного больше времени. Общение при помощи технологий обычно проходит не так быстро, как личное. Говорите немного медленнее (и четче), чем делаете это обычно при встрече лицом к лицу.

### **СОВМЕСТНЫЕ ЗАДАНИЯ**

На видеособеседовании не получится поработать с доской, которую вы могли бы использовать при очной встрече. Поэтому необходимо найти альтернативный способ оценки, используя функционал совместной работы.

Скорее всего, вы будете искать технического специалиста, поэтому вы можете предложить совместно просмотреть фрагмент кода и попросить кандидата внести в него изменения или добавить комментарии, а сами будете наблюдать. Задания должны показывать ход мыслей кандидата — не требуйте от него полноценного и синтаксически безупречного решения. При совместном использовании текстового редактора подход должен быть таким же. Если должность не подразумевает программирования, организуйте процесс так, как будто кандидат находится с вами в одной комнате: попросите его изложить решение на бумаге, а затем показать вам, поднеся лист к камере. Предупредите заранее, что понадобится ручка и бумага, хотя большинство кандидатов и так их подготовят, потому что будут делать заметки.

Такие решения, как Google Docs или функция Live Coding в Visual Code, позволяют совместно редактировать документ в режиме реального времени, и это хороший способ выделить интересующую область документа и обсудить ее.

## **5.2. ОЦЕНКА КАНДИДАТОВ**

Поток собеседований отнимает много сил и энергии. Как бы вы ни верили, что запомните каждого кандидата, со временем вы начнете их путать друг с другом, забывая об их качествах (хороших или плохих) и о том, какое впечатление произвел на вас кандидат во время собеседования. Вы подведете и сами себя, и кандидата, если во время встречи не зафиксируете все, что необходимо, чтобы вы могли не только вспомнить все достоинства и недостатки, но и качественно сравнить кандидатов друг с другом.

В этом разделе я опишу подход с использованием листа оценки, который позволяет составить профиль кандидата и измерить его уровень по простой шкале. Это поможет затем принять решение, основанное на данных, и обеспечит равные условия для всех соискателей.

### 5.2.1. Выбор критериев оценки

Чтобы распознать нужного вам кандидата, вы должны знать, что именно он должен собой представлять. При подготовке описания позиции у вас уже появится список качеств, которые, по вашему мнению, важны. Сюда будут входить социальные навыки (soft skills) — личные качества, навыки общения, выступлений и т.д. и профессиональные (hard skills) — знание языков программирования, навыки работы с кодом и т.д.

Многие интервьюеры испытывают с этим трудности и в конечном итоге указывают слишком общие критерии. Проще всего тут просто представить себе человека, который через несколько недель после начала работы выполняет поставленные перед ним задачи. Подумайте, какие навыки ему нужны для этого. Например, если он взаимодействует с клиентом и выясняет его потребности, нужны будут хорошие навыки общения и способность слушать. Если же он пишет запросы к базе данных, то вам нужно знание SQL.

Эти критерии не должны выглядеть как последовательность вопросов, которые вы будете задавать на собеседовании. Наоборот, они являются результатами собеседования, которые вы определите на основе ответов на ваши вопросы или вопросов, полученных от кандидата. Чтобы не превратить критерии оценки в вопросы, составьте из них список целей, например такой:

- Социальные навыки:
  - внешний вид;
  - речь и общение;
  - деловой этикет / отношение к работе.
- Профессиональные навыки:
  - жизненный цикл разработки ПО;
  - паттерны проектирования;
  - знание определенного языка программирования.

Для разных ролей критерии будут разными, поэтому уделите достаточно времени, чтобы у вас получился список из 10–20 пунктов. По каждому из них вы будете оценивать кандидата по шкале от 1 до 10 — чем выше балл, тем лучше оценка.

Теперь сложная часть: разработка шкалы. Для каждого из критериев необходимо определить, какой уровень будет соответствовать той или иной оценке. Рассмотрим пример. Допустим, вам нужны навыки работы с AWS. Вы можете проранжировать их следующим образом:

- 0  
Нет опыта работы с AWS.
- 1–3  
Использовал отдельные сервисы, такие как S3 или EC2.



- 4–6

Создал и настроил собственную учетную запись, использовал IAM для управления пользователями.

- 7–10

Имеет опыт создания в VPC подсетей, использования групп безопасности и шлюзов NAT, а также работы с ролями в IAM.

Оценка 10 означает, что кандидат ответил на все ваши вопросы об AWS и обладает более чем достаточными знаниями в этой области. Она не означает, что кандидат знает вообще все об AWS — цель здесь не в этом, а в том, чтобы выяснить, знает ли он достаточно для выполнения ваших задач.

Разработав критерии оценки, разошлите их всем, кто будет взаимодействовать с кандидатом, и проверьте, что все понимают эти критерии и те качества, которые необходимо будет измерить. Не каждый сможет оценить кандидата по всем параметрам — это нормально. Чем больше данных вы сможете собрать, тем лучше.

### 5.2.2. Выставление оценок

Можно делать заметки в листе оценки во время собеседования (не привлекая к этому лишнего внимания), можно заполнить его и сразу по завершении, пока впечатления еще свежи. Главное — не затягивать с этим, иначе будет сложно вспомнить все детали. Пример листа оценки показан в табл. 5.1.

Как видите, лист оценки довольно прост. В примечаниях можно в нескольких словах пояснить, почему вы поставили такую оценку — на что конкретно вы обратили внимание? Так вы не забудете необходимые подробности, когда будете сравнивать кандидата с остальными.

**Таблица 5.1.** Лист оценки кандидата

Критерий	Оценка/10	Примечания
Внешний вид		
Речь и общение		
Знание AWS		

Этот механизм оценки основывается на данных и упрощает выбор подходящего кандидата. Но не обязательно принимать на работу именно кандидата с наивысшим баллом, потому что на решение могут влиять и другие факторы, не отраженные в листе оценки, которые вы можете сформулировать при обсуждении в команде. У каждого интервьюера есть свой взгляд на сильные и слабые стороны кандидатов, и эти мнения необходимо учесть. Интересные дискуссии возникают, когда разные интервьюеры сильно расходятся в своих оценках. В этом случае, возможно, кто-то пересмотрит свое мнение после обсуждения. Его эффективность зависит от следующих условий:

- Точно определены наиболее важные критерии для должности.
- Для каждого критерия описаны диапазоны оценок.

- Каждый интервьюер понимает принципы оценивания кандидата.
- Лист оценки для кандидата был заполнен своевременно.

Итак, теперь у вас есть общая схема для обсуждения, позволяющая избежать личных предпочтений в процессе принятия решения и сделать выбор на основе качеств, необходимых для вашей позиции.

Со временем вы станете оценивать кандидатов точнее и эффективнее. Вы и команда узнаете, что действительно работает, а что нет и как получить от кандидата ту или иную нужную вам информацию. Как всегда, для совершенства необходима практика.

### 5.3. КАК СКАЗАТЬ «НЕТ»

В процессе подбора много ошибок совершается при общении с кандидатами, которых необходимо уведомить об отказе. Причина проста — мы не любим сообщать плохие новости. В этом разделе рассмотрим, как справиться с этой важной задачей, чтобы у соискателя остались положительные впечатления, несмотря на отказ.

Никогда не игнорируйте не прошедших отбор кандидатов, делая вид, что их не существует (не отвечая на их электронные письма или звонки) в надежде, что они сами поймут, что не подошли. Так вы вызовете в человеке обиду и враждебность по отношению не только к вам лично, но и к компании в целом. Вы не знаете, где этот человек в дальнейшем окажется и с кем он будет разговаривать. В конце концов, он может устроиться к одному из ваших клиентов. Или к конкуренту. В целом такое поведение неприемлемо, свидетельствует о трусости и может привести к непредвиденным последствиям.

Итак, мы выяснили, что сообщить об отказе необходимо; теперь разберемся, как это лучше всего сделать. Обычно тех, кто не прошел отбор, можно разделить на две категории. Первая — это те, кто абсолютно не соответствует культуре вашей компании или не имеет необходимой квалификации, и этих людей вы никогда не будете рассматривать в качестве сотрудников. Вторая группа — это те, кто может подойти компании — пусть не сейчас, но когда-нибудь в будущем. У них есть необходимые навыки и качества. Но независимо от того, к какой группе относится кандидат, он заслуживает уважения и честного отношения.



#### 5.3.1. Определенно «нет»

Человек может не подходить по множеству причин, и, если ваша система критериев хорошо продумана, она ясно покажет, что кандидат далек от вашего идеала. Поэтому говорите просто и по делу. Поблагодарите за проявленный интерес к вакансии и скажите, что ему не удалось пройти отбор.

Хороший кандидат, который действительно хочет учиться на опыте, спросит, почему он потерпел неудачу. Не пугайтесь этого вопроса. Однако у кандидата не должно сложиться впечатление, что вы торгуетесь и у него еще есть шанс.

Лучше всего ответить на этот вопрос честно, но в общих чертах. Укажите, в чем кандидат не оправдал ваших ожиданий — возможно, эта позиция ему вообще не подходит.

Иногда человек будет упорствовать и говорить, что он может стать лучше. Подобный поворот беседы только подтвердит, что это не тот человек, которого вы ищете. Не спорьте, еще раз отметьте, что уровень кандидата недостаточен для вас, и вновь поблагодарите его за интерес к вашей компании. Старайтесь избегать разбора социальных навыков и сосредоточьтесь на профессиональных. Оценка социальных навыков более субъективна, и ее может быть сложнее обосновать, если кандидат начинает спорить.

Вы сообщили кандидату достаточно, чтобы он принял эту информацию к сведению и воспользовался ею в будущем для других карьерных возможностей. Не углубляйтесь в подробности. Если, например, у кандидата недостаточный опыт работы с Java, просто так и скажите, потому что, если перейти к деталям, может оказаться, что вы на самом деле не помните всех подробностей и не сможете обосновать ваше решение. Будьте профессиональны и кратки.

### 5.3.2. Определенно «может быть»

В эту группу входят те, кому не хватило совсем немного или кого просто превозмогал более сильный кандидат. Любой из этих кандидатов потенциально может стать вашим сотрудником, и если где-то ему не хватает знаний, то его можно обучить, чтобы он соответствовал вашим стандартам. Относитесь к этой группе как к кандидатам на будущее, которые уже прошли проверку. Вы потратили на них свое время и энергию — пусть они будут потрачены не напрасно.

Вы также должны быть готовы к тому, что ваш основной кандидат скажет «нет». Не всегда мяч попадает точно в корзину, и тот, кого вы выбрали, не всегда принимает ваше предложение. Возможно, он уже согласился на работу в другой компании, его не устроила зарплата или льготы или ему не понравилось то, что он увидел на собеседовании (помните: кандидаты тоже оценивают вас).

В этом случае вам придется обратиться ко второму, третьему или четвертому подходящему варианту. Если они не полностью соответствуют вашим критериям, но при этом находятся в категории «может быть», решите, достаточно ли они сильны, чтобы рискнуть, с учетом того, что им может потребоваться дополнительное обучение. Если в данный момент вы не можете себе позволить подобный риск, придется начать подбор заново. Бывает и такое.

Однако предположим, что наиболее подходящий кандидат принял ваше предложение. Вам придется сообщить остальным, что они не прошли отбор. Лучше всего сделать это прямо. Скажите, что они были хороши, но есть более сильный кандидат. Спросите их, не возражают ли они, если вы сохраните их данные, чтобы оставаться на связи и иметь возможность пригласить их в будущем, если что-то изменится.

Если у вас есть профиль в LinkedIn, попросите разрешения добавить их в вашу сеть контактов, это подтвердит искренность ваших слов. В целом, это полезно, особенно в случае кандидатов, которые, возможно, переоценили свои навыки и были не совсем готовы к повышению, на которое они претендовали, но со временем будут.

В следующий раз вы сможете очень быстро и с минимальными затратами привлечь кандидатов из этого пула, поскольку вы уже собеседовали и оценивали их, и познакомили их с компанией. Также не исключено, что вместо одного человека вы решите принять на работу сразу двоих, потому что их адаптацию и обучение можно провести одновременно. Естественно, это решение необходимо согласовать с генеральным и финансовым директором.

## 5.4. АДАПТАЦИЯ

Представьте, что вы приехали в отель, а за стойкой регистрации никого нет; когда же сотрудник приходит, он не может найти для вас номер. После того как номер найден, вы обнаруживаете, что он не убран после предыдущих постояльцев, постельное белье не заменено и повсюду мусор. Вы возмущаетесь, и совершенно справедливо. Тем не менее именно так многие компании встречают новых сотрудников. Как и отель, компания за несколько дней или даже недель знала об их прибытии, так что у нее нет оправдания.

В этом разделе мы рассмотрим тему, которую часто упускают из виду: как встретить нового сотрудника и ввести его в должность таким образом, чтобы создать условия для продуктивной работы. Это требует планирования, подготовки и правильного выполнения. В крупных организациях есть отдел по работе с персоналом, который, возможно, уже организовал этот процесс во всех деталях, но часто он берет на себя только базовые вещи, оставляя адаптацию отделу, куда выходит сотрудник.

Первое впечатление важно. В первые несколько дней работы новый сотрудник либо убедится, что принял правильное решение, либо начнет сомневаться в нем. Скорее всего, у него были и другие предложения, к которым еще не поздно вернуться, если ему не понравится у вас.

### 5.4.1. Вводим в курс дела

Прежде чем заказывать сотруднику ноутбук или выделять рабочее место — необходимо рассказать ему о жизни вашей компании и клиентов. Это кажется мелочью, однако только изучить сайт компании недостаточно. Но не стоит и раскрывать какие-то ваши секреты — просто более подробно расскажите, что вы делаете для своих клиентов и как вы это делаете. Можете дать больше информации о каких-то проблемах, особенно если именно ваш отдел занимается их решением. Вот наиболее важные вещи:

- Подробно изучить сервис или продукт, который вы разрабатываете.
- Познакомиться с основными понятиями и терминами, которые вы используете, и с их контекстом.

- Какие клиенты являются наиболее важными и с кем вы работаете дольше всего? Какие из них имеют особое значение для компании (часто это первые несколько клиентов)?
- Какие еще отделы есть в компании — особое внимание уделите тем из них, с которыми придется взаимодействовать больше всего.
- Любые правила компании или отдела, о которых нужно знать.
- Краткий обзор основных совещаний, в которых сотрудник может участвовать.

На большинство из этих тем не должно уйти более дня, и в зависимости от размера команды вы можете поручить другим ее членам обсудить их с новичком. Запланируйте задачи в их календаре, чтобы это не стало ни для кого сюрпризом.

Наконец, не забудьте показать новичку, где расположены туалеты, кухня, кофе и т. д. Встретьте его у входа в офис и устройте экскурсию, по дороге знакомя с коллегами. Это поможет новому сотруднику расслабиться и уменьшит стресс первого рабочего дня.

Ни в коем случае не оставляйте новичка одного, вручив ему для изучения пачку документов. Не обманывайте себя, думая, что так он войдет в курс дела. На его месте вы бы встали, вышли из офиса и позвонили в другую компанию, которая сделала вам оффер. Так что помогите ему почувствовать себя нужным и вовлекайте в работу с первой минуты.

### 5.4.2. Ваши ожидания

После вводной части переходите к главному: как работать с вами и командой. Это самая важная часть процесса адаптации — как новичку добиться успеха в вашей команде?

Не оставляйте его разбираться в этом самостоятельно. Объясните принятые в команде способы взаимодействия и общения. Расскажите, в каких ситуациях он может попросить вас о помощи и о чем он должен ставить вас в известность. Какие каналы связи вы предпочитаете — личное общение, видеочат, мессенджер или же электронную почту. Все это может показаться мелочами, но у каждого руководителя свой стиль управления, и успеха добиваются те из них, кто избавляет подчиненных от необходимости строить догадки.

### 5.4.3. Приступаем к работе

Наконец, сотрудник должен приступить к работе. Для успешного старта тоже необходима подготовка. Убедитесь, что вы заранее настроили для новичка корпоративную электронную почту и все нужные аккаунты. Не допускайте ситуации, когда он не сможет толком ничего сделать, потому что то одно, то другое не работает — не настроено или не предоставлен доступ. Это будет тормозить работу как сотруднику, так и вам.

Дайте ему несколько простых заданий, чтобы он познакомился с тем, как вы работаете, и сразу же показал результаты. Разработчику, например, можно дать задание внести в код небольшие изменения, чтобы он спокойно изучил жизненный цикл вашего ПО, — это отличный способ познакомиться со всем.

Уделите достаточно времени на подготовку к этому этапу, потому что по мере роста команды вы будете заниматься этим все чаще и вы не должны выглядеть как курица без головы, у которой нет плана и которая все придумывает на ходу.

## ИТОГИ

- Проведение собеседований — сложная работа. Чтобы научиться это делать, необходима практика; и никогда не относитесь к ним всего лишь как к разговору.
- Собеседование — это больше чем просто часовая беседа с кандидатом. Наблюдая за тем, как кандидат ведет себя до и после собеседования, вы получите важную информацию, которая дополнит ответы соискателя на ваши вопросы.
- Оценивать кандидатов необходимо с использованием единой методологии, чтобы легко и быстро сравнить их после собеседований и исключить любую предвзятость.
- Выбор одного кандидата не означает, что вы должны автоматически отбросить остальных; многое может пойти не так, и ничего не решено окончательно, пока сотрудник не вышел на работу.
- То, как вы говорите «нет» тем, кто не прошел отбор, говорит о вас больше, чем то, как вы поздравляете победителя.
- Адаптация нового сотрудника требует усилий, и, даже если он вышел на работу в первый день, это не значит, что он обязательно вернется на второй. В нашей отрасли у хороших специалистов не бывает проблем с поиском работы. Не слушайте гордыню, убеждающую вас в том, что вы — их единственный вариант.

## ПРОВЕРЬТЕ СЕБЯ

Что из перечисленного ниже вы уже сделали?

- Попрактиковались минимум в нескольких техниках интервью, прежде чем использовать одну из них.
- Разработали четкую стратегию собеседования новых членов команды и точно понимаете, кто вам нужен.
- Создали лист оценки для фиксации того, насколько тот или иной кандидат подходит на вакансию.
- Проинструктировали других членов команды, которые будут участвовать в процессе собеседования.
- Продумали, как сообщать об отказе тем, кто не прошел отбор, чтобы оставить эту дверь открытой на будущее.
- Составили план адаптации для новых сотрудников, чтобы они как можно быстрее освоились и не чувствовали себя потерянными.

# 6

## Управление командой

---

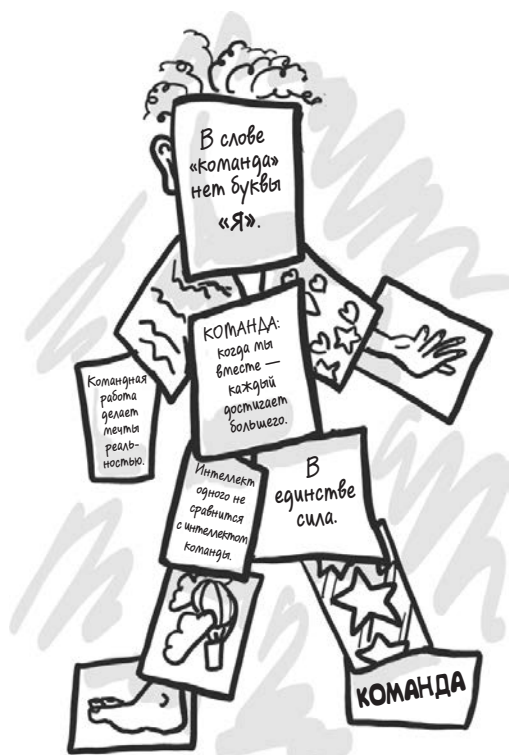
### В этой главе

- ✓ Создание эффективных команд
- ✓ Какие бывают структуры команд
- ✓ Регулярное обучение и тренинги
- ✓ Способы взаимодействия в команде
- ✓ Виды совещаний, правила и порядок их проведения

Стены множества офисов украшают мотивационные постеры с цитатами и слоганами о командах и их силе — например, такие:

- В слове «команда» нет буквы «я».
- Командная работа делает мечты реальностью.
- В единстве сила.
- Интеллект одного не сравнится с интеллектом команды.
- КОМАНДА: когда мы вместе — каждый достигает большего (TEAM: Together Everyone Achieves More).
- Работа в команде помогает обычным людям добиваться необычных результатов.

Какими бы банальными ни были некоторые из них, они во многом справедливы. Хорошая команда может добиться всего, к чему она стремится, и задача технического директора — создавать команды и направлять их, чтобы ваш бизнес и бизнес ваших клиентов был успешным. Существует множество книг по менеджменту, которые подробно описывают, как подобрать идеальную команду



и управлять ею — обязательно добавляйте их в свою библиотеку по мере вашего карьерного роста. В рамках этой книги мы остановимся подробнее на основных задачах и идеях, без понимания которых невозможно будет добиться успеха в командной работе. Главная же цель существования вашей команды очень проста — вы не можете делать все сами.

Чтобы работа команды была успешной, четко сформулируйте ваши требования, зону ответственности команды и ожидаемые результаты. Создание команды похоже на выбор ПО. Вы ожидаете, что программы будут выполнять определенную задачу или набор задач, а вы сможете сосредоточиться на результате и использовать его для достижения ваших целей. Представьте себе, что каждый раз, когда вы открываете таблицу с бюджетом в Microsoft Excel, вам приходится лезть в исходный код, что-то менять в нем и компилировать программу, и только после этого вы сможете увидеть свои цифры.

Команда — это компонент, который должен выдавать определенный результат с заданным уровнем качества и надежности, так, чтобы этот результат можно было использовать как исходные данные для другого процесса или системы. Хорошая команда способна думать сама за себя, брать на себя необходимую ответственность, и ей нужно предоставить достаточно свободы для самостоятельного принятия решений при выполнении задач.

### ОСТОРОЖНО

#### *Проклятие микроменеджера*

Микроменеджер — это тот, кто следит за каждым шагом своих подчиненных, подробно объясняя, как именно что-то должно быть сделано, проверяя каждую мелочь и в целом мешая работе. Большинство людей не видят этой проблемы в себе и оправдываются тем, что таков их способ добиться высокого качества. Консультируя частные инвестиционные компании, я встречаю множество таких руководителей, и, в конце концов, если они не могут изменить свой стиль управления, от них приходится избавляться, потому что они становятся узким местом, блокируют рост и всегда разрушают мотивацию команды.



## 6.1. УСТАВ

Для каждой из своих команд составьте список того, что вы от них ожидаете и чего требуете, — это будет их устав. Он фиксирует цель существования команды и то, каков ее вклад в общий успех отдела.

Затем руководитель команды должен разработать план достижения закрепленных в уставе целей и включить в него все необходимые ресурсы, бюджет и инструменты. Устав должен быть ясным и кратким, не нужно делать его длинным (просто несколько списков, если необходимо), и должен быть доступен каждому в вашем отделе. Таким образом, все сотрудники будут знать свою роль в технической службе.

И, наконец, самое главное — предоставьте руководителям команд возможность выполнить их работу и реализовать все, что написано в уставе, на практике. Позже мы подробнее обсудим, что ответственность за достижение целей несут именно они, а вы должны оказывать им поддержку и руководство.

### ВИЗУАЛИЗИРУЙТЕ

Как генерал перед битвой, я расставляю на большом листе бумаги человечков LEGO, чтобы наглядно представить организацию команд и провести между ними границы ответственности. Я буквально рисую линии между командами и делаю пометки. Этот метод визуализации позволяет быстро заметить, где я что-то упустил или же, наоборот, что в какой-то области ресурсов слишком много. Небольшой совет: если вы используете набор LEGO «Звездные войны», отложите в сторону Дарта Вейдера — никто не хочет им быть.

Если вы унаследовали отдел, скорее всего, в нем уже будут сформированы команды, но формально зафиксированных требований у них обычно нет. В таком случае вам придется потратить время на то, чтобы выяснить, чем занимается каждая команда, и зафиксировать это. Не начинайте вносить изменения, не разобравшись до конца в существующих процессах. После того как вы составите устав, вы сможете его менять на основе вашего опыта и понимания, работают ли те или иные вещи, а также в случае изменений в направлении деятельности компании и соответствующей смены приоритетов.

Когда к команде присоединяется новый сотрудник, он должен первым делом получить представление о своей роли в команде и о функциях команды в целом. Проведите эксперимент: если вы сейчас попросите всех сотрудников одной из ваших команд сформулировать цель команды в одном предложении, сколько разных ответов вы получите? Если все ответы будут совершенно разными, то это не команда, а просто собранные в одном месте люди, каждый из которых отрабатывает свою зарплату. Устав поможет всем им сфокусироваться.

### 6.1.1. Знания

Чтобы команда была успешной, она, во-первых, должна иметь цель. Для этого существует устав. Во-вторых, у нее должны быть ресурсы для достижения этой цели. Ответственность за их определение и согласование лежит на вас и руководителе команды.

Также по мере развития команда накапливает опыт и генерирует знания. Эти знания необходимо фиксировать, чтобы новые члены команды могли ими воспользоваться и работать более эффективно.

Слишком часто команда строится вокруг одного человека, и только этот человек владеет всей информацией, хотя, возможно, это происходит и не по его вине. Если этот человек уйдет — команда понесет огромные потери. В сильной, устойчивой команде уход любого отдельного сотрудника не должен негативно влиять на общий результат. Для этого знания команды можно фиксировать, хранить и делать доступными следующими способами:

- *Вводные/учебные материалы.* Как и в компании в целом, в каждой команде должны быть материалы для адаптации и обучения, соответствующие ее задачам. Например, в команде, ответственной за релизы, можно изучать используемые для этого инструменты (Jenkins/SonarQube и т. п.).
- *База знаний.* В наши дни едва ли найдется компания, в которой нет своей вики или хранилища документов, хотя то, насколько они обновляются и поддерживаются в актуальном состоянии, — это отдельный вопрос. Но база знаний не обязательно должна состоять из документов; в ней могут быть и видеоролики, а их, в зависимости от вашей темы, может быть даже проще сделать.
- *Описанные процессы/правила.* Чтобы команда могла функционировать, она должна иметь правила. Правила не должны быть неявными или само собой разумеющимися, их необходимо описать и донести до всех, чтобы все следовали им для достижения общей цели. Например, для команды разработчиков можно зафиксировать стандарты написания и форматирования кода, а также использование системы контроля версий при выполнении тех или иных задач.

Если просто собрать людей и сказать им, что они теперь команда, — от этого командой они не станут. Чтобы добиться успеха, им нужно предоставить инструменты и правила.

### 6.1.2. Образец устава

Устав не должен быть ни длинным, ни сложным или заумным. Достаточно описать в нем в общих чертах цели команды. Вот пример устава группы поддержки онлайн-сервиса:

- *Цель* — обеспечивать оперативную поддержку конечных пользователей по всем вопросам взаимодействия с сервисом, включая помощь менеджерам из

отдела по работе с клиентами в решении любых проблем, связанных с созданием новых пользователей.

- Команда полностью управляет всеми необходимыми для этого данными, но не вносит никаких изменений в код. При необходимости исправить какие-либо ошибки или добавить новые функции она создает тикеты на команду разработчиков.
- *Клиент* — любой пользователь продукта, как внутренний, так и внешний.
- *Ресурсы*
  - База знаний о структуре внутренних данных в виде внутренней вики.
  - Доступ к данным продакшена только для чтения.
  - Полный доступ к данным продакшена для отдельных сотрудников.
- *Требуемые навыки*
  - Знание веб-технологий, включая системы видеоконференций/чаты.
  - Грамотная письменная и устная речь.
  - Знание SQL.
- *Ключевые показатели*
  - Время до закрытия инцидента.
  - Общее количество обращений за определенный период.

## 6.2. СТРУКТУРА КОМАНДЫ

Сразу оговоримся: не существует идеальной структуры, подходящей для всех. То, что работает для одной команды, не обязательно будет работать для другой. На то, какая структура сложится у вас, влияют многие факторы, в том числе корпоративная культура, личные качества и темперамент сотрудников, важность их работы и необходимость взаимодействовать с клиентами.

Иерархическая структура, предполагающая наличие руководителя или менеджера, к которому члены команды обращаются за помощью и руководством, является более традиционной и пришла из армии. Другой распространенный вариант, особенно в технологических компаниях, — более плоская структура, в которой каждый наделяется большей личной ответственностью, и ожидается, что он без указаний знает, что делать.

Иногда, хотя и не часто, внутри больших технических отделов бывают команды с разной структурой. Например, в службе поддержки может быть руководитель, который за всем следит, а команда, занимающаяся исследованиями, может быть менее формальной.

В популярных реалити-шоу, таких как *Survivor* в США или *The Apprentice* в Великобритании, создают команды с самым разношерстным составом участников

и разной структурой, все это для создания конфликтов и драм с целью развлечения. Но эти конфликты вполне реальны, поэтому стоит позаботиться о подборе структуры, которая лучше всего подойдет для вашей команды.

Начиная обдумывать структуру, учтите два условия, которые можно считать постоянными. Первое, довольно очевидное: есть только один технический директор (это вы), и этот человек отвечает за все, что делает ваш отдел, как за успехи, так и за неудачи. Структура, которую вы создадите, должна обеспечивать прозрачность и помогать видеть, что происходит в команде, чтобы вы никогда не впадали в заблуждения. Второе является скорее ограничением, выведенным на основе многих лет управленческой практики в разных отраслях: количество прямых подчиненных не должно превышать 5–7 человек. Прямой подчиненный — это человек, которого вы наставляете, направляете, обучаете и в целом им руководите. Обычно считается, что невозможно эффективно управлять более чем семью подчиненными, потому что это заставляет слишком сильно распыляться.

Если ваша команда создана недавно и только начинает работу, в ней еще может не быть семи человек, в этом случае сейчас вам беспокоиться не о чем. Но думайте о будущем. При составлении планов считайте, что ваша команда увеличится вдвое.

### ИСКЛЮЧЕНИЯ

#### *Большие команды*

Конечно, цифра семь — это скорее ориентир, чем строгое правило. Часто оно нарушается в группах поддержки и тестирования, которые могут включать большое количество сотрудников одного уровня, выполняющих одинаковые задачи и в обычных условиях почти не требующих управления. Но даже в этом случае менеджер, руководящий этой большой командой, должен иметь соответствующий опыт управления, и, скорее всего, он будет не из технической среды.

С учетом того, что в каждой команде должно быть не больше семи сотрудников, вам понадобится найти способ разделять и властвовать, который будет работать в вашем конкретном случае. Рассмотрим некоторые подходы к организации команд.

## 6.2.1. Вокруг продукта

Если вы разрабатываете один или несколько продуктов, может иметь смысл организовать ваши команды вокруг продуктовых линеек. Каждой линейке продуктов будет соответствовать свой набор подразделений, таких как разработка, тестирование и поддержка, каждым из которых будет руководить отдельный менеджер, все они будут подчиняться еще одному менеджеру, который будет уже отчитываться перед вами.

Это частый сценарий для компаний, у которых есть один или несколько установленных продуктов, каждый из них со своим собственным жизненным циклом и требованиями. Чтобы понять, подходит ли вам такая структура, подумайте, выполняются ли для ваших продуктов какие-нибудь из следующих условий:

- разные стеки технологий;
- непересекающиеся клиентские базы;
- независимые релизные циклы;
- уникальные для разных продуктов и трудоемкие проблемы поддержки;
- значительная разница в прибыли, которую генерируют разные продукты.

Эта структура должна работать как мини-экосистема внутри большой организации и позволять каждой команде обслуживать своих пользователей, не влияя на работу других команд. Эта модель также подойдет, если вы планируете в будущем слияние продуктов или интеграцию одного из них в другой, поскольку она позволит продолжать обслуживать клиентскую базу, пока вы параллельно с этим планируете миграцию.

### 6.2.2. Вокруг жизненного цикла

Это, вероятно, самая популярная модель из используемых компаниями, разрабатывающими собственное программное обеспечение. Такая структура подобна заводскому конвейеру, где каждый отдел отвечает за определенную часть процесса создания и эксплуатации продукта. Например, у вас может быть команда разработчиков, команда тестирования, команда, ответственная за релизы, команда DevOps/инфраструктуры и, наконец, служба поддержки. Каждую из этих команд возглавляет отдельный руководитель, который будет находиться в вашем непосредственном подчинении.

Эта модель хорошо работает, если в организации существует четко определенный и работающий SDLC (Software Development Life Cycle, жизненный цикл разработки программного обеспечения). Скорее всего, на начальном этапе создания проекта каждый из ваших сотрудников будет выполнять сразу несколько ролей. Однако придет время, и вам потребуются специалисты с более глубокими знаниями в каждой области. Тогда вы сможете выделить их в отдельные команды с определенным кругом задач.

### 6.2.3. Вокруг клиента/вертикали

У вас может быть несколько крупных клиентов, или вертикалей, каждая со своими уникальными требованиями, но использующая одну и ту же платформу. Допустим, вы работаете на платформе Salesforce, но для разных клиентов или рынков у вас используются совершенно разные конфигурации. Или же вы одновременно работаете в сферах здравоохранения, финансов и образования. При этом имеет смысл соответствующим образом сгруппировать ваших сотрудников,

чтобы команды могли сфокусироваться на изучении уникальных особенностей каждого из сегментов, что делает их более ценными для вас и ваших конечных пользователей.

#### 6.2.4. Гибридный подход

Наконец, возможна гибридная структура — когда берется понемногу от разных подходов. Такое встречается довольно часто, особенно в крупных организациях, в которых за годы работы появилось большое количество продуктов. Попытка впихнуть всех в общую иерархию создаст только больше проблем, и здесь вам необходимо быть достаточно гибкими и уметь адаптироваться к сложной и постоянно меняющейся среде.

Если вы запускаете проект модернизации и обновления устаревших продуктов, гибридный подход — хороший способ управления в краткосрочной перспективе. Команды в этом случае будут подстраиваться под ситуацию — то, что хорошо работает сегодня, может не иметь никакого смысла через год. Адаптируйтесь и не забывайте присматривать за структурой. Часто бывает достаточно раз в год проверять, оптимальна ли она для текущих задач бизнеса.

### 6.3. НАЗВАНИЯ ДОЛЖНОСТЕЙ

Одна из самых важных вещей, которую вам предстоит сделать, — организовать команду в соответствии с выбранной структурой таким образом, чтобы люди в ней могли развиваться, расти и понимали, чего от них ожидают. Частью этой задачи является выбор названий должностей.

Не стоит недооценивать важность названия. Люди могут говорить, что оно для них не имеет значения, но в действительности это не так, даже если им сложно себе в этом признаться. Должность — это знак признания, достижение и показатель карьерного роста человека. Это также способ сообщить внешнему миру, включая членов семьи, о том, с кем они имеют дело.

*Он не просто визор, он «супер» визор.*

— «Мальчики миссис Браун»  
(*Mrs. Brown's Boys*), BBC TV

К сожалению, не существует универсального стандарта или шкалы, позволяющей точно сравнить навыки сотрудников разных компаний. Такие названия, как «старший специалист», могут подразумевать в разных компаниях самые разные навыки и характеристики. Старший специалист в одной компании по уровню навыков может соответствовать младшему или обычному сотруднику в другой. Позже в этой главе мы составим матрицу уровней сотрудников и соответствующих им названий. А пока просто будем исходить из того, что названия нужны.

### 6.3.1. Что включать в название?

Любое название содержит в себе набор предполагаемых и подразумеваемых характеристик. Когда кто-то называет свою должность, у нас сразу формируются определенные представления — и они могут быть как верными, так и неверными. Например, когда вас знакомят с человеком, занимающим должность генерального директора, вы делаете вывод, что это начальник, находящийся на верхней позиции в иерархии. Хорошее, правильно подобранное название будет содержать следующую информацию:

- обязанности;
- опыт;
- заработная плата;
- уровень в иерархии;
- также оно должно быть понятным внутри вашей отрасли.



Название должности вызывает тот или иной уровень доверия у того, кто его слышит. Вспомните случаи, когда вы слышали название должности — и вам сразу становилось спокойнее. Например, как вам будет комфортнее в самолете: если пилот представится как «капитан» или как «младший пилот»? Находясь в самолете, который катится по взлетно-посадочной полосе и готовится взлететь, мы не хотим слышать ни о каких «младших», потому что это подразумевает отсутствие опыта, отсутствие способности руководить и отсутствие ответственности за свои действия.

#### ЧУВСТВО БЕЗОПАСНОСТИ

Недавно мне пришлось сделать операцию на шее, и, когда я узнал, что мой хирург был «главным хирургом» больницы, я понял, что нахожусь в надежных руках. Возможно, это чувство безопасности было иррациональным, но название должности успокоило меня (поскольку это была моя первая операция, я хватался за любую соломинку).

Хотя названия должностей в вашей команде, скорее всего, не так жизненно важны, они могут иметь большое значение для внешних клиентов, взаимодействующих с вашими сотрудниками. Например, клиент может быть недоволен

тем, что имеет дело с «младшим», потому что считает, что заслуживает большего, учитывая, сколько он вам платит.

Компания может не дать вам большой гибкости в выборе названий, если они соответствуют определенным уровням оплаты труда. Но даже если вы думаете, что у вас связаны руки, остается достаточно места для маневра, и я подробно расскажу об этом позже в этой главе.

### 6.3.2. Ступени к совершенству

В крупных, стабильных компаниях уже может быть разработана должностная иерархия, где каждый уровень имеет определенное значение. Однако в небольших компаниях ее, возможно, еще нет. Тем не менее, даже если она есть, вы, как руководитель технического отдела, должны будете определить, чего вы ожидаете от каждого из уровней. В табл. 6.1 приведена распространенная и считающаяся стандартной в технической сфере иерархия.

**Таблица 6.1.** Должностная иерархия: технологии

Уровень	Опыт	Примечания
Младший специалист (Junior)	0–2 года	Обычно это начальный уровень; недавний выпускник, стартовая позиция
Специалист (Associate)	2+ года	Имеет опыт работы в отрасли, но еще совершенствуется в профессии
Старший специалист (Senior)	5+ лет	Обладает глубокими профессиональными знаниями, эксперт
Ведущий специалист/руководитель (Lead/Manager)	5+ лет	Готов взять на себя ответственность за организацию работы других сотрудников и за достижение ими результатов
Архитектор (Architect)	10+ лет	Обладает глубокими знаниями в целом ряде дисциплин в своей области и отвечает за масштабные задачи
Руководитель департамента/вице-президент (Divisional Manager/Vice President)	10+ лет	Отвечает за несколько команд, работающих в определенной области; обычно участвует в бюджетировании
Fellow <sup>1</sup>	Стаж работы более 15 лет	Назначается за особые заслуги, например выдающиеся профессиональные или научные достижения, в знак признания многолетнего труда

<sup>1</sup> Fellow — высшая ступень инженерного пути развития карьеры, например заслуженный разработчик и т. д. По степени влияния соответствует высшему руководству организации. — *Примеч. пер.*



Каждый из этих уровней в идеале должен указываться для определенной специальности, например «старший разработчик» или «руководитель службы поддержки». Он позволяет выделить отдельных сотрудников в пределах каждой специальности. Но всегда бывают исключения из правил, и ваши названия будут отражать реалии именно вашей компании. Также в некоторых организациях существуют определенные традиции: например, в IBM каждый, оказывается, «вице-президент»! Другие компании могут добавлять «директор» к названиям должностей всех руководителей старшего звена (что сбивает с толку), потому что в Великобритании используется название «управляющий директор» (Managing Director) вместо «генеральный» (Chief Executive Officer) (понятие «директор» в Великобритании юридически закреплено, там директора указываются в учредительных документах компании как ответственные лица). И не стоит за названиями должностей забывать про реальный опыт — пять лет работы младшим специалистом сами по себе не сделают никого экспертом.

### 6.3.3. Частые ошибки

Некоторые компании нажили себе серьезные проблемы из-за того, что должности в них не соответствовали людям, их занимающим. Подавляющее большинство решений, которые мы принимаем как руководители, можно так или иначе поменять. Однако изменение уже имеющегося названия должности может показаться сотруднику понижением, потребует участия отдела кадров и в целом может иметь демотивирующий эффект. Ниже приведены некоторые распространенные ошибки при назначении должностей:

- *Слишком быстрое продвижение* — повышать сотрудника в должности необходимо постепенно и не слишком часто; никогда не повышайте кого-то лишь за отдельное достижение.
- *Должность вместо повышения зарплаты* — Когда вам нужно кого-то поощрить, но бюджет на это не предусмотрен, велик соблазн перевести его на другую должность. Но это даст сотруднику повод начать искать работу на аналогичной позиции с соответствующей ей зарплатой. Вы не сможете продолжать платить старшему специалисту столько же, сколько платили младшему.
- *Должность в награду за многолетний труд*. Тот факт, что сотрудник проработал в компании 20 лет, не делает его автоматически старшим специалистом, он должен еще и соответствовать требованиям к этой должности. Эта ошибка похожа на ошибку постоянного повышения зарплаты кому-то из сотрудников только за то, что он оказывается в нужное время в нужном месте. Обе они ведут к серьезным проблемам и часто достаются новым техническим директорам в наследство от их предшественников.
- *Нестандартные названия*. Как бы заманчиво это ни было, используйте названия, принятые в отрасли. Так вам будет проще набирать сотрудников, ваша команда сможет почувствовать себя частью большого сообщества, а вашим клиентам будет проще понять, с кем они имеют дело.

**ЗАМЕТКИ С ПОЛЕЙ***Директор по знаниям (Chief Knowledge Officer)*

Около 20 лет назад я занимал эту высокую должность в хорошо финансируемом стартапе из Кремниевой долины. В этом стартапе было много специалистов, стоявших у истоков мира Java, и мы выбрали для своих должностей крутые и модные названия... только вот они ничего не говорили внешнему миру. На самом деле я был скорее евангелистом, что говорило другим гораздо больше, чем очень умное (по нашему мнению) название должности. В конце концов мы поняли, что эффект от этих титулов противоположен задуманному — нас избегали, потому что потенциальные клиенты не могли понять, к кому с каким вопросом можно обратиться.

## 6.4. ВЗАИМОДЕЙСТВИЕ С КОМАНДОЙ

После того как вы сформировали свои команды и назначили каждому определенную роль, необходимо организовать работу в этих командах. Поскольку микроменеджмент — это не про вас, и ваши команды будут действовать самостоятельно, необходим способ координировать их деятельность, отслеживать прогресс и узнавать о любых проблемах, с которыми они сталкиваются. Для этого нужно наладить эффективное взаимодействие с руководителями команд, чтобы вы могли в любой момент понимать общее состояние дел в отделе (независимо от того, хорошим оно будет или плохим).

**ВДОХНОВЛЯЙТЕ***Будьте примером для других*

Если вам достались уже сформированные команды — их руководителям нередко не хватает управленческих навыков, или они вообще не изучали менеджмент. Они приобретали опыт в процессе работы и, вероятно, учились на единственном доступном им примере: стиле работы своего непосредственного начальника. Теперь этим начальником будете вы. Ваши действия, как я надеюсь, станут образцом для других. Имейте это в виду, особенно в периоды сложностей или напряженной работы, потому что все, что вы делаете, будет запоминаться и воспроизводиться в командах. Я всегда задаю себе вопрос, был бы я доволен, если бы моим руководителем был я сам? Для протокола: в моей карьере бывали моменты, когда на этот вопрос я отвечал «нет».

### 6.4.1. Определение степени самостоятельности

Хотя вы стремитесь, чтобы ваши команды работали независимо, в какие-то моменты вам придется подключиться к оценке принимаемых ими решений — так же, как генеральный директор оценивает ваши решения. Генеральный директор должен оговорить с вами вопросы, в которых вы не должны принимать решения

без его одобрения (или даже одобрения совета директоров), и вам необходимо установить такие же рамки для ваших команд. У разных команд могут быть немного разные условия, зависящие от их роли в отделе и опыта руководителя, но все должны придерживаться определенного набора общих правил.

Можно выделить три группы вопросов: те, которые команда должна уметь решать самостоятельно; те, о которых вас необходимо ставить в известность; и, наконец, те, которые не должны рассматриваться без вашего участия и по которым окончательное решение принимаете вы. Несколько примеров приведено в табл. 6.2.

**Таблица 6.2.** Уровни вовлечения СТО

Участие не требуется	Информирование	Требуется решение
Собеседование с потенциальными кандидатами	Любые проблемы или вопросы, затрагивающие клиентов	Финальное собеседование с лучшими кандидатами
Внутренние процессы, относящиеся к командной работе	Любые проблемы на продакшене	Увольнение сотрудников и консультации по вопросам управления персоналом
Личные встречи руководителя команды с его подчиненными; отпуска и отгулы	Инциденты или вопросы, связанные с безопасностью	Покупка лицензий и т. п.
Работа над проектом	Изменения порядка взаимодействия с другими командами	Замена технологий или инструментов
	Любые проблемы с членами команды	
	Сроки; изменения в них	

Важно донести эти правила до руководителей команд и найти сбалансированный уровень детализации.

### ЗАМЕТКИ С ПОЛЕЙ

#### *Информация о клиентах*

Я всегда требовал, чтобы меня немедленно ставили в известность о проблемах, связанных с претензиями от клиентов, а также с нарушениями или угрозами безопасности, независимо от времени суток. Более того, я сообщил всей команде свой номер мобильного, чтобы мне позвонили в случае, если я не подтвердил, что прочитал их письмо или сообщение. Я не хочу, чтобы генеральный директор спросил меня о чем-то, о чем я не имею ни малейшего понятия. Всегда лучше сообщить самому руководству о проблеме до того, как ему станет известно о ней от других.

**«НОВОСТЬ ОБ ОТСУТСТВИИ НОВОСТЕЙ»**

Много лет назад Келли Пауэлл (генеральный директор и писатель), занимавшая в то время должность руководителя отдела по взаимодействию с клиентами в Royall & Co, рассказала мне о том, как важно сообщать «новости об отсутствии новостей» в случаях, если от вас ждут информации. Если вы когда-нибудь бывали в ситуации, когда ждешь от кого-то новостей, а их долгое время нет, то наверняка у вас было чувство, что о вас забыли. Такое происходит с каждым, и все испытывают беспомощность и раздражение от того, что ничего не происходит.

Пауэлл требовала сообщать ей о положении дел, даже если результата еще нет и работа продолжается. Она просила направлять ей уведомления по электронной почте или в текстовом сообщении каждые 30–60 минут (в зависимости от серьезности проблемы).

**ЗАМЕТКИ С ПОЛЕЙ**

Я тоже придерживаюсь этого правила и настаиваю, чтобы его соблюдали мои подчиненные. Также я с пониманием отношусь к тем, кто ожидает, что я буду держать их в курсе. Это может выглядеть слишком простым, но я прошу сообщать мне новости в начале и в середине каждого часа. Так проще следить за временем и не нужно запоминать, когда было отправлено последнее сообщение. Не нужно тратить время на подробное описание — достаточно пары слов, что работа идет.

**6.4.2. Метрики**

Хотя это во многом зависит от того, какие конкретно у вас команды, вы, скорее всего, определите те или иные метрики, в соответствии с которыми руководители команд будут измерять эффективность работы и отчитываться перед вами. Эти показатели должны иметь практический смысл, а не просто служить для создания бесполезных графиков. Хорошая метрика — та, которая позволяет оценивать общий прогресс и которую можно включить в отчет для ваших руководителей.

Цифры позволят избежать эмоций и неточности при измерении результатов команды. Однако будьте осторожны: не стоит поощрять соревнование команд между собой, потому что вы, скорее всего, будете сравнивать яблоки с апельсинами — что неправильно. Предусмотрите для каждой команды свой набор метрик и сравнивайте их текущие результаты только с их же результатами в прошлом (или с планом). Для большинства команд подойдет хотя бы один из широко используемых показателей, перечисленных ниже:

- количество открытых/закрытых заявок;
- количество обращений в службу поддержки;
- метрики производительности/нагрузки;
- устранение технического долга;
- процент выполнения задачи/спринта.

Эти метрики должны быть относительно легко измеримы, определение их значений не должно превращаться в целый проект. Скорее всего, вы сможете получать полезные данные с помощью основного инструмента вашей команды (например, JIRA, SonarQube или Jenkins).

Определившись с метриками, решите, с какой частотой вы хотели бы получать их значения. Это может быть один или два раза в неделю. Если выбрать более длительный интервал, то за это время может слишком многое произойти и вы не успеете среагировать и предотвратить проблемы до того, как они возникнут.

### 6.4.3. Ответственность за результат

В хорошо смазанной машине все детали работают слаженно, совместно создавая нужный результат. То же относится и к командам: они должны работать эффективно и предсказуемо, чтобы весь ваш отдел в целом двигался к цели. Для этого руководители команд должны отчитываться о результатах, нести ответственность за неудачи и отмечать достижения вместе с командой.

Согласование метрик и целей устранил эмоции из обсуждений и позволит сделать акцент на данных, измеряющих успех команды. Команда, у которой нет целей, никогда не узнает, соответствует она вашим ожиданиям или нет (а возможно, превосходит их). Это несправедливо по отношению к команде и приводит к управлению на основании субъективных эмоциональных решений.

Выбор подходящих вам информативных метрик может занять несколько недель, особенно если у вас нет в этом опыта. После того как метрики будут выбраны, измерены и проверены, вы можете начать использовать их для аргументированного обсуждения того, как повысить производительность или какие еще показатели можно измерить.

#### ЭТО ОЧЕНЬ ПЛОХО

*С Земли на Луну («From the Earth to the Moon»)*

В этом популярном сериале HBO, повествующем о том, как Соединенные Штаты осуществили экспедицию на Луну, один из эпизодов представляет собой отличную аллгорию управления командой инженеров: «Паук» («Spider»). Все работают круглые сутки, чтобы уложиться в жесткие сроки, и никак не могут понять, почему лунный модуль продолжает разбиваться, пока один из инженеров смущенно не признает, что он допустил ошибку округления. Он идет к начальнику, уверенный, что его уволят или намылят ему шею.

Вот что сказал ему ведущий инженер компании Grumman Aerospace: «Вы молчали? Вы пытались скрыть это? Так нельзя. Это плохо. Это очень плохо. Идите домой, выспитесь, и давайте все исправлять». А теперь представьте, что он бы кричал и ругался. Осмелился бы кто-то в следующий раз признаться в ошибке? Так мы могли бы никогда не добраться до Луны.

Поощряйте открытость и честность на всех уровнях. Это не означает поиск виноватых, если что-то пошло не так. Ваш отдел — это команда, состоящая из команд, и для успешной работы этой команды вам необходим надежный канал связи, основанный на доверии и уважении, чтобы, если случится самое худшее, люди не боялись прийти к вам с плохими новостями.

Прозрачность не означает, что все ошибки и проблемы тут же прощаются и жизнь продолжается. Она означает необходимость брать на себя ответственность за ошибки, докапываться до причины проблем и внедрять методы, процессы и, если нужно, — инструменты, предотвращающие их повторение. Если ничего не меняется или повторяется снова и снова, то прозрачность означает отстранение сотрудников, отвечавших за выполнение задачи, и назначение людей, которые в состоянии изменить ситуацию.

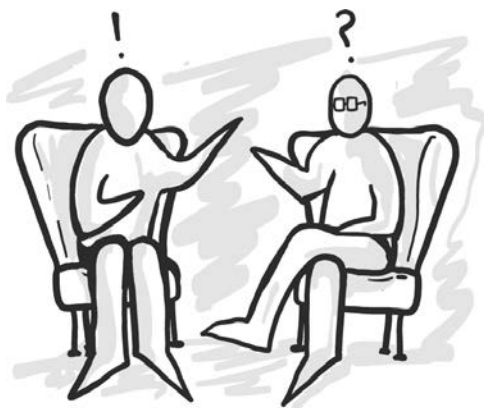
Это ваша обязанность как начальника нескольких команд. Недостаточно указать пальцем и переложить ответственность на некомпетентного менеджера. Это ваш сотрудник — вы назначили его на эту должность, и ваша задача — наставлять его и руководить им, чтобы он мог достичь успеха.

Его ошибка — это и ваша ошибка, которая бросает тень на все, чем вы занимаетесь. Генеральный директор будет наблюдать за вами и контролировать вас, и, если он решит, что вы больше не в состоянии управлять своей командой, он заменит вас, особенно если ваша компания принадлежит фонду прямых инвестиций. У инвесторов нет времени терпеть ошибки, зато есть длинный список опытных, обученных, авторитетных руководителей, которые заменят вас моментально — и глазом моргнуть не успеете.

Ответственность команды за результат означает, что она должна соответствовать метрикам и выполнять обязанности, которые она приняла, взяв на себя свою роль. Она настраивает ваши команды на успех. А их успех — это ваш успех.

## 6.5. ИНДИВИДУАЛЬНЫЕ ВСТРЕЧИ

Среди всех встреч, которые появляются в вашем календаре, есть такие, которые никогда не следует откладывать или отменять, — это классические индивидуальные встречи (one-on-one meeting). Это, как следует из названия, встречи один на один с руководителями ваших команд, которые проводятся регулярно, обычно раз в 1–2 недели. Оптимальная продолжительность этих встреч — 15–30 минут, и они нужны, чтобы убедиться, что у конкретного человека все благополучно.



Эти встречи должны быть строго конфиденциальными, некоторые даже считают их мини-сеансами терапии. На них ваши подчиненные могут обратиться к вам за помощью в решении любой проблемы. Например, попросить подсказать, как им более уверенно общаться с сотрудниками, потому что они слишком волнуются, когда обращаются ко всей команде.

Это также способ для них сообщить о любых неудобствах, которые они могут испытывать при работе с вами. Например, вы можете переходить границы, не отдавая себе в этом отчета, и они должны иметь возможность сказать вам об этом. Скорее всего, другие чувствуют то же самое.

И наоборот, этот формат встречи не следует использовать для обсуждения проектов, сроков или проблем. Для этого есть другие средства. См. табл. 6.3.

**Таблица 6.3.** Формат индивидуальных встреч

Темы, которые можно обсудить	Темы, которые не стоит обсуждать
Личные цели	Статус проекта
Зоны роста	Новые задачи
Вопросы и сомнения относительно работы	Воспитательные беседы
Анализ недавних событий	Сплетни о других сотрудниках
	Новости компании

От вас не ждут ответов на все вопросы. Вы должны выслушать, проявить участие, если это уместно — дать совет. Совещания такого типа бесценны для руководителей групп, которые оказались в новой для себя роли и могут совершать ошибки новичка — несколько предостережений помогут им изменить свой стиль управления.

Это возможность быстро внести коррективы, прежде чем проблема станет настолько значительной, что вам придется думать, правильно ли вы сделали, назначив этого человека менеджером.

#### ЗАМЕТКИ С ПОЛЕЙ

Я обнаружил, что эти встречи — отличный способ побудить людей задумываться о своих результатах и о том, как они хотят, чтобы их воспринимали. Например, если я присутствовал при чем-то, что, по моему мнению, можно было сделать лучше, я начну с вопроса собеседнику, устраивает ли его, как все прошло, — смог ли он донести свою мысль? Считает ли он, что все точно поняли, чего он хотел? Затем, вместо того чтобы говорить, как на его месте поступили бы вы, предложите варианты действий и обсудите, можно ли с их помощью получить другой результат.

Используйте индивидуальные встречи как способ укрепить доверие, предложить наставничество и поделиться вашими знаниями, общаясь на равных и без поучений.

## 6.6. ОБУЧЕНИЕ И ТРЕНИНГИ

Нам повезло работать в одной из самых сложных и динамично развивающихся отраслей, в которой каждые несколько лет появляется что-то принципиально новое. Идти в ногу с актуальными тенденциями, технологиями и процессами — не вопрос желания, а необходимость. Как руководитель, вы должны обеспечить постоянное обучение вашей команды всему новому, чтобы ее члены не только увеличивали производительность и поддерживали вашу платформу в актуальном состоянии, но и повышали свою самооценку и самоуважение.

К счастью, подавляющее большинство людей, работающих в нашей отрасли, понимают, что им нужно продолжать учиться и стремиться к новому. Даже если человек не готов изучать другую специальность, ему придется следить за последними версиями инструментов, которые он использует, изменениями в языке и т. д.

В былые времена считалось, что обучение — это скучное сидение в классе, и на обучение отправляли всю команду на несколько дней, что стоило довольно дорого. Приходилось выпрашивать бюджет на обучение у руководства, которое не видело в этом выгоды. К счастью, эти времена прошли, и одним щелчком мыши можно получить доступ к множеству онлайн-ресурсов, в том числе материалам лучших университетов.

Однако ваша роль заключается не только в том, чтобы обеспечить команду учебными материалами, но и в том, чтобы предоставить им достаточно времени на их изучение. Мы рассмотрим несколько возможных вариантов, как это сделать.

Какой бы путь вы ни выбрали, не прекращайте обучение. Обучение и удержание существующих сотрудников обходится намного дешевле, чем их уход и последующее обучение и адаптация новых.

### 6.6.1. Онлайн-библиотеки/ресурсы

Существует множество платных и бесплатных онлайн-ресурсов, предлагающих разные форматы и пакеты. Многие технические издательства предлагают специальные подписки, дающие неограниченный доступ к их каталогу книг. Специализированные сайты (например, Khan Academy и Coursera) предлагают более формальные онлайн-курсы с упражнениями и тестами, и все это бесплатно. Другие сайты, такие как Udeму, предлагают обучение стоимостью несколько долларов за курс.

Важно провести обзор этих курсов совместно с менеджерами команд, чтобы убедиться, что они актуальны и окупятся. Этот вид обучения идеально подойдет тем, кто предпочитает учиться самостоятельно и в своем темпе.

### 6.6.2. Сертификация специалистов по платформе/ПО

Крупные игроки, такие как Amazon, Microsoft, Cisco и Oracle, предлагают возможность сертификации специалистов для многих своих продуктов. За последние



годы цены на нее значительно снизились, и многие сертификаты стоят всего несколько сотен долларов (по сравнению с тысячами несколько лет назад).

Наличие сертификата подтверждает, что человек прошел обучение и проверку знаний по той или иной теме. Это полезно не только для самого сотрудника, который может добавить информацию об этом в свое резюме, но и для команды, в которой появляются обученные и сертифицированные сотрудники. Почти все программы сертификации предлагают онлайн-курсы с тестами и небольшими упражнениями, а после их прохождения предусмотрен финальный экзамен.

### 6.6.3. Буткемп

Буткемп-курсы, особенно в сфере программирования, стали очень популярными в последние годы, потому что эта сфера привлекает все больше людей. Они могут принимать самые разные формы: от интенсивных сессий с полным погружением длительностью до недели до вечерних или дневных сессий, проводимых в течение нескольких недель. Они организованы в виде очных занятий в классе, поэтому вы общаетесь с преподавателем и другими студентами, перенимая опыт.

Стоимость этих курсов довольно доступна; она рассчитана таким образом, чтобы люди могли сами оплачивать эту возможность развития своей карьеры, поэтому буткемп для сотрудников не должен обойтись компании дорого.

### 6.6.4. Сообщества пользователей

Менее распространенной формой обучения является участие в локальных сообществах пользователей. Обычно они посвящены обсуждению конкретной технологии или языка и организуются раз в месяц (в форме вечерней встречи с пиццей и пивом), при этом приглашенный докладчик рассказывает о новой функции или процессе или подробно разбирает что-то, что уже используется, чтобы другие могли лучше понять и эффективнее использовать это.

Сообщество пользователей может стать отличным инструментом нетворкинга, и вы всегда должны поощрять посещение таких встреч. Всегда полезно знать, кто из ваших соседей работает с теми же технологиями, чтобы, когда возникает проблема или требуется совет, вы могли связаться и обсудить решение.



Большинство сообществ не финансируются и существуют благодаря поддержке местных компаний, которые предоставляют место для выступлений, а также еду и напитки. Если у вас есть возможность пригласить докладчика выступить в вашем офисе, воспользуйтесь ею — это отличный способ не только рассказать другим о вашей компании и о том, чем вы занимаетесь, но и познакомиться с единомышленниками в своем районе.

### 6.6.5. Внутреннее обучение

Учитывая знания и опыт членов вашей собственной команды, почему бы не организовать обмен? Предложите формат, в котором эксперты в той или иной области будут выступать с небольшими докладами по своей теме.

Если вы предпочитаете что-то более неформальное, устройте ланч с едой на-вынос, в ходе которого будет обсуждаться какая-либо тема — идея в том, чтобы каждый принес свой обед и узнавал что-то новое во время еды, хотя, конечно, можно и угостить всех за счет компании.

### 6.6.6. Конференции

Конференции были очень популярны до пандемии COVID-19, посмотрим, вернутся ли они когда-нибудь к прежним масштабам (таким, как AWS re:Invent, когда Лас-Вегас переживал недельное нашествие 30 000 человек). Тем не менее многие конференции успешно перешли в онлайн и смогли сохранить при этом свою ценность, хотя и утратили фактор нетворкинга.

Конференции включают большое количество сессий разного уровня сложности, это дает участникам возможность познакомиться сразу со всем. Отправка на конференцию 1–2 человек может обойтись довольно дорого (особенно если она предполагает оплату проезда и проживания), и вы не сможете предложить это всем сотрудникам (кто-то должен остаться и присматривать за хозяйством). Хороший вариант — предложить эту возможность в награду лучшим сотрудникам, чтобы по возвращении они сделали один-два обзорных доклада для остальных. Можно отправить на конференцию людей с определенной задачей — узнать о конкретной технологии или продукте, а затем поделиться полученными знаниями с другими.

### 6.6.7. Курсы при университетах/колледжах

Некоторые люди хотят сделать значительный скачок в своей карьере, получив ученую степень по выбранному предмету. Это потребует много времени и денег — иногда многих тысяч долларов и несколько лет.

В зависимости от получаемой степени вы можете оплатить сотруднику все обучение или его часть и разрешить посещать занятия в рабочее время. Если вы видите в этом ценность для организации, подпишите соглашение, по которому компания окажет сотруднику помощь, а он после окончания учебы обязуется

отработать в вашей команде определенное количество лет. В случае увольнения он будет обязан компенсировать затраты компании на свое обучение. При этом будьте осмотрительны, чтобы не вызвать недовольство сотрудников, которые не получают подобных преимуществ. Пусть они понимают и разделяют ваше желание поддержать кого-то.

## 6.7. КОММУНИКАЦИЯ

Спросите любого довольного своей работой сотрудника, что конкретно ему нравится, и рано или поздно он произнесет слово *прозрачность*: чувство уверенности от понимания, что происходит вокруг него, ощущение причастности. Прозрачность не означает, что необходимо сразу же всем обо всём рассказывать. От других руководителей высшего звена вы будете получать конфиденциальную информацию, которую, возможно, пока не стоит обнародовать или необходимо дополнительно проработать, чтобы она не вызвала лишних волнений.

Как руководитель команды, вы выступаете посредником между ней и бизнесом, фильтруя и обогащая информацию, относящуюся к команде. Джастин Марквардт (Justin Marquardt), партнер частной инвестиционной компании New Harbour Capital, кратко сформулировал это так: «Крайне важно, чтобы они [СТО] являлись сильными коммуникаторами и эффективно объясняли своей команде, высшему руководству и совету директоров, как их план согласуется со стратегическими инициативами компании, в том числе были способны выйти за рамки технических подробностей и донести связь тех или иных инициатив с ключевыми целями бизнеса».

Прозрачность подразумевает уважительное и внимательное отношение, когда информация сообщается без спешки и своевременно. В этом разделе мы расскажем, как следует организовывать коммуникацию, а также какие встречи и с какой частотой проводить.

### 6.7.1. Электронная почта

Одним из самых популярных способов передачи информации по-прежнему остается электронная почта. Хотя чаты используются все чаще (Slack, Mattermost, Cliq, Teams и т. д.), они предназначены для общения в реальном времени, а не для передачи и сохранения сообщений, которые можно прочесть позже. Не используйте чаты для передачи важной информации, которую должны получить все. Оставайтесь с электронной почтой.

Создайте общий список рассылки, включающий всех членов группы (engineering@xxx.com). Так вы точно никого не упустите. Этот список необходимо обновлять при приеме на работу и увольнении. Вот в каких случаях принято делать общую рассылку:

- *Новый сотрудник в отделе.* Кратко представьте человека, присоединившегося к группе, — перечислите его достижения, сообщите, на какую должность и когда он выходит. Лучше делать эту рассылку заранее, чтобы люди знали о пополнении в команде и могли поприветствовать новичка.
- *Увольнения как по инициативе работодателя, так и по собственному желанию.* Сообщите группе, кто и когда уходит из компании. Предложите другим написать ему свои теплые пожелания. Даже в случае увольнения по вашей инициативе — пожелайте сотруднику всего наилучшего. Достаточно будет такой фразы, как «Мы пришли к взаимному решению завершить наше сотрудничество», не стоит вдаваться в подробности, которые могут быть неприятными. Если человек уходит на повышение, отметьте, что вы все способствовали его карьерному росту.
- *Объявления компании.* Иногда требуется пояснить, что конкретно для вашей группы означает то или иное объявление на уровне компании.
- *Объявления отдела.* Ключевые события, крупные успехи и признание заслуг отличившихся членов команды. А также сообщения о любых сбоях или ошибках (без указания виновных) и мерах, которые были приняты.
- *Тенденции в отрасли, которые могут оказать влияние на компанию.* Важно понимать, что это не просто обмен ссылками на все, что показалось интересным (для этого используйте общий канал в Slack). В письме необходимо рассказывать о серьезных изменениях, которые окажут краткосрочное или долгосрочное влияние на работу вашего отдела.

Не используйте список рассылки для неформальных тем. Получая такое письмо, сотрудники должны понимать, что это не спам, а важная информация, с которой надо ознакомиться.

## СТИЛЬ ПИСЬМА

Честно признаемся, что мы все писали письма или тексты, об отправке которых потом жалели. Либо мы выбрали неверный тон, либо шутки оказались неуместными, либо мы кого-то задели (намеренно или нет). Подобным письмам не должно быть места в вашем наследии для потомков.

Вот несколько советов, как написать хорошее профессиональное электронное письмо для широкой аудитории:

- Пишите естественно; избегайте излишней формальности.
- Не используйте ненормативную лексику, намеки или оскорбления.
- Никогда не переходите на личности.
- Будьте честны: никаких домыслов и преувеличений.
- Оставайтесь профессионалом. Говорите по существу.

Помните, что электронные письма никогда не удаляются. Все чаще компании хранят копии каждого отправленного и полученного письма в отдельном хранилище, данные в котором невозможно изменить. Это может быть обусловлено нормативными требованиями, или же компания может быть особенно

внимательна к юридическим вопросам. Такие службы, как G Suite и Office 365, упрощают настройку подобного хранилища. Поэтому, даже если вы думаете, что удалили электронное письмо — как только оно покидает папку «Исходящие», оно сохраняется навсегда. А поскольку это письмо является собственностью компании, оно может быть использовано как в вашу пользу, так и против вас.

### **НАЙДИТЕ СВОЙ СТИЛЬ**

Лучший совет, который я получил от моего ныне покойного наставника Джереми Джилана (Jeremy Geelan) (журналиста, продюсера BBC и издателя), заключался в том, чтобы писать так, будто текст будет опубликован и запомнится в веках. Прежде чем нажать «Отправить», перечитайте сообщение и подумайте, обрадуетесь ли вы, если оно будет опубликовано? Это простой, но эффективный тест, и каждый раз, когда я отправляю электронное письмо, я мысленно задаю себе этот вопрос, и много раз бывало так, что я редактировал написанное.

## **6.7.2. Совещания**

Совещания — важная часть рабочего процесса. Даже если вы используете видеосвязь, личное общение чрезвычайно важно, и не только с руководителями команд, но и со всеми сотрудниками вашего отдела. В этом разделе мы рассмотрим некоторые распространенные типы совещаний, которые вы будете проводить.

### **СОВЕЩАНИЕ ПО ТЕКУЩИМ ВОПРОСАМ**

Это собрание, на котором вы рассказываете группе, как идут дела. Оно должно проходить раз в 1–2 недели и занимать не более 30 минут. Основные темы: крупные релизы, рост или падение количества пользователей, улучшения процессов и структурные изменения. Формат должен быть постоянным, чтобы люди знали, чего ожидать.

Если вы уложитесь в 30 минут, команда не будет уставать и будет чувствовать полезность каждого такого совещания.

### **ОБЩЕЕ СОБРАНИЕ**

Такие встречи посвящены конкретной теме и проводятся по необходимости. Обычно вначале кто-то один выступает по теме собрания, после чего следует открытое обсуждение в формате «вопрос — ответ».

Это совещание особенно полезно, если вы предлагаете новую инициативу, вносите изменение в продукты или же требуется пояснить какую-либо новость компании. Кроме того, такую встречу можно организовать, чтобы пригласить руководителей другого отдела рассказать об их работе или о том, как повлияет то или иное принятое ими решение на ваш отдел.

### **СОВЕЩАНИЕ С РУКОВОДИТЕЛЯМИ КОМАНД**

Регулярные встречи с руководителями команд служат не только для того, чтобы вы получили информацию, но и для того, чтобы все руководители команд понимали, как продвигается работа у их коллег. Эти совещания следует проводить раз в неделю и посвящать их обсуждению общих вопросов, затрагивающих все команды, не отвлекаясь на конкретные детали. Зачастую обсуждение все же переходит к подробностям — в этом случае отложите вопрос и предложите заинтересованным сторонам либо остаться после встречи, либо запланировать еще одну встречу с необходимыми участниками.

Оставайтесь в рамках 60 минут и выделите руководителям каждой команды одинаковое время для рассказа о прогрессе их команды. Составьте повестку дня, в которой четко укажите порядок выступлений и время на каждое, а в конце встречи оставьте время для других вопросов.

### **ОПЕРАТИВНОЕ СОВЕЩАНИЕ**

Очень многие устанавливают продолжительность таких встреч в 60 минут; это ошибка, которая постепенно высосет все ваше время по мере роста вашего отдела и компании. Назначайте встречи длиной в 15 или 30 минут и просите участников составить повестку дня из одного или двух пунктов, чтобы все знали, о чем пойдет речь.

Совещания — один из главных отвлекающих факторов в корпоративном мире, поэтому с уважением относитесь ко времени, которое люди выделяют для встречи с вами (как виртуальной, так и живую).

### **ПРАВИЛА ЭФФЕКТИВНЫХ СОВЕЩАНИЙ**

Чтобы совещание было эффективным, недостаточно просто собрать несколько человек в одном месте и разрешить им высказываться. Это деловая встреча, а не светское мероприятие. Правила помогут сосредоточиться на поставленной задаче и намного быстрее получить желаемый результат. Вот несколько организационных советов:

- Заранее разошлите участникам повестку дня, включая все нужные материалы.
- Приглашайте только тех, кто будет активно вовлечен в обсуждение; не добавляйте участников только для того, чтобы они были в курсе вопроса.
- Не отвлекайтесь; уберите со стола мобильные телефоны.
- Не заказывайте еду — встреча не должна быть такой долгой, чтобы понадобились закуски.
- Используйте ноутбуки только для презентаций; не отвлекайтесь на проверку электронной почты или мессенджеров.

Совещание должно быть сфокусированным, продуктивным и комфортным. Если это онлайн-встреча, то ее можно записать, но это не всегда нужно. Я обычно записываю встречи, на которых участники демонстрируют что-то, что может быть

полезно для более широкой аудитории. Ведите краткий протокол совещания, достаточно зафиксировать в нем только достигнутые договоренности.

#### **ЧТО ЕЩЕ ПОЧИТАТЬ**

*Meetings Suck, Cameron Herold*

Я очень рекомендую книгу Герольда о том, как повысить продуктивность совещаний. Он разбирает механизмы проведения совещаний, их частоту и составление такой повестки, которой люди действительно будут придерживаться. Я взял у автора интервью в подкасте MacLaurin Group Demystifying Technology, в нем мы подробно обсудили, как сделать встречи более эффективными.

ISBN-10: 1619614146

<https://www.amazon.com/Meetings-Suck-Elements-Business-Valuable/dp/1619614146/>

### **6.7.3. Обратная связь**

Важно, чтобы любой член группы мог задать вам любые вопросы, не обязательно соблюдая иерархию. Манеру общения вы выбираете сами исходя из своих предпочтений, но избегайте чрезмерной открытости в стиле «моя дверь всегда открыта» — люди этим воспользуются.

Вместо этого пусть сотрудники обращаются к вам, и, если чувствуете, что кому-то нужно больше времени, попросите его запланировать 15 минут в календаре и уделите ему внимание, которого он заслуживает. Худшее, что вы можете делать, — проигнорировать человека или выслушать вполоуха, параллельно что-то набирая на компьютере. При этом вас не должны постоянно отвлекать. Поэтому важно четко сформулировать правила и придерживаться их.

#### **УПРАВЛЯЙТЕ ОЖИДАНИЯМИ**

Я никогда не против пройти и пообщаться с человеком или же, если тема важная, найти время в моем календаре. Но я не вступаю в спонтанные обсуждения на месте. Если краткого ответа, состоящего из одного предложения, недостаточно, то я прошу человека забронировать время в календаре. Таким образом, я точно уделю собеседнику достаточно времени и не буду тем, кто сбежал посреди разговора или параллельно переписывался с другими.

Следите, что вы по-настоящему доступны и открыты для общения. Избегайте красивых фраз, которые ничего не значат. За словами должны следовать действия. И люди поймут, если вы объясните, что сейчас вы заняты и что обсуждение этого серьезного вопроса лучше отложить, тогда вы сможете сосредоточиться и уделить ему все свое внимание. Так сотрудник будет чувствовать, что он важен, а не что от него хотят поскорее отделаться.

## ИТОГИ

- Устав с точным описанием задач каждой команды поможет всем понять эти задачи и учитывать их в работе.
- Команды можно сформировать на основе специальностей, продуктов или процессов. Выбор того или иного способа должен основываться на задачах и потребностях команды.
- Вы создаете не просто команду, но также пути роста и развития карьеры каждого из людей. Уделите этому отдельное внимание, так вы покажете, что цените развитие своей команды.
- Названия должностей — это не просто красивые фразы. Это опознавательные знаки, имеющие значение в том числе и за пределами организации, и другие будут подсознательно оценивать человека по тому, как звучит его должность.
- Чтобы гарантировать, что ваши сотрудники не подведут вас, разработайте процесс предоставления отчетности руководителями команд, включая уровень детализации и частоту сдачи отчетов.
- Индивидуальные (one-on-one) встречи — это возможность совершенствовать социальные навыки сотрудников.
- Постоянное обучение важно не только для того, чтобы поддерживать актуальность навыков команды, но и для того, чтобы сотрудники получали знания в новых областях, которые могут быть полезны в будущем.
- Руководителю важно выбрать свой стиль общения, но большая открытость и прозрачность способствуют установлению доверительных отношений с сотрудниками, и они будут больше доверять вашим решениям.

## ПРОВЕРЬТЕ СЕБЯ

Что из перечисленного ниже вы уже сделали?

- Разработали устав, в котором подробно описали цели каждой команды и показатели ее успеха.
- Четко определили структуру команд и каким бизнес-задачам эти команды соответствуют.
- Продумали названия должностей в команде с учетом того, какое значение они будут иметь внутри и вне компании.
- Изложили удобный для вас способ предоставления отчетов о ходе работы от руководителей команд.
- Запланировали регулярные индивидуальные встречи с руководителями команд.
- Сообщили своим сотрудникам, каким образом вы поддерживаете постоянное обучение и совершенствование навыков.
- При планировании встреч указываете четкие цели и результаты, чтобы люди понимали, что их время не будет потрачено впустую.



# Ежегодная оценка сотрудников

## В этой главе

- ✓ Система оценки сотрудников
- ✓ Создание методологии, основанной на данных, чтобы оценка проходила комфортно
- ✓ Если работник совершает ошибки
- ✓ Как уволить сотрудника
- ✓ Правильная организация удаленной работы

Каждый автомобиль нужно раз в год осматривать и обслуживать, чтобы все в нем было в порядке и чтобы выявлять проблемы до того, как они перерастут в трудноустраняемые поломки. То же самое и с командой: необходимо ежегодно проверять, что все всем довольны, и контролировать, в каком направлении хотят развиваться сотрудники, а также в каком направлении вам нужно, чтобы они развивались. При этом вы можете внести изменения, дать сотрудникам понять, в чем они преуспевают и что необходимо улучшить, а также внедрить что-то новое, что интересно вам или вашей команде.

Возможно, вы уже проводите ревью или что-то подобное. Большинство людей не ждет от этих ревью ничего хорошего, потому что они отнимают огромное количество времени, особенно если у вас большая команда, и зачастую проводятся суетливо и поверхностно, не принося никакой пользы, кроме галочки для отдела кадров. Часто ежегодные ревью проводятся перед тем, как в компании определяют выплачиваемые бонусы и премии. Поэтому, с одной стороны, к этому

моменту все запланированные задачи будут выполнены, а с другой — они будут выполнены все и одновременно!

В этом разделе мы рассмотрим, как создать систему ежегодной оценки, которая будет полезна как вам, так и членам вашей команды, чтобы сотрудники готовились к ревью без страха и не тратили на него слишком много времени. Я предложу вам матрицу компетенций, которая поможет основывать процесс ревью на данных, а не на ощущениях или эмоциях.

Если нужно будет уволить сотрудника (по любой причине) — это требует аккуратности, и вы научитесь увольнять так, чтобы никто не остался обиженным.

В некоторых компаниях оценку сотрудников проводят чаще, например раз в квартал или раз в полугодие. Но не следует проводить ее слишком часто, и если у вас в команде уже приняты регулярные индивидуальные встречи с менеджерами (или с вами), то достаточно делать оценку раз в год.

Наконец, мы рассмотрим удаленную работу и как этот формат влияет на проведение оценки.

## 7.1. МАТРИЦА КОМПЕТЕНЦИЙ

Вы, возможно, видели или заполняли классический опросник для ревью, который сформулирован примерно следующим образом: «Назовите три области, в которых сотрудник превосходит других, и три области, которые требуют внимания». Это очень популярная формулировка, но не слишком полезная для развития, потому что вопросы чересчур расплывчатые. Руководителю сложно (и долго) заполнять опросники в таком виде, особенно если у него много дел, а сотруднику трудно их правильно интерпретировать. Подобный формат также предполагает, что всегда должно быть что-то, что нужно улучшить. Так бывает не всегда, но менеджеры действуют по принципу «Я должен что-то придумать, иначе отдел кадров от меня не отвяжется».

Представим для примера члена команды, который желает повышения. Формат, описанный выше, даст ему очень мало информации о том, что нужно делать для этого. Оценка проводится в контексте его текущей роли, и очень мало внимания уделяется советам о том, что ему необходимо сделать, чтобы перейти на следующий этап развития карьеры. Сколько раз мы слышали: «Просто работай усерднее, и тебя повысят»? Это никак не помогает и оставляет неприятный осадок, потому что звучит так, как будто для продвижения нужно работать больше (то есть тратить на работу больше времени без дополнительной оплаты), а не более квалифицированно.

В свою очередь, матрица компетенций позволяет не только оценить, насколько хорошо сотрудники справляются со своими текущими обязанностями, но и четко описывает, что им нужно сделать, чтобы перейти на следующий уровень, — и вам не придется писать простыни текста или придумывать обратную связь ради галочки.

## СОБЕСЕДОВАНИЯ

### Использование матрицы

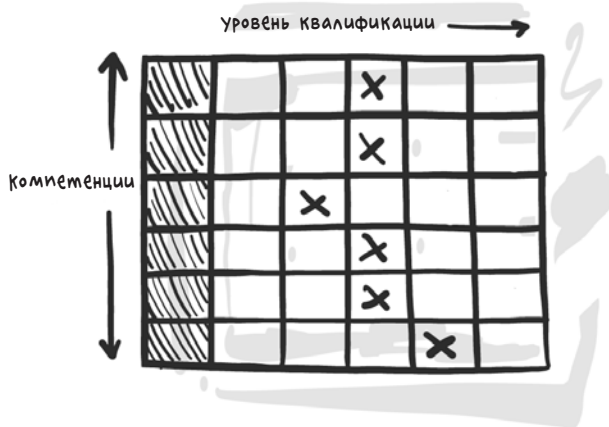
Как вы увидите, хорошо составленную матрицу компетенций можно использовать не только для ежегодной оценки персонала, но и на собеседованиях, как основу для оценки кандидатов. Подготовьте свои вопросы/задания по каждой области, чтобы во время интервью оценить, в каком диапазоне находятся навыки (компетенции) кандидатов.

### 7.1.1. Структура матрицы

Матрица представляет собой сетку, на одной оси которой находятся компетенции, а на другой — уровни квалификации сотрудника. Для каждой должности необходимо разработать отдельную матрицу с указанием ключевых качеств, которые необходимы для успешной работы.

Слева указываются компетенции, важные для рассматриваемой должности, независимо от опыта работы. Вот примерный список компетенций для разработчика/инженера:

- *Знание основной специальности* — технические знания, относящиеся к специализации сотрудника.
- *Понимание задач бизнеса* — знание не только продуктов компании, но также рынка и тенденций.
- *Коммуникация* — общие навыки коммуникации, включая общение по электронной почте / в чатах, навыки ведения переговоров и выступлений.
- *Планирование и организация работы* — управление задачами, их формулирование, планирование и выполнение.
- *Результаты работы и их поддержка* — качество работы / конечного результата и то, как сотрудник поддерживает эти результаты в дальнейшем.
- *Решение проблем и инновации* — способность решать проблемы и внедрять инновации там, где это необходимо.
- *Наставничество и лидерство* — взаимодействие и обмен знаниями с другими.



Вам нужно будет выбрать компетенции, подходящие для вашей организации, но они должны быть достаточно широкими, чтобы охватить как профессиональные навыки (то есть технические знания), так и социальные (межличностное взаимодействие). Например, если ваша команда регулярно взаимодействует с конечными пользователями — важно, какой при этом формируется имидж компании и ее сотрудников.

Тщательно определите компетенции, которые важны для вашей команды, но при этом соответствуют основным ценностям компании. Это облегчит следующий этап. Но не выбирайте и что-то слишком абстрактное, поскольку вам необходимо будет понятным образом объяснить вашим сотрудникам, что они значат.

Для каждой должности необходимо определить уровни квалификации. Они будут разными в зависимости от должности и в крупных компаниях могут быть привязаны к уровню оплаты труда. Для одних должностей может быть меньше уровней, чем для других. Это нормально, но необходимо определить их заранее, чтобы избежать недопонимания в будущем. Этот уровень вы будете определять для каждого человека — представьте, что он будет указан на его визитной карточке. Вот пример возможных уровней квалификации для разработчика:

1. Младший специалист (Junior) / Выпускник
2. Специалист (Associate)
3. Старший специалист (Senior)
4. Ведущий специалист (Lead)
5. Менеджер
6. Директор
7. Fellow

Так вы покажете путь карьерного роста и перечислите возможные в вашей компании уровни — как ответственности, так и оплаты труда.

Мы указали значения для оси X (компетенции) и оси Y (уровни квалификации). Матрица готова.

### 7.1.2. Заполнение матрицы

В ячейках укажем наши ожидания для каждого из уровней всех компетенций. Опишите в 1–2 предложениях критерии успешного достижения этого уровня. Хорошее описание позволит без затруднений обсуждать соответствие тому или иному уровню и приводить конкретные примеры сильных и слабых сторон сотрудника. Хотя вам может потребоваться составить более 30 описаний, все эти усилия будут окупаться в ходе каждого ревью или собеседования в будущем.

В табл. 7.1 приведен пример типичной матрицы для разработчика, включающей три возможных уровня квалификации.

**Таблица 7.1.** Базовая структура матрицы — разработчик

Компетенция	Младший разработчик	Разработчик	Старший разработчик
Знание основной специальности	Базовое понимание	Делится профессиональными знаниями с другими	Следит за отраслевыми тенденциями, знает передовые методы/паттерны
Понимание задач бизнеса	Имеет общее представление о том, чем занимается компания	Имеет общие знания о какой-то конкретной области бизнеса компании	Понимает, чем занимаются все основные подразделения бизнеса
Коммуникация	Может понятно рассказать о своей работе	Может понятно рассказать о работе своей команды, включая изложение технических подробностей	Может рассказывать о работе, включая подробности, слушателям, не являющимся техническими специалистами
Планирование и организация работы	Руководитель объясняет задачу во всех подробностях	Получает задачи от руководителя, самостоятельно определяет подробности	Получает задачу от руководителя в общем виде, самостоятельно определяет подробности, проводит для этого исследования и запрашивает дополнительную информацию
Результаты работы и их поддержка	Результаты проверяет старший разработчик	Результаты проверяет старший разработчик, но без контроля деталей	Результат гарантированно высокого качества; руководитель контролирует архитектуру
Решение проблем	Предлагает решения в рамках своей области	Предлагает широкий диапазон решений	Проактивно выявляет и решает проблемы
Наставничество/лидерство	Принимает инструкции и помощь от других	Обращается за помощью к старшим разработчикам; может быть наставником для младших	Является наставником в своей специализации для разработчиков и младших разработчиков

Со временем определения будут меняться, поскольку команда будет расти и взрослеть. Главное здесь — возможность четко и ясно объяснить сотруднику, почему он находится на том или ином уровне, чтобы он понимал, что это не субъективная оценка, а наглядно видел, как его достижения соотносятся с ожиданиями.

### 7.1.3. Порядок проведения ревью

Определившись со сроками проведения ежегодного ревью, предупредите о нем сотрудника минимум за неделю. Это даст ему время на подготовку. В рамках подготовки направьте ему копию матрицы компетенций для его специальности и попросите самостоятельно оценить свой текущий уровень. Предложите ему также привести примеры своей работы на более высоком уровне.

Вы, как его руководитель, сделайте то же самое. Если это возможно — сравните текущие результаты с прошлым годом, чтобы увидеть, как сотрудник развивается. Оценивайте общую картину, единичные успехи или неудачи не должны влиять на ваши выводы о возможности повышения/понижения сотрудника.

Для каждой из специальностей определите уровень квалификации сотрудника. Если он где-то отстает, запишите, на чем вы основываете такой вывод. Если кто-то близок к продвижению, но еще не выполняет все требования, сообщите ему об этом и подскажите, что ему необходимо сделать, а также назначьте дату повторного ревью, скажем, через шесть месяцев, чтобы сотрудник не ждал своего шанса еще год.

Когда вы будете проводить встречу с сотрудником, первым делом сравните самооценку сотрудника с вашей оценкой. Сопоставьте матрицы и отметьте различия — именно на них следует сосредоточить обсуждение. Если вы сфокусируетесь на конкретных областях и будете использовать в качестве аргументов конкретные требования — будет гораздо легче составить план развития, необходимый для достижения успеха в той или иной области. Вот возможные варианты результатов оценки:

- *Улучшения нет.* Возможно, человек работает на приемлемом уровне, и это устраивает и его, и вас. Не все хотят иметь больше ответственности, это зависит от их жизненных обстоятельств. Таким сотрудникам повышать заработную плату необходимо только на размер инфляции.
- *Чрезмерная самоуверенность.* Большинство людей склонны переоценивать свои способности и, скорее всего, будут оценивать себя на один или два уровня выше, чем они есть, основываясь на примерах ситуаций, когда они действительно добились отличных результатов. Но одна ласточка весны не делает, и здесь важно свести разговор к тому, какие результаты являются типичными. Иногда вы можете потерять того, кто твердо верит, что заслуживает большего. По моему опыту, вам никогда не удастся полностью удовлетворить такого человека, и лучше всего ему продолжить карьеру в другом месте.
- *Неудовлетворительный результат.* Если кто-то не справляется, причинами тому могут быть перенапряжение, переутомление или иногда просто потеря интереса к работе. Найдите эту причину и постарайтесь вместе решить проблему. Не понижайте сотрудника в должности сразу, вместо этого проведите через шесть месяцев повторное ревью, в ходе которого вы окончательно

поймете, требуется ли переводить человека на другую должность, которая позволит ему работать успешно и с удовольствием.

- *Результат выше ожиданий.* Для сотрудника, уровень компетенций которого превосходит требуемый для его текущей должности по всем направлениям, повышение определенно не за горами. Однако будьте осторожны — не перешагивайте слишком много уровней; мой совет — не пропускайте уровни вообще. Независимо от того, насколько хорошо справляется сотрудник, опыт, который он получил бы на пропущенном уровне, может быть необходим для его дальнейшей работы.

Решение о повышении сотрудника необходимо принимать на основании того, насколько важна каждая из компетенций для его должности. Например, человек может показывать отличные результаты только в шести из восьми компетенций, и этого может быть достаточно для повышения. Помните, что матрица компетенций — это вспомогательное средство, а не непрекращаемый свод правил.

В случае повышения сообщите всей команде об этом и о причинах повышения, и все вместе поздравьте сотрудника. Это укрепит вашу репутацию руководителя, принимающего решения на основе понятных критериев и данных, без личных предпочтений и субъективности.

## 7.2. ВАШ ПОДЧИНЕННЫЙ ОШИБАЕТСЯ

В мире технологий возникновение той или иной проблемы — это всего лишь вопрос времени. Аварии случаются в самом неожиданном месте и всегда не вовремя. Системы отказывают, ПО глючит, оборудование ломается, питание пропадает, стихийные бедствия только и ждут своего часа, люди нажимают не те кнопки или не нажимают нужные — поразительно, что при всем этом что-то вообще работает, и работает хорошо.

В этом разделе мы рассмотрим ошибки, вызванные человеческим фактором, и какие существуют способы, чтобы справиться с ними системным и управляемым способом. Большинство ошибок возникает из-за того, что кто-то не предусмотрел их возможность. Например, регулярно происходят перебои электроснабжения, и ответственный за это сотрудник должен принять меры для обеспечения резервного питания — а вы должны выделить на это бюджет. В любом случае, ответственность за нарушение бизнес-процессов несет человек.



### 7.2.1. Первая реакция

Если кто-то ошибся, никогда не обвиняйте этого человека, особенно публично. Неважно, что он сделал, в конце концов, его ошибка — это ваша ошибка. Известная фраза, получившая широкое распространение благодаря 33-му президенту США Гарри С. Трумэну: «Фишка дальше не идет», — больше чем просто слова. Это олицетворение менеджера, берущего на себя ответственность за неудачи своей организации. Многие менеджеры заявляют, что придерживаются этого стандарта, но мало кто доказывает это на деле.

Когда вы быстро обвиняете других и объявляете проблему следствием чьего-то недосмотра или некомпетентности — это говорит больше о вас как о руководителе, чем о ком-то другом. Итак, вот главные вещи, которых вы никогда не должны делать:

- Кричать, ругаться или угрожающе жестикулировать.
- Угрожать или оскорблять.
- Немедленно увольнять кого-либо.

Неважно, насколько вы нервничаете, злитесь или разочарованы — ничто не может оправдать подобное поведение.

Прежде всего необходимо установить первопричину проблемы или, если проблема еще продолжается, прекратить действие этой причины, чтобы вы могли продолжить ее исследование. Вы должны разработать план предотвращения подобных ситуаций, или же, если полностью предотвратить их невозможно, то план минимизации тяжести последствий в будущем.

### 7.2.2. Расследование

По мере расследования проблемы вы сможете отнести ее причину к одной из двух категорий:

- ошибка/некомпетентность;
- непродуманность / недостаточное планирование.

Чтобы установить первопричину проблемы, часто используют итеративный сужающий подход «пять почему», разработанный основателем концерна Toyota. Он не всегда работает, потому что иногда первопричин несколько, но благодаря ему вы сможете начать с общей картины и перейти к более узким вопросам.

Если это была человеческая ошибка, определите, является ли она следствием недостаточного обучения. Прошел ли тот, кто ее совершил, необходимое обучение? Если да, были ли это просто неправильные действия или же какой-то сложный пограничный случай, к которому обучение не может подготовить? Как конкретно произошла эта ошибка: был ли исполнитель перегружен задачами, или его что-то отвлекло — и в результате он сделал то, чего не должен был делать? Были ли пропущены какие-то действия?



Ответив на эти вопросы, вы получите более широкое представление о том, в каком месте произошел сбой и какие шаги можно предпринять, чтобы предотвратить его в будущем. Можете ли вы внедрить дополнительную защиту в виде проверок и резервирования? Или достаточно провести дополнительное обучение и проверку знаний, чтобы убедиться: все знают, что делать?

### ЗАМЕТКИ С ПОЛЕЙ

*Не то окно*

Я работал с клиентом, когда однажды утром началась суматоха, сработали все сигналы тревоги, предупреждая о серьезном сбое на продакшене. Попытки войти на серверы (это было время, когда к серверам подключались по SSH) ничего не дали — серверы как будто исчезли. И так и было! Системный администратор случайно «удалил» продакшен кластер веб-серверов. Он открыл слишком много окон в терминале и думал, что работает с тестовыми серверами и удаляет ресурсы, которые больше не используются. После этого случая были введены дополнительные меры безопасности, но самой эффективной из них стало выделение окон терминала с подключением к продакшен-серверам невыносимо ярким цветом, чтобы их сразу было видно.

Составьте описание инцидента и его причин. Задokumentируйте шаги по исправлению.

Если сотрудник вновь допустит такую ошибку, возможно, эта работа ему не подходит, особенно если другие в команде без проблем справляются с теми же задачами. Также изменения в команде могут понадобиться, если для какой-то позиции тот или иной навык является ключевым, а уровень владения этим навыком у текущего сотрудника недостаточный. Увольнение не следует рассматривать как наказание или возмездие, это просто признание факта, что человек не подходит для должности, на которую он был принят. Да, увольнять неприятно и некомфортно, но используйте этот инцидент в дальнейшем как часть процесса собеседования, чтобы оценить, как другие кандидаты справятся с подобной ситуацией.

Если же проблему можно было предотвратить более тщательным планированием — это другое дело. Невозможно предусмотреть любую случайность, но, несмотря на это, вы вправе ожидать от ответственных людей выполнения основных требований. Вот некоторые примеры:

- Взлом вашей системы, поскольку сотрудник, ответственный за безопасность, не настроил файрвол или не обновил его вовремя.
- Потеря данных из-за того, что сотрудник, ответственный за базы данных, как следует не проверил работоспособность резервных копий.
- Прекращение оказания услуги из-за того, что сотрудник пропустил электронное письмо о проблеме с оплатой.
- Отказ оборудования из-за того, что у сотрудника, ответственного за оборудование, не было плана его обновления или обеспечения безотказной работы.

Мы всегда хороши задним умом, и то, что кажется очевидным после разбора, не всегда понятно или кажется важным до аварии. Удивительно, как часто самые простые и базовые вещи могли бы предотвратить катастрофу, если бы были сделаны, но вместо них всегда находилось что-то более важное.

Поскольку вы тут главный, спросите себя, не была ли ошибка следствием того, что вы не согласовали решение или не ответили на вопрос, который несколько месяцев пролежал на вашем столе. Если безопасник не смог установить файервол из-за того, что вы не согласовали оплату, то его вины здесь нет.

В любом случае задокументируйте событие и поговорите с ответственным за него. Если проблема будет повторяться — возможно, дело в сотруднике, и ее можно решить, проведя дополнительное обучение или, в крайнем случае, заменив сотрудника на более подходящего.

### 7.2.3. Выводы

На любой ошибке есть чему поучиться. Технический директор, который никогда не сталкивался с аварией, на самом деле не руководил по-настоящему. Важно собрать данные и задокументировать проблему для дальнейшего комплексного анализа. Вот что стоит зафиксировать:

- *Проблема повторяется?* Это говорит о том, что процессы, которые вы внедряете для предотвращения проблем, не работают.
- *Проблема появляется лишь для отдельных клиентов?* В компаниях, сфокусированных на отдельных клиентах, проблема может быть связана с чем-то уникальным для конкретного клиента и не проявляться для других.
- *Проблема сезонная?* Если проблема возникает в одно и то же время, определите внешние факторы: например, не связан ли сбой с переходом на летнее время?
- *Проблема возникает у определенных сотрудников?* Если проблема продолжает возникать у одних и тех же людей, то, возможно, им не хватает знаний и требуется дополнительное обучение или замена этих людей.
- *Каков масштаб проблемы/сбоя?* Оценка серьезности проблемы позволит определить, сколько усилий необходимо приложить для ее решения; иногда проще оставить все как есть, особенно если проблемы кратковременные или редкие.

Документируя проблемы, возникающие в вашем отделе, вы будете лучше понимать, насколько хорошо он работает. Никто не может полностью избежать ошибок, но уроки, которые вы из них извлечете, помогут вам эффективнее реагировать на будущие проблемы.

## 7.3. УВОЛЬНЕНИЕ

Вероятно, любой, кто собирается стать руководителем, считает самой сложной и неприятной обязанностью, противоречащей инстинкту человека быть

дружелюбным, увольнение подчиненных. Многие в этом случае обращаются за помощью, надеясь, что существуют уловки или обходные пути, позволяющие сгладить связанный с этим дискомфорт. Увольнять людей приходится по ряду причин:

- Сокращение отдела.
- Какая-то из специальностей больше не нужна.
- Плохие результаты.

Мы рассмотрим каждый из этих сценариев и то, как можно подготовиться и организовать процесс, чтобы он прошел успешно как для вас, так и для сотрудника. Как мы уже не раз говорили, решения о принятии в команду новых сотрудников или увольнении никогда не должны приниматься легкомысленно или в спешке. Вы влияете в том числе и на то, будут ли у этих людей средства к существованию, и, принимая важные решения, влияющие на способность человека обеспечивать себя и свою семью, проявляйте соответствующее уважение.

#### **ЗАМЕТКИ С ПОЛЕЙ**

Я считаю, что вопрос увольнения не должен быть проблемой и нужно рассматривать его так же, как и любой другой вопрос управления командой — только на основе данных, исключив эмоции или личные предпочтения. Если же вы хотите уволить кого-то просто потому, что он вам не нравится, могу поспорить, что проблема в вас, а не в нем, и вы позволяете эмоциям влиять на профессиональные решения. В этом случае вам необходимо обратиться за советом в отдел кадров и к генеральному директору.

На всякий случай повторим: вы не должны принимать решение об увольнении в одиночку. Необходимо заручиться одобрением генерального/финансового директора и на каждом этапе привлекать отдел кадров, если он есть. Если вы не убедили их в том, что решение верное, возможно, вы действуете немного импульсивно и стоит еще раз все обдумать.

### **7.3.1. Сокращение отдела**

Не все получается так, как мы рассчитываем. Мы работаем в отрасли, которая постоянно меняется по мере того, как испытываются новые технологии и исследуются новые области. Иногда что-то срывается, а иногда компании приходится списать убытки и прекратить проект или расформировать команду. Это может произойти из-за того, что изменился рынок или конкурент вырвался вперед. Когда это случится, вы обнаружите, что у вас слишком много сотрудников, которых некуда перевести. У вас больше нет для них работы, хоть и не по их вине.

Прежде всего убедитесь, что вы исчерпали все возможности и не можете перенаправить эти ресурсы на другой проект. Как мы знаем из предыдущих глав, подбор и обучение обходятся дорого, поэтому лучше лишний раз не проходить

эти процессы. Изучите все варианты, в том числе переобучение или перевод на вакансии в другом отделе компании.

Если вы были открытыми и честными, то для сотрудников сокращение не будет сюрпризом. Они будут знать, что такой риск существовал и что это было непростое решение для вас. Совместно с генеральным директором и отделом кадров вы разработаете график закрытия проекта, и после этого финансирование будет прекращено. Соберите команду и максимально подробно расскажите о том, чего вы смогли достичь и почему проект больше нежизнеспособен.

Скорее всего, сотрудникам будет предложен компенсационный пакет в зависимости от их стажа и требований местного законодательства. Это зона ответственности финансового директора или отдела кадров, и данный вопрос выходит за рамки книги.

Поговорите с каждым сотрудником, поблагодарите их за работу и ценный вклад и заверьте, что в сложившейся ситуации нет их вины. По возможности постарайтесь помочь людям найти новую работу — предоставьте рекомендации, дайте советы по трудоустройству, позвоните знакомым из других компаний.

Зачастую другая компания готова принять нескольких (или всех) сотрудников, которых вы сокращаете. В этом случае она получает сработавшуюся команду, обладающую знаниями в той или иной предметной области. Если вы знаете, что команда представляет определенную ценность, поспрашивайте свою сеть контактов, возможно, кто-то захочет пообщаться с вашими сотрудниками.

### **БУМ ДОТКОМОВ**

*Коммодитизация (commoditization) программного обеспечения*

К сожалению, в 1990-х закрытие отделов, а иногда и целых компаний было обычным явлением. На заре развития интернета существовало множество конкурирующих технологий для решения одинаковых задач при отсутствии стандартов. С внедрением стандартизации некоторые компоненты вошли в состав фреймворков с открытым исходным кодом, что привело к исчезновению компаний и групп, бизнес которых был сосредоточен вокруг этих компонентов. В качестве примера можно привести веб-серверы, серверы обмена сообщениями и служебные библиотеки. Это привело к оживлению на рынке труда — компании разом увольняли большие команды инженеров, и их поглощал следующий дотком, считающий, что уж он-то точно знает, что делать.

### **7.3.2. Навыки не требуются**

Еще одна частая причина увольнений — отсутствие потребностей в навыках сотрудников. Такое часто случается после модернизации устаревших систем и перехода на новую платформу или язык, когда прежние навыки оказываются неактуальны.

Этот сценарий схож с предыдущим, но в данном случае увольнение не станет ни для кого сюрпризом, поскольку вся команда уже давно работала над модернизацией. В ее ходе крайне важно регулярно общаться с людьми, чьи навыки больше не понадобятся после успешного завершения модернизации. Дайте им возможность переобучиться по ходу работы над проектом, чтобы они могли переключиться на новую технологию.

Не все хотят переучиваться; некоторые полностью удовлетворены тем, чем они занимаются. К сожалению, вам придется отпустить таких людей в свободное плавание. Этот разговор должен пройти относительно легко и не стать сюрпризом для сотрудника. Еще раз поблагодарите его за работу и окажите ему максимальную помощь в поиске новой должности.

Очень важно обеспечить людям, знающим, что проект подходит к завершению, возможность подыскать запасной аэродром. Однако будьте осторожны, не допускайте ситуации, когда все сотрудники найдут новую работу еще до закрытия вашего проекта. Если вы считаете, что такой риск есть, предложите сотрудникам бонус, чтобы они продолжили работать до завершения проекта, таким образом вы не останетесь без ресурсов. Относитесь к сотрудникам так, как вы хотите, чтобы относились к вам.

### 7.3.3. Плохие результаты

Бывает так, что человек просто не обладает навыками, необходимыми для выполнения работы. Это проявляется одним из следующих способов:

- смерть от тысячи порезов;
- инцидент.

#### **СМЕРТЬ ОТ ТЫСЯЧИ ПОРЕЗОВ**

Эта хорошо известная в мире боевых искусств фраза означает, что смерть наступила не от одной крупной раны, а от множества мелких повреждений. В нашей менее жестокой и воинственной среде это означает, что человек не способен показывать нужный результат сразу для нескольких требований.

Поначалу это может быть трудно заметить, особенно если это новый сотрудник, который хорошо прошел собеседование, но, после того как он проработал некоторое время, становится понятно, что он преувеличил или переоценил свои способности.

Если вы или один из ваших менеджеров начинаете подозревать, что сотрудник не соответствует требованиям, определите области, в которых он показывает неудовлетворительный результат, и подробно опишите, что должно измениться, чтобы требования выполнялись. Привлеките отдел кадров или генерального/финансового директора (в зависимости от размера компании). Сообщите им, что вы собираетесь провести оценку такого-то сотрудника, и если они согласны с вами, значит, вы на правильном пути.

Обсудите ситуацию с человеком, приводя конкретные примеры — это очень важно. В конце концов, вы действительно не хотите его терять и заново начинать процесс поиска и адаптации. Составьте список того, что ему нужно улучшить и к какому сроку. Он не должен включать ловушки в виде заведомо нереальных целей. Покажите, что вы желаете сотруднику успеха, и предложите ему любую необходимую помощь.

Если сотрудник все равно не справляется, то в согласованную дату проведите с ним встречу и расскажите, что в таких-то направлениях улучшения отсутствуют, поэтому места в вашей команде для него больше нет. Это не должно стать для него неожиданностью или выглядеть как ваша эмоциональная реакция. После того как все было зафиксировано и проговорено, он точно знает и ваши требования, и то, где он им не соответствует.

### ПОНИЖЕНИЕ В ДОЛЖНОСТИ

Стоит отметить, что существует альтернатива увольнению, которая может успешно работать в разных ситуациях, — понижение в должности. Если сотрудник находится на позиции старшего разработчика, но вы понимаете, что на самом деле он соответствует уровню младшего, скорректируйте требования до уровня, на котором он сможет их выполнять. При этом также может измениться и его зарплата, но для обеих сторон так будет лучше. Я не раз наблюдал ситуации, когда человеку давали шанс возглавить команду инженеров, но он понимал, что на самом деле ему это не нравится и он хочет просто писать код.

### ИНЦИДЕНТ

Последствия катастрофического сбоя, произошедшего в результате халатности или некомпетентности, могут быть огромными как для вас, так и для компании. Халатность — это когда ситуацию можно было предотвратить или смягчить, но этого не было сделано. Обычно виной тому является пренебрежение сотрудником его основными обязанностями. Вот несколько примеров:

- Ответственный за безопасность не смог обеспечить базовую защиту системы, что было его основной обязанностью.
- Сотрудник ИТ-службы ухитрился потерять критически важные данные, поскольку решил, что жесткие диски абсолютно надежны и резервные копии не нужны.
- Специалист службы поддержки проигнорировал проблему ключевого клиента, в результате чего клиент ушел к другому поставщику.

Здесь необходимо хорошо подумать, прежде чем действовать. Никогда никого не увольняйте только из-за разочарования и гнева или для того, чтобы отомстить. Или же просто потому, что вы должны как-то среагировать. Проведите

тщательное расследование и узнайте как можно больше о том, что привело к проблеме. Это сбой в системе или же вина человека, который не сделал то, что должен был? Можно ли было этого избежать? Находится ли на ответственной позиции правильный человек?

Если вы пришли к выводу, что у человека нет необходимых навыков, сообщите об этом генеральному/финансовому директору и отделу кадров и, если они согласны, планируйте увольнение.

В эту категорию входит также (что относится ко всем сотрудникам компании) и неприемлемое поведение. Грубость или оскорбления, употребление наркотиков или алкоголя, воровство, обман, злоупотребления, связанные с чьим-то полом или расой, — в большинстве организаций за подобное сразу увольняют. В таких случаях немедленно подключайте отдел кадров.

### 7.3.4. Процедура увольнения

При увольнении сотрудника не забудьте о следующих вещах:

- *Отработка.* Будете ли вы требовать от сотрудника отработать еще какое-то время? Иногда может понадобиться передача знаний. Если же это не нужно, то в большинстве случаев присутствие сотрудника на рабочем месте до момента увольнения не рекомендуется.
- *Выбор времени.* Извечный вопрос: «Когда сообщить об увольнении?» Лучше сделать это с самого утра и ни в коем случае не в последний рабочий день недели. Всегда лучше дать человеку возможность что-то предпринять для поиска работы уже на следующий рабочий день, а не заставлять его ждать понедельника.
- *Коммуникация.* Четко определите причины, по которым вы увольняете человека, потому что их необходимо будет озвучить. Также подготовьте сообщение для остальной команды. В большинстве случаев достаточно просто указать на несоответствие навыков, не вдаваясь в подробности, и пожелать человеку всего наилучшего.
- *Организация встречи.* Сообщайте об увольнении только в присутствии свидетеля. Обычно это будет представитель отдела кадров, но также может быть и финансовый директор или даже генеральный директор. Если кого-то увольняет один из ваших подчиненных, с ним в кабинете должны находиться вы и представитель отдела кадров, который может помочь с организационными вопросами. Свидетель нужен, чтобы подтвердить, что все необходимые шаги были сделаны и сотруднику известны и понятны причины увольнения. Говорите только то, что необходимо и достаточно для надлежащего информирования сотрудника.
- *Закрытие доступов.* Наконец, выполните заранее подготовленную процедуру исключения сотрудника из рабочих процессов и закрытия всех доступов.

### 7.3.5. Обратная связь от увольняемого сотрудника

Некоторые люди положительно относятся к опросу при увольнении. Это последняя возможность для сотрудника оставить обратную связь перед уходом из компании. Другие считают, что это всего лишь возможность выговориться и ничего конструктивного в этом нет. Хотя некоторые компании и проводят эту процедуру, многие из них игнорируют подавляющее большинство сообщений, используя ее лишь как возможность для бывшего сотрудника пожаловаться и выпустить пар.

Если вы хороший руководитель, из этого опроса вы не узнаете ничего нового. Все это вы и так должны знать из встреч с командой и индивидуальных встреч, если вы проводите их регулярно и правильно. Вам необходимо создать среду, в которой люди будут чувствовать, что могут высказать свои взгляды и мнения, не боясь последствий. Вы должны узнавать о назревающих проблемах, прежде чем они выйдут на поверхность и станут явными.

## 7.4. УДАЛЕННАЯ РАБОТА

Удаленная работа, или работа из дома, позволяет совмещать работу и личную жизнь, выполняя рабочие обязанности вне офиса. Условия для удаленной работы возникли именно в сфере технологий, и когда-то такой формат работы считался привилегией. По мере роста пропускной способности интернета стало возможным более тесное взаимодействие с удаленными командами с помощью простой в использовании видеосвязи. Настоящий бум удаленной работы случился в 2020 году, когда разразилась глобальная пандемия COVID-19, вынудившая сотрудников во всем мире покинуть свои офисы и отправиться на домашний карантин. На протяжении большей части 2020 года в отрасли происходил массовый переход от модели, при которой из дома время от времени работали только некоторые сотрудники, к тому, что все начали использовать чаты и видеоконференции для командной работы.

Не все получают удовольствие от удаленной работы. Кто-то предпочитает физическое разделение офиса, в котором человек выполняет свои профессиональные обязанности, и дома, куда он возвращается, чтобы отключиться от работы и уделить внимание семье. Кто-то, наоборот, предпочитает гибкость, особенно сотрудники с молодыми семьями, которые хотят видеть, как растут их дети, и при этом активно заниматься карьерой. Политика в отношении удаленной работы обычно принимается на уровне компании, но технические команды часто больше подходят для нее или, по крайней мере, для гибридного формата, чем другие.

Управление удаленными сотрудниками имеет свои плюсы и минусы. В этом разделе мы рассмотрим, как продолжать успешно обеспечивать потребности бизнеса, если вы решите отправить на удаленку всю команду или будете использовать гибридную схему, когда часть сотрудников находится на удаленке, а часть — в офисе.



### 7.4.1. Техническое обеспечение

Для успешной удаленной работы у ваших сотрудников должны быть все необходимые ресурсы, в том числе стабильный интернет, необходимая мебель и, конечно же, достаточно мощный компьютер. Компании нередко оказывают содействие в обустройстве удаленного рабочего места — покупке мебели и оплате высокоскоростного интернета.

Как минимум компания должна приобрести для сотрудников две вещи, которые мгновенно окупятся, — хорошую веб-камеру и гарнитуру, а также научить людей эффективно ими пользоваться. Решающее значение имеет выбор платформы для видеоконференций, предусматривающей возможности шеринга экрана, гостевого доступа и записи трансляции, особенно если вы можете связать ее с другими вашими системами, чтобы организовать единый вход. Отлично подойдут такие решения, как Google Meet, Microsoft Teams, Zoom и Ring, — они надежны и имеют интуитивно понятный и простой в использовании интерфейс. Ваша цель здесь — создать такие условия, чтобы все члены команды могли выйти на связь, просто кликнув по ссылке.

В офисе также необходимо создать конференц-зал с необходимым оборудованием, чтобы те, кто работает в офисе, имели возможность быстро присоединиться к видеовстрече. К счастью, это недорого.

Еще одним важным компонентом для удаленной работы является общий чат. Этот способ виртуально собрать всех в одном месте быстро стал использоваться как основной канал связи вместо электронной почты. Поощряйте использование сотрудниками функции «статус», чтобы показать, что они отошли от рабочего места. Не думайте, что если люди не в офисе, то дома они прикованы к своим столам — в офисе же никто не сидит за столом, не отходя, весь день. Пусть технологии помогают людям быть на связи, где бы они ни находились. Slack, Mattermost, Cliq и Teams — отличные примеры мощных чат-систем, интегрированных с другими продуктами.

Если вы способны организовать (виртуальное) собрание так же быстро, как вы можете собрать всех вокруг доски в офисе, то у вас есть фундамент для успешной удаленной работы.



### 7.4.2. Управление ожиданиями

Управлять удаленными сотрудниками необходимо точно так же, как и теми, кто физически находится с вами в офисе; вы должны доверять команде выполнение работы в соответствии с указаниями, и если она этого не делает, то это равнозначно тому, что сотрудники приходят в офис каждый день, но не выполняют поставленную задачу.

Однако такой формат работы требует от вас и от менеджеров в вашем подчинении особенно тщательно ставить задачи и регулярно контролировать команду, работающую удаленно. Это не значит, что вы должны требовать отчетов каждые 15 минут. Однако, поскольку вы не видите людей и можете не заметить, что они испытывают трудности, вам теперь нужно спрашивать об этом. В зависимости от задач может быть достаточно связываться 1-2 раза в день в чате.

Проводите видеозвонки и поощряйте тех, кто нервничает или не привык находиться в кадре, использовать их регулярно. Чем чаще вы их проводите, тем больше людей начнут их постоянно использовать. Видеозвонок — это не публичное выступление, его не нужно репетировать, и от вас не ожидают безупречности. Если сотруднику нужна помощь, не используйте чат или электронную почту, а позвоните ему лично по видеосвязи. Так вы сможете видеть друг друга и получить больше информации, что ускорит решение проблемы.

При управлении удаленной командой зафиксируйте ваши требования. Например, установите время, в течение которого все члены команды должны быть доступны онлайн, как если бы они были в офисе. Выберите способ сообщать об отсутствии на рабочем месте, когда вы отходите на перерыв, обед или по другим причинам. Никто не должен воспринимать вашу удаленную команду как колл-центр, который в любое время доступен для любых запросов.

Удаленная работа не обязательно означает гибкий график (когда вы сами выбираете часы работы). Он может быть частично гибким — все зависит от того, насколько тесное взаимодействие в режиме реального времени требуется от команды. Скорее всего, удаленные сотрудники будут работать по обычному графику. Гибкий график должен обсуждаться и согласовываться отдельно.

### 7.4.3. Предотвращение злоупотреблений

Удаленная работа создает дополнительную нагрузку как на руководителей, так и на рядовых сотрудников. Это не означает большую продолжительность рабочего дня, однако приходится прикладывать усилия, чтобы соответствовать вышеуказанным ожиданиям. Вам, как руководителю, нужно дополнительно убедиться, что вы не имеете подсознательных предубеждений в отношении своих сотрудников, работающих удаленно, и не предъявляете к ним завышенные требования.

С 2011 по 2020 год правительство Великобритании проводило обширное исследование влияния удаленной работы на людей (<http://mng.bz/PoQg>). Некоторые из его результатов довольно поучительны:

- Удаленные сотрудники на 50% реже получали повышение.
- Удаленные сотрудники чаще работали до позднего вечера.
- Удаленные сотрудники реже брали больничный, чем офисные работники: 0,9% против 2,2%.
- Бонусы для удаленных сотрудников были на 38% ниже, чем для тех, кто работает в офисе.

Это красноречиво подтверждает поговорку «С глаз долой — из сердца вон». Сотрудников, которые не находятся на виду, чаще обходят при обсуждении повышений и бонусов, но, как ни странно, от них ожидают, что они будут работать больше, чем их коллеги в офисе (и они действительно работают больше).

Это неравенство, скорее всего, нивелируется после COVID-19, учитывая, какое огромное количество компаний приняло удаленный формат работы в качестве основного, но важно отслеживать его по мере роста организации. Контролируйте, что к удаленным сотрудникам относятся так же, как и к людям в офисе.

## ИДЕЯ

### *Виртуальные мероприятия в команде*

Одна из реальных проблем удаленных работников — ощущение того, что они исключены из социальной жизни команды. Эту проблему помогут решить виртуальные мероприятия, организуемые по видеосвязи и на которых нельзя обсуждать рабочие вопросы. Прекрасный вариант — «счастливые часы», когда люди могут взять в руки бокал и просто болтать о чем угодно. Используя такие инструменты, как Jackbox.tv и YouTube, мы также успешно проводили онлайн-игры и викторины. Уделяйте таким встречам полчаса каждые несколько недель в конце рабочей недели — и вы получите эффективный инструмент тимбилдинга.

## ИТОГИ

- Ежегодная оценка персонала не должна отнимать много времени и проводиться ради галочки для отдела кадров. Хорошо подготовленные и продуманные ревью могут стать важным и нужным событием для всех.
- Рано или поздно что-нибудь пойдет не так, и то, как вы поведете себя, особенно перед вышестоящим начальством и другими отделами компании, покажет, какой вы руководитель.
- Увольняя людей, убедитесь, что вы принимаете не эмоциональное, а обдуманное и основанное на данных решение. Это только укрепит ваш авторитет.
- Управление удаленной командой мало чем отличается от управления командой, работающей в офисе, но учтите, что необходимо приложить дополнительные усилия, чтобы воспользоваться всеми преимуществами этого нового, развивающегося формата командной работы.

## ПРОВЕРЬТЕ СЕБЯ

Что из перечисленного ниже вы уже сделали?

- Четко понимаете особенности каждого уровня и каждой специальности в организации.
- Составили матрицу, с помощью которой можно оценить компетенции членов вашей команды на основе данных.
- Разработали механизм проведения ежегодной оценки персонала, который помогает сотрудникам понять их возможности для роста.
- Разработали механизм для анализа ошибок и информирования о них, а также для извлечения уроков из ошибок и аварий.
- Определили порядок увольнения сотрудников на основе точных данных и аргументов.
- Выстроили политику эффективной удаленной работы, включающую четкие требования к сотрудникам.

# 8

## Технологические решения

---

### В этой главе

- ✓ Важность долгосрочных стратегических решений, например: разработать самим или купить?
- ✓ Выбор между облачными сервисами или собственным оборудованием и вопросы отказоустойчивости
- ✓ Различные типы хранилищ данных
- ✓ ПО с открытым исходным кодом и выбор подходящего языка/фреймворка

Одна из ключевых обязанностей технического директора — выбор технологий, используемых в компании. Хотя нам нравится думать, что мы работаем в быстро развивающейся отрасли, которая постоянно меняется и развивается, реальность такова, что отдельные системы могут работать в продакшене десятилетиями без значительных изменений. Просто подумайте: то, что вы создаете сейчас, может еще продолжать использоваться, когда вы соберетесь на пенсию. Как такое может быть? Вот несколько причин, по которым система может работать так долго:

- Стек технологий таков, что его невозможно обновить постепенно и по частям, только заменить целиком.
- Техническое руководство годами не желает (или не умеет) прикасаться к священной корове.
- У руководства есть опыт работы только с этими технологиями.
- Система настолько важна, что ее замена или модернизация становится слишком рискованной или дорогостоящей.

- Высокомерное убеждение, что система была построена идеально, а совершенство улучшить невозможно.

Скорее всего, когда-нибудь вы займете другую должность, поэтому управлять тем, что вы проектируете и создаете сегодня, будет кто-то другой. Учитывая это, как минимум полезно для вашей же собственной репутации оставить в наследство систему, которую ваш преемник сможет положить в основу своей работы и за которую вам скажут спасибо. Не делайте так, чтобы после вашего ухода пришлось ломать все, что вы создали, и начинать заново — что иногда бывает при смене руководства.

### ЗАМЕТКИ С ПОЛЕЙ

#### *Системы 40-летней давности*

Я участвовал во многих проектах модернизации после приобретения, когда новые владельцы желали потратить деньги на приведение платформы к современному стеку. Устаревшие системы страдают от высоких эксплуатационных расходов, отсутствия безопасности и ограниченной масштабируемости. Самой старой системе из тех, что видел лично я, было 46 лет (она запускала RPG от IBM через терминалы с зеленым экраном), хотя чаще встречаются системы возрастом от 30 до 40 лет, работающие под управлением Cobol. Я также часто вижу 30-летние системы, созданные с использованием языков четвертого поколения с графическим интерфейсом, таких как Visual Basic, Gupta и Delphi. Все их объединяет одно: они — ключ к успеху компании. Если бы не они, бизнес не выжил бы.

При выборе технологии следует учитывать следующие факторы:

- Какие перспективы у нее в будущем?
- Что произойдет, если технология устареет или больше не будет поддерживаться?
- Какие зависимости нужны для ее работы? Что будет происходить с ними со временем?
- Будет ли в дальнейшем кто-то использовать эту технологию?

Например — старые приложения на Visual Basic, созданные в середине 1990-х годов с использованием сторонних библиотек, которые давно перестали поддерживаться и работают только на Windows 95/XP, и поэтому их нельзя портировать на современные версии Windows. «Майкрософт» отказалась от попыток обеспечить обратную совместимость и вместо этого предоставляет для Windows 10 полноценную виртуальную машину Windows XP, чтобы дать хоть какое-то будущее этим устаревшим приложениям.

Существуют примеры того, как устаревание технологий вынуждает компании предпринимать корректирующие действия. Когда-то в моде были плагины для веб-браузеров. Вы могли использовать Adobe/Macromedia Flash, Microsoft Silverlight или Java-апплеты Sun, чтобы добавить на веб-страницу богатый

«десктопный» функционал. С появлением HTML 5 и значительным усовершенствованием встроенного JavaScript стало очевидно, что эти плагины устарели (и имеют проблемы с безопасностью), и в течение нескольких лет они полностью исчезли из новых версий браузеров.

Мораль этой истории в том, что необходимо учитывать не только внутренний программный стек вашей компании, но и все сторонние системы, на которые вы полагаетесь. Чрезвычайно важно регулярно обновлять программное обеспечение, о чем мы подробно поговорим в главе 13, посвященной поддержке и обслуживанию.

## 8.1. ИЗБЕГАЙТЕ ПОПАСТЬ В ЗАВИСИМОСТЬ

Неважно, выбираете ли вы поставщика, язык или инфраструктуру, всегда избегайте решений, внедрив которые вы становитесь зависимыми от них и не можете в случае необходимости что-то изменить. Сохраняйте возможность для внесения изменений без полного переписывания проекта. Конечно, проще сказать, чем сделать, и многие технические директора / архитекторы, действовавшие из лучших побуждений, на этом погорели. Проблема в том, что выбрать путь, который предоставит больше возможностей в будущем, бывает труднее, а иногда и дороже. Думайте об этом как о страховке: если что-то случится, у вас будет страховой полис, выплата по которому уменьшит ваши страдания.

### ЗАМЕТКИ С ПОЛЕЙ

#### *Flash*

Недавно у меня был клиент, чье ПО для администрирования SAN (сети хранения данных) не обновлялось годами, и, когда дело дошло до внесения небольшого изменения, ничего не вышло: ПО было создано с использованием уже не существующей технологии Macromedia Flash. Пришлось разыскивать старый ноутбук, давно не подключающийся к интернету, чтобы на нем не было обновлений. Как только мы подключились, SAN обновилась и консоль администратора перешла с Flash на HTML5.

Java — хороший пример базового языка, не ограничивающего ваш выбор. Этот язык был создан для работы на множестве процессоров, применяемых для создания встраиваемых систем. Использование абстрагированной виртуальной машины позволило разработчикам писать на едином стандартном языке, а JVM (виртуальная машина Java) стала связующим звеном между оборудованием / операционной системой и вашим кодом. Java предлагает ряд высокопроизводительных JVM (Oracle, IBM, OpenJDK, Amazon), которые можно использовать, не меняя ни единой строчки в коде на Java. Независимо от того, где вы запускаете Java-приложение — в Linux, Windows, Amazon, Google или Microsoft, — код остается тем же. Вы сохраняете контроль над своей операционной системой,

виртуальной машиной и всем остальным, и можете заменить что угодно, не отвлекая команду разработчиков.

Если вы выбрали решение, которое требует расходов на лицензию, вам необходимо подготовиться к возможному увеличению ее стоимости — насколько сильно это повлияет на ваш бюджет? Организации часто жалеют, что купили лицензии Oracle / SQL Server только чтобы увидеть, что с годами цены растут при меньшей гибкости (например, невозможно переехать в облако), и в то же время понимая, что никаких уникальных функций этого продукта они не используют.

Не выбирайте платформы или ПО, для которых нет альтернативы. Иначе вы окажетесь во власти цен и набора функций конкретного поставщика; это касается в том числе бэк-офисных инструментов, таких как Office 365, Google Workspace и Salesforce. Однако в некоторых случаях это может быть приемлемым компромиссом, особенно при использовании крупных поставщиков, которые обычно не меняют цены резко.

Обеспечьте себе несколько «аутов», как если бы вы играли в покер. Хороший игрок в покер никогда не будет гнаться за одной картой, надеясь собрать руку из ничего. По ходу игры он должен быть уверен, что у него есть несколько вариантов. Планирование работы компании основано на том же принципе. Вы не знаете, что ждет вас завтра — бизнес может решить развернуться и двинуться в другом направлении, или же перемены могут заставить всю отрасль бороться за выживание. Ответственность за надлежащее реагирование лежит на техническом директоре. И бесконечно правильнее заниматься организацией и финансированием небольших постепенных обновлений, чем полномасштабной гигантской переделкой.

## 8.2. РАЗРАБАТЫВАТЬ ИЛИ ПОКУПАТЬ

Подобно участникам классического спора, что лучше — табуляция или пробел, многие технические специалисты мучаются дилеммой «создать или купить». Это решение не только оказывает долгосрочное влияние на технический департамент, но и становится предметом споров на уровне совета директоров. Начинается перетягивание каната между менеджерами и технарями, при этом менеджмент считает, что покупка позволит быстро получить результат, а технические специалисты страдают синдромом «изобретено не нами» и уверены, что могут сделать лучше за меньшие деньги. Истина — где-то посередине.





Программное обеспечение для бизнеса — это не Microsoft Word. Никакое готовое ПО не сможет стать основой для вашей компании. Если бы могло — бизнесу не нужен был бы технический директор. Обычному ресторану не нужно специальное ПО — он может использовать онлайн-сервисы, разработанные для управления современными ресторанами, включая автоматическую интеграцию с такими сервисами, как DoorDash и Uber Eats. Однако, если вы McDonald's и управляете франшизами по всему миру, вам потребуется индивидуальное решение (и технический директор для его запуска).

И наоборот, нет смысла изобретать велосипед. Для создания действительно гибкого решения вы изобретаете и разрабатываете только то, что необходимо, а для остального используете готовые инструменты/сервисы или библиотеки с открытым исходным кодом.

Чтобы понять, что именно нужно разрабатывать, в первую очередь определите, на что в вашей компании распространяются ПИС (права интеллектуальной собственности) — какой у вас «секретный соус». Это та часть продукта, которая уникальна для вас и отличает вас от конкурентов. Это может быть код, который вы разрабатываете, или способ объединения компонентов в готовое решение. Все остальное — отвлекающие факторы: решение проблем, которые уже решены. Независимо от того, к какому варианту вы склоняетесь, важно провести оценку обеих частей уравнения с одинаковым вниманием и энергией.

### 8.2.1. Покупка

При принятии решения о покупке основной технологии для вашей компании необходимо учитывать следующие соображения:

- Стоимость установки/лицензии, включая как ежегодные платежи, так и оплату за использование.
- Профессиональные услуги по настройке.
- В случае запуска системы на своем оборудовании:
  - затраты на оборудование и дополнительные лицензии на ПО;
  - затраты на обслуживание оборудования и программного обеспечения (резервное копирование, обновления безопасности и т. д.);
- Время на внедрение и настройку программного обеспечения.
- Можно ли периодически экспортировать данные.

Не стоит недооценивать время, необходимое для исследования рынка, составления списка подходящих решений и тщательную оценку каждого решения. Критически относитесь к обещаниям поставщиков. Как и в случае с разработкой, где все работает безупречно только на доске, их решение всегда будет лучшим, самым быстрым и самым легко настраиваемым, пока вы не подпишете договор. На выбор решения могут уйти месяцы труда всей команды, вынужденной отвлекаться от основной задачи.

Для простоты составьте список требований, которым должен соответствовать приобретаемый продукт. Укажите, обязательным или желательным является каждый из пунктов. Вы получите чек-лист для оценки предлагаемых решений, и, когда вы выберете два наиболее подходящих кандидата, можно будет использовать этот список для проведения тестирования.

**Таблица 8.1.** Покупка: преимущества и недостатки

Преимущества	Недостатки
<ul style="list-style-type: none"> <li>Проверенная/протестированная платформа</li> </ul>	<ul style="list-style-type: none"> <li>Вы не будете владеть правами на интеллектуальную собственность для всего стека решений</li> </ul>
<ul style="list-style-type: none"> <li>Быстро внедряется</li> </ul>	<ul style="list-style-type: none"> <li>Привязка к поставщику</li> </ul>
<ul style="list-style-type: none"> <li>Профессиональная поддержка</li> </ul>	<ul style="list-style-type: none"> <li>Негибкий/ограниченный набор функций</li> </ul>
<ul style="list-style-type: none"> <li>Если это SaaS платформа, то не нужно беспокоиться о масштабируемости, исправлениях безопасности или обновлениях программного обеспечения</li> </ul>	<ul style="list-style-type: none"> <li>Может быть сложнее найти специалистов, владеющих этим инструментом</li> </ul>
	<ul style="list-style-type: none"> <li>Исправления ошибок / выход новых функций ограничены графиком поставщика</li> </ul>
	<ul style="list-style-type: none"> <li>Конкуренты имеют доступ к тем же инструментам</li> </ul>

Помните: если вы найдете решение, которое работает именно так, как вам нужно, легко настраивается и которое можно быстро начать использовать в продакшене, то его может использовать и ваш конкурент. И в таком продукте может быть не так много вашей интеллектуальной собственности, как вам хотелось бы.

### ЗАМЕТКИ С ПОЛЕЙ

#### *Крупная покупка*

Если у вас крупный договор с поставщиком, вы можете попросить у него подтверждения, что его продукт будет решать ваши задачи, для этого его инженеры должны будут подготовить для вас демонстрационную версию. Это часть процесса продажи, и у крупных поставщиков ПО есть отдельные команды, осуществляющие предпродажную и послепродажную поддержку, — как в автосалоне, где вы можете взять автомобиль на тест-драйв или на все выходные.

### 8.2.2. Разработка

Предположим, вы решаете выделить ресурсы для разработки уникального продукта, отвечающего требованиям компании. Это может занять больше времени, но определенно удовлетворит потребности бизнеса. Разработка — это большое обязательство, особенно для организации, у которой нет наготове необходимых ресурсов; такой путь обычно выбирают компании, в которых уже создана инфраструктура для разработки (см. табл. 8.2).

**Таблица 8.2.** Разработка: преимущества и недостатки

Преимущества	Недостатки
<ul style="list-style-type: none"> <li>Полный контроль над набором функций продукта</li> </ul>	<ul style="list-style-type: none"> <li>Необходимость управлять разработкой продукта, включая проектирование, создание архитектуры, реализацию и внедрение</li> </ul>
<ul style="list-style-type: none"> <li>Контроль над процессом разработки и внедрения</li> </ul>	<ul style="list-style-type: none"> <li>Больше затраты времени на разработку</li> </ul>
<ul style="list-style-type: none"> <li>Проще найти сотрудников</li> </ul>	<ul style="list-style-type: none"> <li>Более трудоемкое планирование и организация</li> </ul>
<ul style="list-style-type: none"> <li>Вы владеете всеми правами интеллектуальной собственности</li> </ul>	<ul style="list-style-type: none"> <li>В меньшей степени протестировано; специфический продукт</li> </ul>
<ul style="list-style-type: none"> <li>Быстрое реагирование на ошибки и запросы новых функций</li> </ul>	<ul style="list-style-type: none"> <li>Необходимость обеспечивать поддержку внутри компании</li> </ul>

Хорошее сочетание — «купить» то, что можно, и «разработать» то, что требует большей гибкости и соответствия вашим требованиям. Как мы отметим позже в этой главе, когда будем обсуждать облачные решения, разработка не означает, что вы должны создавать вообще все. Вы можете использовать и интегрировать многие сторонние компоненты. Например, для отправки электронной почты можно использовать стороннюю службу, или если вам нужна подробная аналитика, то лучше отдать эту задачу отдельному инструменту бизнес-аналитики (BI), а вы будете загружать в него исходные данные.

Помните также: будьте осторожны с решениями low/no code. Это современные версии Visual Basic, позволяющие не-разработчикам создавать ПО с помощью инструментов point-and-click («наведи-и-щелкни») или drag-and-drop («перетащи-и-брось»), без написания кода. Эти инструменты создают обманчивое впечатление, что вы экономите кучу времени и быстро решаете 80 % задачи, но оставшиеся 20 % могут делаться гораздо дольше, потому что окажется, что проблема только усложнилась. В худшем случае вы идете на компромиссы, чтобы оправдать выбор именно этого инструмента, хотя он и не обеспечивает необходимой гибкости. Пойдут ли ваши конкуренты на такой же компромисс или же обойдут вас с более широким набором функций?

### 8.2.3. За и против

То, какую сторону вы займете, во многом зависит от среды, в которой сформировался ваш опыт. Хотя и считается, что все технические директора склонны выбирать разработку, возможны разные варианты:

- *Сторонники покупки.* У таких СТО мало или совсем нет опыта управления командой разработчиков, и они считают, что потери в функциональности и гибкости более чем компенсируются стабильностью платформы.
- *Сторонники разработки.* Эти СТО имеют опыт управления командами разработчиков и знают, как создавать продукты под конкретные задачи. Обычно их раздражает негибкость больших готовых решений.

Успешные технические директора знают, как взять все лучшее из обоих миров, ничего не исключая и подробно изучая каждый вариант.

#### ЗАМЕТКИ С ПОЛЕЙ

##### *Создание систем уровня предприятия*

При проектировании крупных систем я начинаю с предположения, что абсолютно все компоненты будут разрабатываться. Пока я не принимаю конкретных решений, но исхожу именно из этого, поскольку мне необходимо разбить каждую проблему на части, с которыми можно будет работать. Когда я узнаю требования к каждой области, туман начнет рассеиваться и станет понятно, какие компоненты можно купить.

Мир не черно-белый — всегда есть оттенки, которые всё усложняют. В частности, существуют своего рода гибридные платформы, например Salesforce, включающая целую экосистему готовых компонентов, которая удовлетворяет потребности большинства организаций. Но помимо этого она предлагает полноценную среду для создания вашего собственного кода, а он может составлять значительную часть проекта.

Главное правило оценки таких решений возвращает нас к зависимости от поставщика, о которой мы говорили ранее: если вы не можете взять свой код, перенести его на другую платформу и запустить там, вы отдаете себя на милость поставщика и успешность вашей работы будет зависеть от него. Если у него случится сбой или все станет работать медленнее — вы ничего не сможете с этим сделать и вам останется только ждать.

### 8.2.4. Чек-лист

Чтобы тщательно оценить оба варианта, подойдите к этому системно, а также подготовьте аргументы для защиты вашего выбора перед генеральным или финансовым директором, чья поддержка вам понадобится. Вы должны решить, что в конце концов важно для бизнеса. Каким критериям должен соответствовать подходящий вариант? Именно здесь пригодится чек-лист, который поможет вам

определился, что лучше — разрабатывать или покупать. В таблице 8.3 приведен пример чек-листа для проекта.

**Таблица 8.3.** Разработка или покупка

Критерий	Разработка .... Покупка
Необходимо быстро реагировать на изменения и запросы новых функций от клиента	..   .....
Решение должно избавить от необходимости управлять инфраструктурой	.....  ...
Желательно использовать существующую команду разработчиков	.....  .....
Права интеллектуальной собственности не важны; этот аспект не является ключевым для бизнеса	.....  .....
Другие ваши системы должны иметь прямой доступ к данным	.....  .....
Решением будут управлять люди, не являющиеся техническими специалистами	.....  .....
Время внедрения решения	.....  .....

Если, пройдя по чек-листу, вы склоняетесь к решению «купить», то сфокусируйтесь на исследованиях и выборе решения в этой области. Разумеется, выбор должен быть основан на данных, а не на ощущениях.

Стоит отметить, что если вы пойдете по пути покупки, то ров вокруг вашей организации станет значительно уже. Это может быть не важно для организаций, чья интеллектуальная собственность не является технологией. Термин ров (moat) в сфере инвестиций определяет, насколько сложно конкурентам копировать то, что вы делаете. Если вы покупаете сторонние инструменты, ничто не мешает вашему конкуренту делать то же самое.

## 8.3. ОБЛАКО ИЛИ СВОЕ ОБОРУДОВАНИЕ

Во многих современных организациях вопрос использования собственных физических серверов и их обслуживания даже не рассматривается. Другие по-прежнему вкладывают большие средства в оборудование, что затрудняет переход к облачной инфраструктуре. В этом разделе мы рассмотрим плюсы и минусы облачных сервисов и посмотрим, как использовать существующее оборудование и как избежать некоторых проблем, с которыми столкнулись те, кто перешел в облако. Для начала необходимо определить, что же мы имеем в виду под словами «облако» и «собственное оборудование»:

- *Облачный сервис* — вариант инфраструктуры, в которой ресурсы ЦП, памяти, сети и хранилища предоставляются поставщиком и оплачиваются по факту использования, иногда с точностью до секунды. Могут быть доступны как

конфигурации, похожие на обычные серверы, так и отдельные ресурсы, полностью отвязанные от железа, на котором они работают.

- *Свое оборудование* — инфраструктура, в которой базовое оборудование (ЦП, память, сеть и хранилище) принадлежит компании и управляется ею.

Говоря об облаке, можно провести аналогию с электричеством. Мы привыкли вставлять вилку в розетку и потреблять столько энергии, сколько нам нужно, а в конце месяца получать счет на соответствующую сумму. Это намного экономически эффективнее и удобнее, чем содержать собственную электростанцию. За многие годы совершенствования и развития вся наша техника пришла к тому, что эту энергию можно быстро и легко получить с помощью стандартной трехконтактной вилки. Точно так же существуют облачно-ориентированные приложения, которые специально разработаны для использования ресурсов, выделяемых по запросу, с возможностью масштабирования по требованию. Однако в некоторых случаях бывает целесообразнее использовать собственное оборудование. Вот возможные аргументы:

- *Подходят для облака* — приложения, предполагающие пиковые нагрузки: ежедневные, ежечасные, ежедневные или сезонные.
- *Подходят для использования собственного оборудования* — приложения, предполагающие интенсивные вычисления, с постоянной загрузкой ЦП на 90 % и более, например для обработки сложных данных или графики (или майнинга биткоинов!).

### 8.3.1. Облако

В этом разделе мы рассмотрим основные плюсы и минусы использования облака для развертывания вашей системы.

#### СОПРОТИВЛЕНИЕ ПЕРЕХОДУ В ОБЛАКО

За последние 20 лет облачные технологии значительно усовершенствовались и учли все критические замечания. Тем не менее вы все равно услышите аргументы «против», которые приводят компании, упорно отказывающиеся даже пробовать облачные сервисы. Рассмотрим ответы на некоторые из таких возражений:

- *Облако небезопасно.* Когда-то этот аргумент действительно было трудно оспорить, но благодаря значительному развитию технологий мониторинга, изоляции сетей, шифрования, контейнеров и управления доступом маятник качнулся в другую сторону. Сделать систему на собственном оборудовании такой же безопасной без значительных финансовых затрат и усилий будет трудно. Это не означает, что облако стало синонимом безопасности; оно все еще требует реализации всех необходимых (а также разумных) механизмов безопасности. Однако хороший архитектор облачных решений проделает это без особых усилий, и это неотъемлемая часть создаваемых им проектов.

- *Мы не сможем контролировать наши данные.* Для подобного убеждения не существует никаких оснований, но оно все еще живо. При использовании облака все ваши данные хранятся у поставщика облачных услуг, а доступ к ним предоставляется с помощью инструментов и API поставщика. Организация хранения может выглядеть по-разному — как в виде файлов, так и в виде баз данных. Контроля над данными у вас будет не меньше, чем если бы они находились в вашей локальной SAN. Да, за загрузку большого объема данных будет взиматься плата. Но доступу к данным никто не препятствует.

Подобные представления возникли из-за иррационального страха пропустить ежемесячный счет на оплату облачных сервисов. Позвольте прояснить: должно пройти много (действительно много) недель, прежде чем крупные поставщики облачных услуг начнут сами удалять ваши данные.

- *Это слишком дорого.* Да, такое возможно, если вы используете поставщика облачных услуг так же, как традиционный ЦОД. Спустя некоторое время эксплуатация собственного оборудования будет казаться более выгодной (при условии, конечно, что вы учтете все расходы — об этом позже). Однако правильно спроектированное облачное приложение будет экономить значительные средства из года в год, особенно с учетом того, что стоимость облачных сервисов продолжает снижаться.

Чтобы облако стало по-настоящему эффективным, как и все остальное, оно должно использоваться правильно. Само по себе оно не является панацеей от всех проблем.

### **Топ-10 ОШИБОК ПРИ ИСПОЛЬЗОВАНИИ ОБЛАКА**

Некоторые с гордостью заявляют, что работают в облаке, но затем выясняется, что, хотя действительно собственных серверов у них нет, использование облака не приносит им никакой реальной пользы и в большинстве случаев скорее обходится даже дороже. Если вы хотите знать, правильно ли вы работаете с облаком, ответьте на вопросы из этого списка:

- *У вас всегда одинаковые расходы на облако.* Если каждый месяц вы платите за облачные сервисы примерно одинаковую сумму, особенно когда у вас сезонный бизнес с периодами высокого и низкого спроса, это значит, что вы не используете эластичность облака и возможность платить только за то, что вам нужно, и только тогда, когда вам это нужно.
- *Вам требуется прерывание сервиса для его обновления.* Если инженеры планируют выключение сервиса для каждого релиза — значит они используют статические серверы, и ПО необходимо обновлять на каждом из серверов. Современный облачный подход предполагает вместо этого создание новых серверов для новой версии (с использованием контейнеров, таких как Docker), постепенный перевод трафика на них, а затем удаление старых серверов.
- *Вы не можете добавить ресурсы (или убавить их), чтобы соответствовать нагрузке.* Уоррен Баффет сказал: «Девять женщин за месяц ребенка не родят».

А вот облако может, и если у вас не получается добавить еще одну «мать», чтобы справиться с нагрузкой, то ваше приложение не готово к работе в облачных сервисах.

- *У вас заканчивается место на диске.* Сообщения о том, что на серверах заканчивается свободное место, указывают, что используются статические серверы, которые требуют ручного обслуживания. Хорошее облачное приложение должно использовать практически неограниченное хранилище с оплатой по факту использования.
- *Происходит потеря данных.* Никому не нравится слышать (или сообщать) о потере данных, и этого не должно происходить в правильно спроектированной облачной инфраструктуре (если только пользователь намеренно их не удаляет). В противном случае это говорит о наличии точек отказа и отсутствии реальной эластичности и резервирования инфраструктуры.
- *Вы исходите из того, что с вашими серверами ничего не случится.* Поскольку в вашем новом виртуальном мире отсутствует физическое оборудование — легко начать думать, что аппаратные сбои вас не коснутся. Однако хотя они случаются редко, но все же бывают, и, если такой сбой приводит к отказу части вашего сервиса и вызывает серьезное нарушение работы, скорее всего, ваши облачные компоненты плохо изолированы друг от друга и отсутствуют их резервирование.
- *Вы используете какой-то собственный инфраструктурный компонент, например балансировщик нагрузки.* Современные провайдеры облачных сервисов предлагают широкий спектр услуг, которые избавляют от необходимости самостоятельно управлять отдельными серверами. Такие сервисы, как балансировка нагрузки, логирование, кэширование и управление конфигурацией на собственных серверах, традиционно приходится настраивать самостоятельно. Поэтому, если вы включаете их создание в вашу архитектуру, это может свидетельствовать о недостатке знаний или понимания того, какие облачные сервисы вам доступны.
- *В облаке вы используете механизмы обеспечения безопасности, предназначенные для традиционных ЦОД.* Способы обеспечения безопасности традиционных аппаратных серверов отличаются от способов обеспечения безопасности в облаке. Если вы вручную настраиваете файерволы на каждом сервере или у вас отсутствует реальное разделение сетей, вы упускаете целый уровень безопасности, предлагаемый облачным провайдером.
- *Вы по-прежнему рассматриваете аварийное восстановление как переключение на резерв.* Если вы до сих пор считаете, что отказоустойчивость — это возможность переключиться на запасную копию сервиса, которая простаивает в ожидании этого маловероятного события, то вы точно не готовы к работе в облаке. В грамотной облачной архитектуре отказоустойчивость обеспечивается по умолчанию.
- *Вы запускаете несколько сервисов на одном сервере.* Железо стоит дорого, поэтому на одном сервере обычно запускают несколько сервисов (вспомните



классический стек Linux Apache MySQL PHP). В облаке таких ограничений нет. В облачной среде серверы должны быть максимально легкими, чтобы их было легко заменять, обслуживать и масштабировать.

Если вы используете облако так же, как центр обработки данных с традиционными серверами, без эластичности при изменении нагрузки, то оно потеряет большую часть своей гибкости и будет обходиться вам дороже, чем собственное оборудование. Многие облачные решения стоят больше, чем аналогичное железо, просто потому, что при переходе на них компания не понимала, что такое облако на самом деле.

### **ЧАСТНОЕ ОБЛАКО**

Большинство людей представляют себе облако в первую очередь как общедоступный инструмент, но часто, особенно в организациях, которые вкладывают большие средства в существующее оборудование, применяется частное облако. Использование на собственном оборудовании облачной ОС с выделением ресурсов по мере необходимости поможет полностью раскрыть потенциал оборудования и вдохнуть в него новую жизнь. Вы можете использовать как коммерческие решения, так и продукты с открытым исходным кодом и выбрать необходимый вам уровень функциональности. Этот путь рекомендуется только в том случае, если у вас есть ресурсы для управления частным облаком. Хотя вы выиграете в стабильности и надежности (поскольку сможете легко перемещать ресурсы с неисправных серверов), вам все равно придется нести накладные расходы на оборудование.

Логичным развитием этой идеи является гибридное облако — сочетание частного и публичного. Когда ресурсы частного облака исчерпываются, публичное может восполнить дефицит, при этом потребители ничего даже не заметят.

### **РЕЦЕПТ УСПЕХА**

Тех, кто привык к традиционному оборудованию, может пугать переход в облако, но бояться не стоит. Вы сможете использовать все имеющиеся у вас навыки и опыт; нужно лишь взглянуть на проблему по-новому. Люди, успешно перенесшие свои проекты в облако, ставили вопрос так: «Вот таким образом я действовал бы при настройке оборудования; как сделать то же самое в облачном сервисе?» Рассмотрим три правила, которые необходимо помнить для успешного перехода:

- *Облако не гарантирует безопасность.* Защиту необходимо продумать, спроектировать и внедрить в каждую часть решения. Не полагайтесь на какой-либо уровень безопасности только потому, что вы используете сторонний сервис. Каждый раз существует риск неправильной конфигурации, поэтому тщательно проводите тестирование.
- *Сбои неизбежны.* Облако привлекает тем, что нет необходимости поддерживать оборудование, и создает впечатление, что так вы избавитесь от сбоев. Но облако от них не защищено; аварии будут. Планируйте их, создавайте и отработывайте сценарии действий. Благодаря грамотной архитектуре сбоев

могут пройти незаметно и не нарушить работу пользователей, что избавит вас от значительных дополнительных затрат.

- *Создать проблемы проще простого.* В облаке очень легко настраивать и использовать ресурсы. Но их так же легко и уничтожить. Будьте осторожны. Понятие «продакшена» здесь существует только в вашей голове. Облачный провайдер предполагает, что вы знаете, что делаете, и не задумываясь выполнит ваш запрос на удаление рабочей базы данных.

Облако избавляет от множества проблем, с которыми вам пришлось бы справляться, работая с традиционным оборудованием. Однако не думайте, что станет легче. Ваши проблемы теперь просто приняли другой характер.

### ЗАМЕТКИ С ПОЛЕЙ

#### *Производство меда*

Одна из аналогий, которую я использую, описывая разницу между традиционными и облачными приложениями, — это обычная пчела. Мать-природа сделала так, чтобы производство меда не зависело от отдельной пчелы; каждая пчела рано или поздно погибает. Смерть одной пчелы не влияет на общий процесс. Именно так устроено хорошее облачное приложение — сбой является частью его ДНК. С другой стороны, большинство компаний, бизнес которых основан на использовании ПО, устроены как одна-единственная огромная генно-модифицированная пчела, постоянно производящая мед. Однако, если эта «пчела» умрет, производство остановится. И в нашем мире слишком много таких генно-модифицированных пчел.

## 8.3.2. Собственное оборудование

Использование собственного оборудования (on-premises) — это общий термин, означающий покупку и обслуживание физического оборудования для удовлетворения потребностей организации. Оборудование необязательно должно при этом находиться в офисе — оно может размещаться в специализированном центре обработки данных. В этом разделе мы рассмотрим вопросы создания нового комплекса, а также, что более актуально, поддержки существующего.

Раньше альтернатив не было, и, если компании требовались серверные мощности, — приходилось покупать собственное оборудование. Правильно оценить потребности в нем было трудно, поэтому покупалось либо слишком много железа, которое потом не использовалось, либо, наоборот, его было недостаточно, что приводило к повышенной нагрузке на оборудование (и на сотрудников, которым приходилось надеяться, что ничего не сломается). Серверы нужно купить, доставить, подготовить и установить — на это могут уйти недели. Быстро масштабировать систему невозможно. И, конечно, все это только при условии, что вы убедили финансового директора, что без этих больших капиталовложений обойтись нельзя.

Сегодня эти проблемы можно частично решить, объединив оборудование в частное облако и распределяя ресурсы по мере необходимости. Например, если вам нужны дополнительные серверы на время рекламной компании, которая вызовет всплеск посещаемости сайта, вы можете отключить часть внутренних серверов и использовать эти освободившиеся ресурсы в частном облаке для развертывания дополнительных серверов для сайта. Тем не менее ваши ресурсы по-прежнему ограничены (потому что оборудования больше не станет).

### **РАЗМЕЩЕНИЕ ОБОРУДОВАНИЯ**

Существует два основных варианта размещения оборудования:

- *В офисе* — оборудование находится в том же здании, что и сотрудники: в отдельном кабинете, шкафу или даже под столом кого-нибудь из разработчиков (такое тоже бывает, хотя так делать не нужно).
- *В дата-центре* — оборудование размещается в выделенном центре обработки данных, обычно в стойках, вместе с другими серверами.

Как растению для выживания нужны солнечный свет и вода, так и серверу нужны электричество, охлаждение и подключение к сети. Хотя современное оборудование не выделяет столько тепла, сколько старые системы, оно по-прежнему требует охлаждения для нормальной работы. Вы почувствуете это, если разместите его в шкафу без кондиционера — любого, кто откроет шкаф, обдаст жаром!

Потребление электроэнергии — еще один аспект, которым часто пренебрегают. Мы рассчитываем, что просто вставим вилку в розетку, а энергетическая компания сделает все остальное. Хотя, по сути, так и есть, вы должны убедиться, что не перегружаете сеть и что в ней отсутствуют скачки напряжения, которые могут привести к повреждению оборудования. Кроме того, необходимо обеспечить бесперебойное питание на случай, если упомянутая компания какое-то время не сможет поставлять электричество.

### **ЗАМЕТКИ С ПОЛЕЙ**

#### *Важность чистоты*

Я помню, как в начале своей карьеры, только что окончив университет, я мирно сидел в кабинете, занимаясь своей работой, а в соседней комнате гудели серверы. Потом вдруг все зашумели — люди стали бегать, кричать, задавать вопросы. Дело было в том, что все серверы одновременно выключились — совершенно неслыханная ситуация. Их гудение смолкло, и единственный звук, который доносился поверх человеческих голосов, — это ритмичный рев пылесоса, которым энергично возила по полу уборщица. Она работала первый день и ничтоже сумняшеся выдернула из розетки провод, которым были подключены серверы, чтобы включить пылесос. С тех пор я перестал недооценивать важность электропитания.

Наконец, сеть — кровь любого сервера, без которой он не сможет выполнять свои функции. Подключить сеть — это не просто воткнуть кабель в разъем RJ45 в надежде, что на сервер начнет поступать трафик. Необходимо убедиться, что трафик безопасен, и настроить резервное подключение на случай, если текущее соединение прервется. Существует множество сложных мультисерверных инсталляций, которые полагаются только на один шестизильный кабель.

Выделенный центр обработки данных (ЦОД) может позаботиться обо всем этом (и возьмет с вас соответствующую плату). Стоимость ЦОД обычно зависит от объема занимаемого оборудованием места в стойке (измеряется в юнитах, или U; один U равен 1,75 дюйма) и пропускной способности сети (доступного вам подключения к интернету). Большинство центров обработки данных включают в свои тарифы стоимость электроэнергии в расчете на использование стандартных блоков питания. Серверы обычно бывают длинными и плоскими и размещаются в стойке плотно один над другим, поэтому их называют «коробки для пиццы». Такие серверы (или серверы для размещения в стойке) стоят намного дороже, чем серверы в обычных настольных корпусах.

В ЦОД также есть все необходимое сетевое оборудование для создания вашей внутренней сети с автоматическим резервированием подключения к интернету. Вам остается только подключить сетевой кабель.

Если же вы планируете размещать оборудование у себя в офисе, учтите несколько моментов. Вам дополнительно потребуются маршрутизаторы, коммутаторы и, возможно, фаерволы, которые тоже будут нуждаться в охлаждении и резервном питании.

### **ПОКУПКА И ОБСЛУЖИВАНИЕ**

Прежде чем покупать оборудование, независимо от того, где вы планируете его размещать, необходимо определить его конфигурацию (ЦП, память, диск), стараясь обеспечить возможность обновления в будущем. После того как оборудование придет к вам, надо будет установить на него операционную систему. Затем необходимо удалить с него все ненужные программы, настроить средства безопасности и сеть. Все это требует времени.

Если вы размещаете оборудование на своей территории, лучше выделить для него отдельное запирающееся (чтобы сотрудники компании не имели туда свободного доступа) и хорошо проветриваемое помещение с независимым электроснабжением. Тщательно спланируйте расположение оборудования, маркируйте и закрепите кабели.

Разработайте систему инвентаризации: важно отслеживать все оборудование любым способом — от подписывания корпусов маркером до наклеивания распечатанных штрихкодов. Необходимо указывать следующую информацию:

- *Дата покупки/установки* — когда было приобретено и подключено оборудование?

- *Источник покупки* — где было приобретено оборудование (включая полные контактные данные продавца)?
- *Конфигурация* — каковы параметры оборудования? Опишите их достаточно подробно, чтобы не приходилось разбирать устройство, чтобы узнать, что внутри. Укажите также, сколько имеется свободных слотов для будущих обновлений.
- *Время наработки на отказ* — в руководстве производителя должен быть указан ожидаемый срок службы устройства. Расчетный срок службы оборудования, имеющего подвижные детали, такие как вентиляторы и жесткие диски, обычно составляет несколько лет.
- *Способы доступа* — есть ли на устройстве консоль? Если да, то как настроена сеть и где хранится имя пользователя/пароль?

Эти данные значительно облегчат работу с оборудованием и его профилактическое обслуживание, например замену отдельных блоков. Проще всего записать их сразу во время установки, пока все данные у вас под рукой, чем через несколько лет, пытаясь вспомнить, что и когда было установлено.

#### **ЗАМЕТКИ С ПОЛЕЙ**

##### *Часть мебели*

Я встречал много чудовищных вариантов размещения оборудования, но настоящий ужас я испытал, когда увидел, что главный сервер (да, он же единственный) находится под столом одного из разработчиков и регулярно используется как подставка для ног. Он был подключен к удлинителю вместе с другими приборами, без какой-либо пометки о том, что его нельзя отключать. Излишне говорить, что всё исправили немедленно.

Запуск собственных серверов (в том числе в облаке) подразумевает их регулярное обслуживание. В него входит и обновление программ, являющихся частью операционной системы. Однако про ПО, обслуживающее аппаратное обеспечение, обычно забывают. Например, если у вас есть сетевое хранилище, маршрутизатор или блейд-серверы, то у них есть и собственное ПО управления, которое тоже необходимо регулярно обновлять.

Обслуживать само оборудование сложнее. Трудно обосновать замену компонента, который еще не вышел из строя, но заменять детали заранее — это полезная привычка. Как минимум вы должны знать, где их достать, чтобы в случае отказа иметь возможность быстрой замены. Аппаратное обеспечение быстро устаревает и не имеет такой степени обратной совместимости, как нам хотелось бы. Это вынуждает организации искать б/у запчасти на eBay.

Ведя подробные записи, вы получите четкое представление о будущих затратах на обслуживание аппаратной инфраструктуры. Таким образом, для финансового директора не будет сюрпризом, когда вы обратитесь к нему с просьбой выделить деньги на замену компонентов.

## 8.4. АВАРИЙНОЕ ВОССТАНОВЛЕНИЕ

Определим, что такое *аварийное восстановление* (disaster recovery). Поскольку сбой в работе вашей системы (независимо от его причин) нанесет значительный ущерб вашим клиентам, необходимо подготовить план, в соответствии с которым когда (а не «если») такой сбой произойдет, предприятие сможет восстановить свою работу. Обычно подобный план предусматривает перевод нагрузки в место, географически отличающееся от основного, чтобы учесть такие ситуации, как пожар, наводнение или кража.

### ЗАМЕТКИ С ПОЛЕЙ

*Годзилла атакует*

Еще в начале карьеры, когда я впервые услышал термин «аварийное восстановление», я сразу же представил себе Годзиллу, шагающего по городу и разрушающего все здания на своем пути. Как деревенский житель, еще не привыкший к жизни в городе, я задавался вопросом: неужели подобное случается так часто, что обязательно иметь план действий? Со временем я понял, что немного увлекся и нужно думать о более приземленных ситуациях, таких как пожар, наводнение или кража. На самом деле наиболее вероятной проблемой будет сбой питания, сети или оборудования (ничего общего с Годзиллой, но до сих пор я не могу слышать слова «аварийное восстановление» без улыбки, потому что это первое, что приходит мне на ум).

Здесь нужно отметить, что для грамотно спроектированного и реализованного облачного приложения возможность аварийного восстановления предоставляется бесплатно, поскольку этот функционал является частью нормальной работы приложения, а не задействуется явным образом в случае особого события. Но это не означает, что только потому, что вы используете облачные сервисы, вы уже готовы к аварийным ситуациям. Для целей этого раздела предположим, что ваши приложения не облачные.

Как технический директор, ответственный за все технологии в компании, вы должны поддерживать их работоспособность, что бы ни случилось. Внесем ясность: реализовать отказоустойчивость непросто. Большинство организаций имеют плохую (читай: «не имеют никакой») стратегию аварийного восстановления, а те, кто думает, что она у них хорошая, никогда не проверяли ее в реальных условиях, и она является не более чем плацебо, страховочной сетью, которая никак не закреплена.

В этом разделе мы рассмотрим все соображения и вопросы, на которые вам необходимо ответить, чтобы составить грамотный план аварийного восстановления в своей организации. Вот три основных вопроса, которые будут определять вашу стратегию:

- Каково допустимое время простоя?
- Допускается ли частичное предоставление услуг на время восстановления?

- После отказа вы полностью перейдете на новый основной комплекс или вернетесь к прежнему?



ОСНОВЫ СТРАТЕГИИ АВАРИЙНОГО ВОССТАНОВЛЕНИЯ

Для наглядности предположим, что у вас есть две зоны: одна основная и одна резервная, готовая подключиться в аварийной ситуации. Необязательно ограничиваться одной резервной зоной — их может быть несколько. Помните, что если она одна, то, как только вы перейдете в состояние восстановления (начнете использовать резерв), у вас больше не будет резерва для уже задействованного резерва.

### 8.4.1. Допустимое время простоя

Генеральному директору легко требовать нулевого простоя. Хотя такое возможно, это очень дорого и очень сложно, особенно если в инфраструктуре существуют компоненты, которые нельзя запустить в нескольких экземплярах. Яркий пример — обычная база данных: в большинстве организаций таблицы не спроектированы для легкой реализации резервирования в реальном времени (если вы полагаетесь на стенированные базой данных автоинкрементные первичные ключи, а не на GUID, то вам придется несладко). Старые системы или плохая архитектура просто не позволяют иметь дублирующую систему, на которую можно быстро переключиться.

При использовании для вашей системы двух географически разнесенных зон сложность (и стоимость) решения для аварийного восстановления будут определять два фактора:

- Технология синхронизации данных в обеих зонах.
- Способ перенаправления трафика (пользователей) на новую основную зону.

Синхронизация данных между двумя разнесенными в пространстве зонами требует времени — буквально. Для передачи данных по сети требуется время, и даже небольшая задержка в 50–100 мс может иметь огромное значение, когда речь идет о синхронизации баз данных. Даже если сеть в рабочем состоянии и не перегружена, вы будете ограничены временем, которое требуется для пересылки байтов. При выборе места для размещения резервной системы учитывайте схему пиринга между ЦОД (они должны быть связаны друг с другом напрямую, чтобы избежать влияния общей перегрузки каналов связи).

Хотя современные БД (такие как SQL Server и Oracle) имеют мощные функции синхронизации (из-за чего лицензия на них стоит дороже),

скорость передачи данных по-прежнему ограничена физическими возможностями сети.

Таким образом, в зависимости от организации вашей инфраструктуры и конфигурации сети, нулевое время простоя может быть недостижимо. Отставив эту высокую цель, подумайте, какое время простоя допустимо для ваших клиентов: минуты, часы или дни? Определить эту величину можете только вы и генеральный директор, которому вы предоставите данные о финансовых потерях в результате простоя.

Существует два подхода к аварийному восстановлению: горячий и холодный резерв. Горячий — более дорогой, он предполагает малое время простоя, и для него необходимо поддерживать серверы в рабочем состоянии и постоянно синхронизированными, готовыми начать обслуживание пользователей, как только на них начнет поступать трафик. При горячем резервировании постоянно потребляется электроэнергия и загружается сеть, а также сокращается срок службы резервного оборудования.

В отличие от горячего, холодный резерв — это режим, при котором работает минимальное количество оборудования, а при аварийном переключении оно включается по мере необходимости. Это более дешевое решение, но и более медленное, потому что приходится ждать, пока серверы загрузятся и синхронизируют данные. Иногда холодный резерв является более чем достаточно дешевым способом обеспечить необходимый уровень отказоустойчивости.



### 8.4.2. Полное или частичное обслуживание

Второй фактор, влияющий на выбор стратегии аварийного восстановления, — необходимость продолжать предоставлять полнофункциональный сервис в случае аварии. Можно ли обойтись частичным сервисом? Представьте, что что-то пошло не так, и спросите себя, какие услуги из тех, что вы предлагаете, можно оставить временно недоступными или предлагать в режиме только для чтения.

Например, представьте, что вы отвечаете за Twitter и у вас очень ограниченный бюджет. В режиме частичного предоставления сервиса можно позволить людям просматривать ранее опубликованные твиты, но временно ограничить возможность писать и публиковать новые. Или, если у вас онлайн-магазин, позвольте клиентам просматривать и добавлять товары в корзину, но уберите возможность оформить заказ. Почти всегда существуют области, которые какое-то время могут работать с ограниченным функционалом.



Если вы предоставляете услугу в ограниченном объеме, подумайте, что можно временно отключить из сервисов, невидимых для конечного пользователя: например, ведение журналов, создание отчетов или какие-то асинхронные служебные задачи.

Обеспечить полное предоставление услуги гораздо сложнее, в том числе потому, что может понадобиться четко определить текущее состояние системы, которое необходимо воспроизвести в новой системе. Может ли что-то задублироваться или быть обработано повторно — потому что резервная зона не знает, что уже было сделано или какая-то информация полностью потерялась? В качестве примера можно привести уведомления по электронной почте, генерируемые системой. Если сбой происходит во время отправки пачки электронных писем, обязательно ли выяснять, какие письма уже были успешно отправлены, или допустимо, чтобы часть пользователей получила письма еще раз?

### ЗАМЕТКИ С ПОЛЕЙ

*Дорого не всегда значит качественно*

Мне нравится, как новые клиенты рассказывают мне о своих стратегиях аварийного восстановления, если они у них есть. Как гордые родители, они с энтузиазмом сообщают, сколько денег тратят на защиту данных клиентов и обеспечение бесперебойной работы. У меня был один клиент, который слишком доверял одной компании с названием из трех букв, и его ежегодные расходы на лицензии на ПО и оборудование превышали шестизначную сумму (в долларах). Его ожидания были высоки, и руководство верило, что, если оно столько платит, значит все в порядке. Все спокойно спали по ночам. Когда я начал задавать вопросы, оказалось, что всё не так радужно, и, если вкратце, в течение нескольких недель мы сократили расходы до четырехзначной цифры и создали конфигурацию, которая действительно работала и использовала AWS. Частая проблема: клиент слишком доверяет аутсорсингу, предполагая, что компания с названием из трех букв делает намного больше, чем есть на самом деле. Вопрос аварийного восстановления сложен, и это не то, что можно просто передать на аутсорсинг и забыть.

Для некоторых систем действительно трудно создать решение для аварийного восстановления без большого количества ручных операций с базой данных и файлами.

Создавая ограниченную версию сервиса, помните: необходимо предусмотреть возможность запустить на ее основе полное решение. Если в вашей основной зоне отключили электричество, скорее всего, энергетическая компания рано или поздно восстановит электроснабжение. Однако если произошел пожар, наводнение или кража, то ваша урезанная версия должна будет стать основным, полнофункциональным комплексом. Представьте ее как лампочку: она не должна погаснуть и оставить вас во тьме, но, скорее всего, вы потеряете, если какое-то время света будет меньше.

### 8.4.3. Переход вперед или возврат назад

Предположим, вы перешли на резервную зону в полной или частичной конфигурации. Что вы будете делать после восстановления основной зоны? Если вы обнаружите, что не знаете ответа, то обратите на это внимание. Это очевидный признак того, что в вашей организации аварийное восстановление никогда не происходило (или не отработывалось) и никто не думал о том, что будет происходить после того, как прошел Годзилла, и о том, что и как необходимо восстановить. Тем не менее все это вполне может произойти в реальности.

В зависимости от вашей стратегии аварийного восстановления и конфигурации оборудования и при условии, что зоны взаимозаменяемы, вы можете просто назначить резервную зону основной. Зачем организовывать еще один переход, если он не требуется? Займитесь решением проблем с неисправной основной зоной и сделайте ее готовым к использованию резервом. Это называется конфигурацией аварийного восстановления без отката.

Однако если у вас нет взаимозаменяемых комплексов, вам придется после решения проблемы вернуться к основной зоне — так обычно происходит, если вы предоставляете частичный сервис во время восстановления. Хорошая новость в том, что необязательно делать это немедленно. Вы можете запланировать второй перерыв в работе сервиса и вернуться к основной зоне спокойно, в удобное для вас время, обычно это можно сделать в выходной или ночью.

### 8.4.4. Организация реагирования на аварии

До сих пор мы говорили о различных вариантах конфигурации аварийного восстановления, рассматривая отдельные части инфраструктуры, чтобы понять, что подойдет для вашей организации. Процесса переключения с одной зоны на другую мы пока не касались. Здесь, как и в случае со сбережениями на черный день, которые у всех нас есть, самое сложное — определить, наступил ли этот черный день, чтобы можно было открыть кубышку. Переход в режим аварийного восстановления — серьезное событие, и такое решение не должно приниматься легкомысленно.

В некоторых случаях это понять проще (вспомните о запуске «Аполлона», когда думаете, начинать ли процедуру аварийного восстановления), в некоторых — сложнее. Например, полный отказ оборудования, пожар, наводнение или кража — это события, после которых нельзя быстро восстановиться, поэтому имеет смысл запустить аварийное переключение.

Однако другие события не столь очевидны. Например, ввод питания, который периодически выключается, но при этом для поддержания работы ключевых серверов как раз хватает ресурса ИБП (источника бесперебойного питания), или сбой сети, когда провайдер уверяет вас, что занимается проблемой и подключение будет восстановлено «очень скоро». Когда — «скоро»? Стратегия перехода в режим аварийного восстановления, включающая такие критерии,

как затронутые сбоем службы и время, в течение которого они остаются недоступны, избавит от неопределенности, и вы будете четко понимать, когда пора щелкнуть воображаемым переключателем.

Определив, когда переключаться, спланируйте порядок перехода. В общем случае необходимо сделать следующее:

- Убедитесь, что все контактные данные, параметры сети и пароли хранятся в надежном и легкодоступном месте.
- Определите, какие службы необходимо запустить (если они находятся в режиме холодного резервирования) и заполнить актуальными данными.
- Определите все сетевые адреса, которые необходимо будет изменить, включая как публичные, так и внутренние записи DNS. Для внесения изменений в DNS потребуется время, в зависимости от срока кэширования записей зоны, поэтому предусмотрите это в плане.
- Определите все роли и параметры настройки, а также какие тесты необходимо выполнить для проверки резервной зоны и кто должен будет это делать.
- Если вы планируете возврат к исходной конфигурации, продумайте план перехода к основной зоне после устранения сбоя.

### ЗАМЕТКИ С ПОЛЕЙ

#### *Непреднамеренные последствия*

Для работы сотрудников с серверами, расположенными в офисе, в компаниях обычно используются виртуальные частные сети (VPN). Но что произойдет, когда вы переключитесь на новую зону с другими точками доступа? Что ж, у одной портфельной компании, с которой я работал, с помощью аварийного восстановления отлично получилось перепилить сук, на котором она сидела. Ей удалось полностью заблокировать себе доступ к резервной зоне, потому что правила файервола были прописаны только для основной зоны. Положительный момент: средства безопасности сработали отлично!

Сделав все это, составьте необходимые документы, разошлите их заинтересованным лицам и разместите в легкодоступном месте. Люди не должны судорожно искать пароль к консоли, когда им понадобится переключиться на резерв.

Крайне важно регулярно тестировать процедуру аварийного восстановления в нерабочее время, чтобы убедиться, что вся документация и процессы доступны и понятны. Системы развиваются, в них вносятся изменения, компоненты добавляются и удаляются. Процесс аварийного восстановления тоже должен развиваться и всегда соответствовать текущему состоянию системы, потому что об авариях обычно не предупреждают заранее.

Рекомендуется регулярно проверять средства аварийного восстановления, по крайней мере, раз в неделю, чтобы убедиться, что все данные при переходе на резерв по-прежнему передаются туда, куда нужно. Раз в квартал или каждые шесть месяцев отрабатывайте полное переключение, чтобы сотрудники освоили

все процессы и не воспринимали сбой как экстраординарное событие. К тому же, как гласит старая поговорка, повторение — мать учения.

### 8.4.5. Облако и аварийное восстановление

Хотя мы исходили из того, что система, для которой вы создаете стратегию аварийного восстановления, не работает в облаке, это не означает, что облако нельзя использовать как часть решения. Вполне допустимо при необходимости рассматривать облако как временный центр обработки данных, и это может оказаться весьма выгодным.

Облако идеально подходит для «горячего» или «холодного» резерва, поскольку вы можете передавать в него данные и оплачивать только тот объем хранилища, который вы используете. Создать взаимозаменяемую конфигурацию в облаке немного проще, потому что вы платите только за то, что вам нужно, и тогда, когда вам это нужно. Облако упрощает тестирование различных конфигураций и стратегий при ограниченном бюджете.

## 8.5. РАБОТА С ДАННЫМИ

Данные — это кровь нашей отрасли. Они поступают в организацию и выходят из нее, генерируют данные о движении данных, перерабатываются в другие данные, и все это необходимо сортировать, преобразовывать, хранить и обрабатывать. Данные — это больше чем файлы на жестком диске или строки в таблице. У них есть значение, контекст и ценность. И наоборот, в то же самое время они могут быть избыточными, отвлекающими и бесполезными. Необходимо уметь отличать полезные данные от информационного шума.

Раньше хранение данных стоило очень дорого. Сейчас, в облачной парадигме бесконечного хранилища, хранить данные стало дешево и просто как никогда (и стоимость продолжает снижаться с каждым годом). Из этого вытекают два вопроса: (а) стоит ли хранить вообще все и (б) как организовать хранение, чтобы данные оставались доступными?

Эффективная работа с данными и их защита — непростая задача. Безопасности в этой книге посвящена отдельная глава, поэтому сейчас мы лишь слегка коснемся некоторых ее вопросов. За данные компании отвечает СТО, и, по моему опыту, технические директора организуют работу с данными не так хорошо, как могли бы, в основном потому, что не понимают истинной ценности этого актива или не знают обо всех доступных инструментах и методах.

В этом разделе мы рассмотрим возможные способы хранения и основные типы данных в компании, а также вопросы безопасности, соответствия нормативным требованиям и прав собственности, с которыми вы можете столкнуться. Попутно мы познакомимся с актуальной терминологией в области данных, которая поможет вам разобраться со множеством поставщиков и решений в этой сфере, которые предлагаются на рынке.

### 8.5.1. Хранение

Когда мы думаем о данных, мы, естественно, представляем их в виде папок с файлами. Именно так представляют себе данные типичные руководители высшего звена, причем большинство из них ограничивается электронными таблицами. Это нормально, учитывая, что топ-менеджеры в основном выходцы из финансовой среды. Они могут говорить о базах данных, на самом деле имея в виду большие листы Excel. Задача технического директора — помочь топ-менеджерам понять, что их самый ценный актив представляет собой нечто большее, чем электронные таблицы. Другая задача СТО — продумать и спланировать долгосрочное размещение данных, чтобы бизнес мог использовать их эффективнее.

Как мы уже сказали, хранение данных никогда не стоило так дешево, как сейчас, и больше нет оправданий для того, чтобы не сохранять их. Но что это на самом деле означает? И действительно ли нужно хранить всё? Чтобы ответить на эти вопросы, для начала рассмотрим основные типы хранилищ, в которые можно загружать данные:

- *Файлы/папки.* На сегодняшний день файл — автономный именованный набор данных, находящийся на физическом носителе, — по-прежнему является наиболее универсальным способом хранения данных. Существует множество популярных форматов файлов, таких как JSON, XML, CSV и TXT, и этот способ хранения данных универсален, потому что почти каждый инструмент способен открывать файлы и работать с ними. Файлы легко копировать, шифровать и перемещать, и ими легко делиться. Не существует ограничений на виды данных, которые можно хранить в файле.
- *Реляционные хранилища (базы данных SQL).* Данные, которые имеют общие атрибуты (столбцы) и связаны друг с другом (строки), прекрасно вписываются в структуру, которую мы называем таблицами. Таблицы могут быть связаны с другими таблицами, а наборы таблиц известны как базы данных. К таблицам можно выполнять запросы, используя специальный псевдоязык (SQL), который описывает команды для получения данных, их изменения или записи в таблицу новых данных. Это очень популярный механизм для хранения больших объемов одинаково структурированной информации.
- *Хранилища NoSQL.* В хранилищах NoSQL, или документных, отсутствуют структурные ограничения, свойственные реляционным хранилищам. В них сохраняется только то, что вы передадите, и не остается пустот на месте отсутствующих данных, как в реляционных СУБД. Группы документов образуют коллекции, в которых каждый документ может значительно отличаться от другого.
- *Озеро данных.* Озеро данных — это место для хранения большого количества файлов разной структуры и разного размера с целью резервного копирования или последующего анализа. Оно обычно используется в качестве промежуточного хранилища для данных, поступающих в организацию, пока для них еще не назначено постоянное место.

- *Корпоративное хранилище данных (Data warehouse)*. Хранилище — это специально разработанная база данных, которая в основном используется для анализа и бизнес-аналитики (business intelligence, BI). В нем хранятся большие наборы данных, организованных таким образом, чтобы упростить составление отчетов и годовой анализ.

До появления облака создание и обслуживание хранилищ любого из этих типов было непомерно дорого, и на каждом этапе подготовки принимались решения о сокращении количества данных до минимально необходимого на текущий день. Если бизнес развивался и решал, что ему нужны более подробные данные, то сохранять их можно было только начиная с этого момента, а исторические данные были утеряны.

Файлы и базы данных распространены повсеместно. В каждой организации есть своя база данных — SQL Server, Postgres, MySQL или Oracle. Тем не менее зачастую БД плохо организованы, например, одна БД может использоваться одновременно для нескольких областей (скажем, для клиентского веб-сайта и для предоставления отчетов о состоянии бизнеса). Вместо этого у каждой области бизнеса должна быть своя БД, и данные должны реплицироваться между ними. Таким образом, когда финансовый директор задумает сформировать масштабный отчет о квартальных продажах, он не положит весь сайт.

Базы данных NoSQL (такие как MongoDB, CouchDB, DynamoDB и Elasticsearch) становятся все более популярными, особенно на ранних этапах разработки проектов, когда еще нет полного представления об итоговой структуре. Их производительность с годами увеличилась настолько, что при правильно выбранной организации данных они способны работать так же эффективно, как и их реляционные аналоги. Будьте осторожны и не увлекайтесь: некоторые команды строят всю систему на архитектуре NoSQL. В этом случае возрастает риск нарушения единообразия данных (то есть в разных документах одни и те же данные будут храниться в разных форматах, например, число 2 может быть представлено в строковом и целочисленном форматах, потому что где-то при обработке данных они не были приведены к одному виду). NoSQL не обеспечит вам ни целостность, ни единую структуру данных.

Озеро данных (Apache Hadoop, Amazon S3) — это одно большое пространство для хранения файлов. Однако это не просто крупное файловое хранилище. В озере данных есть инструменты и механизмы для просмотра файлов, чтобы вы могли выбрать то, что вам нужно. Например, оно отлично подходит для размещения журналов продакшена. Вам не нужно беспокоиться о формате файлов, он может меняться со временем. При необходимости озеро данных предоставит удобный механизм для перебора файлов журналов, чтобы найти заданное ключевое слово или строку (этот подход называется map/reduce).

Наконец, корпоративное хранилище данных (Amazon Redshift, SQL Server) — это место, куда поступают данные, необходимые для отчетности. Оно специально разработано для упорядочивания данных таким образом, чтобы их можно было

очень быстро проанализировать для получения отчетов, необходимых бизнесу. Данные в хранилище обычно агрегируются, обезличиваются и выстраиваются таким образом, чтобы отвечать на вопросы бизнеса. Такое хранилище будет полезно, только если накоплено значительное количество данных. Традиционная реляционная СУБД тоже может обрабатывать запросы к данным, но придет время, когда ее окажется недостаточно.

### ЗАМЕТКИ С ПОЛЕЙ

#### *Ограниченные знания опасны*

Я встречал множество глупейших ошибок в конфигурациях баз данных, в том числе сделанных теми, кто уверен, что нельзя использовать соединение таблиц. Однако особенно примечателен один случай, когда мою команду позвали помочь с очень медленно работавшей базой данных, из-за которой зависал весь сайт. Покопавшись, мы увидели множество операций дискового ввода/вывода, связанных с БД и указывающих на большое количество операций записи на диск, что было странно, поскольку данных сохранялось не так много. Оказывается, один из технарей вычитал, что для ускорения работы нужно добавить к таблице индексы. Исключительно из лучших побуждений он создал индексы для каждого столбца в каждой таблице. База данных утонула в собственных метаданных, поскольку в ней стало больше данных о данных, чем самих данных. Действительно, информация может нанести вред, если окажется не в тех руках!

В облаке все эти хранилища данных можно организовать без дополнительных затрат на запуск и управление серверами (и лицензиями). Подавляющее большинство команд, с которыми я работаю, только выиграли от переноса хранилищ данных в облако, где они масштабируются по требованию, поэтому нехватка места осталась в прошлом, а данные защищены гораздо лучше благодаря множеству инструментов, доступных по умолчанию.

Организация должна иметь хотя бы какую-то разновидность озера данных. Представьте, что это чердак — место, где хранится все, что вы не можете заставить себя выбросить, потому что однажды оно может вам понадобиться. Озеро данных — это место хранения данных, у которых нет постоянного дома, поэтому, если они однажды кому-то понадобятся, вы, по крайней мере, будете знать, что они у вас есть. Хотя поиск на чердаке может занять некоторое время, вы их там найдете.

## 8.5.2. Типы данных

Не все данные в организации одинаковы. Хотя это кажется очевидным, на удивление многие компании не осознают этого простого факта и в результате наживают множество проблем. Важно разбираться в основных типах данных, за которые вы отвечаете.



Хотя это и не исчерпывающий список, его можно взять за основу для выделения групп данных:

- *Система учета* — данные, которые считаются ключевыми для вашей компании, обычно они содержатся в БД. Данные этой категории считаются «истиной» для организации.
- *Конфиденциальные/охраняемые данные* — в эту категорию попадают данные, требующие дополнительной защиты. Примеры: паспортные данные, номера водительских прав, банковская информация, данные кредитной карты.
- *Транзакционные данные* — данные, которые генерируются в процессе продаж или оказания основных услуг; их часто называют операционными данными.
- *Данные бизнес-аналитики* — данные, которые используются для аналитики и отчетов; обычно это агрегированные/обезличенные копии транзакционных данных.
- *Данные журналов* — данные, генерируемые различными компонентами и программами, обеспечивающими работу системы. Эти данные используются для расследования ошибок/сбоев и аудита, а также для анализа производительности.



Для каждого класса нужно разработать правила, которые будут учитывать особенности вашей организации. Необязательно вдаваться в подробности, но определите по крайней мере следующие характеристики:

- *Права собственности.* Кто владеет данными этого класса? Проще всего предположить, что владелец — вы. Наличие прав собственности означает, что вы можете делать с данными все, что пожелаете (продать, удалить или даже сделать общедоступными, если захотите). Проведя анализ, вы, вероятно, обнаружите, что у вас есть данные клиентов/пользователей, которыми вы просто управляете. Также данные могли быть арендованы вами на ограниченный период времени (например, списки рассылки или наборы данных для анализа). Определение истинного владельца данных позволит вам выяснить, для чего и как разрешено их использовать.
- *Хранение.* Как долго предполагается хранить данные? Этот вопрос касается как оперативных хранилищ, так и резервных копий (если правами собственности это предусмотрено). Сюда же относится политика устаревания: как долго хранить данные до перемещения в долгосрочное (более дешевое)



хранилище? В каких случаях можно удалить все данные, включая резервные копии (например, если клиент уходит от вас)?

- *Видимость.* Кто может видеть данные как внутри компании, так и вне ее и какими средствами будет обеспечено ограничение видимости (шифрованием, списками доступа и т. д.)? Когда данные устаревают или перемещаются в другой класс (например, когда оперативные данные становятся данными бизнес-аналитики), здесь будет определяться, какие изменения необходимо внести (например, обезличить данные пользователей), чтобы соответствовать правилам нового класса.
- *Соответствие нормативным требованиям.* Должны ли данные соответствовать нормативно-правовым или другим особым требованиям? Если да, необходимо внедрить дополнительные процессы аудита и обеспечения безопасности, которые могут различаться в зависимости от региона, где работает компания. Особого обращения требуют, например, персональные, медицинские и финансовые данные.

Данные — один из самых ценных активов, которыми компания управляет и владеет. Чем больше данных у вас есть, тем больше их генерируется. Грамотная работа с данными позволит компании избежать этических и правовых коллизий.

Особо следует выделить данные, для которых необходимо обеспечить соответствие нормативным требованиям. В основе любых ограничений (РП, HIPAA, PCI и др.) лежит требование безопасного и бережного обращения с данными. Если у вас есть такие данные, то, управляя ими, внедряйте все возможные способы защиты. Хотя для некоторых типов данных это может быть излишним, но это гарантирует, что вы выработаете в команде привычку относиться к этим данным ответственно, поэтому возможный аудит не станет потрясением и неожиданностью. Также планирование архитектуры с учетом обеспечения безопасности данных и возможности аудита обеспечит страховку на случай любых будущих изменений.

## 8.6. МИКРОСЕРВИСЫ ИЛИ МОНОЛИТ

Существуют два основных принципа разработки корпоративных приложений: монолитная архитектура и более современная микросервисная. Многие системы являются сочетанием этих двух методов. В этом разделе мы рассмотрим их основные различия и то, почему микросервисный подход стал основным при проектировании и поддержке современной архитектуры.

Концепция монолитной архитектуры старше микросервисной и предполагает создание и деплой системы (или ее больших частей) как единого целого. Как правило, для нее характерна единая кодовая база, не предназначенная для использования кем-либо еще, незначительная степень повторного использования кода и слабая изоляция компонентов.

Рассмотрим таблицу 8.4. Если вы отметите «да» для трех или более признаков, то, скорее всего, вы работаете с монолитной архитектурой.

**Таблица 8.4.** Признаки микросервисной и монолитной архитектуры

Единая кодовая база, без внутренних библиотек	Да / Нет
Одна база данных, используемая для всех систем	Да / Нет
Сбой в одном месте приводит к полному прекращению обслуживания	Да / Нет
Односерверная архитектура	Да / Нет
Обновление программного обеспечения требует планового отключения	Да / Нет

Микросервисы — это искусство разделения системы на отдельные «мини-приложения» с узкой областью применения, выполняющие определенное действие или функцию. Взаимодействие между ними осуществляется через API, служебную шину (service bus) или очередь сообщений. Надежный микросервис — это независимый компонент со своим собственным хранилищем данных, средой выполнения (на основе подходящего языка/библиотеки) и релизным циклом, который не влияет на другие части системы.

Микросервисы привлекательны главным образом тем, что снижают риски и позволяют использовать в каждом компоненте наиболее подходящие для него технологии. Они стали очень популярны, когда возможности технологии догнали теорию, а скорость выполнения вызовов API стала достаточно высокой и приблизилась к быстрдействию вызовов внутри кода.

Для технического директора важно, что микросервисы позволяют разбить большую систему на ряд более мелких независимых модулей. Все они могут обновляться по собственному расписанию, и, учитывая, что есть крупные части системы, которые изменяются редко, можно спать спокойно, зная, что они не пострадают в результате внесения каких-либо изменений.

С другой стороны, монолитная архитектура характерна для устаревших систем, что делает внесение изменений в нее очень рискованным из-за высокой опасности непредвиденных последствий. Обновление даже небольшой библиотеки, на первый взгляд невинное, может запустить цепную реакцию, поскольку части кода плохо изолированы друг от друга.

Подвох микросервисов в том, что без жесткого управления и контроля они могут превратиться в набор маленьких монолитных приложений. Такое может происходить с течением времени, если компоненты обрастают все большим функционалом, и, хотя разработчик руководствовался исключительно благими намерениями и, возможно, даже до сих пор убежден, что созданная им архитектура — микросервисная, на самом деле так будет только на словах.

Микросервисы дают возможность внесения изменений в будущем и позволяют отложить технологические решения в какой-то области. Создав надежный, согласованный API, вы можете развивать реализацию сервиса

со временем, по мере необходимости. Если, например, вам нужно сохранять и раздавать файлы для всей системы — вы можете создать микросервис для работы с файлами (File MS).

Этот File MS будет иметь два простых API: принимающий файл для сохранения и возвращающий файл. Первая версия реализации может просто хранить файл в локальном каталоге (с резервной копией). По мере развития сервиса она может хранить файл в сети SAN с большей емкостью. Если сервис становится очень востребован, можно использовать для его реализации облачный сервис, такой как Amazon S3.

Микросервис является слоем абстракции, который скрывает конкретную логику хранения файлов и предоставляет остальной части системы единым образом способ для работы с ними (API). Микросервис похож на библиотеку, используемую в коде, только гораздо более изолированную. Это дает возможность развивать отдельные части вашей системы с такой скоростью, которая нужна бизнесу.

Монолитная архитектура тоже имеет свое применение, но необходимость отключать крупные части системы для проведения обновлений не идет на пользу компании. К сожалению, иногда с монолитами приходится иметь дело, поскольку технология, лежащая в основе некоторых устаревших систем, оказывается несовместимой с современной модульной архитектурой. Тем не менее, проводя модернизацию, всегда стремитесь разбивать компоненты системы на отдельные части — это эффективный способ наведения порядка.

Иногда монолитная архитектура может быть предпочтительна, особенно если вы находитесь на этапе проверки концепции (или работаете в стартапе), когда важно не то, как вы будете воплощать идею, а то, чтобы эта идея была рабочей и бизнес пришел к выводу, что ее дальнейшее развитие целесообразно.

Еще один побочный эффект микросервисной среды заключается в том, что вы можете организовать команды вокруг отдельных модулей, чтобы каждая команда отвечала за разработку, поддержку и управление своим модулем. Это распространенная стратегия в крупных организациях, которые рассматривают пользователей своего API как клиентов.

Звучит знакомо? Да — именно так работают все современные облачные провайдеры. AWS — это всего лишь набор микросервисов, каждый из которых предоставляет свой функционал через API-интерфейсы, которые вы оплачиваете по мере использования.

Подавляющее большинство сервисов AWS возникло благодаря необходимости поддерживать основные сайты торговой площадки Amazon. Библиотека микросервисов AWS стала настолько мощной, что появилась возможность продавать функциональность внешним разработчикам — то есть нам с вами. Хотя ваша компания, может быть, и не станет следующей AWS, надежная микросервисная архитектура позволит вам стать ближе к клиентам и предоставлять им доступ к вашим API, чтобы они могли выстраивать свою собственную экосистему на основе предоставляемых вами услуг.

## 8.7. ПО С ОТКРЫТЫМ ИСХОДНЫМ КОДОМ

ПО с открытым исходным кодом (open source) — это ПО, которым члены сообщества разработчиков делятся друг с другом на безвозмездной основе и которое доступно каждому желающему, по крайней мере для просмотра, и, возможно, для использования в собственных проектах, с некоторыми ограничениями в зависимости от лицензии. Именно эти ограничения мы здесь и рассмотрим, поскольку многие компании не разбираются в их различиях и сталкиваются с юридическими трудностями при продаже или взаимодействии с отдельными клиентами или организациями.

ПО с открытым исходным кодом позволяет ускорить разработку любого программного проекта за счет использования готовых компонентов. Популярными примерами ПО с открытым исходным кодом являются операционные системы (яркий пример — Linux), серверы баз данных (MySQL, Postgres) и настольные приложения (Firefox, Open Office), а также программные библиотеки (jQuery, Gson). Учитывая степень распространения ПО с открытым исходным кодом, вы найдете в нем что-то полезное для себя на любом уровне.

### 8.7.1. Виды лицензий

Хотя у вас есть доступ к исходному коду этого ПО в интернете, это не значит, что у вас есть все права на его использование. Хотя я и могу сыграть Let It Be (на самом деле не могу) на фортепиано, это не значит, что мне можно продавать записи своего исполнения, не выплачивая гонорары правообладателям — Маккартни и Леннону. Так и с исходным кодом. Исходный код может быть защищен авторским правом, как и любое написанное слово, и обычно его использование ограничивается лицензией, которая определяет, что вы можете и что не можете с ним делать. Лицензии для исходного кода делятся на две большие категории:

- копилефт (Copyleft);
- разрешительные.

ПО с копилефт-лицензиями обычно считается неподходящим для бизнес-использования. Этот тип лицензий (наиболее популярной из них является Универсальная общедоступная лицензия (general public license, GPL)) подразумевает, что любое программное обеспечение, включающее код с лицензией GPL, должно само распространяться с лицензией GPL, включая все его модификации. Другими словами, если ваш секретный алгоритм использует ПО с лицензией GPL, то вы юридически обязаны сделать свой исходный код доступным для всех желающих, включая конкурентов.

Разрешительные лицензии (например, BSD, MIT или Apache) не имеют таких ограничений, поскольку они разрешают включать ПО с открытым исходным кодом в коммерческое программное обеспечение или дорабатывать продукты с открытым исходным кодом, а также объявлять производный продукт своей

частной собственностью и закрывать исходный код, что, естественно, гораздо предпочтительнее для бизнеса.

Некоторые разрешительные лицензии обязывают вас публично уведомлять об использовании библиотеки в конечном продукте. Подобные уведомления можно увидеть в разделе «О программе» на некоторых устройствах (например, для телефонов на Android перечислены все использованные библиотеки с открытым исходным кодом).

Каждый из указанных типов лицензий содержит четкие формулировки в отношении авторских прав, патентной защиты и производных работ, различия между которыми достаточно незначительны и имеют значение, только если вы вносите большие изменения или создаете новый продукт на основе ПО с открытым исходным кодом.

Хотя GPL является исключительно копилефт-лицензией, существует ее облегченная версия (LGPL). Этот тип лицензии не требует делать код общедоступным в случае, если его связывание с лицензируемым ПО динамическое, а не статическое. Это гораздо более удобно для бизнеса, чем GPL.

Этот обзор различий ни в коем случае не стоит считать исчерпывающим, поэтому, если вам нужно подробнее углубиться в эту тему, проведите собственное исследование.

#### ПРИМЕЧАНИЕ

Стоит отметить еще один тип лицензии, называемый «общественное достояние» (public domain). Это самая свободная из существующих лицензий, она не устанавливает права собственности и авторские права и не содержит абсолютно никаких ограничений на использование. Она встречается не так часто, хотя примеры кода в книгах часто публикуются под этой лицензией.

### 8.7.2. Рекомендации по использованию

Старайтесь по возможности использовать ПО с открытым исходным кодом, особенно если вы работаете на Java, JavaScript или Python, поскольку на GitHub и SourceForge доступно множество ответов на вопросы и готовых утилит. Прежде чем вы дадите зеленый свет разработчикам, ознакомьтесь с рекомендациями о том, как лучше включать ПО с открытым исходным кодом в ваш продукт:

- Составьте список лицензий, которые может использовать команда, — скорее всего, это будут лицензии типа Apache/MIT/BSD.
- Библиотеки необходимо использовать целиком — не выделяйте отдельные фрагменты и не включайте их таким образом, как будто они были разработаны у вас. Воровство недопустимо.
- Создайте централизованный список всех внешних библиотек (например, общую электронную таблицу) со следующей информацией:

- имя библиотеки;
- URL веб-сайта;
- тип лицензии;
- компонент, в котором она используется.

Разработайте и передайте команде инструкции по использованию open source, чтобы она могла быть уверена в том, что вы поддерживаете, уважаете и поощряете использование открытого исходного кода.

### 8.7.3. Публикация ПО с открытым исходным кодом

Есть еще один способ работы с ПО с открытым исходным кодом — это поделиться своим кодом с другими. На это могут быть следующие причины:

- Вы много лет использовали результаты работы других людей и теперь хотите отблагодарить сообщество.
- Вы хотите повысить надежность своего продукта и просите других помочь в его отладке.
- Вы хотите установить стандарт в определенной нише и ускорить разработку, приглашая других присоединиться к ней.

Крупные корпорации, в том числе Google, Oracle, Microsoft, IBM, Netflix, Amazon и многие другие, выпускают ПО с открытым исходным кодом (под разными лицензиями). Например, Microsoft владеет GitHub — одним из крупнейших ресурсов, на котором размещаются open source проекты.

Если вы решите открыть исходный код библиотеки, разработанной вами или вашей командой, важно понимать, что вы не просто делаете ее общедоступной. Проект с открытым исходным кодом должен поддерживаться и обновляться. Никто не захочет участвовать в разработке библиотеки, которая годами не обновлялась. Чтобы ваша публикация пользовалась успехом, ее необходимо поддерживать так же, как внутреннюю библиотеку, и продолжать исправлять ошибки и добавлять функции.

#### ВКЛАД В ОБЩЕЕ ДЕЛО

Я сам успешно публиковал множество малых и больших проектов с открытым исходным кодом и получал отличные отзывы. Я также принимал участие в разных проектах с открытым исходным кодом, исправляя ошибки и добавляя новые функции. Лично для меня в этой работе важна возможность ответить добром на добро. За те годы, что я занимаюсь разработкой, я использовал множество open source библиотек и хочу хотя бы частично вернуть долг сообществу.

Перед тем как создавать учетную запись компании на GitHub и публиковать там код, убедитесь, что генеральный директор (и юристы) понимают, что

именно вы собираетесь делать и зачем вы это делаете. Продумайте, какой код вы опубликуете и в каком формате. Делясь кодом, будьте очень внимательны. Убедитесь, что вы не раскрываете ничего, что защищено правами интеллектуальной собственности или содержит коммерческую тайну компании. Кроме того, разумеется, публикуемый код должен принадлежать вам.

Для публикации отлично подходят небольшие библиотеки, которые ваша команда создавала, разрабатывая основной продукт. Это полезно еще и тем, что позволяет понять, в правильном ли направлении вы движетесь, особенно если вы встретите похожую библиотеку.

### 8.7.4. Использование кода из интернета

Неважно, насколько хороша ваша команда разработчиков — рано или поздно в поисках решения они полезут в интернет. Пытаясь выяснить, почему данная библиотека ведет себя так, а не иначе, или как что-то написать на данном языке, они в конце концов увидят какой-то код в своем браузере. Ответ на все их вопросы уже ждет, когда его выделят на странице — CTRL+C и затем CTRL+V в IDE или редакторе. Задача решена.

Не торопитесь: сам по себе факт, что код находится на StackOverflow.com или найден в сообщении в чьем-то блоге или статье, не означает, что его можно просто взять и включить в ваше решение. Помните, что любой код, созданный вашей командой, будет считаться активом и интеллектуальной собственностью вашей компании. Он должен быть создан добросовестно, не скопирован и не должен содержать элементов, защищенных авторскими правами. Вы ходите по юридическому минному полю, но, если будете руководствоваться здравым смыслом, поможете команде избежать ошибок.

#### ПРОВЕРКА КОДА

*Код можно и нужно проверять*

При продаже компании может проводиться технический due diligence, при котором проверяются права собственности на разработанный в ней исходный код, особенно если он является основным ресурсом компании. Существует множество инструментов для поиска плагиата (спросите любого профессора информатики). Я видел, как расторгались договоренности о сделках, когда выяснялось, что крупные части кода, который компания выдавала за свой собственный, на самом деле были взяты из open source проектов. Такие находки ставят под сомнение достоверность и законность вообще всего, что делает компания. Еще подозрительнее, если в ходе due diligence разработчики утверждают, что они вообще не используют ПО с открытым исходным кодом: тогда я копаю глубже, чем обычно.

В целом «копирование» кода с форумов разработчиков — это нормально. Все, что превышает 5 или 10 строк, стоит переписать, используя алгоритм только для

вдохновения (просто переименовать переменные недостаточно). Чаще всего код на этих форумах демонстрирует примеры использования библиотеки или API и не является чьей-то собственностью.

Части кода из библиотеки с открытым исходным кодом можно копировать, только если это разрешено лицензией библиотеки, и в этом случае следует действовать так, как если бы вы интегрировали всю библиотеку (то есть зарегистрировать ее использование в вашем списке).

## 8.8. ЯЗЫКИ И ФРЕЙМВОРКИ

Выбор языка и фреймворка — еще одно важное решение, которое может принять технический директор. Это решение будет иметь далеко идущие последствия, возможно, даже более далекие, чем ваше пребывание в компании. Однако в большинстве случаев выбор языка уже сделан за вас, потому что вы унаследовали стек технологий от своего предшественника.

Бизнесу по большому счету все равно, на каком языке разработана система, пока она решает проблемы и пока клиенты довольны. Тем не менее переход на другой язык (или даже фреймворк) — задача не из легких. Помимо того, что в существующую команду разработчиков были вложены определенные инвестиции, а теперь потребуется ее переобучение, изменения затронут все процессы внедрения ПО и всю базу знаний, поддерживающую бизнес. Смена языка повлечет за собой переписывание больших компонентов системы, которое будет сложно обосновать.

Как вообще происходит выбор языка? Во многих компаниях, возглавляемых владельцами или основателями, основным языком обычно является тот, которым владел основатель или первый «технический директор». Они использовали его для написания всего, даже того, для чего он никогда не был предназначен. Это был не выбор языка, а отсутствие выбора.

В свою очередь фреймворк — это набор библиотек или стандартов для конкретного языка, который облегчает выполнение повторяющихся рутинных задач (например, обеспечения безопасности пользователя после авторизации или работа с файлами). Примеры фреймворка — Java Enterprise Standard или Spring для языка Java либо ReactJS для JavaScript и Express.js для Node.js. Зона ответственности фреймворков гораздо шире, чем у языков программирования, и поэтому они могут накладывать гораздо больше ограничений на способы реализации того или иного функционала. Разработчики, специализирующиеся на конкретном фреймворке, хотя и владеют базовым языком, могут столкнуться с трудностями при переходе на другой фреймворк для того же языка. Но знание фреймворка так же важно, как и знание языка.

Разные языки и фреймворки то становятся модными, то выходят из моды. Фреймворки создаются поверх языка, поэтому чаще именно они сменяют друг друга. Будет довольно неприятно, если компания вложила значительные средства в основной фреймворк, а через несколько лет он устарел. Тогда компания



столкнется с необходимостью поддержки как решения, так и фреймворка, на котором оно построено.

Языки тоже устаревают, хотя и гораздо медленнее. Чаще всего перестают использоваться языки, связанные с конкретной компанией. По возможности не используйте их. Если компания-разработчик будет приобретена или разорится, язык умрет. Так произошло с Gupta, PowerBuilder, Delphi и бесчисленным множеством других языков, появившихся на рубеже этого века.

Выбирая язык, важно ориентироваться на пятилетнюю перспективу. Вы собираетесь сделать большие вложения, нанимая разработчиков, создавая базу знаний, разрабатывая процессы внедрения ПО и стандарты кодирования — все это в надежде, что язык будет развиваться, чтобы обеспечить вам больше возможностей.

### 8.8.1. Устаревший язык

Если ваш язык или фреймворк перешел в стадию заморозки, постарайтесь как можно скорее его заменить. Чем его заменять — мы рассмотрим в следующем разделе. Но язык нельзя менять только потому, что вы с ним не знакомы. Смотрите шире и принимайте решение по совокупности факторов, включая то, достаточно ли на рынке труда специалистов по этой технологии. Если ваша команда разработчиков стареет, а молодая кровь не прибывает, значит, используемый вами язык умирает.

Замена языка потребует значительных усилий и ресурсов. Как уже говорилось, одна из эффективных стратегий здесь — определить области, которые легко преобразовать в микросервис. Это даст возможность вырезать куски кода из системы, а не полностью переписывать ее.

### 8.8.2. Выбор языка для перехода

Единственно верного ответа на вопрос, какой язык использовать, не существует. Это хорошо и плохо одновременно. Наша отрасль постоянно создает новые и лучшие технологии, и по сравнению с ними все существующее кажется хуже, чем оно есть на самом деле. Вы в любом случае будете испытывать сомнения покупателя, который, что бы ни купил, думает, что есть вариант получше, но если вы основываете свой выбор на результатах анализа, то он будет правильным для своего времени. Следующие вопросы помогут вам оценить языки/фреймворки:

- *Кто является его разработчиком?* Движущая сила, стоящая за языком, важна, особенно если это конкретная компания. Какая часть ее доходов зависит от успеха этого языка? Если условия для бизнеса изменятся, выживет ли она и что с ней произойдет при этом? Языки, созданные вокруг независимого комитета по стандартизации, обычно имеют больше шансов на долгожительство.

- *Как давно он существует?* Зрелость — важный фактор. Ваша цель не стать защитником языка, а обеспечить решение задач и стабильность для ваших клиентов. Старайтесь избегать языков, существующих менее двух лет, — за этот срок другие успеют пройти по всем граблям в языке и станет понятно, сможет ли он расти и расширяться.
- *Насколько активно тематическое сообщество?* Когда у разработчика возникает вопрос, он обращается к поисковику. Но если язык новый, данных о нем мало. Изучите, насколько активно интернет-сообщество пользователей данной технологии: сколько результатов поиска выдается по запросу, существуют ли активные группы пользователей, есть ли у языка живое и деятельное сообщество open source, сколько книг было опубликовано по этому языку?
- *Учат ли его в вузах?* Убедительным доказательством перспективности языка служит то, что его изучают в университетах. Вузы обязаны думать о долгосрочной перспективе, поскольку студенты должны получать знания, которые останутся актуальными еще несколько лет после выпуска. Как правило, языки, привязанные к одной компании, в университетах не изучаются.
- *Насколько широко эта технология представлена на рынке труда?* Представьте, что завтра вам нужно будет подобрать большую команду. Сможете? Просмотрите сайты вакансий, позвоните в агентства и оцените не только количество вакансий, в которых этот навык требуется, но и количество кандидатов, обладающих этим навыком. Если язык только начинает развиваться, вам придется значительно переплачивать узкому кругу экспертов.

Принять правильное решение бывает сложно; однако с последствиями ошибки вы столкнетесь очень быстро. Потратьте достаточно времени на анализ и выбирайте с умом. Не поддавайтесь соблазну выбрать технологию, которая использовалась для проверки концепции, потому что обычно она не слишком хорошо масштабируется при создании полноценного продакшена.

## ИТОГИ

- Решения, принимаемые в области технологий, имеют долгосрочные последствия.
- Важно выбрать решение, которое не привязывает вас к конкретному поставщику.
- Не всегда очевидно, когда лучше разрабатывать самим, а когда покупать.
- Продавцы попытаются убедить вас, что их решение избавит вас от всех проблем.
- Облачные сервисы избавились от детских болезней и достигли такой степени развития, что при правильном использовании способны обеспечить большую гибкость и экономию средств.
- При наличии специфических требований вы можете использовать собственное оборудование, но такая необходимость возникает все реже.

- Аварийное восстановление (без использования облака) намного сложнее и требует больше опыта, чем думают в большинстве компаний.
- Обдумайте стратегии переключения на резерв и отката назад в случае аварийной ситуации.
- Данные — один из ваших самых ценных активов. Определите, к каким классам они относятся.
- Классификация данных позволяет определить права собственности, а также жизненный цикл данных и требования к их защите.
- Микросервисы — это еще один способ разбить систему предприятия на управляемые небольшие части.
- Монолитная архитектура по-прежнему актуальна, особенно в экспериментальных проектах.
- ПО с открытым исходным кодом — отличный способ воспользоваться помощью сообщества; однако на возможность использования такого ПО сильно влияют лицензионные ограничения.
- Вы можете внести свой вклад в open source сообщество разными способами, особенно если вы хотите создать новый стандарт параллельно с разработкой коммерческого продукта.
- Языки и фреймворки влияют на подбор сотрудников и будущий рост компании, поэтому важно выбирать стек с расчетом на будущее.

## ПРОВЕРЬТЕ СЕБЯ

Что из перечисленного ниже вы уже сделали?

- Перестали полагаться на одного поставщика решений для ключевой части вашей системы.
- Составили список вопросов для принятия решения о разработке или покупке.
- Не совершаете ни одной из 10 ошибок при использовании облака.
- Если у вас есть собственное оборудование, размещаете его в специализированном центре обработки данных или, по крайней мере, в изолированной среде в офисе.
- Создали стратегию аварийного восстановления, в том числе определили, будете ли вы продолжать работать на резервном комплексе или откатитесь назад.
- Отрабатывали аварийное восстановление в последние 12 месяцев.
- Определили характеристики всех данных организации, включая правила безопасности, доступа и срок хранения.
- Ознакомились с типами лицензий ПО с открытым исходным кодом и рассказали команде о различиях между ними.
- Составили понятный список основных языков/фреймворков, включая стандарты их использования в продакшене.

# Разработка

## В этой главе

- ✓ Управление проектами и информирование бизнеса о ходе работы (как успешном, так и нет)
- ✓ Развитие методов разработки в условиях роста команды
- ✓ Управление качеством и тестирование для обеспечения беспроblemных релизов
- ✓ Как избежать привязки решений к требованиям одного клиента

Многие технические директора приходят из инженерной среды, в которой они занимались проектированием и разработкой ПО. В этой среде они развивались и делали карьеру, по мере того как брали на себя все более крупные проекты и большую ответственность. У разработчиков есть собственные подходы и способы оставить в работе свой след. Это и привлекает в нашей сфере деятельности — свобода творчества, выбора средств реализации каждой из задач. Но когда дело доходит до формирования команды и роста компании — приходится выходить за пределы этих представлений и мыслить шире, уже с коллективной, а не индивидуальной точки зрения. Решение, которое сегодня кажется изобретательным и крутым, в будущем придется с большими усилиями поддерживать и адаптировать к изменениям.

В этой главе основное внимание уделяется важным вопросам, на которые технический директор, прошедший путь от разработчика до руководителя, должен как можно скорее обратить внимание, потому что их решение заложит основы

для успешного дальнейшего развития проекта. Как и многие темы этой книги, каждый вопрос здесь представляет собой отдельную дисциплину. Техническому директору не обязательно в совершенстве владеть каждым из них, но он должен иметь представление о них и о том, какую роль они играют в работе команды. Вы должны систематизировать и привести к готовому для промышленной эксплуатации виду все: от управления проектами и информирования о ходе работ (чтобы бизнес мог положиться на вас и вашу команду в том, что продукт будет выпущен в срок) до профессионального и предсказуемого управления релизами.

Простые вещи, например создание единых стандартов написания кода, делают код более читабельным и таким образом упростят работу с ним. И каждый из таких небольших шагов в значительной степени влияет на вашу работу. Мы также рассмотрим, как создать стандартный процесс сборки продукта, чтобы выпуск каждого обновления не превращался в кошмар для бизнеса.

В целом эта глава посвящена созданию профессионального, стандартизированного и надежного процесса разработки, который позволит подключать новых сотрудников и получать от них первые результаты уже за несколько часов или дней, а не за недели, которые ушли бы у них на преодоление подводных камней и борьбу с устаревшими методами работы.

#### **ЗАМЕТКИ С ПОЛЕЙ**

*Продакшен, управляемый разработчиком*

Красивый и качественно сделанный сайт может скрывать под собой целый мир хаоса, заплаток и костылей. Один из аудитов *due diligence*, в которых я участвовал, напомнил мне, что ошибки при принятии простых решений в начале работы могут стоить миллионы (буквально) долларов в будущем. У SaaS-компании, о которой идет речь, был отличный продукт, но технический директор не установил никаких стандартов и не внедрил процессы выпуска релизов. Результатом стал полный бардак в коде и библиотеках. Релизами занимался только один человек (сам СТО). Не приведя работу команды к виду, необходимому для промышленной эксплуатации, технический директор создал такой большой технический долг, что стоимость исправлений оказалась равна стоимости компании. У этого технического директора просто не было знаний за пределами его опыта разработчика, необходимых, чтобы управлять растущей командой.

## **9.1. ПЛАНИРОВАНИЕ ПРОЕКТА**

Одна из самых сложных задач технического директора — сообщать бизнесу сроки. Работаете ли вы над крупной инициативой либо проектом или же над небольшой функцией, другие должны знать о ходе работы, чтобы иметь возможность планировать свои задачи, зависящие от ваших результатов. Неумение технарей придерживаться сроков печально известно. Мы с легкостью назначаем срок,

а потом его и близко не соблюдаем. Причина в том, что мы вечные оптимисты, которые верят, что вещи не так сложны, как на самом деле. В общем и целом мы правы; однако в своих прогнозах мы не учитываем внешние обстоятельства, которые вмешиваются и нарушают идеальный план (если бы клиенты постоянно не надоедали, мы бы совершенно точно все успели!).

Поэтому прогнозировать сложно. А учитывая внешние обстоятельства, новые данные и расползание функционала, порой это кажется невозможным. Однако это не обязательно должно быть так. Чтобы даже самый требовательный генеральный директор терпеливо ожидал результатов от вашей команды, вам понадобится планировать все немного более тщательно, а также изменить оптимистичный стиль общения с бизнесом.

### 9.1.1. Руководитель проектов

Организации бывают двух типов: те, в которых есть хороший руководитель проектов (проджект-менеджер), и те, в которых его еще нет. Причинами, по которым нет выделенного руководителя проектов, могут быть как недостаточный бюджет (уважительная), так и отсутствие необходимости в нем, потому что технический директор сам может управлять проектами (неуважительная). Но независимо от причины его отсутствия — руководитель проекта необходим всегда, когда у вас больше пяти технических специалистов. Если разработчиков меньше — в большинстве случаев они сами эффективно координируют свои действия.

Для целей этой книги предположим, что скрам-мастер и руководитель проектов выполняют одни и те же задачи. Хотя можно найти различия, принципиально эти роли очень схожи. Руководитель проектов помогает достичь ваших целей, управляя всеми ресурсами и сообщая их статус, планируя и поддерживая в актуальном виде сроки исходя из текущей ситуации. Вопреки распространенному убеждению, руководитель проектов не должен ставить задачи и назначать ответственных — это функция лидов. Если есть риск несоблюдения сроков, он оперативно сообщит вам об этом, а также о причинах задержки, что даст вам возможность принять меры, если нужно. Также он проводит периодические собрания и следит за тикетами, чтобы в реальном времени видеть состояние работы. Хороший проджект-менеджер поощряет открытый обмен информацией, независимо от того, хорошие новости или плохие, чтобы постоянно держать руку на пульсе проекта.

Для этой роли нужны врожденные способности организатора в сочетании с практичным взглядом на жизнь и методическим подходом к процессам. Мы, технари, обманываем себя, думая, что в сутках больше 24 часов, а проджект-менеджеры знают, что на самом деле в обычном рабочем дне всего шесть часов (с учетом перерывов, обедов, походов в туалет, размышлений о вечном и т. д.). Они знают, что тому, кто вернулся из отпуска, потребуется время, чтобы войти в рабочий ритм, тогда как мы верим, что он начнет работать сразу по возвращении из двухнедельного отпуска, в течение которого он лежал на пляже, полностью

отключившись от шума и суеты офиса.

Управление проектами — это не наука, а искусство, потому что необходимо учитывать постоянно меняющийся ландшафт, отфильтровывать лишнее и непрерывно корректировать путь к цели. Каждый, кому доводилось управлять дроном, вероятно, оценил его софт, который поправляет траекторию полета, когда порыв ветра сбивает его с курса. Кажущаяся легкость, с которой дрону удается сохранять свое положение, управляя воздушными винтами в реальном времени, впечатляет. Руководитель проектов подобен такому дрону в том, что он анализирует все входные данные и прогнозирует, что произойдет, если пойти по той или иной траектории, и затем сообщает эту информацию техническому директору для принятия корректирующих мер. Менеджер проекта поможет:



- Контролировать результаты работы каждого сотрудника.
- Определить основные этапы и цели.
- Запланировать сроки и регулярно корректировать их для повышения точности.
- Координировать релизный цикл.
- Управлять передачей промежуточных результатов в процессе работы, а также коммуникацией с внутренними/внешними командами.
- На основании данных о ходе проекта предлагать действия, направленные на более эффективное управление временем.
- Отчитываться о состоянии всего; следить, что ничего не упущено и не забыто.

В компаниях, где объем работы по управлению проектами невелик, эти сотрудники обычно также берут на себя роль бизнес-аналитика, который помогает детализировать задачи со стороны бизнеса. Однако руководитель проектов в любом случае в той или иной степени занимается такой детализацией, поскольку одна из его задач — разбить проект на понятные этапы.

Очень важно поддерживать с проджект-менеджером постоянную связь — например, ежедневно. Менеджер не должен бояться сообщать вам плохие новости, поэтому вам необходимо воздерживаться от непрофессиональной реакции на них или обвинений. Также он будет напоминать вам о том, что необходимо сделать, и будет вносить это в планы — например, про ежегодное техническое обслуживание, которое требует выделения части команды для выполнения обновлений. Если подобные события будут запланированы и ожидаемы, то команде не придется постоянно решать неожиданные проблемы и она сможет

работать как отлаженный механизм. А ваше общение с генеральным директором и правлением будет содержать гораздо меньше сюрпризов.

### **СИСТЕМА УПРАВЛЕНИЯ ЗАДАЧАМИ**

Чтобы эффективно управлять командой любого размера, важно всегда знать, над чем она работает сейчас и какая работа запланирована. Технари, как правило, любят доводить дело до конца — нам нравятся списки задач, и мы получаем удовлетворение, когда ставим галочку о том, что та или иная задача выполнена. Сопоставив эти условия, вы поймете, почему использование системы управления задачами (или тикетами) является лучшим способом управления в IT. Отраслевым стандартом здесь является Atlassian Jira, которая выводит работу с тикетами на совершенно новый уровень. Существует и множество альтернатив, но все они работают по одному и тому же принципу: в них есть тикеты с уникальными идентификаторами — это единицы работы, назначаемые конкретному исполнителю на конкретное время. У тикетов есть определенный жизненный цикл: от статуса «открыт» через промежуточные статусы к статусу «закрыт». Статус «закрыт» может означать, например, что задача была завершена или что она не была взята в работу, поскольку уже неактуальна. Статус «открыт» означает, что задача будет показываться исполнителю и ее необходимо выполнить.

Задачи могут быть сгруппированы в эпики, истории или проекты (о гибких (agile) подходах к разработке мы поговорим позже), а также могут включать подзадачи. Вы сами устанавливаете правила работы с тикетами, но, какими бы они ни были, их необходимо донести до команды, чтобы ни у кого не возникало вопросов.

Например, тикет должен представлять собой такую единицу работы, выполнение которой занимает не более определенного периода времени (пусть это будет четыре часа). Если требуется больше времени, то тикет следует разделить на подзадачи меньшего размера. У этой техники есть дополнительный бонус, заключающийся в возможности не тратить время понапрасну. Если человек начинает работать над задачей и в середине работы выясняется, что он не сможет ее завершить, то он может остановиться и попросить помощи. Возможно, он слишком усложняет работу над задачей, или же сотрудник, поставивший ее, недооценил ее сложность и ее нужно разбить на ряд более мелких задач. Выполнение тикета не должно занимать несколько дней или недель — это значительно усложняет управление проектом. Но какой бы уровень детализации вы ни выбрали, он должен подходить для вашей организации, и о нем необходимо сообщить другим. Если вы научите команду работать с тикетами, актуализировать приоритеты и планировать каждый день — это будет постоянно приносить дивиденды в будущем, поскольку все смогут видеть загрузку команды, а сотрудники смогут самостоятельно управлять своими списками задач.

Еще одно большое преимущество централизованной системы управления тикетами в том, что она устанавливает единый язык для обсуждения всего,



что связано с задачами. Я с огромным удовольствием слушаю, как команда перебрасывается номерами тикетов, обсуждая, что предстоит сделать, или когда кому-то нужна помощь. Номер задачи позволяет любому найти ее, посмотреть ее историю, прочитать комментарии, увидеть код, связанный с тикетом, и получить всю информацию о проблеме, чтобы оказать помощь максимально эффективно.

Система управления задачами, если ее правильно поддерживать, также может стать живой, самообновляющейся базой актуальных знаний, особенно если она используется в качестве системы поддержки клиентов. Для этого тот, кто закрывает тикет, должен подробнее расписать шаги для решения проблемы, и этой информацией смогут воспользоваться другие сотрудники.

Разработчики и инженеры не всегда следят за актуальным статусом задач. Здесь также на помощь приходит руководитель проектов, который следит за своевременным обновлением тикетов, а лучшие проджект-менеджеры показывают всем важность этого и таким образом приучают всех делать это без напоминаний. Подавляющее большинство своих отчетов и текущей работы руководитель проектов выполняет в системе управления задачами, поэтому в его интересах добиваться актуальности данных в его главном инструменте. Также в этой главе я покажу, как система управления задачами может помочь в планировании релизов.

### 9.1.2. Что считать проектом

Общее определение проекта ни у кого не вызывает возражений: это набор связанных друг с другом задач, выполнение которых приводит к желаемому результату. Разногласия возникают при обсуждении масштаба: насколько большая задача может считаться проектом? Проект может быть рассчитан как на несколько дней, так и на годы. Независимо от продолжительности, у проекта должна быть конкретная цель — основная причина, по которой он существует. Цель не говорит о том, какими способами и в какие сроки она должна быть достигнута. Она всего лишь определяет критерии успеха, выполнение которых означает, что проект завершен.

Цели проекта необходимо сформулировать языком, понятным не-разработчикам. Это обеспечит прозрачность и более эффективную коммуникацию, особенно с генеральным/финансовым директором, которым нужно знать, на что вы и ваша команда тратите время и деньги. Ниже приведены два разных описания целей одного и того же проекта:

- *Внутреннее описание.* Переделать процесс интеграции новых клиентов, убрав зависимость от Windows XP и реализовав его в виде Java-приложения, предназначенного для использования в бессерверной архитектуре.
- *Внешнее описание.* Модернизировать процесс подключения новых клиентов, переписав его для облака, чтобы повысить гибкость и скорость работы.

Определение результата проекта — это легкая часть. Теперь необходимо понять, как добиться этого результата. Для этого нужно разбить основную задачу на более мелкие, выполнение которых приведет нас к цели. В ИТ существует два основных подхода к реализации проектов. Традиционный подход называется каскадным, или же «водопадом» (waterfall), а целый набор более современных подходов — гибкими методологиями разработки, или agile.

### **КАСКАДНЫЙ ПОДХОД (ВОДОПАД)**

Эта модель предполагает максимальную степень планирования до начала работ. Идея здесь в том, что, как в водопаде, все начинается сверху и движется только вниз. Основной акцент делается на анализе предметной области, при котором выявляются все задачи и проблемы, которые необходимо решить. Затем принимаются основные решения, касающиеся архитектуры и деталей реализации. Каскадная модель ставит целью получить как можно больше данных, прежде чем выделять ресурсы для проекта.

На предварительное планирование уходит много времени, но чем четче определен проект — тем проще спрогнозировать для него сроки и стоимость работ. Кроме того, такие проекты благодаря своей продуманности отлично подходят для участия во внешних тендерах.

Однако то, что на детализацию и планирование проекта уходит много времени, является и одним из основных недостатков каскадного подхода, особенно если проект крупный, потому что, пока планирование не будет завершено, приступать к реализации нельзя. Представьте проект, цель которого — доехать из Майами в Сан-Франциско: команда не сможет начать движение, пока маршрут не будет проработан во всех деталях.

Еще более важный недостаток каскадной модели — ее неспособность работать с изменениями. Изменения появляются по самым разным причинам: какие-то технологии или требования могут потерять актуальность, в проект может добавиться новый функционал. Иногда в больших проектах такие проблемы просто игнорируются и все продолжают следовать плану, поскольку внедрить изменения чрезвычайно сложно. Случается, что реализованный проект уже не решает изначальную задачу и основан на устаревшей технологии (такое чаще встречается в государственных проектах или в акционерных компаниях, поскольку в этих сферах любые изменения требуют слишком большого количества согласований, которые блокируют работу).

При использовании каскадной методики создается большое количество предварительной документации, которая может включать макеты и прототипы и обычно разрабатывается совместно экспертами по предметной области, бизнес-аналитиками, архитекторами и руководителями проектов.

### **ГИБКИЙ ПОДХОД (AGILE)**

Если водопадная модель предполагает, что все детали должны быть известны до начала проекта, то на другом конце спектра находятся гибкие методологии, при

использовании которых проблемы решаются по мере их появления. (На самом деле это упрощение — бывает, что только отдельные проблемы не исследуются или не решаются до начала разработки.) *Гибкий* (agile) — значит способный адаптироваться к изменяющейся среде, поэтапно приближающийся к конечному результату, быстро и успешно реагирующий на изменения. Например, вернемся к нашему проекту поездки в Сан-Франциско, на этот раз используя гибкую методологию: поняв, как ехать по штату Флорида, мы разрешаем команде двигаться к его границе, одновременно с этим планируя следующие шаги. Гибкие методологии — это набор руководящих принципов, обеспечивающих непрерывное внедрение результатов работы и вовлечение в процесс всех заинтересованных сторон, включая конечного клиента или пользователя, для обеспечения высокого качества, поскольку при этом проблемы выявляются и устраняются гораздо быстрее.

Многие считают, что гибкий подход — это просто отсутствие планирования, что абсолютно неверно. Такие проекты точно так же требуют планирования, просто не такого подробного, как при каскадном подходе. Для них определяются промежуточные цели (или, в agile-терминологии, *эпики*), они затем превращаются в набор задач, которые можно брать в работу. Затем эти задачи объединяются в ограниченные по размеру периоды работы, или спринты. Обычно они продолжаются 1–2 недели, и после каждого спринта проводится разбор итогов. Как правило, по результатам каждого спринта какая-то часть функционала должна быть уже готова к использованию, чтобы другие по возможности могли начинать работать с ней.

Agile может быть ориентирован в первую очередь на процессы, что поначалу может пугать. Но это скорее набор руководящих принципов, чем строгих правил, которым необходимо обязательно следовать, — то есть структура, обеспечивающая необходимый уровень порядка и последовательность действий там, где это нужно. Некоторые организации настолько заикливаются на правилах, что работа лишается всякого удовольствия и творческого подхода. Секрет успеха гибкой методологии состоит в том, чтобы найти баланс, подходящий именно для вашей команды. В качестве одного из примеров гибких подходов к разработке можно привести методологию скрам (scrum).

Гибкий подход значительно упрощает внесение изменений, поскольку позволяет увидеть, работает ли решение, до того, как на него будет потрачено слишком много ресурсов. Способность адаптироваться — вот ключ к успеху agile. Для того чтобы внедрить ту или иную методологию, сначала определите и формализуйте области, в которых не потребуются менять привычный вам порядок работы, а затем постепенно адаптируйте остальные процессы, позволяя команде привыкнуть к новой организации работы по мере того, как она начинает приносить плоды. Более подробный обзор гибкой и каскадной методологий можно найти здесь: <https://www.guru99.com/waterfall-vs-agile.html>.

## 9.2. СТАНДАРТЫ РАЗРАБОТКИ

Хотя велик соблазн просто позволить командам разработчиков работать как им удобно, без стандартов и правил, это создаст проблемы в будущем по мере роста проекта. Вы не захотите иметь дело с большими блоками кода, от которых зависит успешная работа всего бизнеса, но которые при этом до такой степени сложные и хрупкие, что только горстка людей осмеливается вносить в них изменения. В каждой компании есть свои «священные блоки кода», обросшие мифами из-за их легендарной сложности, и когда кто-нибудь предлагает их переписать — ответом ему бывают долгие, тяжкие вздохи, а старожилы команды качают головами и советуют вообще не трогать этот код, не то что пытаться его переписать.

Как уже неоднократно отмечалось, технический руководитель должен не только решать текущие проблемы, но и думать о будущем. Разрабатываемые сейчас системы будут нуждаться в поддержке и расширении еще долгие годы, когда вы уже, возможно, покинете свою должность.

Стандарты помогают разработать и продвигать общий подход к тому, как писать и структурировать код. Добиться читабельного и легко расширяемого кода непросто, в том числе потому, что многие разработчики считают, что их код неприкосновенен и не должен никем проверяться. Если вы столкнетесь с таким подходом — искореняйте его, а если люди не хотят меняться — придется с ними поподробнее.

Программирование является в большей степени творчеством, чем строгой наукой, и разработчик всегда выбирает какой-то вариант реализации из множества возможных. При этом то, что один считает логичным и естественным, для другого выглядит надуманным и неуклюжим. У программистов есть дурная привычка: если они чего-то не понимают, они просто полностью переписывают это. Слышали поговорку «Из огня да в полымя»? Поэтому всем выгодно, чтобы код было легко прочитать и понять и чтобы в нем использовались одни и те же подходы и паттерны.

Хорошая новость в том, что эта проблема давно известна. Для всех основных языков программирования существуют стандарты, поэтому вам не нужно будет изобретать велосипед. Соблюдение базовых стандартов легко обеспечить с использованием сторонних инструментов, совместимых с большинством IDE. Также существуют программы, которые можно использовать в процессе сборки приложения для автоматического анализа кода перед деплоем. SonarQube — популярный инструмент с открытым кодом, поддерживающий все распространенные языки, который находит в том числе неиспользуемые переменные, работу с неинициализированными указателями, логические тупики (области, из которых нет выхода), а также многие другие проблемы.

Стандарты кодирования касаются не только форматирования кода в текстовом файле. Они могут определять общие шаблоны проектирования — например, как взаимодействовать с базами данных или регистрировать события в журнале.

Что еще более важно, они должны помогать структурировать код, определять размер функций, иерархию классов, вплоть до правил выбора названий.

Стандарты можно создавать и внедрять постепенно, шаг за шагом. Необходимо организовать постоянную проверку кода (review), и никто не должен исходить из того, что его код безупречен, включая и вас, если вы, будучи техническим директором, все еще программируете.

Код должно быть легко читать и поддерживать, и, что самое главное, в нем должно быть как можно меньше ошибок. Стандарты и ревью кода помогут вам добиться этого.

## 9.3. СИСТЕМЫ КОНТРОЛЯ ВЕРСИЙ

Система контроля версий (version control system) — это централизованное хранилище и одновременно машина времени для всех ваших цифровых активов. Когда-то контроль версий использовался только большими командами программистов, но сейчас процесс разработки ПО уже невозможно представить без него, и он даже проникает в другие области, такие как Google Docs, который сохраняет историю всех изменений в ваших документах.

Системы контроля версий за годы эволюционировали от CVS до SVN и современной системы Git, наиболее популярной среди разработчиков (а сервис GitHub является одним из крупнейших репозиторий ПО с открытым исходным кодом). Тем не менее основной принцип остается неизменным: обеспечить возможность параллельной разработки как одним, так и несколькими разработчиками, чтобы для этого им не приходилось настраивать отдельные окружения для работы с разными версиями проекта. Представьте, что вы находитесь в середине масштабной разработки новой версии проекта и вам понадобилось исправить ошибку в продакшене. Для этого вам нужен будет исходный код версии, которая работает в продакшене (не содержащий никаких изменений из новой версии), чтобы надежно исправить ошибку и сделать релиз, не опасаясь, что новый код попадет наружу.

Некоторые команды не используют контроль версий. Иногда они хранят старые версии в отдельных папках или в каких-нибудь файлах в формате zip. Предлоги не использовать контроль версий бывают самыми разными — что это просто не нужно, не поддерживается для их языка программирования, замедляет работу, — но все это неправда. Основная проблема в том, что они просто боятся нового.

На верхнем уровне система контроля версий представляет собой набор отдельных репозиторий. Их можно рассматривать как простые папки с файлами, но это неправильно, потому что их возможности несравненно больше. Это скорее компоненты, блоки, из которых строится ваш проект. Каждый репозиторий полон жизни и позволяет вам путешествовать во времени, это настоящая мультивселенная, в которой одновременно доступны все ваши действия, все решения, вся эволюция каждой из частей за всю историю проекта.

**ЗАМЕТКИ С ПОЛЕЙ***Цена отсутствия контроля версий*

Еще в начале своей карьеры я работал в Риме для Организации Объединенных Наций — мы писали Java-сервлеты для обработки спутниковых фотографий сельскохозяйственных земель в Африке. У них уже были CGI-скрипты, разработанные на Perl, чтобы ученые могли получать доступ к фотографиям в браузере Mozilla, и нас пригласили попробовать новый стандарт Java Servlet и посмотреть, как он будет работать при реальной нагрузке. Я провел там уже несколько месяцев, и в одну из недель усердно трудился над новым модулем, чтобы закончить его до того, как я сяду в самолет и отправлюсь домой на несколько дней. 25 с лишним лет назад у нас не было стандартов или процессов работы с кодом, не говоря уже о контроле версий. По ходу работы я сам сохранял резервные копии в отдельном сетевом каталоге (и думал, что этого более чем достаточно). Я периодически тестировал, что получается, и, в общем, один из процессов в нашем приложении удалял временные файлы (в те дни место на диске было на вес золота). Однако в спешке я что-то напутал, и он удалил все исходные файлы как локально, так и с моего резервного сетевого диска. Восстановить их я не смог, как ни пытался. Я своими руками удалил результат своего же недельного труда, и другого выхода не было, кроме как остаться на выходные и написать все заново. Эта неделя стала для меня настолько поучительной, что с тех пор подобное не повторялось.

Выбор того, что именно выносить в отдельные репозитории, зависит от множества факторов, и универсального ответа тут не существует. Следующие вопросы помогут понять, что может быть отдельным репозиторием:

- Есть ли у этого компонента отдельная функция или роль?
- Требуется ли он отдельным специалистам с особым набором навыков?
- Насколько он связан с другими компонентами?
- Как делаются релизы этого компонента и всего проекта?

Старайтесь избегать слишком большого количества репозиторий, но и не стоит использовать только один, в котором лежит вообще все. Если репозиторий много, то накладные расходы, связанные с ними, оказываются слишком велики, а если их мало, становится сложно разобраться в отдельных коммитах. Это классическая дилемма Златовласки: «Не слишком горячо и не слишком холодно»<sup>1</sup>.

Подобно тому, как файлы и папки никак не влияют на их содержимое, вложенность и детализацию данных, так же и сама по себе система контроля версий не определяет какой-либо стратегии использования — это просто подход и набор

<sup>1</sup> Дилемма Златовласки названа по аналогии с детской сказкой «Три медведя» (в русском варианте «Маша и медведи»), в которой девочка по имени Златовласка (Маша) пробует три разные миски с кашей и выбирает ту, которая не слишком горячая и не слишком холодная. В психологии обозначает проблему выбора оптимального варианта из имеющихся. — *Примеч. пер.*

инструментов. Существует, однако, ряд общепринятых и всем понятных методик, и вы можете выбрать ту, которая лучше всего подходит для вашей команды и проекта. Вот пара примеров:

- *Gitflow* — использует две основные ветки: master (код в ней всегда готов к деплою) и develop. Для разработки каждой новой функции вы создаете из develop новую ветку (feature branch), изменения из которой после проверки вносятся обратно в develop. Чтобы сделать релиз, вы создаете из ветки develop новую ветку (release branch) — это то, что можно деплоить. Все изменения, попавшие в ветку релиза, надо сменить в мастер. Никакие изменения не должны вноситься напрямую ни в master, ни в develop.
- *GitHub flow* — более простая версия Gitflow, из основных веток в ней используется только master. Для новых функций создаются новые ветки, после ревью и тестирования изменения из них вносятся в master, и их можно деплоить.

При использовании любой методики вы можете сами определить способ интеграции с тикетной системой, правила именования веток, допустимое количество изменений или время жизни для веток, релизный цикл и т. д. Все это нужно выбирать исходя из требований вашего проекта, чтобы вспомогательные инструменты и отчеты, которые вы будете создавать для вашей системы контроля версий, были максимально эффективны.

## 9.4. ОБЕСПЕЧЕНИЕ КАЧЕСТВА (QA)

Обеспечение качества (QA), или тестирование, как и все темы, рассматриваемые в данной главе, является целой отдельной дисциплиной. Цель QA очень проста: проверить, будет ли то, что вы только что разработали или внедрили, делать то, чего вы от него ожидаете?

Тестирование — сложная область, возможно, это одно из самых сложных направлений в вашем отделе. Предсказать все варианты использования и действия, с которыми придется столкнуться вашему ПО, почти невозможно. Тесты как деньги: неважно, сколько их у вас уже есть, вам всегда нужно немного больше.

Одно совершенно точно: мы, инженеры, плохие тестировщики, особенно если речь идет про результаты нашей собственной работы. Поэтому всегда рекомендуется, чтобы кто-то (или что-то) со стороны проводил независимое тестирование. Выделите отдельную команду контроля качества, не имеющую отношения к разработке, чтобы они проверяли исправления и новый функционал с точки зрения конечного пользователя. Скорее всего, вам понадобится внедрить тестирование следующих двух видов:

- ручное;
- автоматизированное.

### 9.4.1. Ручное тестирование

Как следует из названия, ручное тестирование предполагает участие человека, который подтверждает успешное прохождение одного или более тестов. Обычно оно проводится, когда автоматизированные тесты для каких-то сценариев слишком сложны в реализации или ненадежны.

Как правило, есть выделенная команда QA, которая очень хорошо знает продукт и разработала набор планов тестирования как минимум для областей, которые используются чаще всего. Хорошей практикой является создание плана и методологии тестирования, чтобы любой мог выполнить тесты и не возникало ситуации, когда только один человек знает, как правильно тестировать код.

Тестирование может занимать очень много времени, и, в зависимости от тестируемой области, быть довольно утомительным. Это может приводить к тому, что, например, если какие-то тесты всегда проходят успешно, в очередной раз тестирующий может пропустить их, предположив, что все работает. Это создает риски для проекта, но отслеживать подобное очень сложно.

К участию в тестировании можно подключать новых разработчиков, прежде чем давать им задачи по написанию кода. Так они не только познакомятся с продуктом, но и получат представление о том, как организуется тестирование проекта и что необходимо для обеспечения высокого качества.

#### ЗАМЕТКИ С ПОЛЕЙ

##### *Тестирование тестирующих*

Я знаю нескольких технических директоров, которые намеренно дают новичкам в команде QA вместо реального тестирования заведомо проблемные компоненты, чтобы посмотреть, найдут ли они эти проблемы. Это может быть что угодно, от очевидно нерабочего функционала до несоответствия стандартам (касающимся процессов или внешнего вида и поведения). На первый взгляд это выглядит нечестным, но таким образом новички лучше понимают требования, а руководство может быть уверено, что и люди, и процессы надежно работают.

### 9.4.2. Автоматизированное тестирование

Мечта любого технического директора, инженера или владельца продукта — добиться полного покрытия кода автоматизированными тестами. Автотесты работают без участия человека, и их можно выполнять как часть процесса сборки и деплоя приложения. Они могут включать в себя модульные тесты (unit tests), тестирование API, тестирование функционала для конечных пользователей — все, что последовательно проверяет кодовую базу без участия человека.

Сложность создания и поддержки автоматизированных тестов состоит в том, что для правильного выполнения бизнес-логики требуется разработка соответствующих сценариев и тестовых данных. Например, вам нужно протестировать



удаление учетной записи пользователя. Часть теста заключается в том, чтобы выполнить удаление учетной записи и убедиться, что она удалась. Ничего сложного? Так и есть, но, чтобы иметь возможность удалить учетную запись, сперва ее нужно создать. Вот такой порочный круг — добро пожаловать в тестирование.

### ЗАМЕТКИ С ПОЛЕЙ

*Не ведитесь на количество*

Глен Мартин (Glen Martin), который в свое время был менеджером по продукту спецификации Java Enterprise, преподавал мне ценный урок о важности качества, а не количества тестов. Он проиллюстрировал это утверждение, спросив меня, какой автомобиль я выберу: тот, который прошел 10 000 тестов, или тот, который прошел только один тест? Естественно, как молодой наивный инженер, я выбрал первый вариант. И попал в классическую ловушку, предположив, что количество равняется качеству. Глен ответил, что автомобиль, прошедший 10 000 тестов, не прошел тест запуска двигателя, а единственным тестом, который прошел другой автомобиль, был именно этот тест. Простой и очевидный пример, который я помню спустя почти 25 лет.

Покрытие кода — важная метрика, которую следует учитывать при оценке качества тестов. Она показывает, какая часть исходного кода фактически выполняется при прохождении теста. Но не весь код одинаков — какие-то фрагменты кода гораздо важнее других, особенно те, которые выполняются чаще или предоставляют критически важные сервисы.

Поэтому при оценке покрытия кода необходимо определить наиболее важные компоненты. Для них необходимо обеспечить максимальное тестирование. 100 % покрытия кода достичь почти невозможно; хорошим результатом можно считать 50–70 %.

Автоматизация исключает из ручного тестирования повторяющиеся действия, помогая избежать утомления, при котором пропускаются важные шаги. Разработка системы автотестов требует большого количества работы; кроме того, ее необходимо поддерживать в актуальном состоянии. Добавление каждой новой функции может привести к невыполнению тестов, которые приходится обновлять.

Если сроки проекта поджимают, чаще всего пренебрегают тестами. Может возникнуть ситуация, когда сборка приложения блокируется невыполняющимися автотестами. Вы убеждаете себя, что тесты не проходят из-за новой бизнес-логики и никаких ошибок нет. Таким образом, вместо того чтобы доработать тесты, вы выключаете их, что позволяет успешно выполнить сборку и уложиться в срок.

Время летит, дни и недели проходят, а автотесты так и остаются выключенными и неактуальными. Мы всё это проходили — еще один пример технического долга в системе. Использование автоматизированного тестирования требует

большой дисциплины, но оно того стоит, а актуальный набор тестов всегда будет приносить вам дивиденды.

## 9.5. НЕПРЕРЫВНАЯ ИНТЕГРАЦИЯ И ВНЕДРЕНИЕ (CI/CD)

Создание и внедрение программного обеспечения — сложное дело, и не верьте тем, кто утверждает обратное. Команды растут (и становятся более рассредоточенными), все больше рук одновременно работает с исходным кодом, люди приходят и уходят. Как при таком уровне сложности и текучки в команде организовать инфраструктуру таким образом, чтобы гарантировать, что каждый сотрудник при необходимости сможет собрать готовое приложение? Ответ — с помощью пайплайнов непрерывной интеграции и внедрения (CI/CD).

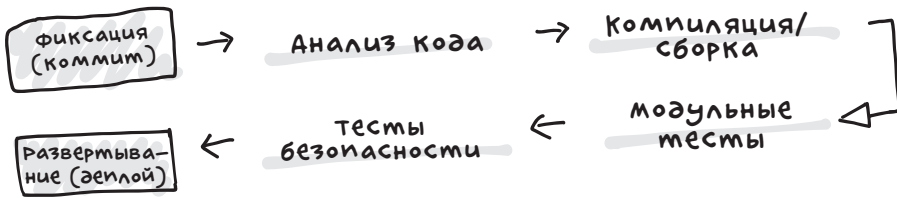
### ЗАМЕТКИ С ПОЛЕЙ

*Только Фред может обновить эту службу*

Если для компиляции, сборки и деплоя кода на продакшен вы используете только один конкретный компьютер (или одного человека), то вы попадаете в зависимость от них, что является проблемой. В подобную ситуацию попасть совсем несложно, и, поскольку все годами работает нормально, она не кажется чем-то серьезным и требующим немедленного решения. В портфельных компаниях, возглавляемых их основателями, я часто вижу, что за обновления отвечает один человек. Помню случай, когда ответственный сотрудник обновил библиотеки.NET на компьютере, после чего проект перестал компилироваться. Это вызвало общую панику, потому что теперь компания не могла обновить свой основной продукт. В качестве краткосрочного решения мы настроили для них виртуальную среду, что позволило сделать процесс сборки более управляемым и предсказуемым.

Как технический директор, вы несете ответственность за то, чтобы обеспечить возможность обслуживания проекта в долгосрочной перспективе, что означает возможность надежно компилировать, обновлять и деплоить ваше ПО в любое время, без необходимости участия в процессе каких-то конкретных людей. К сожалению, во множестве мелких организаций возможность выпускать релизы зависит от того, не находится ли тот, кто умеет это делать, в отпуске.

Философия CI/CD, как следует из названия, заключается в том, что как только код прошел ревью, одобрен и интегрирован в общую ветку, он автоматически проходит этапы компиляции и тестирования, а затем развертывается в указанном окружении. Об ошибке в любой части этого процесса немедленно уведомляются ответственные, с тем чтобы эти ошибки устранялись. Пример типичного пайплайна CI/CD показан ниже.



Он запускается автоматически, когда кто-то делает коммит в определенную ветку в репозитории, обычно это `develop`, после проверки и мержа пулреквеста (pull request merge). Это запускает следующую последовательность действий, или шлюзов (gates), через которые проходит процесс деплоя и каждый из которых должен завершиться успешно:

- *Фиксация* (или коммит, от англ. commit). Событие, запускающее пайплайн, обычно это успешный коммит/слияние кода с заданной веткой.
- *Анализ кода*. Исходный код в ветке проходит серию проверок с использованием таких инструментов, как SonarQube.
- *Компиляция/сборка*. Затем код компилируется (если нужно) и создается артефакт, готовый к деплою.
- *Модульные тесты*. После сборки код проходит модульные тесты, проверяющие, что в кодовой базе не появилось ошибок.
- *Тесты безопасности*. Любые тесты или проверки для обеспечения безопасности.
- *Развертывание (деплой)*. Наконец, код деплоится на серверы, например на серверы для разработки или для тестирования. В зависимости от ветки это может быть и деплой на продакшен.

Разумеется, вы можете добавить в пайплайн столько этапов, сколько необходимо для вашего проекта. Это мощный инструмент автоматизации, который гарантирует, что ни один шаг не будет пропущен и что каждый компонент или библиотека могут быть собраны не только на настольном компьютере разработчика. Таким образом, вам не придется выслушивать классические оправдания: «Ну, на моем компьютере все работает».

Настроить пайплайн CI/CD уже не так сложно, как раньше. Такие инструменты, как GitHub, Bitbucket и CodeCommit, в той или иной форме поддерживают пайплайны, а если вам требуется решать более сложные задачи, вы можете внедрить инструмент оркестрации, например популярный продукт с открытым исходным кодом Jenkins, позволяющий управлять всеми пайплайнами для многих репозиторий и сервисов.

По мере того как проект будет развиваться и вы будете привыкать к использованию пайплайнов, вы будете стараться встраивать больше проверок. Если для вас это в новинку, не мудрите и не переусердствуйте — простая схема *коммит* > *сборка* > *деплой* отлично работает во множестве проектов.

Если что-то пойдет не так и выполнение одного из этапов завершится ошибкой, вы будете точно знать, кто автор проблемы, благодаря истории изменений

в системе контроля версий. Но используйте это не для того, чтобы обвинять кого-то, а для того, чтобы создать открытую среду, в которой не будет избранных, чьи действия не обсуждаются, и все работают сообща, чтобы обеспечить самое высокое качество вашего ПО.

Надежный пайплайн CI/CD поддержит быстрый рост вашего отдела и обеспечит уверенность в том, что все сотрудники будут действовать правильным и стандартным способом. Вот типовые проблемы, которые могут приводить к ошибкам сборки:

- Ошибка компиляции.
- Отсутствие библиотеки, от которой зависит проект.
- Нарушение требований/стандартов.
- Ошибки при прохождении тестов.
- Не выполнены требования безопасности (в коде обнаружены пароли или ключи доступа).
- Несоответствие структуры базы данных.

Чем больше у вас тестов, пусть и мелких, тем больше уверенность в качестве релиза и тем меньше риск. Надежный набор автоматических проверок способствует продвижению философии «выпускайте понемногу и часто». Вам решать, насколько часто вы хотите делать релизы, но какой смысл задерживать выкладывание исправлений, особенно если ваша инфраструктура позволяет делать релизы без прерывания сервиса (например, вы используете бессерверную среду)?

Некоторые команды делают несколько релизов в неделю, некоторые — один в день, а некоторые — много раз в день (одна крупная компания из списка Fortune 100 выпускает тысячи релизов в день в разных частях своей инфраструктуры). При правильной настройке пайплайнов никаких ограничений здесь нет, особенно если вы используете облачные сервисы, в которых все, что нужно для сборки проекта, запускается по вашему запросу.

## 9.6. ТЕХНИЧЕСКИЙ ДОЛГ

Представьте себе автомобиль, который никто не обслуживает (не меняет масло, фильтры и шины). Конечно, в краткосрочной перспективе это экономит немного денег, но каждое пропущенное ТО увеличивает «долг» владельца перед автомобилем. В какой-то момент этот долг может превысить стоимость самого автомобиля, а затем в один прекрасный день посреди дороги у него заклинит двигатель или же лопнет колесо, последствия чего обойдутся еще дороже. Таким образом, дешевле исправлять мелочи, чтобы избежать крупных, неожиданных аварий или затрат.

То же самое происходит и в информационной системе предприятия, только вместо автомобиля здесь исходный код и отдельные компоненты. Каждый раз, когда для какой-либо проблемы используется обходное решение (прямо

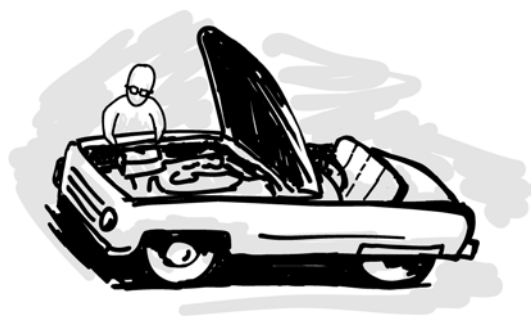
в коде указывается фиксированное значение, не выполняется проверка параметров функции, игнорируется необходимость обновления или применения патча безопасности), вы увеличиваете технический долг.

Технический долг возникает естественным образом, если ваша система развивается быстро. Это нормально — и способствует более быстрому выполнению задач. Однако к такому долгу необходимо относиться так же, как к кредитной карте: деньги с нее тратить можно, но в конце месяца обязательно погасите задолженность.

Сколько времени стоит тратить на устранение технического долга? Это во многом зависит от вашей компании, но, как правило, от 10 до 30 % рабочего времени вашей команды посвящается избавлению от долгов.

Измерить количество проблем в исходном коде можно довольно просто с помощью инструментов контроля качества, включенных в ваш пайплайн (SonarQube, опять же, хороший пример). Если вы замечаете проблемы во время ревью кода, а также во время релизов и обновлений, помечайте их и добавляйте в бэклог. Например, инструмент анализа мог не указать на какой-то фрагмент кода, потому что синтаксических проблем в нем нет, но вы знаете, что это было временное исправление, которое необходимо обязательно переделать, чтобы оно соответствовало общим требованиям.

Хороший СТО всегда знает уровень технического долга в своем проекте. Полезный совет, особенно если вы используете спринты, — через каждые 2–3 спринта выделяйте некоторое время для работ по устранению технического долга.



## 9.7. РЕЛИЗ

Без результатов ваша работа не имеет никакого значения. Выпуск релизов вашего продукта или обновление всей активно используемой корпоративной платформы может стать безболезненным и будничным делом, но, если вы хотите, чтобы это не требовало усилий, — вам придется как следует поработать.

Прежде обновление программного обеспечения было целым отдельным проектом — для него планировался технологический перерыв, обычно в нерабочие часы, направлялись уведомления клиентам, готовился порядок действий, которые затем выполнялись в надежде, что все пройдет по плану, потому что об откате изменений не хотелось даже думать. Сравните с современной архитектурой, в которой обновления при необходимости выполняются несколько раз в день без прерывания сервиса.

В этом разделе мы рассмотрим составляющие грамотной стратегии выпуска релизов, а также что необходимо подготовить и какие коммуникации наладить для ее успешного выполнения. «Выпускать понемногу и часто» — это не просто красивые слова, это достижимая цель.

### 9.7.1. Релиз с прерыванием сервиса

Первый тип релизов выполняется с простом: для него необходимо отключить систему. Это может быть связано с необходимостью перезапуска каких-то компонентов либо с необходимостью выполнить миграцию данных для перехода к новой версии. Такие релизы требуют больше усилий и предварительного планирования. Процедура их проведения может включать следующее:

- Определите, что нужно вывести в офлайн: серверы, компоненты.
- Что будут видеть конечные клиенты: нужно ли создавать специальную страницу с информацией о работах?
- Рассчитайте следующие сроки:
  - время на выключение сервиса;
  - время на резервное копирование;
  - время на обновление программного и аппаратного обеспечения;
  - время на проверку, что все работает;
  - время на приведение системы в доступное для пользователей состояние.
- Для каждого из этих этапов распишите точный порядок действий.
- Для каждого этапа назначьте ресурсы и определите процесс перехода к следующему этапу.
- Определите совместно с бизнесом наиболее удобное время проведения релиза, желательно в период наименьшего использования системы.
- Составьте список тех, кому необходимо предоставлять отчеты о ходе работы, и тех, кого необходимо уведомить о завершении релиза.

Теперь у вас есть план выполнения релиза. Предусмотрите запас времени на случай возникновения проблем. Например, если выключение сервиса на все выходные не является проблемой для бизнеса и при этом вы ожидаете, что вам понадобится всего 2–4 часа, все равно планируйте выключение на все выходные. Если вы справитесь раньше — отлично, но, если что-то пойдет не так, у вас будет время решить проблему без необходимости дополнительно это согласовывать.

При проведении релиза необходимо информировать всех заинтересованных лиц о выполнении каждого этапа. Не скрывайте плохие новости — если все затянется или пойдет не по плану, лучше сказать об этом, чем устраивать радиомолчание и заставлять людей думать, что процессом никто не управляет.

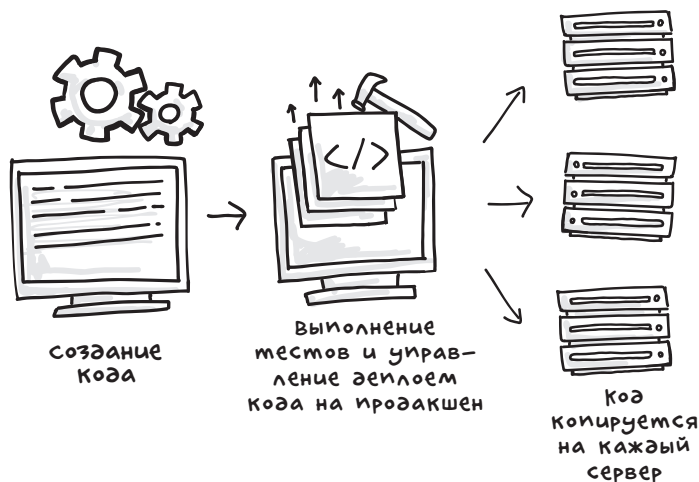
В идеале, особенно если вы делаете релизы часто, найдите время для тестовых прогонов. Это не всегда возможно, в зависимости от того, какая система требует

обновления. Например, миграция бэк-офисной системы (такой как электронная почта) с одной платформы на другую занимает много времени, и проверить ее в полном объеме может быть сложно.

После завершения релиза проведите ретроспективу, чтобы отметить, что прошло хорошо, и изучить, что пошло не так или что можно сделать лучше в следующий раз.

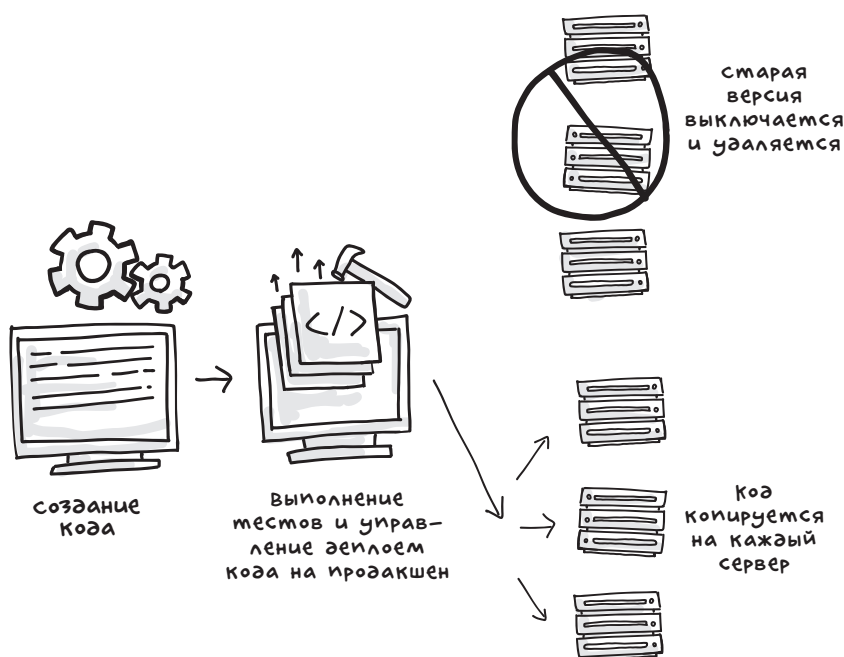
### 9.7.2. Сине-зеленое (blue-green) внедрение

Традиционно, когда кто-то думает об обновлении, он представляет себе замену кода на сервере (то есть буквальную замену файлов в файловой системе). Это сопряжено с большим риском и, конечно же, усложняет откат к предыдущему состоянию, потому что в файловую систему были внесены изменения.



Появление виртуализации, благодаря которой запуск новых серверов стал занимать минуты или даже секунды, открыло возможности для радикального переосмысления многих подходов к работе, и, в частности, к выпуску релизов. Вместо обновления текущих серверов вы можете создать совершенно новую копию с новой версией ПО, полностью настроенную и готовую к работе. После развертывания (если вы используете Docker или что-то подобное, можно создать скрипты для его ускорения) вы можете проверить работу новой версии, а «текущий» набор серверов пока продолжит обслуживать запросы от пользователей.

Убедившись, что все в порядке, вы можете начать направлять трафик на новую версию, и, когда он будет переведен полностью, старую версию можно будет удалить, так что продолжит работать только новая версия. Если же что-то пойдет не так, вы можете удалить новую версию и начать сначала — пользователи при этом даже ничего не заметят.



Такой способ разворачивания называется сине-зеленым (blue-green), и если вы его освоите, то сможете проводить релизы спокойно и безболезненно. Дополнительным преимуществом этого способа является постоянная проверка скриптов для разворачивания инфраструктуры. Чем сложнее ваша архитектура, тем сложнее в реализации будет этот способ деплоя, однако ничего невозможного тут нет. И он идеально подходит для веб-приложений.

Одна из самых сложных вещей в любом релизе — изменение схемы базы данных. Правило здесь простое: только добавляйте, никогда не удаляйте. Удаление делает невозможным откатить обновления, не прибегнув к восстановлению из резервных копий. Вы сможете удалить таблицы, столбцы и индексы позже в будущих релизах, когда будете уверены, что этот функционал работает нормально и его не придется откатывать.

Когда вы добьетесь того, что релизы будут хорошо отработаны и будут проходить безболезненно, вы избавитесь от стресса и рисков, связанных с обновлениями, и они перестанут быть значительным событием. Независимо от того, какую стратегию вы выберете, всегда создавайте описание релиза (release notes), в котором описывайте все изменения, — это очень полезная привычка. Это может быть просто текстовый файл, документ в вики или же страница в Atlassian Confluence. Некоторые системы управления задачами позволяют привязывать тикеты или спринты к версиям ПО и автоматически создавать описание релизов.



Это очень полезный документ, особенно если конечный пользователь должен понимать, какие обновления были выпущены. В этом случае описание релиза можно изложить языком, понятным для конечного пользователя.

## 9.8. ЗАПРОСЫ ОТ КЛИЕНТОВ

При управлении системой крупного и успешного предприятия вы обязательно будете со всех сторон получать запросы на добавление новых функций. Более важные клиенты, скорее всего, будут в приоритете, и именно они, если вы не справитесь, станут вашим худшим кошмаром. Бизнесу необходимо, чтобы его ключевые клиенты были довольны, и, естественно, он будет стремиться удовлетворить их потребности, чтобы оставаться на плаву.

Поэтому, если вы будете выполнять все такие запросы, то можете обнаружить, что согласуете расходы, которые делаются в интересах только одного конкретного клиента, что создает дополнительную нагрузку и в конечном итоге привязывает вашу платформу к этому клиенту. Что же делать современному техническому директору, который не хочет все время говорить «нет»? Отказы с вашей стороны не добавят бизнесу (или клиентам) любви к вам. Лучше рассматривайте каждый запрос на новый функционал в контексте всех ваших клиентов. Учитывайте следующее:

- Будет ли новая функция представлять ценность для других клиентов? Если да, то она в любом случае должна быть в вашем плане.
- Можно ли дополнительно проработать или обсудить эту функцию, чтобы она стала полезной для более широкого круга пользователей?
- Если эта конкретная функция не имеет ценности для остальных клиентов, есть ли другой способ решить проблему клиента?

Например, технология экспорта/импорта данных позволит клиенту создать внешнее решение, также он может использовать ваше API, если оно есть. Если ваша система имеет возможность настройки для отдельных клиентов — добавьте в нее поддержку плагинов, чтобы ее функционал можно было расширять, не затрагивая основные компоненты. Главное — не позволяйте себе оказаться во всем знакомой ситуации:

```
if (Client == XYZ) {  
    // делаем что-то для конкретного клиента  
}
```

Так выглядит самый быстрый и верный способ создания технического долга. Он приводит к усложнению кодовой базы, созданию компонентов, которые крайне сложно тестировать, и появлению кода, который нельзя будет трогать, потому что от него зависит работа клиента.

**ЗАМЕТКИ С ПОЛЕЙ***Клиент всегда прав*

Я встречал множество примеров в стиле «если клиент такой-то — делать то-то» в портфельных компаниях разного возраста и размера, и все это печальные примеры, потому что с каждым таким решением все сложнее свернуть с порочного пути и каждый такой релиз увеличивает сложность продукта. Другая крайность — когда для конкретного клиента создается отдельная копия всего репозитория. Это самая опасная ситуация, которая гарантированно ведет к катастрофе. Вы сами удваиваете свою рабочую нагрузку. Любые обновления или исправления функций необходимо будет вносить и туда, и сюда, и потом тестировать. Если вы поймаете себя на том, что произносите фразу: «Это же только для одного клиента/просьбы», — немедленно бросайте это дело! Потом вы обязательно скажете себе спасибо.

**ИТОГИ**

- Определить сроки выполнения проекта сложно из-за внешних факторов и непредвиденных событий.
- Если вы разобьете проект на более мелкие части, например с использованием методологии agile, планировать работу станет проще.
- Хорошо регулируемая и структурированная система управления задачами (тикетами) поможет вам и вашей команде четко понимать состояние проекта.
- В растущей команде жизненно важно иметь сильного руководителя проектов, который будет управлять работой на всех этапах.
- Внедрение простых базовых стандартов разработки поможет сделать код более читабельным и таким образом упростить его поддержку, что особенно важно для растущего проекта.
- Система контроля версий — ваш верный союзник в борьбе с постоянными изменениями.
- Подходящая стратегия работы с бранчами позволит вам вести разработку параллельно без необходимости ждать или составлять сложные графики работ.
- Признание технического долга — первый шаг к тому, чтобы начать его уменьшать.
- Существуют разные способы тестирования, как ручного, так и автоматизированного. Это сложная и трудоемкая работа, но лучше иметь хотя бы какие-то тесты, чем вообще никаких.
- Выпуск релизов не обязательно должен быть сложным и рискованным; весь секрет здесь в планировании, отработке на практике и коммуникации.
- При использовании виртуализации и облачных сервисов вы можете значительно уменьшить риски при выпуске релизов, внедрив сине-зеленое развертывание.

- Поощряйте и приветствуйте обратную связь от клиентов и пользователей; однако это не должно приводить к превращению вашей кодовой базы в беспорядочный набор случайных функций.

## ПРОВЕРЬТЕ СЕБЯ

Что из перечисленного ниже вы уже сделали?

- Планируете работу сотрудников и управляете ей (если у вас нет руководителя проектов).
- Отслеживаете работу команды в системе управления задачами.
- Придерживаетесь определенного подхода к разработке и планированию проектов.
- Описали и внедрили четкие и понятные стандарты разработки.
- Внедрили устойчивую культуру работы с системой контроля версий, все члены команды понимают важность и цели ее использования.
- Наладили воспроизводимый процесс сборки и деплоя, который не привязан к отдельному человеку или компьютеру.
- Имеете представление о размере технического долга и составили план его уменьшения.
- Создали выделенную группу тестирования, не являющуюся частью команды разработчиков.
- Наладили процесс выпуска релизов, который позволяет выполнить откат системы в случае необходимости.
- Избавились от конструкций вида «если клиент такой-то, то...» в кодовой базе.

# 10

## *Работа с договорами*

---

### **В этой главе**

- ✓ Договоры с поставщиками и подрядчиками
- ✓ Техническое задание
- ✓ Оценка поставщиков
- ✓ Лицензии и договоры на поддержку

Договоры, соглашения и лицензии — это определенно не самая увлекательная сторона работы технического директора. Мы не подписывались на возню с юридическими документами — мы инженеры! Однако, к сожалению, этим приходится заниматься, потому что технический отдел растет. Вам может потребоваться привлечь сторонних поставщиков, проверить параметры оказания услуг или изучить лицензию на использование ПО, поэтому полезно получить общее представление о том, какова роль этих документов в вашей работе.

В этой главе мы рассмотрим основные договоры, которые могут оказаться у вас на столе, их цель и то, как заставить их работать на вас. Уточним: это юридические документы, составленные характерным юридическим языком. Я настаиваю, чтобы вы всегда консультировались с юристом, прежде чем подписывать или составлять такой документ. Слишком часто самоуверенные инженеры, считающие, что полностью понимают содержание документов, подписывают то, что подписывать не стоит. Юридические документы — это не программный код: в них отсутствует логичная структура и единая понятная цель. Хороший технический директор отдает себе отчет в том, что это не его сфера компетенции, и оставляет юридические формулировки юристам.

Пара слов о хранении: обычно все бумажные копии (так называемые оригиналы) договоров с подписями и печатями хранятся у финансового директора. Тем не менее рекомендуется иметь и электронные версии в вашем хранилище файлов (Google Диск, Dropbox, Office 365), чтобы при необходимости в них можно было быстро что-то найти и проверить.

Итак, рассмотрим самые распространенные типы договоров, которые могут использоваться для фиксации духа и буквы ваших соглашений.

## 10.1. ДОГОВОР ОКАЗАНИЯ УСЛУГ

Рано или поздно вы начнете пользоваться услугами сторонних компаний — а может быть, уже пользуетесь. Они могут привлекаться в следующих целях:

- Пополнение штата сотрудников.
- Выполнение конкретного проекта.
- Оказание поддержки.
- Управление системой.

Хороший договор здесь — это такой, из текста которого обе стороны четко понимают свои обязательства, а также стоимость оказания услуг. Не обязательно для этого расписывать много страниц юридических тонкостей, но как минимум ваши ожидания должны быть четко изложены.

Подобно книге, договор должен иметь начало, середину и конец. В первой части указывается дата начала оказания услуг, любые предварительные

условия, которые должны выполняться, и стоимость работ. Если предполагается создание какого-то продукта, то четко пропишите передачу прав собственности на результаты, она может быть привязана к ежемесячным платежам.

В середине описываются детали того, что вы ожидаете от подрядчика, в том числе результаты работ, ответственность, отчетность и необходимость соответствовать тем или иным требованиям. Этот раздел не должен быть ни слишком подробным, ни, наоборот, слишком общим.

В конце договора указываются условия передачи результатов и завершения сотрудничества. Эта часть также может включать положения о досрочном прекращении действия договора по инициативе сторон.

Расторжение договора не обязательно является негативным событием. Причиной может быть изменение требований бизнеса. Часто расторгаются договоры на поддержку устаревших систем, которые перестают использоваться: цель замены или доработки этих систем как раз и состоит в уменьшении расходов.



**ЗАМЕТКИ С ПОЛЕЙ**

*Читайте, что написано мелким шрифтом*

Философу Юджину Льюису Фордсворту (Eugene Lewis Fordsworthe) приписывают высказывание: «Допущения — источник всех ошибок», — и это действительно так, когда речь идет о договорах. Я участвовал в аудите due diligence от имени частной инвестиционной компании, которая намеревалась приобрести компанию, занимающуюся разработкой ПО. У этой фирмы не было разработчиков в штате, и все свои проекты она передавала стороннему подрядчику. Предполагалось, что правами на все результаты работы владеет заказчик. Однако тут была проблема: в договоре не была прописана передача прав интеллектуальной собственности на конечный продукт, и заказчик не мог его продавать. Полагая, что это просто недоразумение, и учитывая их прекрасные многолетние отношения, компания посчитала, что урегулировать юридическую сторону дела не составит проблем. Однако люди ведут себя непредсказуемо, когда речь идет о больших деньгах. В итоге они не смогли договориться. Подрядчик воспользовался ситуацией и начал выдвигать новые требования. Излишне говорить, что инвестор отказался от сделки, и все потому, что кто-то не прочитал то, что было написано в договоре мелким шрифтом.

Тем не менее иногда деловые отношения не складываются как из-за невыполнения обязательств, так и из-за личных конфликтов или неудовлетворительного качества услуг. Пункты о возможности прекратить действие договора необходимы для того, чтобы расторжение произошло обоснованно, своевременно и профессионально. Как правило, при этом предусматривается срок, в течение которого другая сторона может найти альтернативного поставщика или нового заказчика, обычно от двух до восьми недель. Если это договор на привлечение сотрудников — чаще всего устанавливается месячный срок.

Договор оказания услуг часто дополняется следующими документами:

- техническим заданием;
- описанием параметров оказания услуг.

**10.1.1. Техническое задание**

Выбор поставщика услуг мало чем отличается от подбора нового сотрудника в команду. Первым делом необходимо составить список ожидаемых результатов. Он может иметь вид или *технического задания* (ТЗ), или заказа на поддержку. В документе должно быть достаточно подробностей: участвующие в тендере компании должны понимать, что им нужно будет сделать, чтобы условия контракта считались выполненными.

Эти документы обычно обсуждаются с потенциальными поставщиками после подписания NDA (соглашения о неразглашении), которое дает обеим сторонам возможность вести переговоры, не опасаясь утечки конфиденциальной

информации. Тем не менее наличие обязательств о неразглашении не означает, что вы можете рассказывать все — в любом случае лучше ограничиться как можно меньшим количеством информации и сообщать только то, что имеет отношение к обсуждаемому договору.

### ОПРЕДЕЛЕНИЕ

Соглашение о неразглашении (NDA) — это мини-договор, который заключается между двумя сторонами в целях обеспечения конфиденциальности обмена информацией без риска утечки или неправомерного использования интеллектуальной собственности. Обычно оно ограничено определенной областью и имеет конкретный срок действия. Полезно иметь готовый шаблон соглашения о неразглашении, подготовленный и проверенный юристом вашей компании. Этот документ, как правило, подписывается финансовым или генеральным директором и защищает ваши коммерческие тайны и технологии, что позволяет вам обсуждать их с партнерами в объеме, нужном вам для решения текущих задач.

Типовое техническое задание содержит следующую информацию:

- Объем работ: что нужно делать, что не нужно.
- Предоставляемые результаты: исходный код, библиотека или сервис.
- Момент передачи прав интеллектуальной собственности на результат работы.
- Доступы, необходимые для выполнения работ.
- Разрешения/ограничения на использование определенных сотрудников или привлечение субподрядчиков.
- Процесс согласования дополнительных работ, лицензий или командировок.
- Гарантйные обязательства.
- Срок действия договора.
- Условия продления / внесения изменений / прекращения действия договора.
- Ограничения возможных вариантов реализации.
- Условия оплаты.

Степень детализации ТЗ зависит от планируемого взаимодействия. Для ресурсов, предоставляемых быстро и по запросу (таких, как привлечение разработчиков), характер выполняемой работы обычно указывается кратко. Например: «Услуги разработки на языке Java по запросу менеджера проекта», — без перечисления конкретных задач. Однако в этом случае обычно обозначаются минимальные сроки работ и добавляется условие, что компания не может напрямую нанимать на работу персонал подрядчика без предварительного согласования в течение, например, двух лет после завершения срока действия договора.

Если это договор на поддержку ПО — в ТЗ может оговариваться, для каких версий ПО предоставляется поддержка, с какой частотой выполняются обновления и каков порядок взаимодействия с бизнесом при необходимости выключения ПО на какой-то период.

Хорошее техническое задание должно быть достаточно подробным, чтобы обе стороны четко понимали свои обязательства, но при этом гибким, чтобы адаптироваться к изменениям по мере вашего развития. Ни одна из сторон не должна чувствовать себя связанной чрезмерными обязательствами, поэтому старайтесь не заключать соглашения на слишком длительный период. Примеры ТЗ см. на странице <https://www.projectmanager.com/blog/statement-work-definition-examples>.

### 10.1.2. Параметры оказания услуг

*Параметры оказания услуг*, или SLA (service-level agreement), обычно определяют верхние или нижние границы различных характеристик услуги или продукта. Например, ожидаемое время доступности сервиса и какова компенсация в случае, если оно не обеспечивается.

AWS в SLA для службы Elastic Compute Cloud (EC2) указывает, что доступность сервиса в месяц составляет не менее чем 99,99% времени, а в случае невыполнения этого требования клиенту будет предоставлена скидка в размере 10%. Это значит, что время недоступности сервиса может составлять около шести минут (точнее, четыре с половиной минуты — прим. редактора) в месяц, если оно будет больше — Amazon должен будет вернуть вам 10% от суммы вашего счета за EC2 (<https://aws.amazon.com/compute/sla/>).

Цель SLA — обеспечить вашу уверенность в том, что на сервис можно положиться. Представьте, что Amazon не оговаривает параметры оказания услуг или, что еще хуже, доступность сервисов у них составляет всего 50%. Это означает, что их сервисы могут быть недоступны до двух недель каждый месяц без каких-либо компенсаций. Устраивает ли ваш бизнес такое качество обслуживания?

Документы SLA очень распространены для облачных сервисов и моделей X-as-a-Service (что-либо как услуга). SLA применяется ко всем сферам, в которых привлекаются сторонние сервисы, и определяет, на какую их доступность вы можете рассчитывать, что, в свою очередь, позволяет вам предоставлять своим клиентам информацию о возможностях вашего продукта.

SLA предусматривают разные уровни: чем больше девяток (0,999999) надежности вам нужно от службы, тем дороже она будет. Выбор оптимального варианта во многом зависит от ваших потребностей и может быть разным для разных областей. Некоторые части вашей экосистемы способны выдержать большее время простоя, чем другие.

SLA также определяет, сколько времени может пройти с момента сообщения об ошибке или проблеме до ее решения. Чем важнее сервис, тем выше требования SLA.

#### ОПРЕДЕЛЕНИЕ SLA

С другой стороны, если ваш продукт предоставляет API или работает по модели X-as-a-Service, вам необходимо сформулировать ваши параметры оказания услуг, которые вы должны будете соблюдать. Они оговаривают уровень сервиса,



который вы планируете предоставлять своим пользователям, и компенсацию за его невыполнение.

Если вашим сервисом пользуются не сотрудники компании, а внешние клиенты, то размер компенсации устанавливается совместно с бизнесом. Ваша основная задача в этом случае — разработать параметры SLA, выполнимые в рамках разумного бюджета.

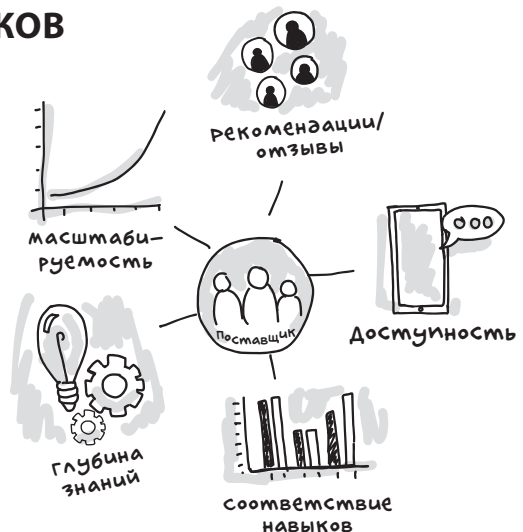
Первое, что нужно отметить: стопроцентный уровень доступности сервиса недостижим — слишком много факторов влияет на него. Вот что следует принять во внимание при определении вашего уровня доступности:

- Какой из сторонних сервисов, используемых вами, имеет самый низкий уровень SLA? (Вы сильны ровно настолько, насколько сильно ваше самое слабое звено.)
- Какова длительность простоя, необходимого вам для выпуска релизов или обновлений системы?
- Способны ли вы обеспечивать выполнение SLA 24 часа в сутки каждый день, или же только в рабочее время?
- Каковы сроки решения той или иной проблемы?
- Достаточно ли у вас сотрудников для обеспечения поддержки, с учетом отпусков?
- Каким был уровень доступности вашего продукта до сих пор?
- Какова самая слабая часть или единая точка отказа в вашем решении, которая может повлиять на его бесперебойную работу?

Никогда не берите цифры из воздуха. Они должны быть достижимы, и вы должны быть готовы обосновать и защитить их.

## 10.2. ОЦЕНКА ПОСТАВЩИКОВ

Вы определили, в какой области вам требуются услуги стороннего поставщика, подготовили основные условия договора и сформулировали требования к параметрам услуг. Пора переходить к выбору поставщика. Этот процесс, проведение тендера, заключается в составлении списка потенциальных поставщиков и их оценке. Кажется очевидным, но я все же отмечу: цена не является определяющим фактором. Окончательное решение следует принимать с учетом следующих параметров:



- *Соответствие навыков.* Определите, соответствуют ли навыки поставщика вашим потребностям. Поставщик не должен обучаться по ходу работы, его навыки должны удовлетворять вас с самого начала. Чтобы убедиться в этом, вы можете побеседовать с теми, кто непосредственно будет работать над вашим проектом.
- *Глубина знаний.* Поверхностные знания в необходимых областях вам не подойдут. Вы привлекаете поставщика в первую очередь потому, что вашей команде не хватает той или иной экспертизы. Чтобы убедиться, что поставщик сможет справиться с любой вашей проблемой, дайте ему несколько тестовых заданий.
- *Доступность.* Потребуется ли вам обслуживание в нерабочее время (обычное условие в договорах на техническую поддержку)? У вас могут быть определенные периоды повышенной нагрузки или ответственности, в которые поставщик должен уделять вам больше внимания. Вы должны оценить, как он обрабатывает запросы (как в рабочее, так и в нерабочее время), скорость ответа, стиль общения, сообщает ли он статус работ и как организует обмен сообщениями.
- *Масштабируемость.* Сможет ли поставщик привлекать дополнительные ресурсы и адаптироваться к вашим потребностям по мере роста вашей компании? Будьте осторожны, если у него есть всего один нужный эксперт. Что, если этот эксперт уйдет в отпуск или, что еще хуже, уволится? Вы ожидаете от поставщика, что вся его команда будет владеть необходимыми знаниями, чтобы обеспечить непрерывное обслуживание.
- *Рекомендации/отзывы.* Попросите рекомендации от клиентов со схожими требованиями. Если поставщик предоставляет услуги поддержки определенного продукта, узнайте, существует ли для этого продукта список рекомендованных поставщиков или партнерские программы и входит ли в них ваш кандидат?

Прочие соображения касаются, конечно, цены и срока действия соглашения — сможет ли поставщик предоставлять вам необходимые услуги и масштабироваться вместе с вами по мере вашего роста? Учет всех параметров и простановка оценок для них позволят вам выбрать правильного поставщика.

Когда вы получите окончательный договор — подключите финансового или генерального директора (и по возможности юриста). Обсуждение важных деталей, излагаемых мелким шрифтом, требует второй пары глаз, которая не имеет прямого отношения к предмету сделки, поскольку такая независимая сторона будет критически оценивать каждый пункт и переводить текст с юридического языка, чтобы понять, насколько он в действительности выполним.

## 10.3. ЛИЦЕНЗИИ НА ИСПОЛЬЗОВАНИЕ ПО

Покупка лицензии на программное обеспечение — это способ оплаты его использования в виде загрузки приложения, пользования сервисом или операционной системой. Наиболее распространенной моделью сейчас стала оплата фактически используемых ресурсов, в том числе и для десктопных приложений (которые раньше обычно оплачивались один раз — в момент покупки). Но какую бы модель оплаты вы ни применяли, вам необходимо отслеживать использование ПО — как для целей бюджетирования, так и для контроля выполнения различных требований, в том числе условий лицензии.

Например, используете ли вы в коммерческих целях лицензии, предназначенные для студентов или индивидуальных пользователей? Такие ситуации часто вскрываются при проверках — компании считают, что они по крайней мере стараются двигаться в правильном направлении, поэтому их совесть чиста, однако у закона иное мнение на этот счет.

Существуют приложения, предназначенные для учета использования лицензий. Если ваша организация не настолько велика, чтобы тратить деньги на специальные инструменты, начните с простого и ведите учет всех лицензий в общедоступной электронной таблице. Фиксируйте в ней как минимум следующие детали:

- Дату приобретения.
- Дату/стоимость продления.
- Контактные данные поставщика лицензии.
- Контактные данные вашего персонального менеджера, такие как электронная почта/телефон/адрес.
- Комментарии, для чего конкретно используется лицензия.
- Ключи/коды лицензий.

Чтобы поддерживать список в актуальном состоянии, его следует пересматривать не реже чем раз в шесть месяцев, а также при каждой покупке или продлении лицензий.

Часто компании недоплачивают или переплачивают за лицензии. У крупных поставщиков программного обеспечения (таких как Microsoft, Oracle, IBM) лицензии сложные и часто меняются. Раньше с этим было проще, потому что лицензии привязывались к отдельным физическим серверам. Но с появлением многоядерных процессоров и виртуализации уже невозможно исходить из того, что на одном физическом сервере будет размещаться, например, 10 виртуальных серверов. Как в связи с этим должна измениться лицензия? И что, если сервер будет работать только в определенное время, например в общедоступном облаке? Поставщики опасаются, что им будут недоплачивать, и соответственно адаптируют лицензии.

По мере роста вашего проекта довольно велик риск столкнуться с отзывом какой-то из лицензий. Поэтому важно при изменении лицензии внимательно

проверять все условия. Обратитесь к менеджеру по работе с клиентами вашего поставщика, который будет рад вас проконсультировать.

Лицензии на использование операционных систем и систем управления базами данных могут быстро стать очень затратными, если вы не будете внимательны. Использовать Microsoft SQL Server для разработки ничего не стоит, но для использования этой СУБД в продакшене вам придется покупать дорогую лицензию, стоимость которой в некоторых случаях может равняться зарплате разработчиков.

Большинство поставщиков с радостью предоставят вам полный доступ к своим инструментам для целей разработки, потому что так они могут зацепить вас или привязать к своему решению. Взвешенное принятие технических решений, с учетом всех особенностей лицензирования, принесет вам дивиденды в будущем.

Поставщики публичных облачных решений в некоторой степени позволяют преодолеть недостатки лицензирования. И Amazon, и Azure предлагают виртуальные серверы, в которых стоимость лицензии включена в поминутную оплату. Запуск экземпляра Windows Server EC2 будет стоить всего на несколько центов в минуту дороже, чем запуск экземпляра Linux. При этом вам не придется беспокоиться о том, нарушаете ли вы закон с точки зрения использования лицензий.

Выше мы рассматривали преимущества решений с открытым исходным кодом, такие как преимущества Linux по сравнению с Windows, MySQL/Postgres по сравнению с SQL Server и так далее. Но что, если вам нужна коммерческая поддержка — номер телефона, по которому можно позвонить, если что-то пойдет не так? К счастью, все популярные проекты с открытым исходным кодом имеют собственную экосистему, включающую различные варианты поддержки. Цены и детали услуг различаются, но найти то, что подходит вам, не составит труда.

## 10.4. ДОГОВОР НА ПОДДЕРЖКУ

Вопрос, который возникает снова и снова: когда стоит приобретать услуги поддержки? Вопрос разумный, потому что большинство программ работает абсолютно нормально. Так зачем тратить лишние деньги?

### ОПРЕДЕЛИТЕ ВСЕ ОСНОВНЫЕ КОМПОНЕНТЫ ВАШЕГО КОМПЛЕКСА

Для целей этого упражнения компонентами будем считать части информационной системы, которые необходимы для обслуживания клиентов и обеспечения непрерывной работы. Теперь предположим, что с одним из этих компонентов что-то случилось: к кому вы обратитесь?

Если в вашей команде есть кто-то опытный и знающий — отлично. Но что, если нет? А что делать, если этот специалист не сможет решить проблему (или окажется недоступен: скажем, в отпуске)? Вы подвергаете бизнес риску невозможности обслуживать клиентов.

Все зависит от уровня навыков и опыта членов вашей команды. Договор на поддержку имеет смысл заключать, когда команде не хватает опыта или пришло время обновлять ПО. Еще один способ определить, нужна ли вам поддержка, — выполнить упражнение ниже.

Не обязательно заказывать поддержку для всего — это может обойтись очень дорого. Кроме того, вам, скорее всего, это не нужно, если у вас сильная команда, которая знает, что делает, и вы не полагаетесь на одного человека. Купите поддержку тех частей, в которых ваша внутренняя экспертиза слабее.

Отдельно упомянем аппаратные компоненты, такие как серверы и сетевое оборудование: если вам приходится управлять значительным объемом инфраструктуры, то необходимо обеспечить надлежащую поддержку на случай сбоев. Оборудование ломается — даже то, в котором нет подвижных частей. Надежный контракт на техническую поддержку даст вам уверенность в том, что, если что-то произойдет, компоненты (например, жесткие диски) будут быстро доставлены и заменены (хотя хороший партнер будет заменять расходники до того, как они выйдут из строя). Поддержка аппаратной инфраструктуры может стоить дорого (и это еще одна причина подумать о переходе в облако); но, к сожалению, без нее не обойтись.

### ЗАМЕТКИ С ПОЛЕЙ

#### *Ныряем в мусорные баки*

Как известно, время идет своим чередом. Годы пролетают незаметно, и вдруг оказывается, что оборудованию, купленному, казалось бы, только в прошлом году, уже 10 лет и оно больше не поддерживается. У меня были клиенты, застрявшие на устаревшем оборудовании, которые копались на сайтах по продаже подержанных вещей, таких как eBay, в поисках подходящих деталей. Торговля старой техникой — это целая отдельная сфера, поскольку компании пытаются выжать из своего оборудования как можно больше. Если вы оказались в такой ситуации, обязательно включите в ваш план модернизацию аппаратного обеспечения.

## ИТОГИ

- Техническое задание (ТЗ) — это документ, в котором подробно описывается оказываемая услуга.
- ТЗ должно защищать интересы обеих сторон, из него каждый должен четко понимать все требования.
- Документ SLA (параметры оказания услуг) определяет ожидаемый уровень оказания сервиса.
- Он также определяет последствия в случае несоблюдения требований SLA.

- Определив требования, проведите тендер на оказание услуг, чтобы привлечь потенциальных поставщиков.
- Оценка поставщиков похожа на процесс собеседования с кандидатами и проверку рекомендаций.
- Лицензия на использование определяет, что вы можете и что не можете делать с программой, оборудованием или сервисом.
- Контракты на техническую поддержку являются вашей страховкой на случай необходимости обслуживания системы.

## ПРОВЕРЬТЕ СЕБЯ

Что из перечисленного ниже вы уже сделали?

- Проверили, что во всех договорах со сторонними поставщиками четко указана передача прав интеллектуальной собственности заказчику после оплаты или завершения работ.
- Подготовили шаблон соглашения о неразглашении на случай взаимодействия с третьей стороной.
- Понимаете SLA всех сервисов, которые вы используете, и как они влияют на доступность вашей информационной системы.
- Определили SLA ваших сервисов как для внутренних, так и для внешних клиентов.
- Разработали и описали методологию оценки поставщиков.
- Организовали учет лицензий на все используемое вами ПО, в том числе контроль выполнения ограничений.
- Если это необходимо — заключили договоры на поддержку основных компонентов.

# 11

## Документация

---

### В этой главе

- ✓ Почему документация важна
- ✓ Виды документации
- ✓ Диаграммы и рисунки как эффективное средство коммуникации
- ✓ Стратегии включения документации в текущие процессы

Эта глава не расскажет вам ничего нового: у нас, технарей, все плохо с созданием документации. Редко встретишь компанию, у которой документация в порядке, и чаще всего благие намерения по ее созданию так и остаются лишь намерениями.

Документация похожа на тесты: ее никогда не бывает слишком много. Но сколько ее должно быть? И как узнать, когда ее достаточно?

В этой главе мы выделим области, для которых может хватить и поверхностного описания, а также области, которые должны быть задокументированы во всех подробностях, в том числе те, что имеют решающее значение для бесперебойной работы команды



и обеспечения преемственности. Хотя список не будет исчерпывающим, он поможет вам задуматься о том, какую документацию следует создать, чтобы можно было поддерживать работоспособность необходимых для компании систем даже после того, как любой из сотрудников покинет команду.

Документация, если ее регулярно обновлять, не требует особого внимания. Однако, если забросить ее на долгое время, затем придется изрядно попотеть.

### ЗАМЕТКИ С ПОЛЕЙ

*Один из моих лучших сотрудников*

В одной из компаний, где я был техническим директором, возникла необходимость подготовить большой объем документации — как внутренней, так и внешней — для передачи партнерам и обучения. Я нанял опытного технического писателя, имеющего также опыт разработки. Его задача заключалась не в том, чтобы документировать вообще все, а в том, чтобы вести заметки и набрасывать схемы (иногда буквально на салфетке), затем структурировать эти заметки, систематизировать их и приводить схемы к общему формату. Он также определял, когда что-то из этого требовало переработки, и проверял описание каждой новой версии, чтобы понять, какие документы могут нуждаться в обновлении. Прием на работу этого сотрудника оказался одним из моих лучших решений, и я советую всегда иметь технического писателя, если в команде 20 или более инженеров.

## 11.1. ЗАЧЕМ СОСТАВЛЯТЬ ДОКУМЕНТАЦИЮ?

Зачем эти хлопоты? Ведь ваша команда знает, как все устроено, — это знание передается из поколения в поколение. Кроме того, ваш технологический стек так грамотно спроектирован и внедрен, что в нем разберется любой толковый специалист. Если вы так думаете, то, вероятно, еще и уверены, что ваш код не содержит ошибок и, следовательно, не требует тестирования! Звучит абсурдно, но многие действительно так считают.

Документация — это преемственность. Документация — это масштабируемость. Документация — это свобода. Это не признак слабости и не формалистская рутина, которую приходится терпеть и которую можно откладывать на самый последний момент.

Независимо от того, насколько современна ваша архитектура и читабелен ваш код, существуют правила и логика, применимые только к вашей организации. Нельзя понять, почему были приняты те или иные проектные решения, только читая код. Документация — это руководство по эксплуатации вашей системы, и без документации никто не сможет в полной мере воспользоваться всеми ее возможностями.

Люди забывают детали. По мере того как дни складываются в недели, недели — в месяцы, а месяцы — в годы, забывается причина, по которой что-то было



сделано именно так, а не иначе. Нестандартное бизнес-правило или неочевидное ограничение, которое в свое время направило архитектуру/реализацию в ту или иную сторону (и в то время это решение было верным), в будущем может оказаться не столь понятным. С решениями, причины которых затерялись в прошлом, произойдет одно из двух:

- Новый сотрудник увидит его, подумает, что можно сделать все проще, переделает — и в итоге что-то сломается.
- Люди будут бояться трогать решение или тем более пытаться его переделать. Оно приобретет мифические свойства и будет считаться слишком сложным или важным, чтобы рисковать и вносить в него изменения.

Оба сценария несут большую опасность, и обоих можно полностью избежать, если оставить краткие пояснения. Как однажды заметил один мудрый разработчик: «Комментарии [в коде] — это любовные записки будущему самому себе».

### 11.1.1. Целевая аудитория

Документация — это не единая универсальная вещь, подходящая сразу для всех читателей. Это набор документов, предназначенных для разных групп получателей, в зависимости от того, как они взаимодействуют с системой. Вот какими могут быть эти группы:

- *Конечные пользователи.* Пользователи системы, взаимодействующие с готовым продуктом.
- *Техподдержка.* Сотрудники службы поддержки, которые помогают конечным пользователям с их вопросами.
- *IT/DevOps.* Команда, отвечающая за поддержание системы в рабочем состоянии.
- *Разработчики.* Команда, которая создает и исправляет функционал.

Чем больше и старше система, тем больше у каждой группы будет особых требований. В небольших командах, которые только начинают развитие своего проекта, одни и те же люди могут выполнять широкий спектр обязанностей, совмещая несколько ролей, а документация для конечного пользователя может вообще отсутствовать.

По мере роста у каждой группы появятся собственные процессы и зона ответственности, и ей будут требоваться поддержка и документация от более низкого уровня, на который она опирается. Наличие надежного источника информации поможет новым членам каждой группы (от конечного пользователя до разработчика) быстрее и успешнее адаптироваться к своим ролям.

Составляйте документацию на языке, понятном каждой группе, чтобы информация была более доступной. Например, описание алгоритмов не будет иметь особого смысла для конечного пользователя или сотрудников службы поддержки, но поможет разработчикам развивать и дорабатывать ПО. Точно так же подробные описания интерфейса и последовательности скриншотов

не помогут команде IT/DevOps, потому что им на самом деле все равно, как выглядит приложение. Им необходимо знать, как, например, делать резервное копирование. Обращайтесь к конкретной аудитории.

### 11.1.2. Формат

Чаще всего документацию оформляют в классическом текстовом формате: такие документы легко создавать, читать, хранить, они не занимают много места, их легко искать и на них можно делать ссылки. Однако в последнее время набирают популярность видеоролики, записи сеанса работы с каким-либо инструментом, как будто ты смотришь через плечо человека, который это делает. Доступные инструменты для создания и редактирования коротких видео позволяют быстро записать и переслать или опубликовать их — а 5-минутный ролик может заменить несколько страниц подробного текста.

Хорошая документация должна быть легко доступна при необходимости. Какой бы хорошей ни была вещь, она практически бесполезна, если ее нельзя найти в тот момент, когда она нужна.

Документация в файлах PDF или DOC была актуальна 20 лет назад, но не сейчас. Документ, особенно предназначенный для обучения или описывающий работу платформы, должен быть живым. Его необходимо обновлять с каждым релизом или при появлении новых данных. И такое обновление не должно превращаться в отдельную большую задачу, оно должно делаться быстро и без усилий.

Когда Тим Бернерс-Ли (Tim Berners-Lee) создавал интернет, его целью было обеспечить удобный обмен информацией. Таким образом, логично организовать хранение документов на основе технологий интернета. Такие инструменты, как вики, Atlassian Confluence, Google Docs и Office 365, отлично подходят для работы в браузере. При выборе платформы для документации учитывайте следующее:

- *Прямые ссылки на контент.* Убедитесь, что пользователь может попасть в нужный раздел по ссылке и не требуется совершать несколько шагов, чтобы найти его.
- *Организация связанного контента.* Инструмент должен позволять легко объединить связанные области, чтобы читателю было удобно искать дополнительную информацию. В идеале это должно делаться автоматически, с помощью тегов или семантического анализа текста.
- *Простое обновление.* По мере появления новой информации контент необходимо обновлять, особенно если он предназначен для службы поддержки.
- *Хранение предыдущих версий документа.* Большинство современных инструментов уже поддерживают эту функцию. Возможность вернуться к предыдущей версии позволит постоянно дорабатывать документы, а читатели смогут определить, к какой версии продукта относится тот или иной контент.

- *Безопасность.* Не все документы должны быть общедоступны, поэтому необходимо иметь возможность ограничить доступ к определенным областям или разделам для конкретных групп пользователей.
- *Обратная связь.* Читатели должны иметь возможность оставлять комментарии или примечания, чтобы дополнять текст.
- *Вложения.* Насколько удобно прикреплять дополнительные файлы (изображения, видео, PDF)? Такие данные лучше хранить вместе с документом, к которому они относятся.

Хороший контент не обязательно должен быть безупречно оформлен — главное, чтобы он помогал читателю. Поощряйте каждого вносить свой вклад в формирование базы знаний — это залог того, что она будет полезной. Чем проще будет написание и дополнение документов, тем лучше.

### 11.1.3. Проверка

Мы все знаем о необходимости тестировать код перед релизом с помощью автоматизированных или чаще ручных тестов, проверяющих, что код делает именно то, что должен делать. Если вы хотите создать современную технологическую платформу, нельзя просто «надеяться», что все будет хорошо. То же самое касается и документов. Прежде чем документы начнут использоваться по назначению, необходимо проверить, насколько они полезны.

Все содержимое документов, особенно недавно написанных, кроме автора должен проверить еще кто-то, подтвердить нужность информации и убедиться, что документ не содержит слишком много пропусков или допущений с учетом его целевой аудитории и темы. Необязательно организовывать формальный процесс ревью — просто попросите еще кого-нибудь посмотреть текст.

Возьмем, к примеру, процесс релиза, включающий сборку, проверку и деплой вашей системы. Вместо того чтобы держать все в голове, разработчик этого процесса решил его описать. Как узнать, достаточно ли подробностей он зафиксировал? Нужно поручить кому-нибудь пройти по документу (в идеале тому, кто не очень разбирается в предмете и не будет подсознательно заполнять пробелы в описании). Это лучший способ понять, решает ли документ свою задачу.

Особое внимание уделите описанию работы, которую обычно выполняет один человек, — все ли его действия зафиксированы верно, чтобы любой мог заменить этого сотрудника? Только проверив документацию, вы сможете с уверенностью ответить на этот вопрос.

## 11.2. ТИПЫ ДОКУМЕНТАЦИИ

В этом разделе мы рассмотрим, какие документы могут храниться в организации. Это не исчерпывающий список, а скорее ориентир, чтобы вы поняли, с чего начать.

### 11.2.1. Протоколы собраний

Собрания не должны быть частыми и многолюдными, поскольку они сильно отвлекают от работы и занимают много времени. Если вы решили, что собрание необходимо, оно должно быть максимально продуктивным.

У хорошего собрания должен быть один или несколько вопросов для обсуждения (или план встречи) и понятная цель. Собрания могут быть отличным способом обмена информацией, идеями и предложениями. Тем не менее, если на них почти или совсем не ведутся записи, много ценной информации упускается — и это большая потеря.

Протоколы собраний, подробно описывающие историю принятия тех или иных решений, должны стать частью вашей базы знаний. Некоторые организации ведут отдельный «журнал решений», подробно описывающий, почему договорились что-то делать именно так.

На каждом собрании назначайте кого-то, кто будет вести заметки, и предусмотрите время для их обработки. Можно вести записи в Google Doc в режиме реального времени, фиксируя ключевые моменты и решения. Если это быстрое собрание, посвященное разбору конкретной задачи, зафиксируйте все договоренности в комментариях к тикету в виде списка.

В постпандемийном мире многие собрания перешли в онлайн и проходят в режиме видеоконференций. В таком формате удобно записывать и хранить все ключевые встречи, чтобы не пропустить важные решения.

Сообщения в чате (Slack, Microsoft Teams и т. д.) также могут содержать ценную информацию. Пусть члены команды копируют ее и добавляют или в основную базу знаний, или в тикеты. Можно даже настроить ботов или плагины, чтобы вы могли быстро пометить сообщения, которые нужно отправить в отдельное хранилище.

### 11.2.2. Демонстрации продукта

Вспомните, сколько раз вы проводили демонстрацию продукта или показывали какой-то функционал по шагам. Записывалось ли это? При этом каждый следующий показ будет содержать что-то новое — или новый вопрос от зрителей, или новую информацию.

Полезно посмотреть рекламные ролики вашего продукта — так можно увидеть, каким образом отдел продаж продвигает его, и узнать, как его функции воспринимает клиент/пользователь и чего они ожидают от продукта. Ваши представления об этом могут значительно отличаться от клиентских.

Эти материалы помогают управляющей развитием продукта команде понять, что нужно пользователям, инженерам — что в продукте наиболее важно, и, наконец, новым членам команды — в создании чего они участвуют.

Объем информации со временем будет расти; не бойтесь этого. У таких материалов есть замечательный побочный эффект: они служат машиной времени,

позволяющей новичкам ознакомиться со старыми версиями продукта и понять, как он эволюционировал.

### 11.2.3. Инструкции по использованию

Вероятно, один из самых важных документов — это тот, который объясняет, как поддерживать систему в рабочем состоянии, если хотите — инструкция по использованию. Он не про деплой кода. Это описание всех частей платформы и того, что они делают, как контролировать их состояние и что делать, чтобы снова запустить их в случае отключения электроэнергии (или выключения по другой причине).

Уровень детализации здесь должен быть достаточным, чтобы любой мог понять и выполнить нужные действия. Недостаточно просто написать: «Обязательно перезапустите компонент XYZ». Как перезапустить XYZ, какие команды нужно вводить и куда, как определить, успешно ли все прошло, и что делать в случае проблем? Такие детали часто пропускаются и хранятся только в чьей-то голове.

Например, во время запуска системы в консоли может выводиться много разной информации. Человек, не привыкший читать все эти строки, будет сбит с толку, он не будет понимать, что из этого хорошо, что плохо, а что можно игнорировать. Но если включить в документацию простой скриншот вывода, всего этого непонимания можно избежать и быть уверенным, что все работает правильно. Как известно, лучше один раз увидеть, чем сто раз услышать.

Руководство не должно быть многословным. Вы же не пишете очередную книгу о Гарри Поттере. Вы описываете, какие действия и в каком порядке нужно выполнить. Придерживайтесь простой структуры: нумерованный список шагов, для каждого из которых читатель должен понять, что нужно сделать, успешно ли все выполнилось и можно ли переходить к следующему шагу.

Также необходимо описать, как контролировать состояние системы: по каким признакам можно определить потенциальные проблемы? Сначала это может казаться сложным. Небольшой совет: попросите сотрудника, отвечающего за ту или иную часть комплекса, представить, что он уходит в отпуск, и написать электронное письмо тому, кто будет его замещать. Такой формат более привычен, и людям проще написать подробное электронное письмо, чем новую страницу вики или документ в Google Docs. Возьмите это письмо и добавьте его в базу знаний. Так можно делать много раз, и в результате вы соберете обширную, подробную библиотеку практических руководств.

Никто в команде не должен чувствовать, что единственная причина, по которой он здесь работает, — это его знание той или иной системы. Подобное отношение часто встречалось 10–20 лет назад, но сегодня это редкость. Люди ценятся не за информацию, которой они владеют, а за их вклад в общее дело команды.

Все мы любим работать с новыми модными технологиями — мы же инженеры. В нашем ДНК заложен постоянный поиск чего-то нового и лучшего. Важно описать работу существующих систем на таком уровне, чтобы никому не приходилось возиться с ними только потому, что больше никто этого не умеет.

### 11.2.4. Резервное копирование и восстановление

Часто в документации не уделяется необходимого внимания резервному копированию и восстановлению системы. Необходимо описать не только способ экспорта/импорта базы данных (хотя и это тоже). Речь об информационной системе в целом, от аппаратного обеспечения до операционных систем и ПО.

Что требуется, чтобы вернуть компонент к жизни? Какие части операционной системы нужны для запуска компонента? То, что кажется очевидным сегодня, может быть непонятно через месяцы или годы.

#### ЗАМЕТКИ С ПОЛЕЙ

##### *Дьявол в деталях*

Несколько лет назад я столкнулся с ситуацией, когда попытки настроить новый сервер для генерации уменьшенных версий изображений зашли в тупик, потому что Java-приложение отказывалось работать. Оказалось, что для работы Java-приложения требовалось наличие ImageMagick — это популярная библиотека с открытым исходным кодом. Приложение не выводило достаточно информации для диагностики проблемы, хотя, конечно, должно было бы. Документация или какие-то подсказки также отсутствовали, и для того, чтобы выявить эту зависимость, пришлось декомпилировать приложение.

С ростом популярности контейнеров (таких, как Docker) зависимости на уровне операционной системы теперь обычно описываются как часть процесса сборки. Однако это не отменяет необходимости указать, что от чего зависит. Даже если все находится в контейнерах, необходимо описать, как контейнеры работают и взаимодействуют между собой, чтобы понять, как ими управлять. Должно быть достаточно информации, чтобы любой, кто разбирается в используемых технологиях, мог собрать компоненты системы из исходного кода. Если сейчас это невозможно, то у вас есть информационный долг. Разумеется, документы также необходимо обновлять с каждым релизом, по мере появления новых компонентов и замены или удаления старых.

Как я уже говорил, отрабатывать восстановление системы необходимо хотя бы раз в год, если не чаще. Если вы используете облачные сервисы, это можно делать без проблем, не мешая остальной работе и не подвергая риску продакшен. Полезно поручать новичкам выполнять сборку проекта с нуля. Это эффективный способ познакомить людей с процессом, показать взаимодействие компонентов и проверить актуальность документации.

### 11.2.5. Процесс деплоя

Релизы и обновления систем предприятия должны быть отлажены, предсказуемы и безопасны. Чтобы этого добиться, необходимы соответствующие инструменты и документация. Даже если вы полагаетесь на автоматизированный инструмент оркестрации (такой, как Jenkins), для его настройки в вашей среде все равно требуется руководство.

Оно должно содержать все необходимые шаги для подготовки сборки, с объяснением происходящих процессов, а также описанием критериев успеха или сбоев. Кроме того, следует указать порядок выполнения шагов и на каких этапах изменения можно безопасно отменить. Если у вас есть дополнительные процессы выпуска патчей, фиксов или релизов, то их тоже необходимо подробно описать.

Разработка надлежащей документации поможет информировать сотрудников и гарантирует, что релизы не будут зависеть от одного человека. Бывает такое, что компания может выпускать обновления, только когда старший специалист DevOps не в отпуске.

### 11.2.6. Комментарии в исходном коде

Если вы хотите вовлечь разработчиков в спор, по накалу страстей не уступающий легендарной дискуссии о табуляции и пробелах (кстати: пробелы), спросите, сколько комментариев должно быть в коде. Ответы будут варьироваться от «Моему коду комментарии не нужны — он и так понятен» до «Да уж, без комментариев тут не разберешься».

Чаше всего, если исходный код недостаточно очевиден и требует дополнительных комментариев — возникают вопросы «что» и «как». Реже бывает непонятно назначение чего-то и приходится пояснять «зачем».

#### ЧТО?

Такой уровень детализации предназначен для других разработчиков, использующих эту часть кода. Например, это может быть библиотека с публичными функциями или методами, которые будут использоваться разработчиками других систем. Сюда также относятся API, которые используются какими-либо сервисами.

Для большинства современных языков существуют соглашения или специальный синтаксис, которые позволяют создавать документацию к вашей библиотеке из комментариев к функциям. Таким образом разработчики, импортировавшие библиотеку в свой проект, могут увидеть справку к ее методам с помощью встроенных в IDE механизмов автодополнения кода.

Отличным примером является язык Java, в котором есть стандарт для написания документации для классов и методов, он называется Javadoc. Такая документация не только показывается при использовании автодополнения кода и других функций IDE, на ее основе также можно сгенерировать набор

веб-страниц с описанием исходного кода. Другой пример — библиотека Swagger, которая предназначена для создания документации в формате HTML для различных API, и, кроме этого, содержит инструменты для выполнения запросов к API для его тестирования или изучения.

Большое преимущество подобной документации состоит в том, что она находится в том же файле, что и код, поэтому ее легко обновлять. Документация в исходном коде очень важна, особенно для больших команд или в тех случаях, когда ваше API используется внешними клиентами.

### КАК?

Этот тип документации тоже находится в функциях и методах исходного кода и предназначен только для тех, кто разрабатывает этот код. Он не должен быть чересчур подробным. Работа грамотно спроектированного и хорошо написанного кода должна быть очевидна. Для этого необходимо использовать осмысленные имена функций и переменных, соответствующие их назначению, — в первую очередь необходимо стремиться к читабельности, а не к оптимизации. Тем не менее, если какая-то часть требует дополнительных пояснений, стоит добавить к ней несколько строк комментариев.

Идеальным будет оставлять в комментариях ссылки на тикеты — это поможет следующему разработчику лучше понять, почему компонент делает именно это. Какой-нибудь неочевидный оператор `if` можно легко объяснить, добавив номер тикета, и тогда следующий разработчик не будет его рефакторить в уверенности, что помогает проекту, но на самом деле добавляя еще больше проблем.

Убедитесь, что при ревью кода проверяется и достаточность комментариев. Когда разработчики привыкнут это делать, они уже не смогут работать без документации.

### ЗАЧЕМ?

Если для чего-то недостаточно описать «как», то нужно объяснить «зачем», то есть для чего предназначена та или иная функция или система, обычно на гораздо более высоком уровне бизнес-логики. Обусловлено ли это на первый взгляд понятное техническое решение каким-то требованием бизнеса или это особенность работы одного из партнеров?

## 11.2.7. Схема архитектуры

Когда вам дарят коробку дорогих шоколадных конфет, на что вы смотрите в первую очередь? Конечно же, на схему, на которой указано, где лежат конфеты с разными начинками! Эта схема дает вам шанс избежать фундука, который прячется внутри. А истинные ценители шоколада используют ее для планирования своего гастрономического путешествия, ведь они хотят насладиться максимально насыщенным вкусом. Схемы и карты обеспечивают нас информацией, чтобы мы могли ориентироваться гораздо увереннее.



У вашей системы тоже должна быть такая «карта», или схема архитектуры. Когда вы знакомитесь с системой, всегда просите нарисовать ее схему на доске. Если в ответ вы получите недоуменные вопросительные взгляды, уточните: «Она состоит из прямоугольников и линий».

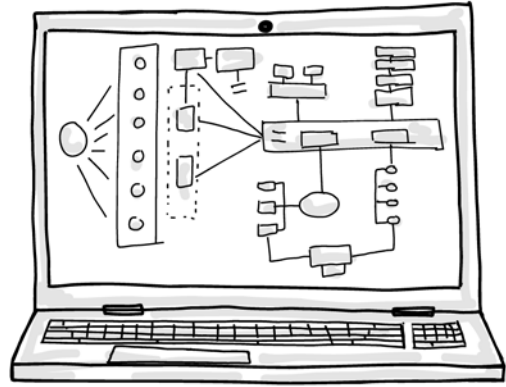
Это не должна быть сложная или формальная схема. Это не строительный чертеж, который должен соответствовать строгим стандартам. Это просто путеводитель, помогающий понять термины, в которых описывается платформа.

Здесь должны быть основные компоненты вашей системы, и если какие-то из них связаны между собой, то их нужно соединить на схеме линиями. Не вдавайтесь в подробности. На этом уровне достаточно знать, что File Microservice используется модулем обработки фотографий профиля пользователя. Сейчас не имеет значения, *как* он это делает — это может быть вызов API, очередь или просто общая папка. Такие подробности должны быть в других документах. Здесь же вы указываете — возвращаясь к аналогии с картой — только границы штатов, автомагистрали, показывающие, каким образом разные штаты связаны между собой, и самые крупные города.

Слишком много деталей тут не нужно, это перегружает карту и делает ее трудной для понимания. Для каждого компонента (или штата, если говорить о карте) создайте отдельную схему, более подробную (уже с указанием местных дорог и городков).

Аналогия с картой особенно хорошо подходит для архитектур, использующих микросервисы и API. Как и в штатах США, крупные города и автомагистрали остаются неизменными, но местные дороги в каждом штате и городе постоянно развиваются.

В продуманных схемах архитектуры отдельные части могут оставаться актуальными даже спустя годы развития проекта. Схема верхнего уровня должна показывать людям, не входящим в вашу команду, как все устроено, не загружая их излишними деталями или сложными терминами. Не удивляйтесь, если она появится в презентации для совета директоров, потому что хорошая схема — отличный способ продать свое видение.



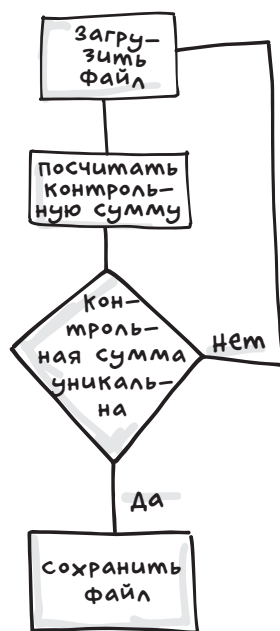
### 11.2.8. Диаграммы процессов

Со схемой архитектуры тесно связаны диаграммы процессов. Они дают представление о том, как данные перемещаются по системе и что с ними происходит,

когда они попадают в тот или иной сервис или библиотеку. Если схема архитектуры — это карта автомагистралей внутри штата или между ними, то диаграмма процесса — это наложенная на нее схема трафика из Google карт, показывающая, где находятся автомобили.

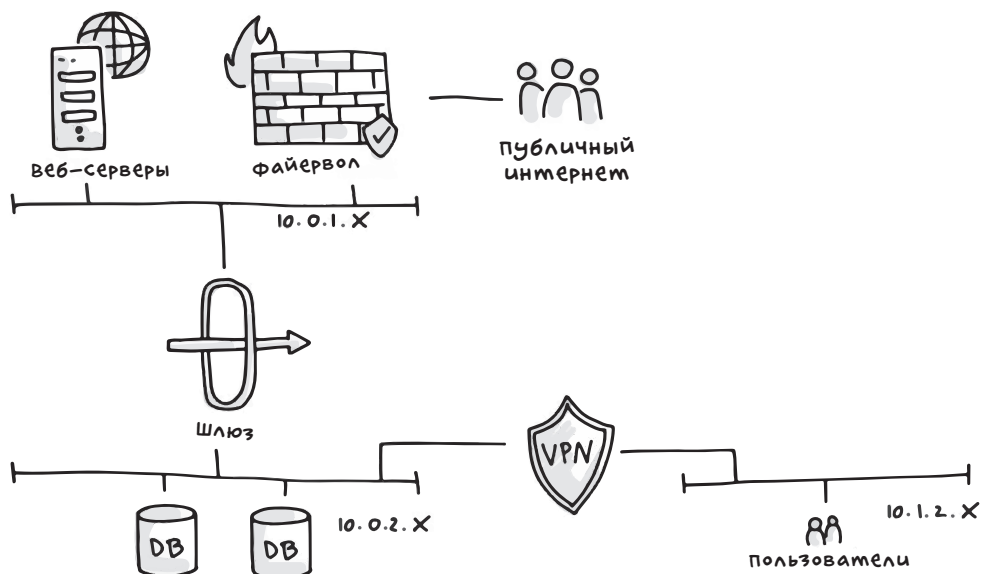
Диаграмма процесса — это логическая схема, показывающая, где данные хранятся, кэшируются и передаются из одного компонента в другой. Опять же, если она будет чрезмерно подробной — схема станет слишком сложной и в ней можно будет увязнуть.

Цель таких диаграмм — наглядно показать движение данных и на общем уровне обозначить пути и способы их перемещения, а также то, как данные попадают в систему и покидают ее.



### 11.2.9. Схемы сети

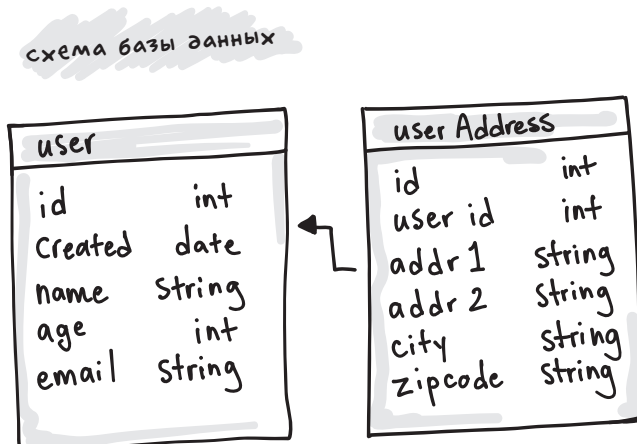
Схема сети похожа на схему архитектуры, но она создается с точки зрения структуры ваших сетей и серверов. Полезно указывать на ней сетевые адреса, маршруты, файерволы и шлюзы, которые соединяют части сети.



Если у вас есть физические серверы, на этой диаграмме нужно показать их местоположение (например, адрес ЦОД и координаты стоек в нем). Характеристики каждого элемента оборудования, в том числе даты покупки, гарантийный срок и настройки конфигурации, можно будет указать в отдельных документах.

### 11.2.10. Схемы баз данных

Базы данных лежат в основе большинства информационных систем, именно они хранят бесценные данные и связи между ними, без которых деятельность компании невозможна. Тем не менее, несмотря на всю важность этой области, документации по ней чаще всего недостаточно, хотя так быть не должно.



Существует, однако, множество инструментов (например, DB Schema), которые могут проанализировать реляционную базу данных и сгенерировать ее схему, при этом на основании внешних ключей (foreign keys) будут определены связи между таблицами. Вы можете дополнить эту схему комментариями, поясняющими бизнес-логику, которая стоит за структурой данных. Также все современные СУБД поддерживают возможность добавления комментариев к таблицам и столбцам наподобие того, как это делается в исходном коде.

Другие хранилища данных могут не поддерживать встроенные комментарии, и, следовательно, для них приходится вручную создавать дополнительные документы. Контроль этого нужно добавить в релизный цикл, чтобы, если что-то изменится, документация была обновлена.

Если ваша документация, описывающая структуру и месторасположение всех данных, составлена так, что будущие сотрудники смогут легко ее найти и использовать, то вы можете быть спокойны.

### 11.2.11. Соответствие нормативным требованиям

Вы можете работать в отрасли, где требуется наличие определенной документации, например описание процессов, результаты аудитов и периодические отчеты. Типичный пример таких отраслей — здравоохранение и финансы. Некоторые правительственные и военные проекты также требуют определенного уровня описания процессов (например, соответствия стандарту ISO).

Подробности, касающиеся соответствия HIPAA, PCI, ISO и т. д., выходят за рамки этой книги. Если вашей компании требуется сертификация — привлечите стороннего эксперта, который поможет организовать этот процесс. В больших аудиторских компаниях, например, этим занимаются целые департаменты. Эта задача на первый взгляд кажется пугающей, но, если подойти к ней правильно, сложностей не будет.

Какими бы ни были нормативные требования — считайте, что за ними стоит здоровое желание добиться, чтобы все стороны действовали прозрачно. Организациям, в которых отсутствуют документация, процессы и регламенты, здесь будет тяжелее всего. Все придется создавать с нуля, и это может быть трудной и долгой задачей. Компаниям с четкой документацией и стандартами намного проще. Да, в них тоже наверняка есть области, требующие доработки, и части, которые не помешает усилить, но все уже движется в правильном направлении.

Соответствие требованиям — это, конечно, больше, чем просто документация, и все, что вы описали, необходимо выполнять на практике. Это может потребовать доработки кода или архитектуры, так что смотрите на это как на возможность для оптимизации и приведения в порядок различных частей проекта.

Большинство нормативных требований уделяют много внимания вопросам безопасности и работы с данными. Если вы считаете, что ваша организация может попасть в сферу действия таких требований, то начинайте изучать их заранее, готовиться и внедрять в команде соответствующие практики. Считайте, что это знак вашей исключительности. Именно вы отвечаете за защиту систем и данных вашей организации. Поэтому прислушивайтесь к любым полезным рекомендациям.

### 11.2.12. Учет лицензий и результатов аудитов

Вопрос, о котором обычно не думают, но рано или поздно он возникает, — как найти все ваши лицензии и купленное ПО. Заведите в базе знаний документ для учета всего используемого вами стороннего ПО, как купленного, так и с открытым исходным кодом. Сообщите команде, что важно вносить туда все, даже то, что кажется незначительным. Каждый раз, когда разработчик Java добавляет новую библиотеку в конфигурацию Maven или разработчик JavaScript подключает новый модуль npm, обновляйте этот документ, включая информацию об источнике ПО и условиях лицензии.

Проверка списка используемого ПО — это обязательная часть аудита due diligence, но на момент написания этой книги я еще не встречал команду, в которой эта информация была бы под рукой. Этот список необходим и в других случаях, например для продления и обновления лицензий, а также для того, чтобы понять, можно ли перенести ваше ПО в облачные сервисы, если вы захотите это сделать.

### 11.3. ДОКУМЕНТЫ ДЛЯ ПУБЛИКАЦИИ (WHITEPAPERS)

Строго говоря, эти документы не относятся к внутренней документации компании, но упомянуть о них стоит. Компания может создавать и публиковать документы, предназначенные для внешней аудитории. В них описываются подходы или процессы компании в той или иной области.

Такие документы удобны, если клиентам требуется обучение для того, чтобы полноценно использовать ваш продукт. Они часто применяются в процессе продаж и прекрасно иллюстрируют профессионализм создателей продукта и глубину вложенных в разработку знаний.

В них не должно содержаться коммерческой тайны или подробного описания технологии. Цель здесь — познакомить читателя со стороны с вашим ходом мыслей, чтобы ему было легче взаимодействовать с продуктом. Вот что можно описать:

- Принципы безопасности API.
- Рекомендации по работе с данными.
- Рекомендуемые методы работы при интеграции API.
- Жизненный цикл данных.

### 11.4. РЕКОМЕНДАЦИИ

Создание документации, возможно, не самая эффектная часть вашей работы, но она является одной из важнейших, и без документации ваша команда не сможет успешно масштабироваться и расти. Вот некоторые лучшие практики и советы, которые помогут вам найти свой ритм:

- Тщательно выбирайте инструмент для вашей базы знаний, он должен предоставлять как можно больше функционала.
- Создайте шаблоны, чтобы можно было быстро добавлять новые страницы. Гораздо проще редактировать что-то, чем создавать с нуля.
- Начинать с малого и двигайтесь не спеша. Это марафон, а не спринт.
- Поощряйте добавление в документацию информации из электронных писем и чатов.
- Быстро создать контент можно, записав видео или скринкаст.
- Описывайте, что делает исходный код, во время его ревью.
- Отрабатывайте и тренируйте все процессы управления вашим комплексом.

## ИТОГИ

- Записывайте знания сразу, пока они еще свежи.
- Выбор правильного инструмента облегчит всем внесение изменений и дополнений.
- Придерживайтесь простого стиля, чтобы те, кто не любит много писать, не боялись участвовать в создании документации.
- Видеоролики, голосовые заметки и схемы отлично подходят для описания деталей.
- Важно перепроверять все документы, касающиеся управления системой и ее поддержки.
- Даже краткие заметки или списки — это лучше, чем ничего.

## ПРОВЕРЬТЕ СЕБЯ

Что из перечисленного ниже вы уже сделали?

- Предоставили команде инструмент совместной работы, в котором можно делиться знаниями.
- Создали документ, объясняющий, как запустить и поддерживать в рабочем состоянии вашу информационную систему.
- Перечислили все компоненты вашей системы, включая сведения о необходимых версиях/лицензиях.
- Разработали простую для понимания схему архитектуры.
- Подробно описали этапы создания резервных копий и восстановления из них.
- Поощряете культуру открытости и обмена знаниями.

# 12

## Безопасность

---

### В этой главе

- ✓ Важные соображения по обеспечению безопасности
- ✓ Безопасная разработка ПО
- ✓ Кризисное управление при атаке или взломе
- ✓ Когда необходим CISO (руководитель направления информационной безопасности)

Безопасность — это одна из сфер, в которых, если все делать правильно, сложностей не будет. Однако, если заниматься ими только для галочки, в минимальном объеме, это чревато проблемами. В некоторых организациях на первый взгляд с безопасностью все хорошо, но если копнуть поглубже, то обнаружится, что в части IT ключ от входной двери у них лежит под ковриком. Также есть опасный эффект: чем дольше вы живете без инцидентов, тем проще убедить себя, что вы защищены, — но это ничем не лучше, чем верить, что ваш дом неуязвим для пожара, потому что до сих пор он не горел.

Разумеется, для создания многоуровневой системы безопасности потребуются сфокусированные усилия и большое количество работы и могут понадобиться серьезные переделки. Тем не менее вы можете многое сделать для защиты существующей платформы, даже если изначально технологии обеспечения безопасности в нее не закладывались. В этой главе мы рассмотрим, какие шаги можно предпринять для значительного улучшения ситуации с безопасностью.

### ЗАМЕТКИ С ПОЛЕЙ

Как наставник, помогающий техническим директорам в их профессиональном развитии, я встречаюсь с такой точкой зрения, что безопасность достаточно организовать один раз и затем можно перейти к другим делам. Обеспечение защиты при этом не считают постоянной задачей, когда необходимо соответствовать непрерывно изменяющимся требованиям. Но безопасность — это не просто установка файервола или выбор надежных паролей. Это непрерывный процесс: оценка поверхности атаки, мониторинг систем на наличие уязвимостей или признаков проникновения, необходимость всегда быть на шаг впереди злоумышленников.

Многие думают так: у нас нет ценных данных, представляющих интерес для хакеров. Это используется в качестве предлога, чтобы ничего не делать для защиты. Да, ваши данные или системы могут не быть конечной целью, но их можно использовать для достижения других целей. Если хакеры могут взломать ваши системы и, например, получить доступ к стороннему сервису, то какой ущерб может нанести атака злоумышленников от вашего имени? Всегда необходимо обеспечивать максимально возможную защиту от внутренних и внешних атак.

Эта глава не содержит исчерпывающего перечня мероприятий, необходимых для обеспечения информационной безопасности, — для этого понадобится отдельная книга. Здесь приведен общий обзор основных вещей, которые нужно иметь в виду, чтобы вы могли внедрить ориентированные на безопасность подходы, сберечь данные ваших клиентов и оправдать их доверие.

## 12.1. ВНЕСЕНИЕ ИСПРАВЛЕНИЙ

В программных продуктах, независимо от того, кто их разрабатывал, всегда присутствуют ошибки. Эти ошибки не преднамеренные, они возникают из-за того, что какой-то фрагмент недостаточно хорошо продумали или протестировали. Ошибки могут быть как безобидными — что-то не совсем правильно отображается — так и способными вызвать катастрофические последствия (в зависимости от проекта). Как только об ошибке станет известно, кто-нибудь обязательно придумает способ, как ее использовать, чтобы получить незаконное преимущество или выгоду, причинить ущерб или просто похвастаться.

Большинство проблем безопасности возникает по следующим причинам:

- Код некачественно разработан/протестирован.





- У обычной функции появился непредвиденный побочный эффект (как случилось, например, в конце 2021 года с библиотекой для логирования Apache Log4j, ниже мы расскажем об этом более подробно).

В современном мире большинство компаний очень быстро выпускают исправления, или патчи, если им сообщат о багах или уязвимостях либо если они обнаружат их своими силами. Никто не хочет быть обвиненным в недостаточном внимании к вопросам безопасности.

Поэтому важно поддерживать вашу систему в актуальном состоянии, своевременно внедряя все исправления, и не допускать использования устаревшего ПО. Для этого необходимо работать в двух направлениях:

- следить за появлением исправлений;
- планировать обновления.

### 12.1.1. Определение наличия исправлений

Для начала необходимо составить список всего ПО (включая библиотеки), которое используется в компании, с информацией о версиях. Этот список понадобится во многих случаях, его важность и работу с ним мы обсуждали в главе о документации. Если не следить за актуальностью этого списка, можно легко пропустить критическое обновление: вы не обратите на него внимания, думая, что оно к вам не относится.

Наличие обновлений необходимо в первую очередь проверять на сайтах разработчиков. У всех крупных поставщиков ПО есть сайты или рассылки, за которыми вы можете следить. Проекты с открытым исходным кодом публикуют описание для каждого из релизов, также у них есть форумы, на которые вы можете подписаться. Лучшим источником информации об уязвимостях является широко известный реестр CVE.

#### ЗАМЕТКИ С ПОЛЕЙ

*Общеизвестные риски безопасности и уязвимости (Common Vulnerabilities and Exposures, CVE) <https://www.cve.org/>*

CVE, онлайн-база данных всех известных уязвимостей, поддерживается с 1999 года, и каждый квартал в ней регистрируется около 5000 уязвимостей. Каждой из них присваивается номер CVE, по которому можно узнать подробное описание, уязвимые версии ПО и ссылки на решение, если оно есть. Этот ресурс — своего рода Википедия нарушений безопасности, и часто, когда вы читаете о проблеме в публичных источниках, вы видите ее номер CVE, по которому можно ознакомиться с техническими подробностями в этой БД. Хорошим вспомогательным сайтом является <https://www.openvuln.io/>: на нем есть функции фильтрации обновлений и подписки на них.

Настройте автоматическое получение обновлений, чтобы их наличие не приходилось постоянно проверять. Будьте внимательны: периодически проверяйте,

что ваша подписка работает. Параметры рассылок и уведомлений со временем могут меняться.

Хотя это обязанность вашего специалиста по информационной безопасности (Chief Information Security Officer, CISO), поощряйте других членов команды тоже следить за уязвимостями. Сигнальный огонь Гондора (во «Властелине колец») мог увидеть всего один человек, этого уже достаточно, чтобы забить тревогу.

### 12.1.2. Планирование

Узнать, какие обновления нужно установить, — это обычно легкая часть. Теперь нужно решить, когда это делать. Есть соблазн проигнорировать какие-то обновления как не относящиеся к вам. Это большая ошибка. Возьмите за правило устанавливать обновления, даже мелкие, часто, по крайней мере ежемесячно: это может повлиять на текущую работу, но защитит вас в долгосрочной перспективе. В идеале у вас должна быть возможность выполнять сине-зеленый деплой (см. главу 9), чтобы упростить тестирование исправлений/обновлений перед внедрением.

Обновления, особенно из проверенных источников, в 99% случаев будут работать без сбоев. Но, каким бы высоким ни был этот показатель, не рассчитывайте на полное отсутствие проблем. Тестируйте, тестируйте и еще раз тестируйте. Используйте для проверки исправлений серверы для разработки или стейдж или, по крайней мере, выкладывайте обновления вместе с релизами вашего ПО. Так ваши тестировщики смогут обнаружить любые отклонения в поведении.

Если у вас нет возможностей для тестирования в изолированном окружении, то придется иметь дело сразу с боевыми патронами — обновлять продакшен. Это относится в том числе к системам бэк-офиса (таким, как серверы электронной почты и файловые серверы). В таких ситуациях всегда лучше проводить обновления в нерабочее время — лучше всего в субботу, чтобы, если что-то пойдет не так, у вас оставался день в запасе (воскресенье) для решения проблемы.

### 12.1.3. Особые случаи

Такое оборудование, как принтеры, сканеры и телевизоры, обычно не относится к компетенции технического директора, однако для него тоже нужно постоянно проверять наличие обновлений безопасности. Современные продукты (например, Google Chromecast/Nest) регулярно обновляются автоматически. Однако не все производители настолько прилежны. Сетевые маршрутизаторы и коммутаторы здесь требуют особого внимания, и способ получения информации об обновлениях для них не всегда очевиден.

Для некоторых устройств приходится вручную заходить в консоль управления и проверять наличие обновлений. Этим действием нельзя пренебрегать,

и оно должно входить в обязанности сотрудников, ответственных за ИТ. Заведите привычку регистрировать такие проверки в общем журнале или отдельных журналах для каждого устройства. Любое устройство, подключенное к интернету, необходимо рассматривать как часть продакшена и уделять должное внимание его безопасности и управлению паролями, даже если оно не вызывает никакого беспокойства. Любое устройство в сети может быть атаковано. Если у устройства есть IP-адрес — оно является потенциальной угрозой.

#### ПРИМЕЧАНИЕ

Поэтому рекомендуется выносить продакшен в отдельную, полностью изолированную сеть. Использование одной и той же сети бэк-офисом и продакшеном увеличивает поверхность атаки за счет бесконтрольного подключения и отключения новых устройств.

## 12.2. ТЕСТИРОВАНИЕ НА ПРОНИКНОВЕНИЕ

Тестирование на проникновение, как следует из названия, заключается в том, что кто-то со стороны пытается взломать вашу систему и ищет уязвимости и слабые места, которые можно использовать для проникновения или причинения ущерба. Из всех проверок безопасности эта больше всего страдает от эффекта плацебо: слишком часто компании пребывают в уверенности, что у них с безопасностью все хорошо, потому что они успешно прошли тестирование на проникновение, хотя это было уже больше шести месяцев назад.

Как только тест на проникновение завершен, он уже устарел. Методы взлома постоянно совершенствуются, и нельзя считать себя неуязвимым только потому, что вы прошли тест. Кроме того, как только вы выпустили новый релиз вашего ПО — результаты всех предыдущих тестов на проникновение становятся неактуальными. Почему? Потому что вы не знаете, есть ли в новом релизе новая уязвимость или незащищенное место. Поэтому тестирование на проникновение необходимо проводить не разово или время от времени, а непрерывно и сделать его неотъемлемой частью вашего процесса непрерывной интеграции и деплоя.

Хороший тест на проникновение должен исследовать все уровни доступа. Самый очевидный из них — это доступ снаружи. Но к чему могут получить доступ или что могут испортить ваши сотрудники, имеющие расширенные права? Это следующие уровни тестирования безопасности, которые часто упускают из виду.

Вы можете добавить в ваш арсенал средств защиты тесты на проникновение с открытым исходным кодом, но не полагайтесь исключительно на них, особенно если какие-то из них не обновляются. Привлеките стороннего поставщика (для этого придется согласовать дополнительные расходы), особенно если вы работаете с конфиденциальными данными, например финансовыми или медицинскими.

## 12.3. СОЦИАЛЬНАЯ ИНЖЕНЕРИЯ

Скорее всего, вы слышали о социальной инженерии — когда сотрудника компании убеждают совершить действие, открывающее доступ злоумышленнику, и при этом жертва будет уверена, что не делает ничего плохого, и, возможно, даже будет рада оказать кому-то кажущуюся необходимой помощь.

Например, вы можете позвонить в службу поддержки, убедить сотрудника в том, что вы кто-то, кем вы не являетесь, и получить конфиденциальные данные. Это может быть так же просто, как проникнуть в офисное здание без пропуска, следуя за другим человеком, причем он, скорее всего, еще и придержит вам дверь. Уверенный в себе злоумышленник может долго и успешно выдавать себя за сотрудника.

Такие случаи легко приводить в качестве примера, и на бумаге все проблемы очевидны. Но реальный мир — это совершенно другое дело, здесь бдительность и подозрительность уступают место естественному человеческому желанию помогать и идти навстречу.

### ЗАМЕТКИ С ПОЛЕЙ

*Это может случиться с каждым*

Один из моих инженеров попался на простой прием социальной инженерии; к счастью, без каких-либо последствий. Как-то в офисе появились два человека, которых он принял за представителей владельцев здания, и попросили пустить их к шкафу за дверью, это было нужно им для каких-то работ в соседнем офисе. У нас было небольшое помещение, и в этот шкаф приходил кабель от оператора связи Verizon, и там же находилось все сетевое оборудование. Наш инженер, всегда очень бдительный в интернете, на этот раз не только не попросил их подтвердить свою личность, но и придерживал дверцу шкафа, пока они все изучали. Когда он сообразил, что допустил двух совершенно незнакомых людей к нашему ключевому оборудованию без какой-либо проверки, он был ужасно растерян и смущен. Эти люди действительно были от владельцев, и все было в порядке, но урок запомнился навсегда — любого можно застать врасплох.

Нас всех когда-то обманывали, в большинстве случаев безобидно. Тем не менее безопасность нельзя основывать на надежде, что мы не столкнемся со злоумышленником.

Потенциальной мишенью является любой сотрудник, но наибольшему риску подвергаются те, кто взаимодействует с людьми со стороны — в первую очередь это служба поддержки, а также менеджеры по работе с клиентами и все, кто помогает продавцам с подключением новых клиентов. Проводите с ними регулярные тренинги по безопасности.

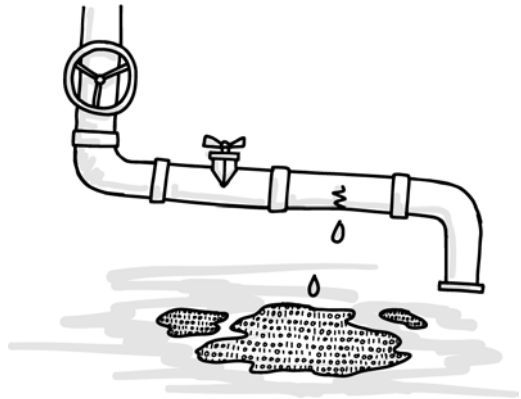
Если вы хотите проверить, насколько ваша команда готова противостоять обману, проведите упражнение в стиле «тайный покупатель». Например, попросите

кого-то, кого не знает ваша команда, но кому вы доверяете, попытаться пройти в офис без пропуска. Это простая и безобидная проверка, но она выявит слабые места, которые вы сможете исправить. Пусть ваш шпион попробует попасть в офис быстрее вас, в лучших традициях Голливуда или телешоу. Не стоит недооценивать человеческое стремление помогать, а также то, насколько далеко люди готовы зайти в злоупотреблении им.

## 12.4. УТЕЧКА ДАННЫХ

Подобно тому, как из треснувшей трубы или ржавого крана утекает вода, из компании понемногу могут утекать данные. На первый взгляд такие утечки могут показаться безобидными, но, если их не остановить, они могут стать серьезной проблемой безопасности.

Обнаружить утечку данных намного сложнее, чем заметить подтекающий кран, под которым собралась лужа: вы не будете слышать звука капающей воды. От вас потребуются сосредоточенность и постоянная бдительность. Найти правильный баланс между доступностью и безопасностью — сложная задача. Определите, что важно для вашей организации, и организуйте защиту этих областей. Данные могут уходить наружу в самых неожиданных местах, поэтому рассмотрим некоторые распространенные источники утечек, чтобы вы представляли, где это может происходить в вашей компании.



### 12.4.1. Журналы событий

Скорее всего, вы фиксируете все события (почему бы и нет?), и это бесценная информация для отладки и анализа производительности. Хорошо. Но, подождите, а что конкретно ваши разработчики сохраняют?

Если вы записываете запросы к API и ответы — что, опять же, обычная практика — то что происходит при этом с конфиденциальными данными, которые в них передаются? Вы уделяете большое внимание тому, чтобы, например, номера паспортов или водительских удостоверений хранились только в зашифрованном виде, но при этом они будут попадать в логи в открытом виде, потому что передаются в API.

### ЗАМЕТКИ С ПОЛЕЙ

Однажды я работал с клиентом, который с гордостью уверял, что не хранит пароли конечных пользователей. Вместо этого он хранил хэши паролей с добавлением случайной строки (соли). Прекрасно. Однако в его системных логах все пароли находились в открытом виде, поскольку там сохранялись все запросы к API, и их было легко найти. Упс.

Журналы событий могут быть настоящей сокровищницей, и обычно они не так хорошо защищены, как базы данных. Тщательно следите за тем, что у вас регистрируется, и используйте средства маскировки конфиденциальных данных перед их сохранением в журналах. Не стоит использовать безопасность как оправдание тому, чтобы вообще не хранить логи, — просто следите за тем, что в них попадает.

### 12.4.2. Сообщения об ошибках

У вас бывало такое, что вы заходите на сайт и вместо нужной страницы видите страницу ошибки, иногда содержащую большое количество диагностической информации для разработчика? Однако вы не разработчик; вы просто пользователь со стороны.

Такая «полезная» страница может содержать много конфиденциальных данных, касающихся подключения к БД и информации о системе. Потенциальные взломщики могут специально вызывать ошибку на сайте, чтобы получить подсказки.

Раньше сайты, работающие на JSP, CFML или PHP, выводили на страницах ошибок полную последовательность вызовов функций (stack trace), значения переменных и параметры сервера. Эта информация не предназначалась для продакшена, но где-то по пути кода из разработки и тестирования ее вывод порой забывали выключить. Хотя такое бывает нечасто, сообщения об ошибках, если за ними не следить, могут привести к утечке большого количества информации.

### 12.4.3. Экспорт данных

Ваша система может позволять пользователям экспортировать данные или выполнять произвольные запросы / получать аналитику через интерфейс или даже через API. Не получает ли пользователь при этом доступ к слишком большому количеству данных?

Часто инструменты бизнес-аналитики или отчетности становятся источником утечки, если данные не изолируются в достаточной степени. То, что видно на экране, может быть лишь частью того, что фактически передается, и в консоли разработчика в браузере можно быстро найти гораздо больше данных (и, возможно, более ценных).

#### 12.4.4. Система контроля версий

Исходный код — это сердце многих компаний, но он может служить и источником утечек. Указанные в нем учетные данные (такие как имена пользователей и пароли для разных сервисов) могут оказаться в самых непредсказуемых местах, например, попасть в инструменты проверки качества кода, которые исследуют его и копируют куда-либо для анализа или более эффективной работы. Некоторые инструменты управления версиями, как коммерческие, так и с открытым исходным кодом (например, GitHub), выполняют базовую проверку на наличие чувствительной информации и отправляют оповещения с описанием того, что они нашли.

### 12.5. ПЕРИОДИЧЕСКАЯ СМЕНА ПАРОЛЕЙ

Пароли по-прежнему являются основным способом аутентификации для большинства систем, с которыми мы работаем. Сама по себе эта технология неплоха. Слабые места, которым она обязана своей плохой репутацией, — это генерация паролей и работа с ними. Если пароль никогда не меняется и указан прямо в исходном коде, то он никак не участвует в обеспечении безопасности — это просто дополнительный атрибут конфигурации.

Правильная организация работы с паролями должна быть такой, при которой сами пароли имеют достаточную длину и сложность (необходимый набор букв и символов), а их изменение происходит легко и без последствий. Пароли, которые нужно указывать в исходном коде или отдельных файлах, будут создавать проблемы. Если пароль находится в файле, то его может прочесть любой, у кого есть доступ к соответствующей файловой системе.

Нередко, особенно в различных старых системах, учетные данные для подключения к базе данных хранятся в файле XML, после изменения которого требуется перезапуск сервера. Это является проблемой для многих серверов приложений. В такой ситуации смена пароля становится кошмаром, поскольку приходится изменять файлы и планировать перезапуск серверов и проверку их работы. Неудивительно, что пароли меняют редко.

Работа с паролями должна быть гибкой, чтобы их можно было менять или регенерировать настолько часто, насколько это требуется. Программное обеспечение, требующее пароль, должно поддерживать его получение из менеджера паролей. Провайдеры облачных сервисов предоставляют такую услугу, и она является основой для надежного облачного приложения (учетные данные как к локальным, так и к сторонним сервисам не смогут быть получены, пока вы не подключитесь к менеджеру паролей).

Скорее всего, вы будете жить в гибридном мире и работать вручную с паролями, хранящимися в обычных файлах (даже если этот файл создается безопасно с помощью скрипта). Определите все места, где требуется комбинация имени пользователя и пароля, и для каждого из них опишите процедуру изменения пароля — что нужно менять и перезапускать.

Для этой цели подойдет простая электронная таблица, хранящаяся вместе с другими результатами инвентаризации ваших систем, в которой будут перечислены все компоненты, требующие учетных данных, как в примере ниже.

Компонент	Программа	Формат	Перезапуск	Назначение
Веб	Tomcat	web.xml	Да	База данных

### 12.5.1. Учет параметров доступа

Меняйте пароли в следующих случаях:

- Сотрудник покидает организацию.
- Возникло подозрение в нарушении безопасности.
- Программное обеспечение было обновлено.
- Периодически (например, раз в месяц).

Поддержание дисциплины в команде и внедрение практики регулярной смены паролей поможет избавиться от связанных с этим страхов и сомнений. Общий совет: если меняете пароли, то меняйте их все. Если вы подозреваете, что скомпрометированы одна или два учетные записи, считайте, что в зоне риска находятся все. Не поддавайтесь искушению ограничиться точечными изменениями.

## 12.6. БЕЗОПАСНАЯ ИНФРАСТРУКТУРА

Безопасность — это не еще одна функция, которую можно добавить к проекту или купить. Это философия и набор принципов, по которым вы живете и работаете, и то, за чем вы постоянно следите. Создание безопасной инфраструктуры сосредоточено вокруг трех основных задач: инвентаризации, защиты и контроля.

### 12.6.1. Инвентаризация

Прежде чем возводить эффективный «забор», необходимо определить области, нуждающиеся в защите. Это называется «определить поверхность атаки» — найти все потенциальные пути взлома или утечки. Как технический директор, вы обязаны максимально сокращать поверхность атаки, а там, где это невозможно, обеспечивать всю возможную защиту. Вот примеры направлений атаки на вашу организацию, а также способы их локализации:

- Серверы продакшена находятся в той же сети, что и серверы для разработки. Слишком многие имеют к ним доступ. Уменьшите поверхность атаки, поместив продакшен в изолированную сеть.



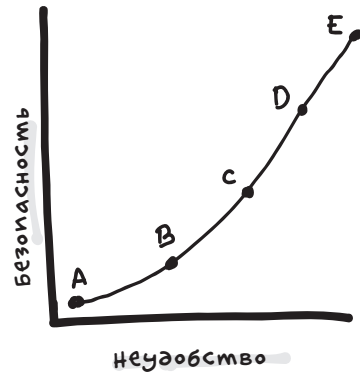
- Все используют одну и ту же учетную запись для доступа к БД, поэтому нельзя понять, кто выполнил то или иное действие, или запретить доступ отдельному пользователю. Уменьшите поверхность атаки, создав отдельные учетные записи только для тех, кому этот доступ действительно нужен.

Это все реальные ситуации, и исправление их не требует запредельных усилий, а увеличение безопасности в результате будет очень существенным. Поверхность атаки недостаточно определить один раз. Она меняется с выпуском каждого нового релиза или добавлением нового сервиса, и вы должны будете заново оценить, достаточны ли ваши средства защиты.

### 12.6.2. Защита

Определив все области, потенциально подверженные риску, переходите к следующему шагу — защите. Это самая сложная часть процесса. Каждый уровень защиты требует финансовых затрат и несет потенциальные неудобства для пользователей. Рассмотрим следующие сценарии доступа к базе данных (или к любой другой системе, требующей аутентификации):

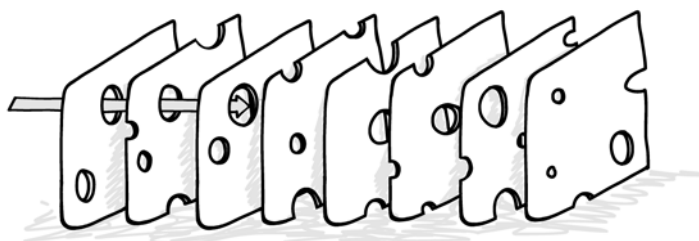
- Имя пользователя и пароль не требуются.
- Все используют одни и те же имя пользователя и пароль.
- У каждого пользователя свои имя пользователя и пароль.
- Для доступа к базе данных требуется подключить VPN.
- Используются MFA (многофакторная аутентификация) и VPN, а также индивидуальные имя пользователя и пароль.



Каждый следующий уровень обходится дороже предыдущего как за счет трудозатрат на его организацию, так и за счет стоимости дополнительного функционала и лицензий. Кроме этого, каждый уровень добавляет неудобств пользователю, которому приходится совершать все больше действий.

Такова плата за обеспечение безопасности. К примеру, самая защищенная база данных — та, которая не подключена к сети и к которой никто не имеет доступа. Но каким бы привлекательным этот вариант ни казался с точки зрения безопасности, для целей бизнеса он не подходит. Данные должны быть доступны, чтобы приносить пользу и обеспечивать работу бизнеса. Баланс между безопасностью и доступностью у каждой системы будет своим, и его поиски требуют изучения и обсуждения с ответственными за бизнес, чтобы все понимали и поддерживали связанные с этим процессы и политики.

Существует много способов достичь цели, особенно в вопросах безопасности. Защита обеспечивается на нескольких уровнях, и ошибочно считать, что всего один уровень может гарантировать безопасность.



Создайте несколько уровней защиты, каждый из них для своей области. Эту модель часто называют моделью швейцарского сыра. Ее большое преимущество в том, что такая защита не зависит от каждого отдельного механизма. Например, фаервол может блокировать всё, за исключением нескольких портов, через которые пользователи попадают только в одну подсеть, и дальнейший доступ требует дополнительной аутентификации. Такое сочетание аппаратной и программной защиты может быть очень эффективным.

### ЗАМЕТКИ С ПОЛЕЙ

*F5 Big-IP, весна 2022*

По моему опыту, безопасность — это не что-то, что можно отдать на аутсорсинг и забыть. Например, логичным шагом выглядит покупка серьезных средств защиты, таких как F5 BIG-IP (можно сказать, что это IBM в сфере безопасности). Тем не менее даже компания, которая сосредоточена исключительно на обеспечении безопасности, со всеми ее многоуровневыми проверками, выпустила продукт, который позволял выполнять произвольные команды от имени пользователя root путем отправки простого запроса HTTP POST. Уязвимость CVE-2022-1388 имеет рейтинг 9,8 из 10 и является одним из самых серьезных нарушений безопасности в современной истории. Безопасность — это всегда сложно, поэтому никогда не полагайтесь на один инструмент и используйте несколько решений, которые дополняют друг друга, чтобы обеспечить наилучшую возможную в данный момент защиту.

Полагаться на один уровень защиты — значит складывать все яйца в одну корзину в надежде, что она никогда не упадет и яйца не разобьются!

### 12.6.3. Мониторинг

После инвентаризации и защиты следующий и последний этап — постоянный мониторинг уязвимостей и атак на случай, если кому-то удастся обойти защиту. Здесь у нас две цели: во-первых, убедиться, что средства защиты работают. Нет ничего хуже, чем обнаружить, что после перезагрузки программный файервол не заработал. Во-вторых, определить потенциальные слабые места, которые можно улучшить. Злоумышленники постоянно совершенствуют свои методы, и ваша цель — чтобы ваша команда и технологии были хотя бы на один шаг впереди них.

Основной источник данных для мониторинга — журналы событий каждого из уровней защиты. Работа с ними сопряжена с определенными сложностями. Каждый из уровней и каждая система защиты имеют собственную консоль управления и свой журнал событий. Вручную заходить в каждую систему для проверки совершенно неудобно, и люди со временем устают и становятся невнимательными.

Нужно объединить все журналы в единый агрегат для комплексного мониторинга. Это позволит увидеть на первый взгляд несвязанные части скоординированной атаки.

Сложность этой панели инструментов во многом зависит от вашего бюджета и от ценности того, что защищается. Например, можно начать с таких инструментов, как Elasticsearch или Splunk, это популярные агрегаторы журналов. Некоторые продвинутые инструменты, использующие искусственный интеллект, могут постоянно анализировать данные.

Каким бы путем вы ни пошли — а это в любом случае будет движение вперед — убедитесь, что вы приучаете команду к постоянной бдительности. Не думайте, что если вы уже внедрили какие-либо средства защиты, то этого достаточно. Обеспечение безопасности — постоянная и важнейшая часть обязанностей команды.

Стоит отметить, что эти агрегаты также можно использовать для проведения проверок. Регистрацией входов в систему (как успешных, так неудачных) часто пренебрегают. Добавление этой информации поможет отслеживать необычную активность, например использование учетной записи сотрудника ночью или попытку доступа к недавно удаленной учетной записи. Уровень детализации будет зависеть от вашей организации и от данных, за которые вы отвечаете.

## 12.7. БЕЗОПАСНАЯ РАЗРАБОТКА

Компаниям, занимающимся разработкой и внедрением ПО, особенно важно обеспечить безопасность:

- исходного кода, который они разрабатывают;
- процесса сборки и внедрения ПО.

### 12.7.1. Создание безопасного исходного кода

Разрабатывать безопасный код сложно. Если бы это было легко, нам не приходилось бы постоянно исправлять и обновлять программное обеспечение. Уязвимости появляются в исходном коде по двум основным причинам:

- Низкое качество кода или наличие ошибок, которыми могут воспользоваться злоумышленники.
- Высокая сложность функционала создает возможности для его незапланированного использования.

В зависимости от языка программирования низкое качество кода влияет на продукт по-разному. В языках, которые требуют от разработчика ручного управления памятью, риск атак с использованием переполнения буфера выше, чем в языках с автоматическим управлением памятью. Другой пример — печально известные «SQL-инъекции», когда один оператор SQL может быть преобразован в несколько операторов путем подстановки в него специально подобранного значения.

#### SQL-ИНЪЕКЦИЯ: ЧТО ЭТО ТАКОЕ

Рассмотрим сайт, на котором есть функция поиска и пользователь может ввести произвольный поисковый запрос. Код, выполняющий соответствующий запрос к базе данных, может выглядеть следующим образом:

```
stmt = "SELECT id, email FROM table X WHERE name='" + inputFromUser +  
      "''";
```

Выглядит безобидно, не правда ли? Однако, если посмотреть внимательнее, видно, что разработчик создал огромную дыру в безопасности, которая позволяет взломать или удалить всю базу данных. Каким же образом?

Что, если в поле ввода на сайте вместо слова `noah` ввести `"2'; TRUNC X; "` и отправить этот запрос в API? В результате будет собрана строка, содержащая две отдельных команды SQL, которые будут выполняться одна за другой (при условии, что библиотека для работы СУБД разрешает это, а в большинстве случаев она разрешает). Первая команда SQL выберет записи, в которых поле «имя» имеет значение 2; это не вызовет никаких проблем. Однако вторая команда, `"TRUNC X"`, удалит все строки таблицы X. После некоторого количества проб и ошибок злоумышленник может устроить хаос в вашей базе данных.

Подумайте только: взломщику даже не нужен какой-либо доступ к сети или к базе данных — он всего лишь использует средства, предоставленные ему самой компанией и включающие удобный интерфейс для выполнения его хакерских SQL запросов.

На заре существования веб-сайтов такие атаки были обычным делом, и многие из них оставались незамеченными. И хотя подобные уязвимости теперь встречаются нечасто, разработчики все еще пишут неграмотный код, особенно те из них, кто недостаточно обучен принципам безопасной разработки и тому, что все

данные, поступающие снаружи, необходимо проверять. И это только один пример. Разработка безопасного и безошибочного кода — отдельная дисциплина, и ей посвящено множество книг.

Вы можете снизить риски, обучая свою команду программистов, внедряя стандарты разработки и проводя внимательное ревью кода. Инструменты контроля качества кода (такие как SonarQube) могут выявлять некоторые типовые проблемы и постоянно совершенствуются, но никогда не заменят взгляд опытного человека.

### ЗАМЕТКИ С ПОЛЕЙ

*Взлом Log4j, конец 2021 г.*

В современном мире, где все переплетено и взаимосвязано, сочетание функций разных систем может создать уязвимости даже без явных ошибок разработчиков, и ничто не иллюстрирует эту проблему лучше, чем история с популярной библиотекой журналирования Log4j. Компания Tenable, специализирующаяся на кибербезопасности, назвала ее «самой большой и самой критичной уязвимостью за последнее десятилетие». Сторонний пользователь мог запустить на уязвимых машинах произвольный код путем простой записи в журнал специально сформированной строки; так работала библиотека, которая фиксировала в журналах поисковые запросы, пользовательский ввод или сообщения из чата. И этот функционал был заложен в нее разработчиками, но никто не подумал, что ее будут использовать для обработки небезопасных строк — вот непредвиденные последствия лучших намерений.

Подобные побочные эффекты возможны в любой системе, которая допускает расширение с помощью сторонних плагинов. Яркий пример — веб-браузеры, в которых плагины или дополнения обеспечивают более широкие возможности для пользователей. Однако это означает постоянное перетягивание каната между возможностями и безопасностью, потому что каждое изменение системы плагинов создает новые возможности для обхода ограничений.

## 12.7.2. Защита процесса сборки

Есть еще одна область, в которой нарушения безопасности могут принести такой же ущерб, как и взлом продакшена: это средства работы с исходным кодом и сборки ПО. При этом защищена она обычно хуже, что понятно: разработчики здесь вносят изменения и экспериментируют с кодом, а безопасность, при всей ее важности, может этому мешать.

Тем не менее у многих команд отсутствуют средства защиты, отделяющие среду для разработки и сборки от продакшена. Предпринимаются ли при этом какие-то меры, чтобы в код незаметно не оказалась добавлена вредоносная библиотека или фрагмент кода?

Если пайплайн сборки не защищен, кто-нибудь может внести в него небольшие изменения. Точно так же, если доступ к вашей системе контроля версий не ограничен надлежащим образом или вы используете небезопасный менеджер зависимостей (например, Maven или prtm, где не фиксируется конкретная версия библиотек), то вы рискуете задеплоить код, не зная, что на самом деле у него внутри.

Так произошла печально известная атака на ПО SolarWinds, которая оказала огромное влияние на все отрасли, поскольку это программное обеспечение использовалось в крупных организациях как часть системы управления ИТ-инфраструктурой. Хакеры нашли идеального троянского коня: им не пришлось взламывать каждую цель, хватило одного сервиса, которым пользовались все. Затем вредоносный код распространялся как часть обычных обновлений ПО, проникая глубоко во внутренние сети и не вызывая срабатывания средств защиты.

Безопасный подход здесь — считать, что любая часть вашей системы является потенциальной мишенью, и создать как можно больше средств защиты, включая вынесение процесса сборки в отдельное окружение, фиксацию конкретных версий для всех сторонних библиотек (и их обновление только после проверки и подтверждения) и повышенное внимание к внезапным изменениям размеров файлов, которые не связаны с большими изменениями в коде (например, если собранное приложение стало занимать в два раза больше места, чем обычно, должен срабатывать алерт мониторинга).

## 12.8. «НАС АТАКУЮТ»

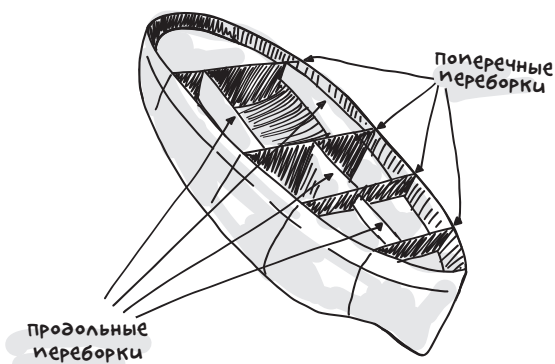
Один из самых кошмарных моментов в жизни технического директора — когда вы узнаете, что вас атакуют. Еще один, ничуть не лучше, — когда вас атакуют, но вы об этом не знаете. Будем надеяться, что такое будет происходить как можно реже, потому что вы предприняли все необходимое для мониторинга угроз в режиме реального времени. Атаки могут быть следующего вида:

- Атака с целью вызвать отказ вашей системы.
- Скачивание данных.
- Удаление или шифрование данных.
- Несанкционированный вход в систему.

Эти ситуации могут быть чрезвычайно стрессовыми, главное тут не паниковать. Окружающие будут ждать от вас разумного и надежного плана действий. Первым делом необходимо определить область атаки и то, как ограничить дальнейший урон. Но это легко только на бумаге.

### 12.8.1. Аварийное отключение

Вместо того чтобы дожидаться атаки и только после ее начала решать, что делать, заранее пройдите по всей вашей системе и продумайте способы реагирования, или *аварийные выключатели (kill switch)*. Это цифровые версии переборок на корабле, которые предназначены не только для обеспечения жесткости, но еще и для герметизации и изоляции, чтобы одна пробоина в корпусе не могла затопить весь корабль. (Если переборки спроектированы неправильно, они создадут больше проблем, чем решат, как это было в случае с *RMS Titanic* — и, раз уж мы заговорили об этом, в фильме «Титаник» на плоту хватило бы места для двоих!)



#### ЗАМЕТКИ С ПОЛЕЙ

*Утечка данных как из пожарного шланга*

У одного из клиентов, с которыми мы работали, была интеграция со сторонним платежным сервисом. Часть интеграции заключалась в том, что API клиента получал уведомление об успешном завершении платежа. Используя средства мониторинга, мы обнаружили странности при работе с этим API: данных приходило гораздо больше, чем следовало, и, кроме этого, было очень много ошибок «неизвестный получатель платежа». При подробном изучении выяснилось, что API получал конфиденциальную информацию о вообще всех платежах, проходящих через этот платежный сервис. Мы тут же сообщили им об этом, и сначала нам не поверили (потому что у них не было соответствующих логов), а когда признали проблему, им потребовался почти день на то, чтобы выключить передачу данных (аварийного отключения у них тоже не было). Если бы у них были логи или хотя бы аварийный выключатель, они могли бы действовать активнее, то есть остановить утечку и уже после этого исследовать проблему. Позже нам сказали, что ошибка была в запросе, в котором отсутствовал нужный оператор WHERE, и поэтому наш клиент получал информацию обо всех платежах, а не только о своих. В этом сервисе явно не проводились ревью и тестирование кода — то есть этот инцидент вскрыл целую пачку проблем — и всё это в публичной компании (так что даже «большие парни» могут допускать такие заметные ошибки).

Цель аварийного отключения не в том, чтобы решить или устранить проблему, а в том, чтобы выиграть время для устранения первопричины проблемы. Вот возможные примеры цифровых аварийных выключателей:

- Отключение сетевого доступа к серверу/компьютеру или подсети.
- Выключение ресурса (или приостановка, если он виртуальный).

Самое важное во время атаки — это ограничить возможный урон. Это обязательно означает останов всех систем в ущерб бизнесу. Если есть возможность, изолируйте атакуемую область, используя заранее предусмотренные аварийные выключатели, но если такой возможности нет, то безопаснее на какое-то время полностью отключить систему, особенно если вы подозреваете, что скомпрометированы конфиденциальные данные пользователей.

### 12.8.2. Коммуникация

Коммуникация важна. Если ваше руководство будет знать, что происходит, это обеспечит вам их поддержку и пространство для маневра. Может показаться, что такая коммуникация будет отвлекать, но, как говорилось выше, периодические новости-об-отсутствии-новостей будут держать всех в курсе, а ваша команда сможет сосредоточиться на проблеме.

После изоляции проблемы перенесите фокус на определение нанесенного ущерба, чтобы затем выяснить, как ликвидировать последствия и предотвратить повторное появление подобных проблем. Ущерб может заключаться как в неправомерном использовании ваших ресурсов (например, если был вызван отказ в обслуживании), так и в скачивании или уничтожении данных либо в их шифровании в целях вымогательства. Необходимо не только определить, что конкретно произошло, но и убедиться, что злоумышленники не смогут продолжить или возобновить атаку после того, как вы с ней справились.

Если ваша команда не имеет соответствующих навыков, то никто не упрекнет вас, если вы привлечете специализированного поставщика услуг безопасности. Его экспертиза позволит не только решить проблему до конца, но и гарантировать, что она не повторится.

Только когда вы определите, каким образом взломщики получили доступ, и устраните слабые места, вы можете быть уверены в том, что после восстановления все не начнется сначала. Иначе вы будете играть в «ударь крота» («whack-a-mole»), думая, что решили проблему, но, как только вы отвлекетесь, она повторится.

Проверка вашей системы потребует усилий, времени и знаний, и вы никогда не будете до конца уверены, что вычистили все следы атаки. Если вы оказались в такой ситуации, я настоятельно рекомендую привлечь стороннего подрядчика. Хотя современные подходы к архитектуре с их большим количеством





взаимосвязей создают дополнительные риски для безопасности, они также упрощают восстановление системы, особенно если вы используете контейнеры и виртуальную инфраструктуру, — просто уничтожьте контейнеры и создайте их с нуля (при условии, что в ваш процесс сборки не проникла библиотека злоумышленников).

### 12.8.3. Устранение последствий нарушения безопасности

Еще одна область, с которой вам, возможно, придется столкнуться, — это последствия нарушения безопасности или кражи данных. В этой ситуации вы должны определить, что было украдено. В отличие от физических активов — похищение которых очевидно, потому что они будут отсутствовать, — цифровые активы можно скопировать, не оставив следов. Будем надеяться, что вы записываете достаточно событий, чтобы следы остались в журналах; иначе как бы вы вообще узнали о краже? В таких случаях часто проводят *компьютерно-техническую экспертизу*.

Самое безопасное здесь — считать, что злоумышленники забрали все, что было им доступно. После того как вы составили список украденного, определите следующее:

- Кого необходимо уведомить?
- Что сделать, чтобы снизить ущерб от похищения или ценность данных?
- Какие меры безопасности предпринять, чтобы предотвратить дальнейшие кражи?
- Влечет ли кража юридические последствия?

Мой совет — не пытайтесь сделать все самостоятельно. Подключите руководство компании, проконсультируйтесь с юристом и воспользуйтесь услугами сторонних специалистов.

Мы все получали электронные письма в духе «беспокоиться не стоит, но...» от генеральных или технических директоров после утечек данных в крупных фирмах. Их рассылают тем, чья учетная запись или данные по результатам расследования могли быть скомпрометированы. Таким образом они, как ответственные за ваши данные, предупреждают вас, чтобы вы были начеку на случай чего-то подозрительного.

Очевидные шаги, которые можно предпринять, включают в себя принудительный сброс всех паролей или требование заново аутентифицироваться для учетной записи, требующей кода подтверждения из электронной почты или SMS. Какими бы простыми эти действия ни казались, они уменьшают ценность украденных учетных данных. Однако с определенными данными, такими как имя, адрес или номер паспорта, вы мало что можете сделать, поэтому их кража может повлечь серьезные последствия. Вы серьезно подвели ваших клиентов, доверивших вам свои ценные данные.

В зависимости от украденных данных вам, возможно, придется публично сообщить об утечке. Как правило, это касается любых финансовых или медицинских данных, но, прежде чем что-либо делать, проконсультируйтесь с вашими юристами.

Чтобы разобраться во всех подробностях атаки или взлома, требуется время. Вы неизбежно обнаружите, что слабое место было именно там, где события не регистрируются или регистрируются недостаточно подробно. Ничего необычного в этом нет, и добавить дополнительное логирование не составит труда.

Проведите тщательное расследование и представьте его результаты так, чтобы они были понятны нетехническим специалистам. Рассмотрим простую структуру отчета об инциденте из четырех частей: инцидент, выявление, влияние, устранение (PIIR — issue, identify, impact, remedy):

- *Инцидент* — что произошло и как это произошло.
- *Выявление* — как вы узнали о проблеме.
- *Влияние* — последствия инцидента.
- *Устранение* — действия для прекращения инцидента и устранения последствий.

Отчет PIIR может выглядеть, например, следующим образом:

- *Проблема.* Через форму на сайте, используемую для изменения адреса пользователя, любой зарегистрированный пользователь мог увидеть адрес любого другого пользователя, просто изменяя через консоль разработчика в браузере значение ID при выполнении запроса к API.
- *Выявление.* Команда системных администраторов обратила внимание на увеличение количества запросов к API, в итоге они выявили подозрительные запросы к API работы с адресом, выполняемые с одного и того же IP.
- *Влияние.* С одного IP-адреса, предположительно принадлежащего одному пользователю, была получена информация о примерно 1000 адресов пользователей, прежде чем системные администраторы заблокировали доступ для него. Конкретного пользователя идентифицировать невозможно, потому что в журнале аутентификации пользователей не сохраняется информация о их IP-адресе.
- *Устранение.* API было доработано, теперь оно возвращает только адрес текущего пользователя, доработка выложена на продакшен. Все остальные API проверяются на наличие аналогичных проблем и дорабатываются. При регистрации событий сеанса пользователя теперь сохраняется IP-адрес. Составлен список всех учетных записей, чьи адреса были просмотрены.

В зависимости от инцидента PIIR могут быть достаточно большими. Эти документы предназначены только для внутреннего пользования и должны содержать достаточно подробностей, чтобы на них можно было опираться, если будет принято решение о публикации информации об инциденте. В них излагаются факты, а не план действий (который находится в зоне ответственности всего руководства компании).

## 12.9. РУКОВОДИТЕЛЬ НАПРАВЛЕНИЯ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ (CISO)

Организации растут и расширяются, и в конце концов задачи обеспечения безопасности, организации мониторинга и соответствия требованиям станут слишком большими и важными, чтобы их могли выполнять другие сотрудники параллельно их основным обязанностям. В этом случае лучшим решением станет выделение отдельного специалиста, который в дальнейшем может сформировать целую команду. Обычно эта позиция называется руководителем направления или директором по информационной безопасности (Chief Information Security Officer, CISO). В некоторых компаниях может быть аналогичная должность с более коротким названием — директором по безопасности (Chief Security Officer, CFO).

До появления этого сотрудника вся ответственность за поддержание и мониторинг безопасности, скорее всего, лежала на вас и вашей команде. Чтобы у нового ответственного не были связаны руки и он мог охватить все вопросы безопасности, он не должен входить в вашу команду.

Энди Ву (Andy Wu), генеральный директор CodeArcs, любит известное латинское выражение *Quis custodiet ipsos custodes*, которое примерно переводится как «Кто устережет самих сторожей?» или «Кто наблюдает за наблюдателями?». В нашем мире это означает, что ответственный за информационную безопасность не должен подчиняться техническому директору. Его возможности и влияние не должны быть ограничены. Он будет работать совместно с вашей командой, и в то же время у него должна быть возможность предъявлять к ней требования.

Хотя это может выглядеть для вас как потеря влияния или ограничение полномочий, на самом деле все совсем наоборот. Это признание роста и зрелости платформы, лежащей в основе бизнеса. Ответственный за информационную безопасность будет необходим в следующих случаях:

- Ваша компания должна проходить процедуры сертификации, связанные с безопасностью, и обеспечивать соответствие требованиям.
- Вы используете данные, к работе с которыми предъявляются особые требования и необходимо проведение аудитов.
- Средств защиты и обеспечения безопасности стало настолько много, что для работы с ними требуется сотрудник на полный день.

Хороший ответственный за информационную безопасность посвятит все свое время и усилия защите компании и принадлежащей ей информации, работая над этим совместно со всеми, кто участвует в обработке или создании данных.

## ИТОГИ

- Установка исправлений — это не только обновление ПО; также требуется тестирование и планирование.
- Тесты на проникновение, если их не проводить после каждого релиза, — не более чем плацебо.
- Сотрудники — самое слабое звено цепи из-за возможности использования приемов социальной инженерии.
- Пароли, которые сложно менять, ни от чего не защищают.
- Создание защищенной среды требует постоянной бдительности и усилий.
- Необходимо обучать разработчиков вопросам безопасности и проверять написанный код на отсутствие слабых мест.
- Процесс сборки приложений может быть уязвимым местом.
- Создание цифровых переборок, или аварийных выключателей, позволит ограничить распространение атаки.
- Реагирование на инциденты безопасности и устранение их последствий требует организованной работы и эффективной внутренней коммуникации.
- Рано или поздно вы дорастаете до уровня, когда для обеспечения безопасности потребуется отдельный сотрудник.

## ПРОВЕРЬТЕ СЕБЯ

Что из перечисленного ниже вы уже сделали?

- Внедрили график обновления всего стороннего ПО и библиотек.
- Провели тестирование на проникновение для текущего релиза вашей системы.
- Проводите непрерывное обучение и информирование сотрудников об угрозах атак с применением социальной инженерии.
- Можете быстро менять все пароли без необходимости перезапуска компонентов.
- Добавили журналы событий и средства мониторинга, достаточно эффективные, чтобы вы могли узнать о нарушениях и утечках данных.
- Предусмотрели несколько аварийных выключателей, чтобы иметь возможность ограничить распространение атаки или утечки данных.
- Организовали обучение разработчиков и ревью кода, чтобы не создавались уязвимости.
- Создали средства сборки и деплоя, которые защищены не меньше, чем продакшен.

# 13

## Поддержка и обслуживание

---

### В этой главе

- ✓ Обслуживание оборудования
- ✓ Мониторинг состояния платформы
- ✓ Методы резервного копирования и восстановления
- ✓ Как уложиться в бюджет

Мы часто считаем, что система или платформа, которую мы создаем, будет работать сама собой. Все усилия мы тратим на ее проектирование, разработку и внедрение и порой забываем о том, что требуется, чтобы поддерживать ее в рабочем состоянии, предотвращать потенциальные проблемы и обеспечить ее быстрое восстановление, если что-то пойдет не так.

В этой главе мы рассмотрим некоторые общие вопросы, которые могут заставить врасплох технического директора. В основном это самые базовые вещи, но иногда даже очевидное можно упустить из виду. Например, многие из нас уверены, что резервное копирование выполняется в полном объеме, но вспомните, когда вы в последний раз делали полное восстановление? Вы уверены, что резервные копии действительно содержат все, что нужно?

Мы упоминали, что при разработке кода может появиться технический долг, но он также образуется, если не следить за состоянием вашей системы в продакшене. Проведение регулярных работ по обслуживанию часто игнорируется, хотя эти работы необходимы, чтобы не было никаких сюрпризов.

## 13.1. ОБОРУДОВАНИЕ

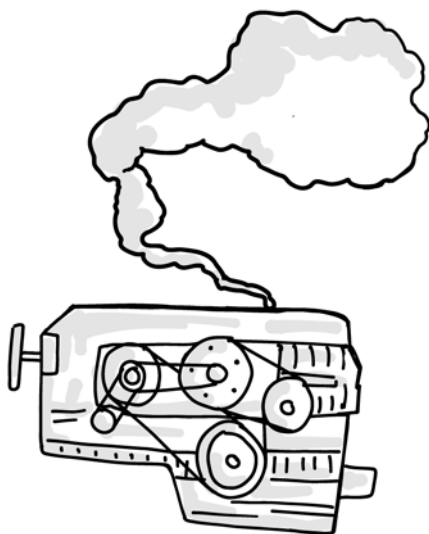
Если вы используете физические серверы или сетевое оборудование, то вам нужен план их обслуживания. Аппаратное обеспечение — это живой организм, требующий внимания и регулярного обслуживания, особенно те его элементы, которые имеют движущиеся части, в том числе вентиляторы. Движущиеся части рано или поздно перестанут двигаться, и это может вызвать негативные последствия. Ваша задача — сработать на опережение, предусмотреть возможную поломку, подготовиться к ней и, когда она произойдет, спокойно ее исправить.

Как мы уже говорили, полная и своевременная инвентаризация оборудования имеет ключевое значение для поддержания его в исправном состоянии. Необходимо фиксировать следующие детали:

- производитель;
- тип/модель;
- дата покупки / установки / следующего обслуживания;
- сведения о поддержке;
- журнал обслуживания;
- информация о местоположении / учетные данные.

Для этого хватит простой электронной таблицы, но существуют и специальные программы. Обслуживание оборудования похоже на ТО автомобиля — да, вы можете не поменять масло и сэкономить немного денег, но в какой-то момент двигатель сломается, и это обойдется вам гораздо дороже, особенно если учесть сопутствующие неудобства и невозможность пользоваться машиной. Масла в компьютерном оборудовании нет, но зато в нем есть жесткие диски (в том числе SSD) и вентиляторы, которые нужно заменять. У каждого компонента есть свой показатель MTTF (mean time to failure — среднее время работы до отказа), который показывает ожидаемый срок его службы. Компонент может прослужить и дольше, а может и меньше. Важно заранее позаботиться о наличии на складе запасных частей.

Проблемы возникают, когда оборудование достигает такого возраста, что найти для него запчасти становится трудным или почти невозможным. В поисках деталей, даже подержанных, компании нередко обращаются к площадкам типа eBay. Заранее продумайте план действий на случай такой ситуации, иначе она станет для вас неприятным сюрпризом.



## 13.2. ДОГОВОР НА ПОДДЕРЖКУ

Договоры на поддержку со временем приобрели плохую репутацию и считаются слишком дорогими, когда за поддержку вам приходится платить больше, чем за саму лицензию на продукт. В частности, этим славятся некоторые из известных поставщиков СУБД.

Смотрите на эти контракты как на страховые полисы: если что-то пойдет не так, они помогут вам восстановить работоспособность. Тем не менее, как и страховые полисы, не все договоры на поддержку одинаковы, и вы можете не получить тот объем помощи или поддержки, который вам действительно нужен. Поэтому, прежде чем подписывать долгосрочный контракт, убедитесь, что он предусматривает все, что нужно для вашей системы. Поддержка может включать следующее:

- Постоянные обновления функционала или безопасности.
- Гарантированное время ответа (не решения проблемы, а лишь подтверждения, что ваш запрос получен).
- Регулярное резервное копирование (если оборудование подключено к интернету).
- Восстановление при отказе и резервирование.
- Замена деталей/компонентов при их выходе из строя.

В зависимости от опыта и возможностей вашей организации вам может потребоваться меньший объем поддержки, чем другим. Чтобы определить, нужна ли вам поддержка (и если нужна, то какая), необходимо провести анализ рисков для каждого из компонентов и ответить на следующие вопросы:

- Если компонент выйдет из строя, повлияет ли это на предоставление услуг конечным пользователям?
- Какой объем знаний имеется внутри компании для решения проблемы и насколько хорошо обучена команда?
- Как часто возникают проблемы?
- Единственная ли это точка отказа?

Иногда наличие или отсутствие платной поддержки влияет только на приоритеты. Например, если будут проблемы с подключением вашего здания к интернету, провайдер рано или поздно их решит, но, скорее всего, в первую очередь он займется клиентами, у которых есть договор на поддержку, и лишь затем всеми остальными.

Все это актуально и для ПО. Поставщики облачных сервисов предлагают базовый уровень поддержки бесплатно, но при этом восстановление системы или ответ на ваш запрос может потребовать несколько дней, а не часы, как для тех клиентов, у которых есть платная поддержка. Такой договор — это спасательный круг; номер, который вы набираете, когда все другие возможности исчерпаны. Обязательно покупайте поддержку для старых систем, которые еще не модернизированы или не заменены, чтобы подстраховаться до тех пор, пока вы сможете что-то сделать с ними.

Также можно заключить договоры на поддержку для ключевых библиотек или ПО с открытым исходным кодом. Многие крупные проекты (Mongo, Postgres, MySQL, Elasticsearch) имеют «коммерческие» версии, которые можно приобрести, но по сути это просто договор на поддержку для того же самого продукта. Бывают недорогие коммерческие версии, где каждый релиз задерживается на некоторое время и обновление происходит только тогда, когда выйдет несколько следующих версий основного продукта. Таким образом, вы не будете сталкиваться с детскими болезнями, которые случаются в свежих релизах (иногда к этому добавляется некоторый дополнительный функционал). Кроме того, такие версии предусматривают оповещение об обновлениях безопасности, чтобы вашей команде не приходилось следить за их появлением на сайтах.

Договоры на поддержку оборудования могут сильно различаться по стоимости и уровню обслуживания. В случае дорогого контракта оборудование будет контролироваться удаленно, при этом инженеры будут самостоятельно заменять детали еще до того, как вы заметите, что они вышли из строя. Такие договоры типичны для больших SAN (storage area network, сетей хранения данных), поскольку срок службы жестких дисков ограничен. В случае более дешевой поддержки о проблеме надо будет сообщать по телефону или электронной почте, и подрядчик вышлет вам комплектующие для самостоятельной установки. За что вы заплатите, то и получите. Не всем нужен полный набор дорогих услуг, но решать тут вам.

### 13.2.1. Предоставление поддержки вашим клиентам

Говоря о поддержке, стоит рассмотреть и другую сторону медали: какую поддержку вы оказываете своим клиентам, когда они сталкиваются с проблемами в работе сервисов, которые вы им предоставляете? В крупных организациях первая линия поддержки обычно подчиняется операционному директору (СОО) или ИТ-директору (CIO). Сотрудники этой линии поддержки напрямую взаимодействуют с конечным пользователем и пытаются диагностировать и устранить неполадку. Если им нужна помощь, они передают информацию в технический отдел. Таким образом, хотя вы напрямую и не участвуете в поддержке конечных пользователей, вам все равно необходимо разработать процессы для предоставления своевременного ответа команде поддержки. Для этого обычно предусматривают SLA (соглашение о параметрах оказываемых услуг), где фиксируются параметры, на которые может рассчитывать конечный пользователь (см. главу 10). Обычно в SLA входит следующее:

- Доступность сервиса за определенный период; например, доступность 99,9% за год означает, что время простоя за этот период не должно быть более 8,7 часа.
- Сроки подтверждения запроса как в рабочее, так и в нерабочее время.



- Возможность удаленного мониторинга.
- Применение обновлений, включая исправления ошибок и обновления безопасности.
- Максимальное время, которое может быть затрачено на обработку отдельного запроса или всех запросов клиента.
- Возможность дополнительных работ и исправления проблем, возникших не по вине поставщика, за отдельную плату.

Это соглашение вы составляете не в одиночку, а совместно с отделом продаж, менеджером по продукту, генеральным и финансовым директором. Его содержимое будет определяться потребностями клиента, а также тем, что отдел продаж сможет ему продать, а вы, как технический отдел, — выполнить. Итак, если вы решили, что поддержка будет оказываться и в нерабочее время, то необходимо будет укомплектовать штат сотрудников (обеспечить отдельные команды в разных часовых поясах, либо наличие дежурного, который будет постоянно доступен по телефону, либо организовать круглосуточное присутствие сотрудников в офисе) и запланировать соответствующие расходы в бюджете.

Никогда не пытайтесь оказывать круглосуточную поддержку 7 дней в неделю, не имея продуманного плана, как конкретно она будет предоставляться. Ее нужно планировать так же, как ужин на День благодарения или Рождество. Полагаться на то, что команда много лет шла вам навстречу, или надеяться, что аварий не будет, — это не стратегия поддержки. Подходите к организации поддержки так же, как вы ожидаете этого от ваших поставщиков.

## 13.3. МОНИТОРИНГ

Самая неловкая ситуация, которая сразу говорит о нашем непрофессионализме, — это узнать о том, что что-то не работает или недоступно, от клиента. Тем не менее в какой-то момент нашей карьеры все мы в ней оказывались. Приложив некоторые усилия, можно создать культуру эффективного мониторинга, которая позволит как вам, так и вашему руководству чувствовать себя более уверенно.

### 13.3.1. Внешний мониторинг

Если ваш основной пользователь или клиент не является частью вашей компании, то необходимо контролировать работу сервиса с его позиции. Как бы очевидно это ни звучало, это означает, что средства мониторинга должны находиться вне вашей сети. Например, если вы используете AWS, то для мониторинга стоит использовать Azure или Google, или, по крайней мере, отдельную учетную запись AWS в другом регионе.

Внешний мониторинг предполагает периодическое выполнение с вашим сервисом действий, имитирующих реальное взаимодействие с клиентом. Эти

действия могут находиться в диапазоне от простого определения доступности сервиса до измерения сетевых задержек и времени отклика. При создании средств мониторинга проектируйте их таким образом, чтобы каждое сообщение о проблеме сопровождалось информацией о том, что конкретно пошло не так.

Например, вы можете просто контролировать доступность веб-страницы входа в систему. Для начала неплохо, но так вы не проверите доступность API для аутентификации клиентов. Если двинуться дальше и использовать для входа тестовую учетную запись, то вы подтвердите возможность аутентификации; однако проверяет ли это работоспособность основного сервиса? Можете ли вы запрашивать еще какие-то API, которые будут выполнять запросы (только легкие) к основной базе данных? Мы добавили всего пару дополнительных шагов, но посмотрите, насколько больше всего теперь проверяется: и сеть, и API, и база данных. Выполняйте проверку каждые 5-30 минут (в зависимости от потребностей), и, если что-то из указанного выше перестанет работать, вы очень быстро об этом узнаете.

Теперь продумайте отправку оповещений. Логично отправлять их в электронном письме, но как конкретно? Если электронная почта работает в той же сети, которая контролируется, то существует риск, что письмо не будет доставлено, потому что та же проблема, которая мешает работе сервиса, может препятствовать и отправке электронной почты. Это еще одна веская причина использовать электронную почту Google или Microsoft — независимо от того, что происходит в вашей сети, почта всегда будет доступна. Если же вы поддерживаете и используете собственный почтовый сервер, то все равно настройте для оповещений адрес не на вашем основном домене, а на бесплатном сервисе (Gmail, Yahoo и т. д.), чтобы он пересылал письма нужным людям.

### ЗАМЕТКИ С ПОЛЕЙ

#### *Ручное обновление страницы состояния*

Одна компания, с которой я работал, взаимодействовала со своим поставщиком через API. У этого поставщика был ужасно нестабильный сервис, но при этом его страница состояния всегда оставалась зеленой. Как оказалось, ее обновляли вручную, и только после того, как проходило собрание и на нем принимали решение опубликовать информацию о сбоях. Компания не хотела показывать, насколько плохой была работа их API. Вести такие игры бесполезно, особенно с учетом популярности публичных сервисов проверки, таких как Downtetector и IsItDownRightNow. Все, чего они в результате достигли, — это доказали, что их компании нельзя доверять. Нельзя скрыть что-то в мире технологий, особенно надолго.

При использовании стороннего сервиса электронной почты придерживайтесь той же схемы: создайте группу или алиас «alert», письма на который будут пересылаться всем, кому нужно. Гораздо проще управлять списком рассылки, чем отправлять письма отдельным получателям.

Еще один полезный инструмент — это публичная страница состояния с индикаторами работы каждого общедоступного компонента. Такие страницы можно видеть, например, на Amazon и Azure. Множество проектов с открытым исходным кодом позволяют создать ее без особых усилий. Опять же, убедитесь, что страница состояния не находится в основной сети организации.

### 13.3.2. Внутренний мониторинг

Взгляд снаружи полезен, но гораздо большее покрытие и детализацию обеспечивает внутренний мониторинг, помогающий предотвратить потенциальные проблемы до того, как они повлияют на клиентов, — например, контроль использования дискового пространства, чтобы освободить место прежде, чем сервер перестанет работать. Это простой пример, но он иллюстрирует эффективность внутреннего мониторинга. Контроль параметров основных компонентов, таких как процессор, память и диск, каким бы банальным он ни казался, поможет обнаружить множество ситуаций, потенциально ведущих к серьезным сбоям.

Определите, какое состояние компонентов следует считать «нормой». Это поможет определить и проанализировать всплески активности или необычное использование ресурса. Если внезапно уменьшился объем используемой памяти, это может говорить о том, что какой-то процесс умер или что он работает нестабильно — запускается и тут же падает. Обращайте внимание на непонятные отклонения от нормы — рост нагрузки, не связанный с увеличением трафика от пользователей или приходом новых клиентов.

С учетом особенностей вашей системы определите критерии, на основании которых вы будете принимать решение об обновлении или расширении ваших сервисов (добавлении ресурсов для веб-серверов или баз данных) или же, наоборот, об освобождении ресурсов, которые больше не нужны. Эти метрики жизненно важны для обеспечения нормальной работы. Другой уровень мониторинга — это журналы, в которых приложения фиксируют все события во время работы. Каждое приложение генерирует такие журналы, которые могут храниться как в отдельных файлах на диске, так и в службе событий Windows или в Amazon CloudWatch. Источников данных много.

Современный способ обрабатывать все эти данные — перенаправить все журналы в централизованную службу (такую как Elasticsearch) и использовать ее для сравнения, поиска аномалий, мониторинга и оповещения. Так вы сможете не только проводить анализ всех журналов в одном месте, но и легко обнаруживать связи между разными компонентами, потому что проблемы в одном из них могут быть признаком другого, более серьезного сбоя. Классический пример: увеличение количества SQL-ошибок в веб-приложении указывает на проблемы с сервером базы данных, где долго выполняющийся запрос мог заблокировать таблицы. Такие зависимости бывает трудно обнаружить, поэтому для анализа причин проблемы придется проверять различные подсистемы.

В идеале все журналы должны в итоге попадать в общее хранилище. Это журналы как операционных систем, так и маршрутизаторов, файерволов, различных приложений, VPN, сетей, а также журналы аутентификации пользователей, клиентов и процессов. На их основании можно сделать удивительно много полезных наблюдений — например, вы можете обнаружить, что каждый раз при входе кого-то в систему повторяется один и тот же набор действий. Информации слишком много не бывает.

### ЗАМЕТКИ С ПОЛЕЙ

#### *Централизованное время*

Для современных ОС это уже обычно не является проблемой, но все же лучше убедиться, что все системы синхронизированы и используют один и тот же часовой пояс, обычно это UTC (всемирное координированное время). Во всех операционных системах есть средства синхронизации времени, которые периодически подстраивают часы. Это очень упрощает ретроспективный анализ журналов, поскольку на единой шкале времени сразу становятся видны причинно-следственные связи.

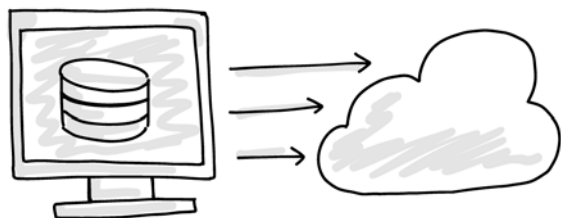
Создание информационных панелей (дашбордов), отображающих информацию о текущем состоянии ваших систем, потребует времени, проб и ошибок. И они будут развиваться по мере того, как ваше понимание работы систем будет углубляться. Ваша цель здесь проста: выиграть как можно больше времени для корректирующих действий, прежде чем проблемы затронут работу компании. Не допускайте ситуации, в которой единственным средством мониторинга станут звонки ваших клиентов. Это не способствует повышению доверия ни к вам, ни к вашей компании.

## 13.4. РЕЗЕРВНОЕ КОПИРОВАНИЕ И ВОССТАНОВЛЕНИЕ

Почти все компании утверждают, что у них есть резервные копии, по крайней мере важнейших компонентов, но, как сетует Райан Берч, директор по информационным технологиям в New Harbour Capital, «выглядит очевидным, но часто забывается, что резервная копия, из которой никогда не проводилось восста-

новление, — это не резервная копия, а просто способ расходования хранилища, что-то, что создает иллюзию безопасности, как страховочный трос, который ни к чему не прикреплен».

Организация резервного копирования — это сложная



задача, и, пока вы не выполните полное восстановление системы из резервной копии, вы не сможете узнать, действительно ли в ней сохраняются все компоненты.

Первый вопрос, на который вы должны ответить: что сохранять? Только данные, или же данные и конфигурацию, или же и данные, и конфигурацию, и само приложение? Ответ зависит в первую очередь от цели резервного копирования. Она может быть следующей:

- Восстановление в случае сбоя.
- Периодическая фиксация состояния для контроля соответствия требованиям.
- Использование для целей тестирования/разработки.

В подавляющем большинстве случаев резервные копии создаются для обеспечения бесперебойной работы, и в этом случае важно время восстановления. С каждой минутой простоя компания теряет деньги, а также репутацию и доверие к бренду. Отработанная стратегия резервного копирования станет залогом вашего душевного спокойствия, поскольку в случае аварии вы будете иметь четкий план действий и понимать сроки восстановления.

Обычно рекомендуется хранить файлы резервных копий в зашифрованном виде в отдельной сети или в отдельной учетной записи в облачном сервисе. Но часто, особенно в небольших организациях, резервные копии хранятся не отдельно, а на том же самом компьютере. Исходите здесь *не* из того, что сбой возможен, а *из того*, что в какой-то момент он обязательно случится.

Необходимо делать резервное копирование не только для тех сервисов, которые вы обслуживаете самостоятельно, но и для всего, что хранится в сторонних сервисах. Например, регулярно копируйте исходный код, размещенный на GitHub или Bitbucket, в безопасное и управляемое вами место. Часто забывают и о Salesforce, особенно если он применяется в основном как облачная база данных. Если вы используете Google Диск, Office 365, Dropbox — любой сервис, в котором вы храните данные, — спросите себя, к каким последствиями приведет потеря доступа к нему или блокировка учетной записи?

Чтобы создать надежную стратегию резервного копирования и восстановления, требуется много усилий; кроме того, понадобится выполнить первое архивирование и копирование данных и настроек. Но после этого все должно работать на автопилоте, а отчеты должны регистрироваться в централизованной системе журналирования, чтобы вы получали оповещения, если резервное копирование не было выполнено или произошла какая-то ошибка.

Отработайте восстановление системы. Создайте план действий и разместите его в доступном месте, чтобы любой, кто имеет необходимые доступы, мог легко собрать и восстановить систему. Проводите учения не реже, чем раз в год (или даже чаще), чтобы убедиться, что есть резервные копии для всех необходимых компонентов.

### 13.4.1. Частота/срок хранения

Два насущных вопроса — как часто следует делать резервные копии и как долго их следует хранить. Рассмотрим каждый из них по очереди.

Частота резервного копирования главным образом зависит от того, как часто меняются данные. Если это происходит только раз в неделю, то при ежедневном резервном копировании архивы почти не будут различаться. Слишком частое резервное копирование может привести к чрезмерному расходованию ресурсов и нагрузке на диски. Если вам требуется восстановление с высокой точностью, стоит учесть это требование при первоначальном проектировании вашего хранилища: например, использовать репликацию данных в несколько разных регионов в режиме, близком к реальному времени.

Раньше срок хранения резервных копий зависел от доступного места. С появлением облачных хранилищ и оплатой по мере их использования лимитов по размеру больше не существует. Теперь акцент смещается в сторону определения, какие данные являются полезными для восстановления.

Раньше рекомендовалось сохранять последние три резервные копии на случай их потери или сбоя копирования. Теперь, с хранилищами большого объема, использованием контрольных сумм и шифрованием, это не проблема. В большинстве случаев будет достаточно одной копии. Однако если вы понимаете, что резервная копия может понадобиться для восстановления только части данных, то вам стоит хранить архивы подольше.

Допустим, вы узнали, что был случайно удален какой-то набор данных, но это обнаружилось только через несколько дней или недель — что нередко случается, когда вашим сервисом пользуется несколько разных клиентов. В такой ситуации лучше иметь возможность вернуться и для части данных восстановить состояние, в котором они были в заданный момент времени. Поэтому хранение данных в течение нескольких месяцев может быть оправдано.

#### ПРИМЕЧАНИЕ

Это не означает, что вам нужно хранить ежедневные копии всех данных за шесть месяцев. Храните разное количество данных в зависимости от их возраста: например, по прошествии семи дней оставляйте только еженедельные архивы; через четыре недели оставляйте только ежемесячные архивы. Так вы снизите затраты на хранение.

### 13.4.2. Безопасность архива

Прежде чем завершить эту тему, необходимо упомянуть о том, как важно обеспечить безопасность полученных архивов: из-за недостаточного внимания к их защите и легкой доступности они могут стать источником утечек конфиденциальных данных. Вы же не хотите, чтобы все усилия по обеспечению безопасности

и многоуровневая система контроля доступа стали бесполезными только потому, что резервная копия доступна для всех, у кого есть полномочия DevOps.

Сотруднику, ответственному за создание и проверку резервных копий, необязательно видеть данные внутри архива. Управляя доступом к ключам шифрования, вы можете ограничивать доступ к архивам резервных копий и обеспечивать их сохранность.

Уделяйте безопасности резервных копий такое же внимание, как и безопасности систем, к которым они относятся. Не позволяйте резервным копиям стать источником утечки.

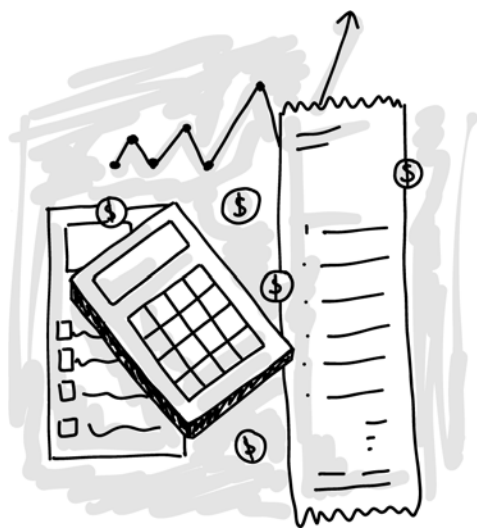
## 13.5. КОНТРОЛЬ БЮДЖЕТА

Как мы уже говорили в предыдущих главах, планирование бюджета позволит как вам, так и бизнесу понять, какие требуются ресурсы. Но это только часть связанной с бюджетом работы. Вторая часть задачи состоит в том, чтобы соблюдать этот бюджет. Для этого необходимо постоянно фиксировать все расходы и траты и не допускать сюрпризов.

Хорошие новости: если вы используете облачные сервисы — в них легко настроить ежедневные предупреждения о превышении заданных лимитов. Расходы здесь могут расти как снежный ком, особенно если разработчики создают новые сервисы (и, возможно, с большим количеством ресурсов, чем нужно), а затем забывают их удалить.

Полезно не реже чем раз в месяц анализировать расходы на облако, проходиться по всем используемым ресурсам и проверять, все ли из них нужны. Пользуйтесь функционалом тегов, чтобы отслеживать услуги, относящиеся к тому или иному проекту или отделу. Очень легко создать новый сервер и затем забыть, для чего он был изначально нужен, а со временем такие серверы становятся неприкосновенными. Люди боятся, что что-то сломается, если их отключить, поэтому проще их не трогать (и продолжать платить за них). Таким образом стоимость облачных сервисов становится заоблачной.

Еще одна причина, по которой могут расти счета, — это увеличение количества используемых ресурсов. Надежный и грамотно спроектированный проект, ориентированный на облачные сервисы, будет автоматически увеличивать и уменьшать потребление ресурсов в зависимости от нагрузки. По мере развития



продукта отдел продаж будет привлекать новых крупных клиентов, поэтому использование сервисов будет расти, а вместе с ним и расходы. Это правильно и нормально. Тем не менее вы должны понимать точную причину увеличения расходов на облачные сервисы, чтобы, когда финансовый директор придет с вопросами, можно было вместе с ним порадоваться тому, что дело в популярности вашего продукта.

В любом случае регулярная проверка и актуализация бюджета позволит избежать сюрпризов, и если вам потребуется внести коррективы в первоначальную смету — не проблема, эти расчеты были всего лишь оценкой и со временем они уточняются.

Если вы не применяете облачные сервисы, то у вас будет больше единовременных расходов — на покупку серверов, добавление памяти или дисков. Принцип здесь тот же, что и для облака: анализируйте загрузку ресурсов, чтобы убедиться, что они используются максимально эффективно. Объем хранилища может увеличиваться бесконечно. Если вы видите, что он постоянно растет — рассмотрите вариант переноса долгосрочного хранилища в облако и платите за тот объем, который будете использовать, без всяких забот об оборудовании.

Стоимость лицензий и расходы на поддержку спрогнозировать и заложить в бюджет легче, но эти расходы тоже могут увеличиваться и увеличиваются из года в год. Стоимость лицензий коммерческих СУБД, как правило, будет расти, поэтому по мере роста организации проверяйте наличие альтернатив (более дешевых или бесплатных).

Если вы начинаете использовать новый программный продукт, стоит заключить договор на его поддержку, но, когда придет время продления договора, оцените, окупились ли затраты и действительно ли вам нужно это ПО. Имейте в виду, что многие компании-разработчики начинают брать деньги не за поддержку, а за использование ПО, то есть продавать подписку на продукт. Как технический директор, вы несете ответственность за расходы своего отдела, и вы должны искать оптимальное соотношение цены и качества в рамках согласованного бюджета.

## ИТОГИ

- Ведите учет оборудования, чтобы эффективнее планировать профилактическое обслуживание.
- Определение срока службы компонентов поможет спланировать закупки.
- Договоры на поддержку — ваши страховые полисы. Убедитесь, что они покрывают все, что вам нужно.
- Мониторинг с позиции клиента поможет выявить проблемы до того, как клиенту придется сообщать вам о них.



- Благодаря внутреннему мониторингу систем вы будете знать их состояние, что позволит вашей команде предотвращать потенциальные проблемы до того, как они станут реальными.
- Периодичность создания резервных копий в основном зависит от того, как часто изменяются данные и насколько старые данные вы хотите восстанавливать при необходимости.
- Необходимо регулярно отрабатывать восстановление, чтобы проверять полноту резервных копий.
- Еженедельная или ежемесячная проверка бюджета позволит вам избежать сюрпризов.

## ПРОВЕРЬТЕ СЕБЯ

Что из перечисленного ниже вы уже сделали?

- Провели инвентаризацию всего оборудования и ведете журнал обслуживания.
- Составили список деталей, для которых закончилась гарантия и которые могут скоро выйти из строя, и определили, есть ли они в наличии.
- Заключили договоры на поддержку для ключевых компонентов вашей системы.
- Используете инструменты внешнего мониторинга.
- Используете инструменты мониторинга внутренних показателей.
- Настроили автоматическое резервное копирование, при этом архивы хранятся за пределами вашей инфраструктуры и надежно защищены.
- Периодически проверяете возможность восстановления системы из резервных копий.
- По крайней мере ежемесячно проверяете расходование бюджета.

# 14

## *Рост компании*

---

### **В этой главе**

- ✓ Каким образом происходит покупка компании или ее рост за счет поглощений
- ✓ Всегда будьте готовы представить компанию в лучшем свете
- ✓ Как провести due diligence и выяснить то, что действительно важно
- ✓ Грамотное принятие и передача должности

Ваша компания со временем будет расти, и это может произойти за счет следующих сделок:

- Покупка другой компанией.
- Покупка частным акционером или инвестиционной группой.
- Поглощение меньшей компании.
- Слияние с другим бизнесом.

Для каждого из этих сценариев существуют свои сложности и стратегии достижения успеха, которые мы рассмотрим в этой главе. И у них есть общая проблема: они могут отнять массу сил и времени, если вы не знаете, как правильно себя вести.

## 14.1. ИНВЕСТОРЫ

Руководство компании, частью которого вы, как технический директор, являетесь, всегда находится в поисках выгоды для клиентов, сотрудников и инвесторов. Успешная сделка служит интересам всех вовлеченных сторон. Независимо от того, покупают ли вас или вы кого-то покупаете, такая сделка — это важное подтверждение того, что ваша организация действует правильно.

Какова в этом роль вас, технологического лидера и стратега организации? Вы здесь отвечаете за презентацию результатов уже выполненной работы и планов на будущее, без искажений и преувеличений.

Если к вам проявляют интерес инвесторы, например частная инвестиционная компания (private equity, PE), то вся ваша работа будет контролироваться гораздо серьезнее. Вопреки распространенному мнению, инвесторы — это не старики с деньгами. Подавляющее большинство этих компаний составляют пенсионные фонды, университетские фонды целевого капитала и семейные трасты. Относитесь к этому так: благодаря вашему успеху пенсионеры получают пенсии, а студенты — высшее образование.

Причина повышенного контроля над вашей компанией понятна: использование чужих денег требует особой осторожности и внимательности. Частные инвестиционные компании управляют и распоряжаются чужими деньгами: им поручают инвестировать определенную сумму в относительно безопасный актив, как правило, на 3–5 лет, и не только благополучно вернуть вложенные средства, но и получить прибыль. Таким образом, при заключении сделки инвестиционная группа зарабатывает деньги или возвращает вложения, продавая свою долю. Обратите внимание на разницу с работой венчурного капитала, который обычно инвестирует в идею и построение компании на ее основе, а не в уже существующий и успешный бизнес.

Разные компании придерживаются разных инвестиционных стратегий. Некоторые стремятся быть единственным инвестором и иметь полный контроль; другие предпочитают действовать совместно с кем-то еще. Есть инвесторы, готовые держать свои вложения дольше стандартных 3–5 лет; есть такие, которые стараются не вмешиваться в управление компанией, оставляя его существующему руководству. Однако инвесторы все чаще стремятся принимать активное участие в развитии компании, предлагая использовать их опыт и связи. Этим они необыкновенно ценны для компании, поскольку позволяют ей воспользоваться чужим практическим опытом в безопасной, контролируемой среде. В языке инвесторов существует понятие операционного партнера — это представитель инвестора, который готов засучить рукава и подключиться к вашей работе. Совет: если вы будете работать с операционным партнером, приветствуйте и максимально используйте возможности, которые это вам предоставляет.

Не бойтесь — партнер не думает отнимать у вас работу. Он хочет поддержать вас, помочь расти и успешно выполнять ваши задачи. Конечно, если он сочтет, что вы не подходите для занимаемой должности, то он может заменить вас другим

человеком. Это наихудший сценарий, но умолчать о нем было бы неправильно. Однако, чтобы до этого дошло, вам нужно будет постараться проигнорировать все получаемые вами советы и наставления. К счастью, у вас есть эта книга, и она поможет заполнить пробелы в ваших знаниях.

## 14.2. ИМПРОВИЗИРОВАННЫЙ ПИТЧ

«Будь готов», — гласит девиз бойскаутов. Невозможно предсказать, в какой момент к вам заявятся люди в костюмах и попросят рассказать, как обстоят дела в компании и, что еще более важно, каковы ее планы и стратегия. В главе 3 мы подробно рассказали о том, как важно всегда иметь стратегию под рукой, — в том числе для таких ситуаций.

### ЗАМЕТКИ С ПОЛЕЙ

*Будь готов*

Часто, особенно на начальных этапах, такие встречи не афишируют и проводят практически без подготовки. Так, однажды генеральный директор в 17:00 попросил меня присоединиться к обеду с инвестиционными партнерами, которые захотели с нами познакомиться. Никаких презентаций и докладов, просто старая добрая беседа за обеденным столом (где для простого айтишника было чересчур много разных столовых приборов).

Не стоит недооценивать влияние и долгосрочные последствия этих встреч: от них зависит, будет ли сделка похоронена прямо сейчас или вы сможете двигаться дальше. Именно вы можете вселить в партнеров уверенность (но не самонадеянность) в то, что у вашей технологии, хотя она может быть и неидеальной, есть будущее, а у компании есть план дальнейшего роста.

## 14.3. В НОГУ С ИНВЕСТОРОМ

Понимание того, как инвесторы расставляют приоритеты и зарабатывают деньги, поможет лучше спланировать и реализовать вашу стратегию. Как мы уже говорили, большинство из них зарабатывают на сделках, которые заключаются обычно через 3–5 лет после первоначальных вложений. Это может быть продажа бизнеса другой инвестиционной компании или стратегическому инвестору (компании, которая заинтересована в приобретении продукта, клиента или талантливых сотрудников).

Если вы сравните продажу компании с продажей дома, то поймете, что необходимо представить компанию, как и дом, в лучшем свете. Не стоит показывать потенциальным покупателям дом в то время, когда на кухне идет ремонт. Они посмотрят на это и сделают логичный вывод, что заканчивать

ремонт придется им. Лучше было вообще его не начинать и предоставить покупателям полную свободу творчества. Но при чем тут вообще кухня, ведь вы технический директор?

Дело в том, что этот принцип следует применять, приступая к любым крупным проектам, особенно длительностью в год и более. Если вы уверены в том, что сделка состоится, вы должны определить, сможете ли завершить проект вовремя. В таких случаях всегда лучше обращаться за советом к генеральному директору, который лучше понимает перспективы.

Если вы сотрудничаете с операционным партнером от вашего текущего инвестора, обязательно проконсультируйтесь с ним. Он лучше всех поможет вам взвесить все «за» и «против» проекта и понять, удастся ли завершить его вовремя, чтобы занять определенную позицию на рынке.

Чаще всего под тщательный анализ подпадают проекты модернизации, в ходе которых все стороны получают выгоды от замены устаревшей технологии на что-то более подходящее. Новый инвестор предпочел бы, чтобы такие проекты завершались, пока все ресурсы и знания есть в наличии, и чтобы никто не нервничал и не отвлекался на новых владельцев.

Даже если вы знаете, что проект не будет завершен вовремя, ему может быть дан зеленый свет. В таком случае тщательно определите его основные этапы. Возможно, вам придется скорректировать план таким образом, чтобы к тому времени, когда произойдет сделка, путь к завершению проекта был ясно виден. Возвращаясь к аналогии с ремонтом кухни — вы можете закончить, по крайней мере, прокладку труб и электропроводки, чтобы кухней можно было хоть как-то пользоваться.

Привлеките к планированию проекта как можно больше заинтересованных сторон. Соберите от них информацию и не обещайте слишком многого. Предусмотрите больше времени на непредвиденные обстоятельства, потому что во время, оставшееся до сдачи проекта, вы и некоторые члены вашей команды можете быть параллельно заняты подготовкой к сделке — документами, презентациями и проведением due diligence.

## 14.4. ПОДГОТОВКА К ИНТЕГРАЦИИ

Одним из распространенных видов приобретения является присоединение (tuck-in). В этом случае большая компания полностью поглощает меньшую. Сделка с небольшой компанией, в отличие от более крупной, не имеет высокой сложности или больших накладных расходов — скорее всего, понадобится оформить лишь десяток сотрудников, работающих в конкретной области. Удобный вариант для инвесторов, поскольку это очень быстрый способ обеспечения роста и масштабирования активов. Это не слияние в классическом смысле, когда обе стороны имеют почти равные права. Рассматривайте присоединение скорее как покупку полностью укомплектованного и самостоятельно работающего отдела.

Тем не менее для успеха здесь необходима полная интеграция компаний, чтобы дальше они могли двигаться в одном направлении. Это требует усилий и тщательного планирования. Никто не захочет получить в результате набор совершенно разных технологических стеков и команд с разными навыками, которые не смогут пользоваться общими ресурсами.

Это не означает, что придется переписывать всю присоединенную систему; но технология должна быть совместима с общей инфраструктурой, такой как оборудование, средства работы с журналами, системы безопасности и инструменты мониторинга. К примеру, предположим, что вся ваша инфраструктура находится в AWS, а присоединяемая компания использует Azure. Целесообразно тоже перевести ее на AWS.

В зависимости от размера компании вы можете либо присоединять, либо быть присоединенными. В любом случае существуют общие принципы, которые необходимо учитывать в вашей стратегии.

Успех присоединения зависит от скорости интеграции и от того, насколько быстро приобретенный бизнес начнет приносить пользу. Время интеграции — важный фактор для инвесторов при оценке целевых активов. Следовательно, ваш стек технологий должен быть способен быстро присоединять другие проекты или быть присоединенным к ним. Этому условию отлично удовлетворяют современная микросервисная архитектура или набор API. Использование API позволяет выиграть время, чтобы определить, следует ли переписывать бэкенд или переносить его на что-то, более соответствующее инфраструктуре нового владельца. Также при интеграции нужно не забыть про объединение систем контроля версий и процессов сборки. Вопрос адаптации команды мы рассмотрим позже в этой главе.

#### ЗАМЕТКИ С ПОЛЕЙ

ИТ

Технический директор обычно не занимается вопросами организации электронной почты и хранения рабочих файлов, но это зависит от размера организации. Если вам все же придется этим заниматься, то учтите, что эти задачи следует решить в первую очередь: ничто не поможет новой команде почувствовать себя как дома лучше, чем привычные электронная почта и чат.

### 14.5. DUE DILIGENCE

Слова *due diligence* могут вселить страх в сердца даже самых опытных технических директоров, уверенных, что это обязательно будет допрос с пристрастием и что все, что они сделали, будет подвергнуто критике. Любой, кто видел классический фильм «Офисное пространство» («*Office Space*», 1999), помнит сцену,

когда два менеджера (обоих зовут Боб) собеседуют всех сотрудников, решая, кого уволить. Вот зловещая фраза, которая, по мнению многих, воплощает в себе всю суть due diligence: «Чем вы вообще здесь занимаетесь?»

Due diligence — это возможность для покупателя понять, что он на самом деле покупает, и в том числе узнать обо всех подводных камнях. Возвращаясь к аналогии с покупкой дома, результат due diligence похож на отчет об осмотре объекта недвижимости, составляемый профессионалами. Они указывают в отчете, на что покупатель должен обратить внимание перед покупкой, чтобы избежать неприятных сюрпризов (например, на протекающую крышу или на то, что кондиционер требует замены).

Зачастую даже крупные консалтинговые компании, проводя due diligence, могут упустить важные детали и контекст. Многие из них забывают интересоваться «зачем». Проведение due diligence не должно быть прохождением по таблице с заранее подготовленными стандартными вопросами, потому что так оно неизбежно превращается в формальное дознание, исключающее возможность отметить достижения и интересные особенности.

Когда кто-то что-то покупает — телевизор, машину, дом или компанию, — он делает это с конкретной целью: решить свою проблему. Если вы планируете использовать телевизор только для Xbox, то вам не нужна возможность установки на него приложений и неважно, сколько каналов он может принимать. Вас интересует только разрешение и частота обновления экрана. Другой покупатель собирается смотреть местные новости и Netflix, и его процедура оценки будет совсем не такой, как у геймера. Это вопрос соответствия цели, то есть «зачем».

Прежде чем начинать due diligence, определите причину покупки, то есть планы и ожидания от нее. Это позволит обеим сторонам сделки сосредоточиться на действительно важных и актуальных вопросах. Таким образом, due diligence — это диалог, а не сцена допроса подозреваемых, прикованных наручниками к столу, из сериала «Полиция Чикаго» («*Chicago PD*»).



Этап due diligence — довольно поздняя стадия сделки по приобретению, и к этому времени «зачем» должно быть уже хорошо понятно. Таким образом, цель проведения оценки проста: узнать, способна ли приобретаемая технология делать то, что нам нужно, а если нет, то сколько денег и усилий потребуется, чтобы она смогла это делать?

Мы рассмотрим процесс due diligence с точки зрения каждой из участвующих в нем сторон. Вы можете как проводить оценку due diligence, так и быть в центре

ее внимания. Начнем с первого варианта, поскольку он дает больше понимания и общей информации, а затем перейдем на другую сторону.

### 14.5.1. Что входит в due diligence

В зависимости от того, на каком этапе находится ваша карьера, вы, скорее всего, хотя бы 1–2 раза уже проводили due diligence. Возможно, процедура называлась иначе — скорее всего, оценкой. Она могла проводиться не для компании, а для конкретного ПО, языка, фреймворка или сервиса. Готовясь к ней, вы заранее изучали все, что могли, формулировали свои мысли и составляли вопросы.

Когда вы общались со специалистами по оцениваемой системе (ее пользователями или же техническим менеджером), вы спрашивали их о вещах, относящихся к проблеме, которую эта система должна решать. У вас наверняка не было таблицы с абстрактными вопросами, ответы на которые ни на сантиметр не приблизили бы вас к принятию обоснованного решения. Вместо нее у вас был список тем, относящихся в основном к эксплуатации продукта, обсудив которые вы бы поняли, в правильном ли направлении движетесь. В этом и состоит разница: так вы понимаете «зачем» сделаны те или иные вещи, и на основании этого можете определить, подходит ли оцениваемый продукт для ваших целей.

Тот же подход применяется при оценке технологий всей компании — в этом случае требуется просто немного больше усилий, но сам процесс будет не сложнее. Вы здесь эксперт по технологиям, который занимается решением практических проблем. Большая часть из того, что вы увидите, будет либо «кока-колой», либо «пепси» (они обе хороши — просто разные), либо какими-то мелочами, которые легко решить. Так что именно вы, а не сотрудник консалтинговой компании, лучше всего подходите для проведения due diligence. Вы способны быстро распознать вещи, требующие особого внимания, поскольку они могут стать проблемными.

Рассмотрим основные области, которые требуют подробного изучения. При проведении обследования помните, что, если сделка состоится, все обнаруженные проблемы или вопросы придется решать вам. Но вполне возможно, что вы также узнаете что-то, что поможет вашей команде и что можно будет использовать в дальнейшей работе. Может быть, вы найдете решение для проблемы, с которой давно боретесь, или узнаете, как делать что-то более эффективно. Всегда найдется что-то полезное.

#### ТЕХНОЛОГИЧЕСКИЙ СТЕК

Насколько отличаются их технологии от ваших? Достаточно ли у вашей команды знаний, чтобы использовать их? Если они применяют устаревшие технологии, то что потребуется для их дальнейшей поддержки, в том числе будут ли проблемы с подбором специалистов или обслуживанием оборудования или операционных систем? Есть ли что-то, что можно оптимизировать или сделать



более эффективным с помощью современных технологий? Если понадобится все переписать — каких усилий и затрат это потребует и вызовет ли это перебои в работе?

### **ПРАВА ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ (ПИС)**

Какая часть их технологий уникальна? Что является объектом ПИС («секретным ингредиентом») — алгоритмы и код — либо это способ применения инструмента? Имеются ли патенты на технологии, и если да, то кому они принадлежат — физическим лицам, компании, кому-то еще? Кто (как ни странно это звучит) на самом деле является владельцем исходного кода? Ответ на этот вопрос не всегда очевиден, особенно если компания отдавала основную разработку на аутсорсинг.

### **ДАННЫЕ**

Как сохраняются и используются данные? Какой их объем доступен и может применяться для аналитики? Каковы основные источники и потребители данных? Кому принадлежат данные и какие есть ограничения на их использование? Как регулируется работа с данными (должен ли выполняться регламент Евросоюза по защите персональных данных (GDPR)?) и есть ли в компании конфиденциальные данные, подпадающие под действие нормативных требований (например, PCI, PCI или HIPAA)?

#### **ЗАМЕТКИ С ПОЛЕЙ**

*Это мои данные — буду делать с ними что хочу*

Я часто сталкивался с компаниями, которые считали, что если адрес электронной почты попал к ним в базу данных, то они могут делать с ним все, что им заблагорассудится. Но это не так. Как были получены эти данные? Если пользователь не давал согласия на использование (или продажу) своих данных, то они бесполезны. Я не раз наблюдал, как люди бледнеют, когда понимают, что данные, которые у них есть, далеко не так ценны, как они думали.

### **КОМАНДА**

Какова структура существующей технической команды? Каковы навыки и опыт сотрудников? В небольших организациях люди часто работают с самого момента основания компании, и всему, что они умеют, они научились на этой работе. Если вы имеете дело с таким случаем, то чему и у кого эти люди учились?

Кто из инженеров является более опытным, чем остальные, и незаменимым? Лучший способ выяснить это — спросить: «Чье отсутствие сильнее всего заметно в компании, когда человек уходит в отпуск?» Большинство команд ответят на этот вопрос быстро и не задумываясь. Небольшие компании слишком

полагаются на верность и трудолюбие одного человека. Иногда это тот самый, кто спроектировал все системы, и, вероятно, именно он отвечает на все ваши вопросы в связи с due diligence.

Привлекает ли компания подрядчиков или аутсорсинговые фирмы для поддержки или разработки? Каковы их зоны ответственности и сроки договоров? Вам необходимо выяснить, могут ли какие-либо договорные отношения вызвать проблемы в будущем (например, с правами собственности или с прекращением договора).

### **ПОДДЕРЖКА**

О каких проблемах обычно сообщают пользователи или клиенты? Сколько времени тратится на поддержку по сравнению с разработкой новых функций?

Сколько усилий или времени обычно занимает в компании подключение новых клиентов? Участвует ли в этом техническая команда? Какие серьезные сбои или проблемы были за последние 12 месяцев? Как они повлияли на клиентов и что было сделано, чтобы они не повторились? Что не дает руководству спать по ночам и что бы они сделали в первую очередь, если бы у них была волшебная палочка?

### **БЕЗОПАСНОСТЬ**

Как в целом в компании организовано обеспечение безопасности? Когда в последний раз менялись основные пароли? Какие доступы имеют разработчики? Отделен ли продакшен от окружения для разработки?

Если компания работает с учетными данными пользователей, то как хранятся пароли в базе данных (открытым текстом или в виде хэша с солью)? Вы удивитесь, сколько компаний до сих пор хранит пароли в открытом виде!

Как организовано тестирование, проводились ли тесты на проникновение, на SQL-инъекции в продакшене? Требования безопасности либо могут учитываться с самого начала, либо про них вспоминают в какой-то момент позже — вам нужен первый вариант.

### **ИСХОДНЫЙ КОД**

Если приобретаемая компания разрабатывает исходный код, очень важно изучить его общее качество и подходы к разработке. Вникнуть в каждую строчку кода, конечно же, не получится. Во-первых, у вас нет времени, а во-вторых, за несколько часов невозможно понять результат разработки, ревью, переписок и исправлений, сделанных множеством разработчиков за годы. Лучше выбрать несколько мест, подробно изучить их и на основе этого сделать вывод об общем состоянии.

Все мы в своей жизни писали код, по которому лучше бы нас не оценивали, поэтому выберите относительно недавний коммит. Найдите его по истории в системе контроля версий или начните с какого-то из тикетов и проследите работу над ним от создания задачи до разработки и деплоя на продакшен.

Еще одна сфера, на которую следует обратить внимание, — безопасность разработки. Изучите, как в коде организована работа с параметрами конфигурации, именами пользователей и паролями. Если в инфраструктуре имеется большое количество API — проанализируйте, как они устроены, единообразно ли используются? Как обрабатываются ошибки и исключения? Код основан на предположениях о том, что ничего не сломается, или в нем предусмотрены различные проверки?

О зрелости кода многое скажет его общая структура. Это просто куча функций или же все организовано в виде объектов, которые можно использовать при разработке нового функционала? И если говорить об использовании готового кода, то какие библиотеки с открытым исходным кодом применяются для решения типовых задач, и применяются ли? Соответствуют ли лицензии этих библиотек требованиям бизнеса, соблюдаются ли условия лицензий?

Наконец, ознакомьтесь с системой журналирования. Как мы увидели в предыдущей главе, она помогает мониторить продакшен в реальном времени и оперативно диагностировать проблемы. Приводятся ли журналы к единому формату, чтобы можно было легко выявлять исключения и ошибки среди информационных и прочих не требующих реагирования записей?

В компаниях, занимающихся разработкой, стоит изучить организацию работы, включая ревью кода, работу с системой контроля версий, тестирование, сборку и деплой. Небольшие команды обычно гораздо меньше соблюдают формальности, и в них все это может быть в зачаточном состоянии. Не нужно осуждать их — просто отметьте это как особенности. Неважно, что и как они делают и нравится вам это или нет: результат их работы кому-то нужен, иначе вы бы даже не и подумали приобретать эту компанию.

## **ПРОДАКШЕН**

Последнее, что необходимо выяснить, — как выполняется деплой и как организован мониторинг продакшена. Вы можете спросить, для эффективности добавив драматическую паузу: «Откуда вы знаете, что прямо сейчас у вас все в порядке?» Ответ на этот простой вопрос, в том числе честное пожимание плечами, говорящее о том, что ваш собеседник понятия не имеет, все ли в порядке, многое скажет о том, как в компании организован продакшен.

Вам нужно понять, что необходимо для бесперебойной работы проекта. Помните: если сделка пройдет успешно, это станет вашей заботой. Итак, выясните, какие требуются ежедневные проверки, какие журналы нужно просматривать и по каким признакам можно понять, что что-то может пойти не так.

Изучите план обеспечения отказоустойчивости. Как работает резервирование при сбоях? Обсудите слабые места и точки отказа и что с ними можно сделать. Сюда также относится вопрос резервного копирования и восстановления — при его обсуждении также выясните, есть ли доступная документация или инструкции по восстановлению.

Еще в рамках этой темы узнайте, кто отвечает за оплату сторонних сервисов и продление подписок и договоров с ними. Нередко (особенно в небольших компаниях) учетные записи таких сервисов, как AWS, GitHub или GoDaddy, «принадлежат» программисту, который начал с ними работать, просто потому, что к ним привязана его банковская карта и с нее списывается ежемесячный платеж, который затем возмещается компанией.

### ЗАМЕТКИ С ПОЛЕЙ

*Google забыл оплатить счет за домен*

В современном мире, где все взаимосвязано, сервисы могут оказаться отключенными по самым нелепым причинам, в том числе если не оплатить продление домена или SSL-сертификата. Такое произошло с Blogger, одним из сервисов технологического гиганта Google. Кто-то не оплатил продление домена blogger.in, поэтому контроль над ним был на некоторое время потерян. См. <https://www.androidpolice.com/2020/07/16/someone-at-google-forgot-to-renew-a-blogspot-domain-still-offline-a-week-later/>.

Большинство из найденных проблем вполне решаемы и рассматриваются в этой книге, но, прежде чем заключать сделку, важно определить, кто чем владеет и на каких основаниях.

## 14.5.2. Проведение due diligence

Все вышеупомянутые темы — это всего лишь то, что нужно иметь в виду, чтобы вы могли начать готовиться к изучению вопроса, насколько приобретаемая компания соответствует вашим целям. Теперь нужно понять «как». Не устраивайте допрос и не задавайте вопросы по заранее заготовленному списку. Придя с перечнем вопросов, вы не построите правильные взаимоотношения и не узнаете нужные вам подробности. Вместо этого просто разговаривайте — пусть два технаря рассказывают друг другу о себе и своей работе. Вашему собеседнику должно быть интересно и комфортно, он должен спокойно делиться информацией, а не жалеть, что не взял с собой адвоката!

Всегда поддерживайте дружелюбный настрой, цените усилия другой стороны и хвалите ее за результаты, которых она добилась. Повторюсь, что сам факт того, что к компании проявляют интерес покупатели, говорит о том, что дела в ней идут скорее хорошо, чем плохо.

Вы пришли не для того, чтобы судить кого-то и решать, что правильно, а что нет. Вероятно, вы найдете много такого, что можно улучшить, но это можно будет сделать позже. Выясните, из-за каких препятствий или обстоятельств все получилось именно так, а не иначе. Например, из-за нехватки финансирования или просто из-за незнания альтернативных вариантов.

Ваша цель здесь — наладить отношения, создать связь. Если все пойдет по плану, то человек, с которым вы разговариваете, станет частью вашей команды. И он должен чувствовать, что способен принести вам пользу.

Во время разговора с активным использованием доски и погружением в код информация начнет передаваться сама собой, без всякого принуждения. Чтобы создать нужный настрой, я обычно использую два сценария ролевой игры. Первый — я выступаю в роли нового клиента и прошу провести меня через процесс интеграции, включая демонстрацию продукта и обучение. Это прекрасный способ создать комфортные условия для вашего собеседника, потому что так он может с гордостью продемонстрировать свой продукт и показать его лучшие стороны. Также вы достигнете следующих целей:

- Узнаете, какие термины приняты в компании для общения с клиентами. Это позволит вам использовать привычные для них формулировки и тем самым уменьшить количество вопросов, которые могут быть неверно поняты, потому что вы выбрали не те слова или фразы.
- Понять, какие функции наиболее важны для клиентов. Фразы наподобие «Мы такое можем, но никто этим не пользуется» сразу отсекут целую массу вопросов, а вы узнаете, что что-то не совпадает с ожиданиями руководства.
- Получите возможность углубиться в детали той или иной области, чтобы выяснить то, что вас интересует. Например, вам показывают интерфейс чата и в нем можно загрузить файл. Вы сможете спросить, куда в итоге попадает загружаемый файл, а также есть ли резервные копии этого места, и если есть, то где они хранятся.

Когда показ для клиента закончится, самое время будет сделать перерыв на обед или кофе. Используйте возможность поблагодарить за разговор и похвалить проект. Будьте здесь самим собой. Позвольте всем немного отдохнуть перед второй ролевой игрой.

В зависимости от того, как устроен отдел разработки, с которым вы общаетесь, она может выглядеть по-разному. Ваша задача — получить как можно больше информации, не вызывая у собеседника чувства, что он наговорил лишнего.

В этой игре представьте себя новым разработчиком, инженером или другим членом команды, проходящим адаптацию, чтобы приступить к работе. Побыв в роли клиента, вы узнали терминологию, принятую в компании, так что теперь можете адекватно общаться и задавать более актуальные и конкретные вопросы, ответы на которые позволят вам стать полноценным членом команды.

Это даст всем возможность углубиться в подробности, при этом ваш собеседник будет находиться в знакомой ему ситуации, особенно если компания недавно нанимала новых сотрудников. Так вы познакомитесь с внутренними процессами, способами взаимодействия и управления задачами, порядком разработки и деплоя кода, а также с основными компонентами, которые обеспечивают работу системы.

Еще одним результатом должна стать атмосфера доверия и взаимопонимания. Вы можете поделиться своим опытом и, возможно, помочь решить какие-то очевидные проблемы, а также оценить сложности, с которыми сталкивается компания. Теперь заполнить любые пробелы в знаниях будет гораздо проще.

Удобно иметь карточку с нужными вам темами, как в лото. Каждый раз, когда вы знакомитесь с каким-то вопросом, ставьте галочку. Эта карточка не определяет ни порядок обсуждения, ни конкретные вопросы. Но она позволит отследить, все ли темы вы охватили в процессе обсуждения. Успешный *due diligence* информативен, познавателен и увлекателен для обеих вовлеченных сторон.

### 14.5.3. Представление результатов

Поговорив со всеми и составив необходимые заметки, изложите свои выводы языком, понятным нетехническим специалистам. Результаты вашей работы будут использовать руководители, банкиры, юристы, инвесторы, операционные партнеры и бухгалтеры. Не перечисляйте технические подробности, а укажите возможные последствия для всего, что вы нашли, и то, какие усилия потребуются для устранения проблем.

Вернемся к аналогии с домом. Предположим, вы обнаружили неисправность выключателя бойлера. Не вдавайтесь в технические детали; просто укажите, что горячей воды нет, на ремонт уйдет три недели и его ориентировочная стоимость составит 2000 долларов. Это более ценная информация для покупателя. Теперь он знает, что ему нужно найти 2000 долларов (либо добавить денег, либо попросить скидку от цены продажи) и он минимум три недели не сможет принимать душ.

С технологиями принцип тот же. Когда мы, технари, излагаем подробности (например, что у основной базы данных нет резервирования и что она работает на одном сервере с публичным IP-адресом), нам кажется, что все тут понятно. Часто мы упускаем из виду вопрос «и что?» и забываем объяснить, каковы последствия этого.

В примере с базой данных мы могли бы сказать: «В случае сбоя (в том числе отключения электроэнергии) компания не сможет работать, а на восстановление может уйти несколько дней. Нужно выбрать и внедрить решение для обеспечения отказоустойчивости, это будет стоить X долларов». Все технические подробности, подтверждающие ваши выводы, должны содержаться в приложениях или сопутствующей документации. Стандартный отчет о *due diligence* включает в себя следующее:

1. Общий обзор объекта покупки.
2. Методология: время проведения, кто присутствовал, какие материалы были предоставлены, а какие были недоступны.
3. Выводы.
4. Дополнительные материалы.

Выводы необходимо упорядочить по серьезности их последствий для бизнеса. Можно составить отдельный список того, что вам понравилось и что можно похвалить.

Пишите так, чтобы содержание документа было понятно каждому, независимо от его уровня технических знаний. Представьте правдивую картину, ничего не искажая. Если целевая компания использует X, а не Y, а вы предпочитаете Y, то это не имеет никакого значения и не должно указываться в качестве проблемы.

Если позволяет время, попросите людей, с которыми вы говорили, просмотреть документ перед публикацией. На это есть две важные причины. Во-первых, точность. Вы могли что-то не понять или перепутать. Проверка дает возможность исправить такие ошибки. Ваше самолюбие здесь никак не задевается, потому что вы излагаете не мнения, а факты. Вторая цель — установить связи и вызвать доверие людей, которые, возможно, будут работать в вашей команде. У них будет возможность увидеть, что вы считаете важным и как вы предлагаете действовать. Вы также продемонстрируете вашу открытость, честность и порядочность. В конце каждой беседы отмечайте, что у собеседника будет возможность проверить точность итогового отчета и что в отчете не будет никаких сюрпризов или того, что не обсуждалось. Нервозность и недоверие никому не идут на пользу.

#### **ЗАМЕТКИ С ПОЛЕЙ**

##### *Светофор для выводов*

Я не раз проводил due diligence для покупателей — как компаний, так и частных инвесторов. За годы я разработал удобную шкалу оценки: на «красные» зоны заказчику нужно обратить пристальное внимание, «желтые» — иметь в виду, а «зеленым» можно порадоваться, если они соответствуют его целям.

#### **14.5.4. Прохождение due diligence**

До сих пор мы обсуждали, как проводить due diligence со стороны покупателя, но что, если ваша компания — объект покупки и опрашивают вас? Теперь, когда вы знаете, как проводить процесс, вы лучше поймете, о чем вас спрашивают. Имейте в виду, что вас не судят, а если вы чувствуете обратное — это ошибка того, кто проводит интервью.

Существует много так называемых экспертов, которые проваливали due diligence, общаясь снисходительно или высокомерно. К сожалению, в этой ситуации вам остается разве что отнестись к происходящему с улыбкой. Вопреки распространенному мнению, не всегда покупает тот, кто предложит самую высокую цену — в большинстве случаев сделки заключаются с теми, кто наиболее подходит по менталитету. И due diligence в том числе позволяет это понять.

Ключом к успешному общению является подготовка. Однако к вам могут прийти уже со списком вопросов, и через 20 минут вы будете умирать от скуки. В этом случае вежливо поинтересуйтесь, не будет ли проще и быстрее, если этот список пришлют вам по электронной почте и вы ответите на все вопросы.

Не обязательно готовить многостраничные документы, но следующая информация должна быть у вас под руками:

- Общие схемы архитектуры/сети.
- Список всех технологий (включая версии).
- Лицензии, даты обновления и стоимость.
- Организационная схема со всеми сотрудниками.
- Описание процесса адаптации новых членов команды.
- Список недавних обращений в поддержку.
- Приоритеты развития.

Проницательный читатель узнает список: это темы, которые освещались в книге и которые должны быть постоянным объектом внимания СТО.

Вам необходимо предоставить как можно больше подробностей, не раскрывая при этом секретную информацию, и помочь покупателю принять обоснованное решение. Возможно, есть вещи, которые вы предпочтете не обсуждать. Причины для этого могут быть разными, но в любом случае необходимо согласовать это с генеральным директором (и, если нужно, с отделом кадров), чтобы вы одинаково понимали, какую информацию можно раскрывать.

Главное — никогда не лгите и не искажайте информацию, ни к чему хорошему это не приведет. Если вам задают действительно неудобный вопрос, тактично переадресуйте его генеральному директору. Но такое бывает редко, особенно в случае небольшой компании, которую вы, скорее всего, представляете.

Тем не менее подробно обсудите вопрос разглашения коммерческой тайны с генеральным директором. В какой-то момент, пусть и не на раннем этапе, вы с ней столкнетесь.

Доступ к исходному коду, если его попросят, предоставить можно и нужно. Вы можете показать его или в виде простого zip-файла, переданного через защищенный канал, или же создать в вашей системе контроля версий учетную запись с доступом только для чтения. Однако, когда вы дойдете до баз данных — доступ к ним необходимо ограничить демонстрацией экрана, без прямого доступа к данным. Они принадлежат вашим клиентам и должны быть защищены. Вы можете показать количество записей в таблицах, какие-то примеры, но не забывайте о безопасности.

Все то же самое относится и к инфраструктуре. Если вы используете облачные сервисы, вы обязательно будете их обсуждать. Заранее подготовьте учетную запись только для чтения, позволяющую посмотреть на все, что необходимо, без возможности внести какие-то изменения.

Открыто и честно говорите о недостатках. Кто знает, возможно, именно эти люди помогут вам решить проблему. И прежде всего старайтесь получить от



общения удовольствие и пользу — как бы вы ни готовились, всегда найдется что-то, что можно улучшить, и вы всегда сможете чему-то научиться.

### ЗАМЕТКИ С ПОЛЕЙ

*Говорите «нет»*

Опытный профессионал, проводя due diligence, никогда не запросит доступ, который может нарушить безопасность системы или внести какие-то изменения. Никто не хочет быть причиной сбоя. А вы должны обеспечивать безопасность, кто бы к вам ни пришел. В конце концов, вы всегда можете сказать «нет» и попросить уточнить, зачем нужен именно такой доступ. Возможно, есть другой способ получить тот же результат. Сообщите об этом генеральному директору и двигайтесь дальше.

Помните: желающих приобрести ваш бизнес может быть несколько, и в этом случае за короткое время вам придется провести не одну встречу с потенциальными покупателями. Не забывайте, однако, про свои основные обязанности. Клиентам по-прежнему нужно предоставлять сервис, командой нужно руководить. Делегируйте часть задач либо выделите достаточно времени, чтобы вы могли отвлечься. Проведение due diligence тоже может занять больше времени, чем вы предполагали, поэтому заранее составьте план на случай вашей занятости.

## 14.6. ПЕРЕДАЧА ПОЛНОМОЧИЙ

Рано или поздно наступит момент, когда вы будете передавать дела своему преемнику. К этому могут привести разные причины, например, вы поймете, что переросли свою роль или она больше не соответствует вашим целям. Еще одна причина, которую мы рассмотрим в этой главе, заключается в том, что в компании не может быть двух технических директоров. Как в классическом фильме 80-х «Горец» («*Highlander*»), останется только один.

В любом случае то, как вы передадите свои полномочия, станет финальным штрихом в вашем наследии и результатах вашего труда. Вам нужно, чтобы новый технический директор мог успешно работать. Даже если он переделает с нуля все, что вы



создали, — пусть; это уже не ваша забота. Вы несете ответственность только за настоящее, а не за будущее, в котором вас нет.

Все свои решения вы принимали, основываясь на информации, которая у вас была на тот момент. Оглядываясь, с учетом всего, что мы с тех пор узнали, мы часто видим, что можно было поступить по-другому. Это не оправдывает необдуманные и спорные решения, но дает понимание ограничений, исходя из которых вы тогда действовали. О них необходимо рассказать новому техническому директору. Такие сведения важны для бизнеса, но самостоятельно о них не сможет догадаться даже самый проницательный СТО.

Как известно, отдел может только то, на что способны его сотрудники. Важно представить новому техническому директору каждого члена команды, подчеркнув его сильные и слабые стороны. Однако это не должно скатываться в обсуждение личных качеств. Можно взять результаты последнего ревью и отметить произошедшие с тех пор улучшения (или ухудшения). Это даст хороший инструментарий для обсуждения именно тех вопросов, которые помогут новому СТО быстро получить представление о команде.

Важно рассказать в общих чертах ваше видение дальнейшего пути компании. Это может показаться бессмысленным, ведь вас уже в ней не будет, а новый СТО может выбрать другое направление. Однако вы проработали в компании не один год, вы хорошо разбираетесь в особенностях бизнеса и имеете опыт решения различных проблем клиентов. Это понимание очень важно для вашего преемника, у которого нет этого контекста.

Кроме стандартного знакомства с техническими особенностями, как принято в большинстве случаев при передаче дел от одного руководителя к другому, покажите новому СТО все задачи, с которыми вы сталкиваетесь. Иногда советуют пройтись по 24 часам типичного рабочего дня, но для полного понимания вашей роли этого будет недостаточно. Поэтому откройте ваш календарь и пройдите по последним 30 дням. Можно немного рассказать о повторяющихся встречах, однако основное внимание уделите встречам с людьми, не входящими в состав вашего отдела. Это позволит новому техническому директору увидеть, кто и как будет с ним взаимодействовать. Рассказав о ходе (и результатах) каждой такой встречи, вы поможете ему начать разбираться в текущих проектах и проблемах.

По какой бы причине вы ни уходили из компании, это всегда печально, что совершенно естественно. Примите это и гордитесь тем, что вы сделали все возможное и направляли вашу команду. Если верить теории эволюции, все изменения — к лучшему, в том числе и смена руководства (хотя не всегда — как вы помните, Apple пришлось в конце концов вернуть Стива Джобса).

Ни в коем случае не напускайте тумана, даже если все сложно. Помните о команде, которую вы покидаете, — у нее должно быть грамотное руководство и возможность учиться. Готовя своего преемника, вы обеспечиваете их успешную работу.

## 14.7. ПРИНЯТИЕ ПОЛНОМОЧИЙ

Обсудив передачу полномочий новому техническому директору, рассмотрим обратную ситуацию, когда вы вступаете в должность или получаете повышение. Предположим, что вам передали дела, как мы описали выше. Теперь вы предоставлены сами себе, команда ожидает от вас руководства, а начальство — что все продолжит работать и развиваться. Что теперь? Все просто — ничего не делайте.

Вам предстоит многое узнать о компании, поэтому не торопясь изучите ее, продукт, клиентов и сотрудников. На самом деле в первые 30 дней от вас ничего не ждут, а какие-то понимание и стратегия начнут формироваться только через 90 дней.

Спротивляйтесь искушению делать изменения только ради изменений. Лучше вносите небольшие улучшения, если они стоят того. Присутствуйте на всех возможных совещаниях и просто слушайте и наблюдайте за участниками. Вспомните первую главу этой книги и Маверика, который целый час сидел и наблюдал за игроками в покер, и примените тот же прием — понаблюдайте за своей новой командой.

Проводите как можно больше индивидуальных встреч с членами вашей команды. На них я люблю задавать один вопрос, чтобы завязать диалог: «Если бы вы стали королем/королевой на день, что бы вы поменяли или исправили?» Ответы будут совершенно разными, в зависимости от того, как конкретный человек видит ситуацию. Большинство из них будут содержать возможные действия, и они, даже незначительные, помогут вам понять, что важно для тех, кто занимается непосредственной работой.

Не предлагайте решений или гарантий. Могут быть веские причины, почему все организовано именно так, а не иначе, и вы, вероятно, их выясните. Пока же вам нужно слушать, впитывать информацию и составлять общую картину.

Чарльз Мун (Charles Moon), который работал техническим директором / руководителем отдела исследований и разработок во многих организациях, дает отличный совет: «Будьте внимательны к внутренней политике, поскольку вас могут окружать те, кто поддерживал предыдущие технические решения. Старайтесь относиться к ним как к положительным изменениям, связанным с развитием технологий и рынка».

Как мы уже говорили в предыдущих главах, познакомьтесь с другими руководителями. Ваше положение новичка, когда вы еще не обязаны знать все тонкости, дает вам возможность без стеснения задавать самые базовые вопросы. Пользуйтесь этим.

## ИТОГИ

- Участие в сделке по продаже может отнять много сил и времени, если не знать, как правильно организовать свою работу.

- Покупка небольшой компании может принести значительную и быструю выгоду, но только если вы как можно быстрее интегрируете или начнете использовать приобретенный актив.
- Архитектура, упрощающая интеграцию, открывает вашей компании дополнительные стратегические возможности.
- Due diligence — это не допрос, а беседа, имеющая конкретную цель.
- Для принятия высокоуровневых стратегических решений необходимо представить результаты due diligence в форме, понятной людям, не являющимся техническими специалистами.
- Принимая дела у предыдущего технического директора, слушайте и изучайте его задачи и проблемы, а не старайтесь как можно быстрее все изменить.
- Передавая свои полномочия новому техническому директору, важно настроить на успех всех сотрудников, включая вашу команду.

## ПРОВЕРЬТЕ СЕБЯ

Что из перечисленного ниже вы уже сделали?

- Готовы в любой момент без предварительного уведомления представить положение дел в компании в реалистичном, но при этом оптимистичном виде.
- Выяснили, есть ли вероятность участия в сделке в ближайшее время.
- Готовы пройти как минимум поверхностный due diligence.
- Понимаете сильные стороны архитектуры системы своего предприятия и не игнорируете области, требующие улучшений.
- Можете ознакомить нового технического директора с причинами, которые привели к тем или иным решениям.

# 15

## *Вы, Inc.*

---

### **В этой главе**

- ✓ Оценка и анализ вашего профессионального и карьерного развития
- ✓ Как идти в ногу со временем и не поддаваться синдрому самозванца
- ✓ Как не мешать работать другим и при этом оставаться незаменимым
- ✓ Планируйте будущее, чтобы не тратить время на работу, которая не приносит удовлетворения

Выполняя свою работу, вы помогаете другим добиваться успеха и делать карьеру. Их результаты — ваша самая ценная награда. Создание и развитие команд, которые способны достичь самых амбициозных целей, и участие в их работе — вот что приносит мне истинную радость. В одиночку вы можете не так много: у вас всего лишь две руки, а в сутках лишь 24 часа.

Однако кто же будет наставлять и поддерживать вас, пока вы занимаетесь созданием команды? Кто поможет вам сосредоточиться на том, чего вы ждете от вашей профессии и работы?

В этой главе мы рассмотрим темы, которые должны помочь вам фокусироваться и получать удовлетворение от работы. Никто не хочет спустя 5–10 лет внезапно осознать, что он не добился того, чего хотел. Это время, кажущееся таким долгим, может пролететь очень быстро, особенно если вы с головой ушли в масштабные проекты.

## 15.1. ОЦЕНИТЕ СЕБЯ

Как технический директор вы будете (по крайней мере, должны) проводить в команде ежегодные ревью, контролировать прогресс сотрудников и следить за тем, чтобы в их карьере не было сюрпризов ни для них, ни для вас. Однако, в зависимости от организации, ревью для вашей собственной позиции может не проводиться. Эту проблему можно решить несколькими способами.

Прежде всего, попросите генерального директора (СЕО) оценить вас. Выделите интересующие вас области, а он со своей позиции может определить, насколько вы в них успешны. Вот несколько примеров:

- *Коммуникация.* Достаточно ли вы информируете бизнес о состоянии проектов, направлениях работы и текущих вопросах?
- *Уровень детализации.* Устраивает ли уровень детализации поступающей от вас информации, бывает ли она слишком подробной или, наоборот, недостаточно подробной?
- *Бизнес-аналитика.* Каков ваш вклад в обеспечение бизнеса необходимыми данными? Отслеживаете ли вы тенденции и изменения в отрасли и обсуждаете ли их внутри компании?
- *Инициатива/видение.* Есть ли уверенность, что выбранные вами технологии обеспечат прогнозируемый будущий рост компании и позволят использовать все новые возможности?

Ваш непосредственный начальник — скорее всего, это СЕО — конечно же, хочет, чтобы вы добились успеха, потому что ваш успех — это его успех. Часто, даже если процесс оценки сотрудников в компании налажен, для руководителей оценка может не проводиться. Не используйте это как оправдание. Старайтесь получать обратную связь и наладьте процесс ее получения. Регулярно проводите встречи с генеральным директором по всем вопросам, связанным с ведением бизнеса. На одной из таких встреч поинтересуйтесь его мнением о вашей работе и прямо спросите, что еще вы можете сделать, чтобы достичь успеха (или же развить его).

### ЗАМЕТКИ С ПОЛЕЙ

#### Офис

Независимо от того, британскую или американскую версию известного ситкома «Офис» («*The Office*») вы предпочитаете, обе они прекрасно изображают ситуацию, когда начальник оторван от своей команды. Дэвид Brent (или Майкл Скотт) верят, что они уважаемы и любимы в коллективе, и считают подчиненных своими друзьями. По испуганным лицам и недовольным взглядам их сотрудников мы понимаем, что это далеко не так. На экране такое распознать легко, но в реальной жизни — гораздо сложнее. Сериал чрезвычайно популярен, потому что на этом заблуждении построена масса комических сцен, и чаще всего они близки к реальности — всем нам в какой-то момент случалось попадать в ситуацию из «Офиса».

Это не поставит в неловкое положение ни вас, ни СЕО. Если критики в ваш адрес не будет, попросите СЕО поставить перед вами цели или определить области, в которых он хотел бы видеть от вас большего. Хорошие генеральные директора заинтересованы в развитии своих руководителей и всегда готовы дать обратную связь. Если ваш директор не такой, помогите ему, подтолкнув его в верном направлении.

Старайтесь также получать обратную связь от своей команды. Поначалу это может быть непросто, потому что подчиненные могут чувствовать себя некомфортно, сомневаться или думать, что вы потом припомните им их слова. Это многое говорит о вашем стиле руководства. Если люди не чувствуют, что могут говорить с вами открыто и честно, то уже это само по себе является показательной обратной связью. Не стремитесь к тому, чтобы править железной рукой, внушая страх и ужас. Осуществляйте руководство уверенно, твердо и беспристрастно, будьте внимательны к людям. Если вы выберете верный тон, то вашим подчиненным не составит труда предоставить вам небольшой отзыв, который будет для вас полезен.

Оценка сотрудников — это ни в коем случае не способ уничтожить их авторитет и самооценку. Ее цель — помочь сотруднику понять, как окружающие воспринимают его действия и способ общаться, чтобы он мог развиваться. Мы всегда думаем, что выражаемся четко и ясно. Однако можем не замечать, что упускаем подробности, которые имеют огромное значение для слушателя. Следите за обратной связью от команды. Этикетку бутылки трудно прочесть изнутри. Вам нужна честная обратная связь от окружающих.

Еще один человек, к которому вы можете обратиться, — это вы сами. Хороший лидер способен анализировать свои действия, быть честным с самим собой и понимает, когда что-то можно было сделать лучше или по-другому. Но не стоит быть слишком суровым и сильно критиковать себя — просто будьте внимательны к себе и исправляйте то, что требует исправления.

### ЗАМЕТКИ С ПОЛЕЙ

*Язык, разделенный Атлантикой*

Когда я начал работать в Соединенных Штатах, приехав из Великобритании, то заметил, что некоторые мои фразы, казавшиеся невинными по другую сторону Атлантики, здесь воспринимаются совершенно иначе. Мне пришлось тесно познакомиться с отделом кадров, потому что они подробно объясняли мне, как что-то может звучать оскорбительно, а не ободряюще или забавно (я вооружился словарем «*Urban Dictionary*», чтобы доказать, что не имел злого умысла). Поэтому мне пришлось научиться читать выражения лиц и язык тела на совершенно новом уровне и тщательно обдумывать все формулировки, прежде чем произносить их вслух.

## 15.2. НАСТАВНИЧЕСТВО

Наставник заботится о ваших интересах, но не имеет прямого влияния на вашу карьеру. В частности, обычно ваш начальник — это не ваш наставник. Босс может направлять и обучать, но не является наставником в истинном смысле этого слова.

Наставник — это тот, кому вы доверяете, к кому обращаетесь за советом по насущным для вас вопросам, кто может предоставить взгляд со стороны и дать непредвзятый совет. Наставником может быть человек из любой сферы, но это должен быть тот, кто знает как ваши сильные, так и слабые стороны. Иногда человеку даже не нужно знать, что он выполняет роль «наставника» — он может выслушать и дать совет и без этого формального статуса.

### ЗАМЕТКИ С ПОЛЕЙ

За всю карьеру у меня было два наставника: один — в самом ее начале, он помог мне сориентироваться в мире бизнеса, когда я вставал на ноги после университета, а второй поддерживал меня и продолжает поддерживать до сих пор в более серьезных проблемах, как профессиональных, так и личных.

Нам всем иногда нужна помощь человека, который даст нам толчок или найдет нужные слова, чтобы поддержать тогда, когда мы нуждаемся в этом. Как найти наставника? Это все равно что спросить: «Как найти спутника жизни?» Важную роль в этом процессе играют личные связи и отношения. Часто наставниками становятся родители, родственники или друзья семьи. Если это ваш случай, оцените компетенцию наставника. Человек, не знакомый с вашей сферой деятельности, не сможет должным образом понять область, в которой вы ищете помощи или поддержки. Можно (и нужно) иметь несколько наставников и советников для разных сфер жизни.

Как часто к ним обращаться — хороший вопрос. Наставник — это не ваш лучший друг (он может им быть, но не тогда, когда выполняет роль наставника). Он будет участвовать не в каждом решении, которое вы принимаете, а только в важных ситуациях, влияющих на всю вашу жизнь. Например, к наставнику по автомобильным вопросам (если бы такой существовал) стоило бы обращаться только тогда, когда вы собираетесь купить новую машину, а не каждый раз, когда вам нужно заправиться. И даже в этом случае автомобильный наставник не скажет вам, какую модель покупать. Он выслушает ваши пожелания, подскажет, что еще стоит учесть при выборе, и поможет вам все обдумать. Возможно, предложит несколько вариантов на рассмотрение.

Наставник не дает ответы — вы должны сами принимать решения. Он поможет вам увидеть общую картину и задаст вопросы, которые внесут ясность. Пока вы находитесь в поиске подходящего наставника, пробел заполняют всевозможные



книги и статьи на тему саморазвития. Просто делайте поправку на то, что их авторы не знают ни вас, ни вашу ситуацию.

## 15.3. ИДТИ В НОГУ СО ВРЕМЕНЕМ

Одно из прекрасных качеств нашей отрасли — ее постоянно меняющийся ландшафт, то, как она заново изобретает себя каждые пять лет. Одно из сложнейших препятствий, с которыми мы сталкиваемся в нашей отрасли, — ее постоянно меняющийся ландшафт, то, как она заново изобретает себя каждые пять лет!

Да, наше величайшее благословение — это также и наше величайшее проклятие. Мы постоянно учимся, гонимся за новинками, осваиваем новые термины и изо всех сил стараемся не отставать. Это изнуряет. Это стресс, особенно когда от нас ждут решений, обеспечивающих соответствие компании современным технологиям.

Именно здесь *синдром самозванца* поднимает свою уродливую голову и подрывает ваше чувство собственного достоинства и уверенность в себе. Что такое синдром самозванца? Это чувство, что вы находитесь там, где не должны быть — не на своем месте. Вы живете в постоянном страхе, что в какой-то момент вас разоблачат и выгонят. Отдел кадров что-то напутал и принял на работу неподходящего человека, и вы теперь изображаете того, кем не являетесь.

Отметим, что в какой-то момент подобным образом себя чувствует большинство людей, в том числе очень известных. У всех случаются приступы неуверенности и возникает ощущение, что задачи им не по плечу.

Чтобы справиться с этим чувством, прежде всего необходимо его распознать. Напомните себе о ваших достижениях, о пути, который вы проделали. Дышите. Копайте глубже, и, с какой бы проблемой или вызовом вы ни столкнулись, черпайте силы в вашем опыте, потому что именно внутри вас находится то, что направит вас, напомнит, что на этом месте должен быть не кто иной, как вы, и подтолкнет вас вперед.

Помните: помощь всегда рядом. Вы не должны чувствовать себя в изоляции. В истории человечества еще не было эпохи, в которой людям было бы так легко взаимодействовать, — и вам обязательно кто-нибудь поможет.

Небольшой совет: обновляйте свое резюме каждые шесть месяцев. Оно служит своего рода дневником, где вы перечисляете основные вехи своей карьеры. Благодаря актуальному резюме вы будете:

- не забывать о своих достижениях;
- готовы к новым возможностям, которые могут появиться в любой момент.

В разгар приступа неуверенности все ваши сильные стороны забываются. Обновляйте свое резюме и держите его под рукой. Все выдающиеся личности (и те, кто хочет ими стать) порой сомневаются в себе и впадают в рефлекссию. Это естественно.

## 15.4. ОТСЛЕЖИВАНИЕ ИЗМЕНЕНИЙ

Будем откровенны: идти в ногу со временем сложно, и для этого требуются значительные усилия. Все меняется — выходят новые версии, появляются новые фреймворки, и, как будто этого мало, иногда происходят глобальные изменения, которые заставляют нас чувствовать, что мы все больше отстаем. Программное обеспечение, оборудование, сети, операционные системы — удивительно, что хоть кто-то поспевает за всем этим!

Сколько больших изменений вы активно отслеживали с самого начала (см. табл. 15.1), за сколькими вы следили вынужденно, и, наконец, какие из них вы совершенно упустили из виду и до сих пор не знаете, что они означают?

**Таблица 15.1.** Технологические изменения

Интернет	Веб 2.0	Метавселенная
Виртуализация	Облако	Бессерверная архитектура
Большие данные	Озеро данных	Бизнес-аналитика (Business Intelligence)
Машинное обучение	Искусственный интеллект	Блокчейн
Цифровая валюта	Шифрование	Многофакторная аутентификация
Цифровые подписи	3G/4G/5G	Интернет вещей
Виртуальная реальность	Игры	Мобильные приложения
Десктоп	SQL/NoSQL	Хранилище данных
Мессенджеры	Открытый исходный код	Low code / no code программирование
«Туманные вычисления»	API	Монолитная архитектура
Макросервисы	Микросервисы	...

Этот список можно продолжить, и это мы еще не касались новых языков и их функций. Например, Java сегодня совсем не тот, что был 25 лет назад. Появляются все новые паттерны проектирования, поскольку мы придумываем новые и лучшие способы использования технологий. Если вы вернетесь на 10 лет назад, то увидите, что большая часть технологий из таблицы 15.1 существовала только в виде прототипов и исследовательских проектов и этим технологиям только предстояло стать распространенными. Теперь же некоторые из них настолько укоренились в современных проектах, что мы уже не можем себе представить, как без них обходились. Так как же уследить за всеми изменениями?

Позвольте мне вас успокоить: ответ на этот вопрос — никак. Точка. Вы должны сосредоточиться на областях, которые имеют значение для вас и для вашей компании. Выберите минимум три источника технологических новостей, относящихся к вашей области, и подпишитесь на новостную рассылку

по электронной почте или в соцсетях — так вы начнете получать информацию. Со временем это будет работать само собой: читайте, принимайте к сведению, снова и снова. Вы начнете распознавать повторяющиеся темы. Когда вы будете видеть, что какая-то информация появляется раз за разом в разных источниках, вы поймете, что ей стоит уделить повышенное внимание. Возможно, имеет смысл поручить кому-нибудь из вашей команды изучить ее глубже и понять, стоит ли за ней следить.

Необязательно быть на самом острие технологий, достаточно не отставать. На острие опасно — потому оно так и называется, что может поранить, если что-то пойдет не так. Не берите на себя ненужные риски, полагаясь на технологии, которые могут не прижиться в будущем.

Следите за конкурентами и за тем, как они позиционируют свои продукты. Они могут выезжать на волне хайпа (вы видели, сколько на рынке блокчейн-решений?), но это повод изучить технологию поглубже, чтобы составить обоснованное мнение на случай, если оно понадобится.

Еще один отличный источник знаний о перспективных тенденциях — подписка на аналитические отчеты исследовательских агентств, если ваша организация может себе ее позволить. Gartner и Forrester — два очень уважаемых источника, и вы, вероятно, встречали поставщиков, гордящихся своим появлением в скандально известном магическом квадранте Gartner. Стоимость такой подписки может составлять несколько тысяч долларов в год, но она даст вам представление о последних исследованиях и разработках, позволит сравнивать продукты и получать скидки на участие в конференциях. Аналитики держат руку на пульсе за вас.

Наконец, приучайте и поощряйте свою команду следить за отдельными компонентами — фреймворком или языком. Пусть они просят выделить какое-то время, чтобы изучить возможности, которые появляются здесь для вашего проекта.

Учиться интересно, но времени следить за новинками всегда не хватает. Не делать этого тоже нельзя, поэтому разделите эту нагрузку с командой и постарайтесь сделать ее естественной частью рабочего процесса.

## **15.5. ПОДГОТОВКА ПРЕЕМНИКА**

Настанет момент, когда вы захотите двигаться дальше. Возможно, изменятся ваши желания или потребности, либо вы перестанете получать удовлетворение от своей работы. Какой бы ни была причина, кто-то должен будет вас заменить.

По мере роста организации вы все меньше будете заниматься прикладными задачами. Основное время вы будете уделять управлению, и то, что делали раньше сами, вы передадите другим. Такая естественная эволюционная передача повседневной рутины является формой преемственности.

Следующий этап — выбрать того, кто мог бы занять ваше место. Это потребует от вас уверенности в себе, ведь мы всегда хотим чувствовать себя особенными и нужными. Активно готовить, наставлять и обучать кого-то вместо себя — большая работа, и относиться к ней нужно серьезно.

Начнем с того, что, вероятно, не от вас зависит, кто именно придет на ваше место. Однако если вы уйдете правильно, то к вашему мнению, советам и мудрости будут прислушиваться и интересоваться ими.

Но вы можете дать человеку, в которого вы верите, прекрасную возможность проявить себя и быть замеченным. Можно подготавливать его планомерно, в течение нескольких лет, в удобном для него темпе. Ему даже не обязательно знать обо всех ваших планах. По мере того как его сфера ответственности будет расширяться, он увидит вашу уверенность в нем и оценит ваше неизменное доверие.

## 15.6. КАРЬЕРНЫЙ УСПЕХ

В чем заключается работа вашей мечты? Что заставляет вас просыпаться по утрам и с нетерпением браться за дело?

Планирование карьеры — это не определение того, сколько денег вы хотите зарабатывать в том или ином возрасте. Планирование будущего — это сосредоточенная работа по выявлению ваших желаний и потребностей и составлению плана их достижения. И будьте уверены: план сидеть сложа руки и ждать, когда зазвонит телефон, не работает.

На эту тему есть бородатый анекдот: «Господи, почему я никогда не выиграю в лотерею?» — «Так помоги мне и хотя бы купи лотерейный билет!» Любой целеустремленный успешный человек скажет вам, что он достиг всего благодаря планированию. Упорная работа, включающая в себя подготовку и планирование, поможет вам не только увидеть возможности, но и пользоваться ими.

Вы читаете эту книгу, потому что хотите стать техническим директором, или же вы недавно им стали и вам нужно разобраться в том, что представляет собой эта должность. Почему это случилось именно сейчас? Как вы к этому пришли? Каковы критерии успеха для роли, в которой вы находитесь? Понимание, как конкретно выглядит успех, к которому вы стремитесь, поможет вам оценить свою удовлетворенность и отслеживать прогресс. Со временем вы будете получать новый опыт и исходя из него менять или адаптировать свои представления и планы.



**ЗАМЕТКИ С ПОЛЕЙ**

Окончив университет, я понятия не имел, что существует такая должность, как технический директор. У меня была одна цель — узнать как можно больше и быть лучшим в том, что я делаю. Я пока не понимал, чем конкретно буду заниматься, но за первые несколько лет работы в отрасли я получил больше информации о ней, смог сфокусироваться и начать планировать карьеру. С тех пор, по мере достижения каждой новой цели, я знакомился с новым миром, осваивался в нем и формулировал свои дальнейшие цели. Работа техническим директором принесла мне больше всего удовлетворения (в профессиональном плане). Возможность создавать и развивать проекты и менять жизни людей — это настоящая честь, и я отношусь к этой работе со всей ответственностью.

Только один человек отвечает за ваш успех — это вы. Вы достигли одной из ваших целей и получили замечательную должность технического директора — но что дальше? Что нужно делать, чтобы добиться успеха в этой должности и открыть перед собой новые горизонты?

Проанализируйте развитие вашей карьеры. Вспомните путь, который вы прошли, и подумайте, какие из принятых вами решений привели вас туда, где вы сейчас находитесь. Если вы добились успеха — значит, вы вовремя принимали верные решения с правильной целью. Извлеките максимум из возможности, которая вам предоставилась, и не забывайте присматривать за «Вы, Инс.» и регулярно обновлять и оценивать ваш бизнес-план.

**15.7. ШАГ ВПЕРЕД**

Желая продвижения по карьерной лестнице, вы резонно зададитесь вопросом: «Как мне стать техническим директором?» Вопрос логичный, особенно если в компании уже есть технический директор с хорошей репутацией, который не собирается никуда уходить, — как занять его место? Никак, если только он не решит сменить работу или уволиться по другим причинам. Итак, если вы чувствуете, что готовы сделать следующий шаг, вам придется начать поиски в другом месте.

Первым делом начните изучать вакансии. Вас, скорее всего, напугают некоторые требования, особенно требование опыта. Классическая дилемма: как приобрести опыт, если везде требуются уже опытные сотрудники?

Попробуйте определить, является ли требование опыта обязательным или всего лишь желательным. Иногда организация хочет укрепить свое лидерство в сфере технологий и решает нанять СТО, хотя раньше этой должности в ней не было. Если у вас есть опыт руководства командами, который подходит для этой позиции, — пробуйте свои силы. Не выдавайте здесь желаемое за действительное, будьте откровенны с работодателем и обратите его внимание, что, хотя

официально вы никогда не назывались техническим директором, у вас есть подходящий опыт и вы готовы сделать шаг вперед.

На должность СТО компании ищут надежного человека, который будет твердо руководить работой команды (занимаясь всем, что описано в этой книге). Руководство компании должно понимать, что вы во всеоружии и готовы принять вызов. Технический директор — это не старший разработчик: работодатель будет оценивать ваши навыки общения и то, как вы собираетесь управлять командой, чтобы она показывала успешный результат. Именно в этих вопросах на собеседовании вам потребуется раскрыть свои самые сильные стороны.

Вакансий технических директоров мало по сравнению с количеством позиций руководителей или старших разработчиков. Поэтому у вас будет много конкурентов, похожих на вас, — тех, кто готов сделать следующий шаг. Отправляя отклик на позицию СТО, тщательно продумайте сопроводительное письмо — оно не менее важно, чем резюме. Это ваша возможность рассказать, что вы можете сделать на этой позиции и почему вы подходите этой компании. В резюме уделите больше внимания вашим навыкам руководства и организации работы, а не техническим навыкам.

Прежде чем отправлять первый отклик — осмотритесь, вникните в каждую вакансию, почувствуйте, что за ней стоит. Оцените, чего каждая из компаний ожидает от технического директора, особенно если это новая должность. Как уже говорилось выше, знайте себе цену, учитывайте ваши потребности и не идите на компромиссы.

## ИТОГИ

- Обратная связь от руководства поможет определить, что вам необходимо улучшить.
- Регулярно запрашивайте обратную связь у своей команды, чтобы убедиться, что вы обеспечиваете ей все, что нужно.
- Наставник поможет вам продумать важные решения, влияющие на всю вашу жизнь.
- Ведите дневник или обновляйте резюме, записывая свои достижения, — это поможет вам преодолеть синдром самозванца.
- Не пытайтесь уследить за всеми новинками — это невозможно. Но вы можете кое-что сделать, чтобы не пропустить самое главное.
- Занимаясь ростом и развитием других, отмечайте, кто в будущем сможет стать вашим преемником.
- Определите свои карьерные цели. Это поможет вам распознавать успехи и достижения.

## ПРОВЕРЬТЕ СЕБЯ

Что из перечисленного ниже вы уже сделали?

- Договорились со своим руководителем о вашем регулярном ревью.
- Внедрили в команде механизм обратной связи о вашей работе.
- Нашли наставника, к которому можно обратиться по важным вопросам.
- Познакомились с новыми технологиями, которые начали широко использоваться в вашей отрасли.
- Подписались на новостные источники, чтобы быть в курсе новых тенденций.
- Определили, кто при необходимости может стать вашим преемником.
- Разработали план развития карьеры и зафиксировали ваши цели.

*Алан Уильямсон*

**Настоящий СТО:**  
**думай как технический директор**

*Перевела с английского М. Трусковская*

Руководитель дивизиона	<i>Ю. Сергиенко</i>
Руководитель проекта	<i>А. Питиримов</i>
Ведущий редактор	<i>Е. Строганова</i>
Научный редактор	<i>О. Логинов</i>
Художественный редактор	<i>В. Мостипан</i>
Корректоры	<i>Л. Галаганова, М. Лауконен</i>
Верстка	<i>Л. Соловьева</i>

Изготовлено в России. Изготовитель: ООО «Прогресс книга».  
Место нахождения и фактический адрес: 194044, Россия, г. Санкт-Петербург,  
Б. Сампсониевский пр., д. 29А, пом. 52. Тел.: +78127037373.

Дата изготовления: 11.2023.  
Наименование: книжная продукция.  
Срок годности: не ограничен.

Налоговая льгота — общероссийский классификатор продукции ОК 034-2014,  
58.11.12 — Книги печатные профессиональные, технические и научные.  
Импортер в Беларусь: ООО «ПИТЕР М», 220020, РБ, г. Минск, ул. Тимирязева, д. 121/3, к. 214, тел./факс: 208 80 01.  
Подписано в печать 01.11.23. Формат 70×100/16. Бумага офсетная. Усл. п. л. 28,380. Тираж 500. Заказ 0000.