

ПРИЛОЖЕНИЕ  
К АУДИОКНИГЕ

# Настоящий СТО

Думай  
как технический  
директор

Алан Уильямсон

*(Приложение содержит таблицы и схемы  
из печатной версии книги)*

# ГЛАВА 3. ДОЛГОСРОЧНОЕ ВИДЕНИЕ

Страницы 82-83

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
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

# ГЛАВА 4. СОЗДАНИЕ КОМАНДЫ

Страницы 111-112

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 помещено</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>▪ Заключив 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>

## Страница 117

Навык	Текущая оценка					Средняя оценка	Общая оценка	Требуемый уровень	Разница
	Кормак	Ноа	Джоэф	Диллон	Престон				
AWS	[7]	0	0	0	0	[7]	[7]	10	-3
JavaScript	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

# ГЛАВА 5. СОБЕСЕДОВАНИЯ, ВЫБОР ПОДХОДЯЩЕГО КАНДИДАТА

Страница 145

**Таблица 5.1.** Лист оценки кандидата

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

# ГЛАВА 6. УПРАВЛЕНИЕ КОМАНДОЙ

Страница 160

**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 влиятельности соответствует высшему руководству организации. — *Примеч. пер.*

## Страница 163

**Таблица 6.2.** Уровни вовлечения СТО

<b>Участие не требуется</b>	<b>Информирование</b>	<b>Требуется решение</b>
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	

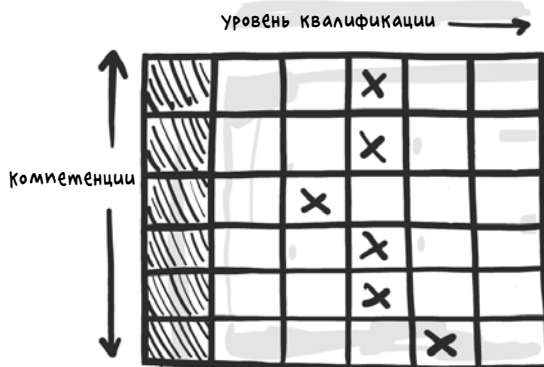
## Страница 167

**Таблица 6.3.** Формат индивидуальных встреч

<b>Темы, которые можно обсудить</b>	<b>Темы, которые не стоит обсуждать</b>
Личные цели	Статус проекта
Зоны роста	Новые задачи
Вопросы и сомнения относительно работы	Воспитательные беседы
Анализ недавних событий	Сплетни о других сотрудниках
	Новости компании

# ГЛАВА 7. ЕЖЕГОДНАЯ ОЦЕНКА СОТРУДНИКОВ

Страница 179





## Страница 181

**Таблица 7.1.** Базовая структура матрицы — разработчик

<b>Компетенция</b>	<b>Младший разработчик</b>	<b>Разработчик</b>	<b>Старший разработчик</b>
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

# ГЛАВА 8. ТЕХНОЛОГИЧЕСКИЕ РЕШЕНИЯ

## Страница 202

**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>

## Страница 203

**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>

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	..... .....



ОСНОВЫ СТРАТЕГИИ АВАРИЙНОГО ВОССТАНОВЛЕНИЯ

## Страница 216



## Страница 226

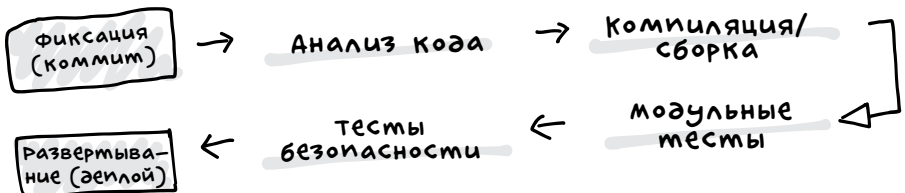
**Таблица 8.4.** Признаки микросервисной и монолитной архитектуры

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

## ГЛАВА 9. РАЗРАБОТКА

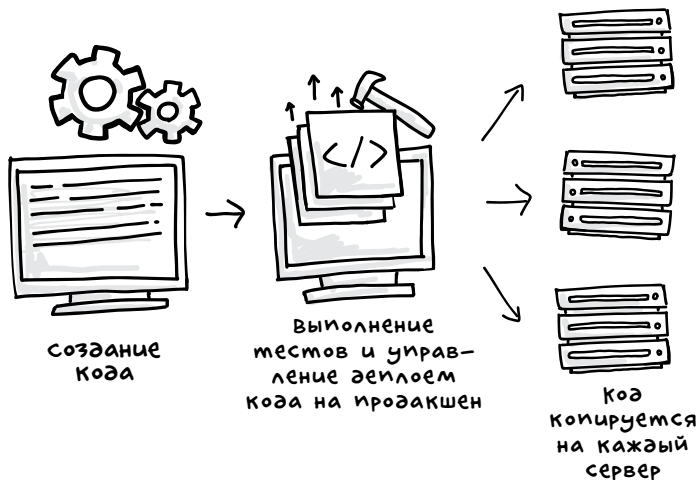
### Страница 251

Пример типичного пайплайна CI/CD:

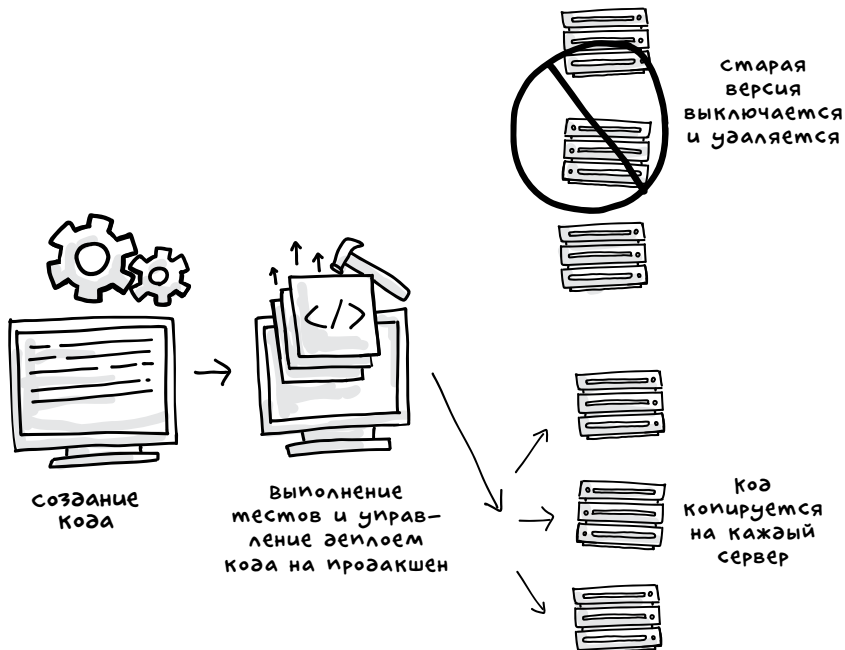


## Страница 255

Сине-зеленое (blue-green) внедрение



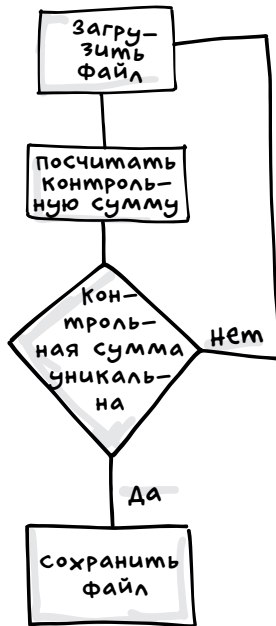
## Страница 256

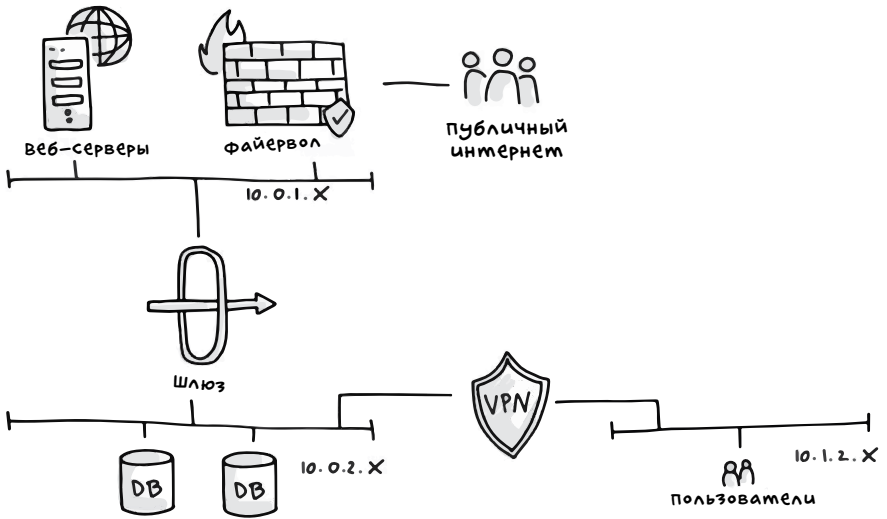


Например, технология экспорта/импорта данных позволит клиенту создать внешнее решение, также он может использовать ваше API, если оно есть. Если ваша система имеет возможность настройки для отдельных клиентов — добавьте в нее поддержку плагинов, чтобы ее функционал можно было расширять, не затрагивая основные компоненты. Главное — не позволяйте себе оказаться во всем знакомой ситуации:

```
if (Client == XYZ) {  
    // делаем что-то для конкретного клиента  
}
```

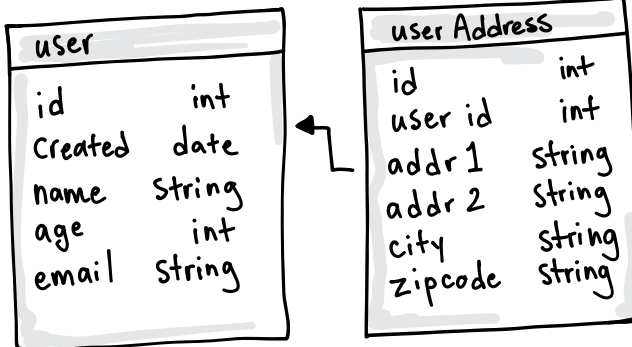
## ГЛАВА 11. ДОКУМЕНТАЦИЯ





Страница 283

схема базы данных



# ГЛАВА 12. БЕЗОПАСНОСТЬ

## Страница 296

Компонент	Программа	Формат	Перезапуск	Назначение
Веб	Tomcat	web.xml	Да	База данных

## Страница 300

### 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 +  
      "'";
```

# ГЛАВА 15. ВЫ, INC.

## Страница 346

**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	...