

# Head First Архитектура ПО

Как было бы здорово иметь книгу об архитектуре программного обеспечения, более увлекательную, чем поход к стоматологу, и более информативную, чем стопка документации! Жаль, что это всего лишь мечты...

Раджу Ганди

Марк Ричардс

Нил Форд



**SPRINT**  
book

2025

## Содержание (сводка)

	Введение	21
1	Архитектура ПО без мистики: <i>приступим!</i>	33
2	Архитектурные характеристики: <i>знайте свои возможности</i>	73
3	Два закона архитектуры ПО: <i>все основано на компромиссах</i>	113
4	Логические компоненты: <i>строительные блоки</i>	155
5	Архитектурные стили: <i>классификация и философия</i>	199
6	Многоуровневая архитектура: <i>разделение обязанностей</i>	217
7	Модульные монолиты: <i>управляются предметной областью</i>	247
8	Микроядерная архитектура: <i>индивидуальная подгонка</i>	279
9	Сделай сам: <i>приложение для путешественников TripEZ</i>	309
10	Микросервисная архитектура: <i>шаг за шагом</i>	335
11	Событийно-управляемая архитектура: <i>асинхронные приключения</i>	381
12	Сделай сам: <i>проверка знаний</i>	435
	Приложение: <i>шесть главных незатронутых тем</i>	499

## Содержание (настоящее)

### Введение

Архитектура ПО – сложный предмет, и ваш мозг будет убеждать вас, что его нельзя освоить. Он будет думать: «Лучше сосредоточиться на чем-то более важном, например, что съесть на обед и есть ли у свиней крылья». Но вы СМОЖЕТЕ убедить его, что архитектура ПО – важный предмет, и в данной главе мы поможем вам сделать это.

	Кому адресована эта книга	22
	Мы знаем, о чем вы думаете	23
	Мы знаем, о чем думает ваш <i>мозг</i>	23
	Метапознание: как мы мыслим	25
	Вот что сделали МЫ	26
	Что можете сделать ВЫ, чтобы заставить ваш мозг повиноваться	27
	Важная информация	28
	Главы «Сделай сам»	29
	Рецензенты	30
	Совместные благодарности	31
	Индивидуальные благодарности	32

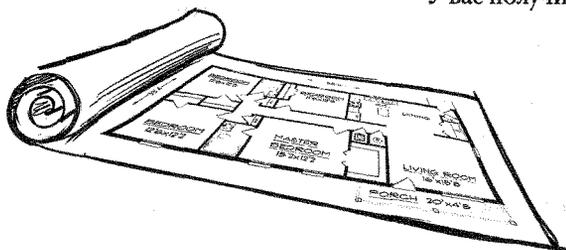
# Архитектура ПО без мистики

## 1

### Приступим!

**Архитектура программного обеспечения лежит в основе успешной работы системы.** Эта глава срывает покров таинственности с архитектуры ПО и рассказывает об архитектурных измерениях и различиях между архитектурой и дизайном. Об этом важно знать, поскольку понимая архитектурные практики и применяя их, вы сможете создавать эффективные системы, которые не просто лучше функционируют, но и отвечают нуждам и чаяниям бизнеса и продолжают работать, когда техническая и бизнес-среда претерпевают постоянные изменения.

Основы архитектуры программного обеспечения	34
План строительства и архитектура ПО	35
Измерения в архитектуре ПО	36
Мозаика из измерений	37
Измерение первое: архитектурные характеристики	38
Измерение второе: архитектурные решения	40
Измерение третье: логические компоненты	42
Измерение четвертое: архитектурные стили	44
С точки зрения дизайна	48
С точки зрения архитектуры	49
Спектр между архитектурой и дизайном	50
В какую часть спектра попадает ваше решение?	51
Стратегические и тактические решения	52
Большие и малые усилия	54
Значимые и незначимые компромиссы	56
Собираем все воедино	58
У вас получилось!	59



## 2 Архитектурные характеристики

### Найдите свои возможности

**Что должна поддерживать архитектура?** Архитектурные характеристики (возможности архитектуры) — фундаментальные строительные блоки любой системы. Без них невозможно принимать архитектурные решения, выбирать архитектурные стили и вообще создавать логичные архитектуры. В этой главе вы узнаете, как определить наиболее распространенные характеристики (масштабируемость, надежность, тестируемость), как они влияют на архитектуру ПО, как помогают принимать архитектурные решения и какие из них наиболее важны в конкретной ситуации. Готовы добавить возможности в архитектуру своего ПО?



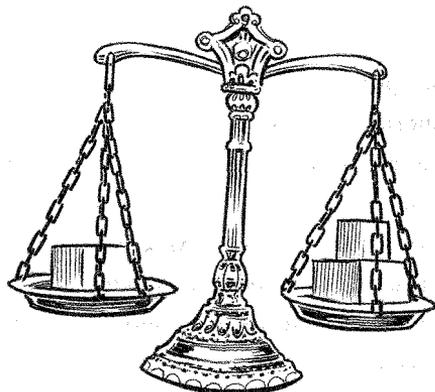
Причины создания социальной сети Lafter	74
Разговор в кабинете	75
Что такое архитектурные характеристики	76
Определение архитектурных характеристик	77
Характеристики не связаны с предметной областью	78
Характеристики влияют на структуру архитектуры	79
Избегайте чрезмерного усложнения	80
Учитывайте явные и неявные возможности	82
Международный зоопарк характеристик	83
Технологические архитектурные характеристики	84
Структурные архитектурные характеристики	85
Эксплуатационные архитектурные характеристики	86
Сквозные архитектурные характеристики	87
Получение архитектурных характеристик из предметной области	92
Получение архитектурных характеристик из знаний об окружении	93
Получение архитектурных характеристик из предметных знаний	93
Составные архитектурные характеристики	95
Приоритеты определяются контекстом	96
Потери при переводе	98
Архитектурные характеристики и логические компоненты	100
Сбалансированность архитектурных характеристик	101
Сокращение количества архитектурных характеристик	102

# 3 Два закона архитектуры ПО

## Все основано на компромиссах

**Что происходит, когда нет «лучших практик»?** Лучшие практики — это относительно безопасные способы достижения определенных целей. Они называются «лучшими» (а не «хорошими») не просто так — они работают. Так почему бы просто не использовать их? Но в архитектуре ПО нет лучших практик. Вам придется тщательно анализировать каждую ситуацию, чтобы принять решение, и объяснять не только «что» нужно сделать, но и «почему».

Как преодолеть этот новый рубеж? К счастью, есть законы архитектуры ПО, которые вам помогут. В данной главе вы узнаете, как анализировать компромиссы и создавать записи архитектурных решений, чтобы фиксировать ответы на вопросы «как» и «почему». К концу главы вы освоите инструменты для навигации по изменчивой территории архитектуры ПО.



Приложение для фанатов кроссовок	114
Что мы уже знаем?	116
Взаимодействие с другими сервисами	118
Анализ компромиссов	119
Анализ компромиссов: версия с очередями	120
Анализ компромиссов: версия с темами	121
Первый закон архитектуры ПО	122
Компромиссы неизбежны	124
Принятие архитектурного решения	125
Что еще делает решение архитектурным?	126
Второй закон архитектуры ПО	128
Записи об архитектурных решениях (ADR)	129
Разговор в кабинете	130
Создание ADR: выбор хорошего заголовка	131
Создание ADR: определение статуса	132
Создание ADR: определение контекста	136
Создание ADR: описание решения	137
Создание ADR: оценка последствий	139
Создание ADR: управление выполнением	141
Создание ADR: примечания	141
Преимущества ADR	144
Two Many Sneakers празднует успех	145

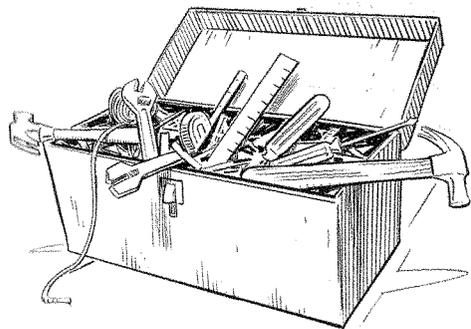
## Логические компоненты

## 4

## Строительные блоки

**Готовы приступить к созданию архитектуры?** Это не так просто, как кажется, и если сделать что-то неправильно, то программная система может обрушиться, как плохо спроектированный небоскреб или мост.

В данной главе показаны несколько приемов определения и создания *логических компонентов* — функциональных строительных блоков системы, которые описывают, как ее части соединяются друг с другом. Эти приемы помогут вам создать надежную архитектуру — фундамент эффективной программной системы.



Еще раз о логических компонентах	156
Запускаем Adventurous Auctions	158
Логическая и физическая архитектура	159
Создание логической архитектуры	161
Шаг 1: выявление основных компонентов	162
Подход на основе рабочих процессов	164
Подход на основе субъектов и действий	166
Ловушка для сущностей	168
Шаг 2: назначение требований компонентам	170
Шаг 3: анализ ролей и обязанностей	172
Стремитесь к связности	173
Шаг 4: анализ архитектурных характеристик	174
Компонент «Прием ставок»	176
Связывание компонентов	177
Афферентные связи	178
Эфферентная связь	179
Оценка связанности	180
Сильносвязанная система	182
Применение закона Деметры	183
Балансирование	185
Несколько заключительных слов о компонентах	186

# 5

## Архитектурные стили

### Классификация и философия

#### Существует множество разных архитектурных стилей.

Каждый из них имеет свои причины создания и принципы использования. Понимание философии стиля поможет вам решить, подходит ли он для вашей области. В этой главе мы коротко поговорим о различных типах архитектурных стилей (рассматриваемых далее в книге), что поможет вам ориентироваться в них, а также в других стилях, с которыми вы рано или поздно столкнетесь в своей профессиональной деятельности.

Существует множество архитектурных стилей	200
Мир архитектурных стилей	201
Организация кода: техническая и предметная	202
Модель развертывания: монолитная и распределенная	204
Монолитные модели развертывания: плюсы	206
Монолитные модели: минусы	207
Распределенные модели развертывания: плюсы	208
Распределенные модели: минусы	209
Вот и все!	212



## Многоуровневая архитектура

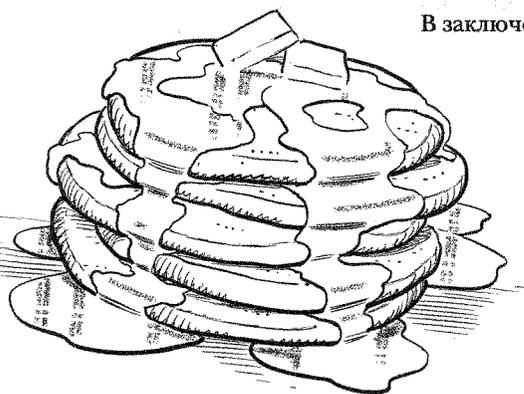
# 6

### Разделение обязанностей

#### Как поступить, если задача проста и время поджимает?

Стоит ли беспокоиться об архитектуре? Все зависит от того, как долго должно просуществовать то, что вы создали. Если решение одноразовое, то отбросьте осторожность. Если нет, то выберите самую простую архитектуру, которая обеспечивает некий уровень организации и позволяет получить выгоду, при этом существенно не влияя на скорость разработки. Например, это может быть *многоуровневая архитектура*, поскольку она проста и легко реализуется с помощью известных паттернов проектирования.

Naan & Pop: сбор требований	218
Назад к паттернам проектирования	220
Многоуровневый паттерн MVC	221
Организация уровней	224
Преобразование уровней в код	225
Предметные области, компоненты и уровни	227
Достоинства многоуровневой архитектуры	230
Уровни и реальный мир: физические архитектуры	231
Компромиссы физической архитектуры	232
Предостережение об изменениях в предметной области	235
Сверхвозможности многоуровневой архитектуры	236
Слабые стороны многоуровневой архитектуры	237
Звездные рейтинги многоуровневой архитектуры	238
В заключение	240



## 7

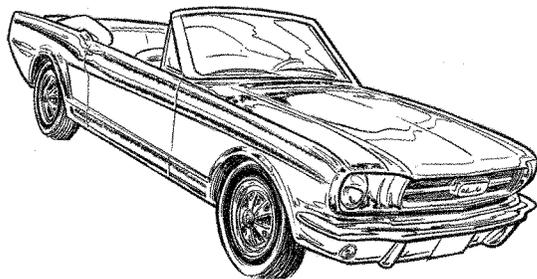
## Модульные Монолиты

## Управляются предметной областью

**Создать монолит можно несколькими способами.** Вы уже познакомились с многоуровневой архитектурой, увязывающей компоненты *технически*. Можно долго развивать многоуровневый монолит, но когда возникает необходимость координировать взаимодействие разных команд, то может потребоваться нечто большее, возможно, даже другой архитектурный стиль.

В этой главе описывается архитектурный стиль *модульный монолит*, делящий приложения по *бизнес-задам*, а не по техническим признакам. Вы узнаете, что это значит и на что следует обратить внимание. Кроме того, мы поговорим обо всех компромиссах этого стиля.

Модульный монолит	251
Изменения в предметной области <i>болезненны</i>	253
Что такое модульные монолиты	254
Покажи мне код!	256
Как сохранить модульность модулей	259
Модульность вплоть до базы данных	263
Остерегайтесь соединений	265
Сверхвозможности модульного монолита	266
Слабые стороны модульного монолита	267
Звездные рейтинги модульного монолита	268
Naan & Pop доставляет пиццу!	270



## Микроядерная архитектура

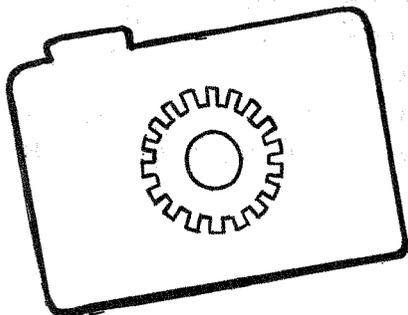


### Индивидуальная подгонка

**Новые возможности можно создавать по одной.** Есть цели, которые особенно хорошо достигаются с помощью определенных архитектурных стилей. Микроядерная архитектура — чемпион по индивидуальной подгонке. Она хорошо подходит для огромного количества приложений. Поняв идею этого стиля, вы начнете видеть его повсюду!

В данной главе вы познакомитесь с архитектурой, позволяющей пользователям делать все по своему усмотрению.

Преимущества Going Green	280
Две части микроядерных архитектур	283
Спектр «микроядерности»	285
Ядро сервиса оценки устройств	287
Инкапсулированные и распределенные плагины	289
Взаимодействие с плагинами	291
Контракты плагинов	296
Going Green включается в работу	297
Сверхвозможности микроядерной архитектуры	298
Слабые стороны микроядерной архитектуры	299
Звездные рейтинги микроядерной архитектуры	300
В заключение	302



Сделай сам

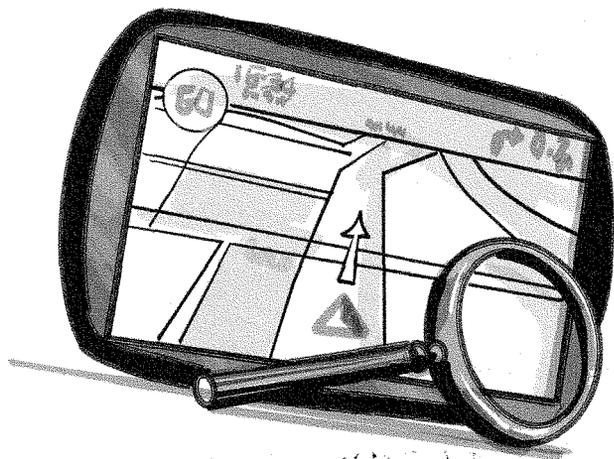
## 9

## Приложение для путешественников TripEZ

## Готовы продолжить знакомство с архитектурой ПО?

В этой главе вы будете играть роль архитектора программного обеспечения. Вам предстоит определять архитектурные характеристики, создавать архитектуру, принимать архитектурные решения и выбирать между многоуровневой, модульной и микроядерной архитектурами. Упражнения в этой главе помогут вам лучше понять, что делает архитектор ПО, и покажут, как много вы узнали. Ваша задача — создать архитектуру для стартапа, разрабатывающего сайт для путешественников.

Облегчаем возможность путешествий	310
Путь пользователя через TripEZ's	311
Планирование архитектуры	312
План действий архитектора	313
Шаг 1: определение архитектурных характеристик	314
Шаг 2: определение логических компонентов	316
Шаг 3: выбор архитектурного стиля	318
Шаг 4: документирование решения	320
Шаг 5: составление схемы архитектуры	322
Нет правильных (или неправильных) ответов!	324

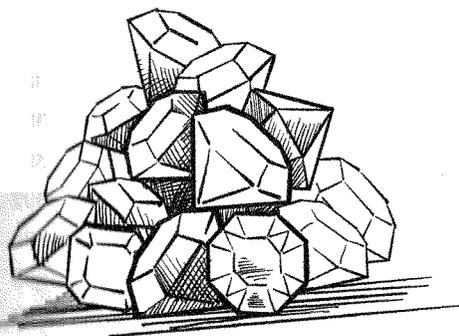


## 10

## Микросервисная архитектура

## Шаг за шагом

**Как упростить изменение архитектуры?** Потребности бизнеса меняются быстро, и архитектуры ПО должны соответствовать им. В этой главе вы узнаете, как создать гибкую архитектуру, которая может меняться вслед за изменениями требований бизнеса, масштабироваться по мере его роста и сохранять работоспособность даже после системных сбоев. Заинтригованы? В данной главе вы познакомитесь с *микросервисами* — архитектурным стилем, решающим все эти и многие другие проблемы.



Хорошо ли вы себя чувствуете?	336
Что такое микросервисы	339
Эти данные только мои	340
Насколько малы эти «микро»	342
Разукрупнение	344
Почему желательно уменьшать микросервисы	345
Укрупнение	346
Почему желательно укрупнять микросервисы	347
Все дело в балансе	348
Общая функциональность	351
Совместное использование кода через общий сервис	352
Совместное использование кода через общую библиотеку	353
Управление рабочими процессами	357
Оркестрация: дирижирование микросервисами	358
Хореография: станцуем	360
Сверхвозможности микросервисной архитектуры	364
Слабые стороны микросервисной архитектуры	365
Звездные рейтинги микросервисной архитектуры	366
В заключение	368

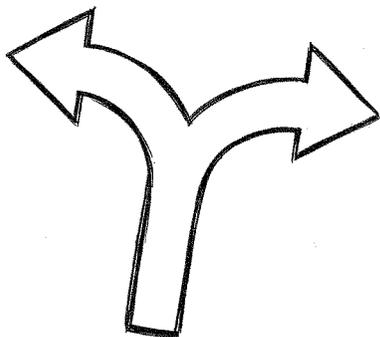
## 11

## Событийно-управляемая архитектура

## Асинхронные приключения

## Может ли архитектура делать много дел одновременно?

Чтобы развиваться, бизнес должен иметь возможность обслуживать все больше пользователей, без задержек и сбоев. В этой главе вы узнаете, как создавать высокопроизводительные системы на основе *событийно-управляемой архитектуры* — популярной разновидности распределенных архитектур, очень быстрых, легко масштабируемых и расширяемых, но при этом довольно сложных. Вы познакомитесь с такими новыми понятиями, как события, сообщения и асинхронная связь. Это поможет вам создать архитектуру, способную решать множество задач одновременно.



Слишком медленно	382
Ускоряемся	383
Der Nile развивается быстрее, чем прежде	384
Что такое событие	386
События и сообщения	388
Иницирующие и производные события	390
Меня кто-нибудь слышит?	392
Асинхронная связь	393
Запустил и забыл	395
Асинхронность ради победы	397
Синхронность ради победы	399
Топологии баз данных	401
Монолитная база данных	402
Базы данных с разделением на предметные области	404
Отдельная база данных для каждого сервиса	406
Сравнение событийно-управляемой и микросервисной архитектур	410
Гибриды: событийно-управляемые микросервисы	414
Сверхвозможности событийно-управляемой архитектуры	416
Слабые стороны событийно-управляемой архитектуры	417
Звездные рейтинги событийно-управляемой архитектуры	418
Собираем все воедино	420
В заключение	421

## 12

Сделай сам

## Проверка знаний

**Готовы проверить свои навыки создания распределенной архитектуры?** В этой главе вам предстоит снова побывать в роли архитектора программного обеспечения. Вы будете определять архитектурные характеристики, принимать архитектурные решения, создавать логическую архитектуру и выбирать, какой подход использовать: микросервисный или событийно-управляемый. Упражнения в этой главе помогут вам лучше понять, что делает архитектор ПО, и покажут, как много вы узнали. Вам предстоит создать архитектуру для системы тестирования учащихся Make the Grade. Удачи! Мы надеемся, что вы получите пятерку!



Добро пожаловать в Make the Grade	436
Процесс тестирования учащихся	437
Планирование архитектуры	438
План действий архитектора	439
Шаг 1: определение архитектурных характеристик	440
Шаг 2: определение логических компонентов	442
Шаг 3: выбор архитектурного стиля	444
Шаг 4: документирование решения	446
Шаг 5: составление схемы архитектуры	448
Нет правильных (или неправильных) ответов!	450

