TEAM TOPOLOGIES

ORGANIZING BUSINESS AND TECHNOLOGY TEAMS FOR FAST FLOW

MATTHEW SKELTON
and MANUEL PAIS

Foreword by Ruth Malan

IT Revolution
Portland, Oregon

МЭТТЬЮ СКЕЛТОН,
МАНУЭЛЬ ПАИС

ТОПОЛОГИИ КОМАНД

2-Е ИЗДАНИЕ

Астана
«АЛИСТ»
2026

УДК 004.42
ББК 32.973.26-018.1

Скелтон, М.

С42 Топологии команд. — 2-е изд.: Пер. с англ. / М. Скелтон, М. Паис — Астана: АЛИСТ, 2026. — 288 с.: ил.

ISBN 978-601-12-9651-9

Книга предлагает практический подход к организационному дизайну в сфере IT, в рамках которого рассматриваются четыре базовых типа команд и несколько паттернов взаимодействия между ними. В основе изложенной методики, получившей название Team Topologies, лежат два ключевых принципа: создание устойчивого потока ценности и управление когнитивной нагрузкой команд. Читатель узнает, как формировать эффективные команды, выстраивать их взаимодействие, снижать избыточные зависимости и адаптировать структуру организации к изменению технологий и бизнес-задач. Авторы показывают, как выстраивать командные структуры и коммуникации так, чтобы они менялись вместе с организацией, а не становились тормозом ее развития.

Для руководителей разработки, IT-менеджеров, архитекторов и лидеров команд

УДК 004.42
ББК 32.973.26-018.1

© 2026 ALIST LLP

Copyright © 2019 by Matthew Skelton and Manuel Pais.
All rights reserved. / Все права защищены.

ISBN 978-1-94278-883-6 (англ.)
ISBN 978-601-12-9651-9 (каз.)

© Matthew Skelton, Manuel Pais, 2019, 2025
© Издание на русском языке. ТОО «АЛИСТ», 2026

Отзывы о книге

TEAM TOPOLOGIES

«Team Topologies предлагает свежие идеи о том, как предвосхищать изменения рынка и технологий и адаптироваться к ним. Чтобы выжить, предприятиям необходимо отказаться от существующих командно-административных структур и передать полномочия лидерам, обладающим наиболее полной информацией для принятия решений и реагирования на ситуации. Эта книга поможет руководителям и бизнес-лидерам сосредоточиться на ключевых стратегиях высокоэффективных команд, чтобы эффективно отвечать на потребности сегодняшнего дня и меняющегося ландшафта будущего».

— **Барри О'Рейли** (Barry O'Reilly), основатель ExecCamp,
бизнес-консультант и автор книг Unlearn и Lean Enterprise

«Нет ничего более фундаментального для управления, чем то, как структурирована организация и какие модели поведения поощряются. Несмотря на это, немногие пытались систематизировать и проанализировать паттерны организационного дизайна ИТ-подразделений, проходящих через цифровые, DevOps- и SRE-трансформации. Скелтон и Пайс не только приняли этот смелый вызов, но и попали точно в цель, создав незаменимый и уникальный ресурс».

— **Деймон Эдвардс** (Damon Edwards), сооснователь Rundeck

«Team Topologies предлагает крайне необходимую основу для оценки и оптимизации организации команд с целью повышения их эффективности. Команды оптимального размера, с четкими границами и правильным уровнем коммуникации готовы приносить ценность компании и удовлетворение участникам команды. Team Topologies сочетает методичный подход с практическими кейсами, раскрывая весь потенциал технических команд».

— **Грег Беррелл** (Greg Burrell),
старший инженер по надежности в Netflix

«Team Topologies Мэттью Скелтона и Мануэля Пайса — уникальная книга. Она окажет значительное влияние на технологические компании. Для непрерывной поставки требуется структурированный и методичный подход к формированию команд, а не копирование каких-то ритуалов Spotify. Это именно та книга».

— **Ник Тьюн** (Nick Tune),
руководитель API-платформы, Navico

«В Condé Nast International [DevOps Topologies] сыграли ключевую роль в понимании нашего текущего состояния DevOps и в формировании видения целевой операционной модели DevOps. Удалось обойти подводные камни и организационные антипаттерны, превосходно описанные в моделях. … Я чрезвычайно рада, что Мэттью и Мануэль развивают успех DevOps Topologies и превращают свои дальнейшие наработки в масштабную книгу Team Topologies, посвященную организационному дизайну».

— **Кристал Хиршорн** (Crystal Hirschorn),
вице-президент по инженерии, глобальной стратегии и операциям,
Condé Nast International

«Высокоэффективная команда — ключевой генератор ценности в современной цифровой экономике. Однако создание и масштабирование адаптивной экосистемы таких команд слишком часто остается труднодостижимой целью. В Team Topologies Скелтон и Пайс предлагают инновационные инструменты и концепции для структурирования операционной модели следующего цифрового поколения. Рекомендуется для CIO, корпоративных архитекторов и стратегов цифровых продуктов по всему миру».

— **Чарльз Бетц** (Charles Betz),
ведущий аналитик, Forrester Research

“Мэттью Скелтон и Мануэль Пайс утверждают, что Team Topologies задумывалась как практическая книга, и это действительно так. Она хорошо структурирована и снабжена навигацией, основана на здравых идеях и предлагает читателю, как и авторам, рассматривать организацию как социотехническую систему или экосистему. Из этого предположения следуют практические рекомендации

без жестких предписаний и умение объяснить подход, обеспечивающий эффективный дизайн технологической и человеческой организационной структуры. Всем, кто работает в области технологического и организационного дизайна, безусловно стоит прочитать [Team Topologies]».

— **Д-р Наоми Стэнфорд** (Dr. Naomi Stanford),
практик организационного дизайна, преподаватель и автор

«Работа Мэттью и Мануэля по паттернам и языку оказалась чрезвычайно ценной как для формирования стратегий трансформации контекста команд со временем в рамках всей организации, так и для помощи бизнес- и технологическому руководству в понимании тем потока и непрерывной поставки».

— **Ричард Джеймс** (Richard James),
руководитель цифровых технологий и инженерии, Nationwide

«Команды — фундаментальный строительный блок организаций; то, как эти команды работают и в какой системе они функционируют, определяет разницу между средним и высоким уровнем эффективности. Эта книга — глубокий источник информации о том, как оптимизировать систему организации с учетом ее текущего контекста».

— **Джереми Браун** (Jeremy Brown),
директор, Red Hat Open Innovation Labs EMEA

«DevOps — это прекрасно. Но как реальные организации должны структурироваться, чтобы его реализовать? Невозможно просто поместить всех в одну несегментированную команду, разместить в огромном офисе с открытой планировкой и ходить вместе на обед или играть в настольный футбол. Team Topologies предлагает практический набор шаблонов для ответа на ключевой вопрос DevOps, который другие руководства оставляют в качестве упражнения для читателя».

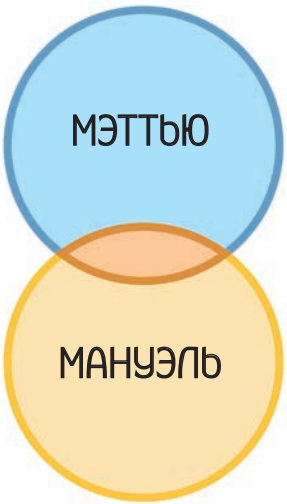
— **Джефф Сассна** (Jeff Sussna),
основатель и CEO, Sussna Associates,
автор книги Designing Delivery

«Если требуется анализ проблем традиционных способов работы, а также практические рекомендации по стратегиям их смягчения (например новые режимы взаимодействия, снижение когнитивной нагрузки и создание соответствующих «Team API»), то эта книга для Вас!».

— **Дэниел Брайант** (Daniel Bryant),
технический консультант/советник
и менеджер новостей в InfoQ

«Team Topologies — увлекательное чтение, поскольку в книге исследуется симбиотическая связь между командами и поддерживаемой ими ИТ-архитектурой. Она выходит за рамки распространенного подхода со статичными оргструктурами или хаотической самоорганизацией и показывает, как эволюционировать систему людей и ИТ-систему совместно».

— **Мирко Херинг** (Mirco Hering),
глобальный лидер DevOps в Accenture,
автор DevOps for the Modern Enterprise



Моей жене, Сьюзи Бек — за всю твою поддержку и вдохновение.

Кэти, моей спутнице жизни и опоре семьи — спасибо за твою бесконечную любовь и поддержку.

Дэну и Бену — ежедневным источникам тепла — надеюсь, эта книга поможет вам понять, чем папа зарабатывает на жизнь.

Содержание

Отзывы о книге Team Topologies 5

Предисловие 15

Вступительное слово 19

Как появилась эта книга 20

Как использовать эту книгу 21

Ключевые влияния, сформировавшие эту книгу 23

Часть I. Команды как средство поставки 26

Глава 1. Проблема организационных схем 29

Коммуникационные структуры организации 31

Мышление в категориях оргсхемы — и есть проблема 31

За пределами оргсхемы 33

Team Topologies: новый способ размышления о командах 35

Возрождение закона Конвея 36

Когнитивная нагрузка и «бутылочные горлышки» 38

Итог: переосмысление структур, назначения и взаимодействий команд 40

Глава 2. Закон Конвея и почему он важен 43

Понимание и применение закона Конвея 43

Обратный маневр Конвея 46

Архитектуры программных систем, способствующие потоку в рамках команды 51

Проектирование организации требует технической экспертизы 53

Ограничение лишних коммуникаций 54

Не каждому нужно общаться со всеми 56

Осторожно: наивное применение закона Конвея 57
Выбор инструментов формирует паттерны коммуникации 57
Множество различных компонентных команд 58
Повторяющиеся реорганизации, создающие «вотчины» или сокращающие штат 59
Итог: закон Конвея критически важен для эффективного проектирования команд в ИТ 59

Глава 3. Мышление «команда прежде всего» 61

Используйте небольшие долгоживущие команды как стандарт 62
Меньший размер способствует доверию 63
Потоки работ — к долгоживущим командам 66
Участникам команды необходимо мышление «команда прежде всего» 68
Приветствуйте разнообразие в командах 69
Поощряйте всю команду, а не отдельных людей 70
Правильные границы минимизируют когнитивную нагрузку 70
Ограничивайте ответственность команды в соответствии с ее когнитивной нагрузкой 71
Измеряйте когнитивную нагрузку через относительную сложность домена 73
Ограничивайте количество и тип доменов на команду 75
Соотносите размер границ программного обеспечения с когнитивной нагрузкой команды 76
Проектируйте «командные API» и содействуйте взаимодействию команд 80
Определяйте «командные API», включающие код, документацию и пользовательский опыт 81
Облегчайте взаимодействие между командами для формирования доверия, осведомленности и обучения 82
Явно проектируйте физическую и виртуальную среду, способствующую взаимодействию команд 83
Предупреждение: инженерные практики являются основополагающими 91
Итог: ограничивайте когнитивную нагрузку команд и облегчайте взаимодействие между командами, чтобы двигаться быстрее 91

Часть II. Топологии команд, обеспечивающие поток.....94

Глава 4. Статические топологии команд..... 97

Антипаттерны команд.....98

Проектирование для потока изменений.....99

Формирование межкомандного взаимодействия для обеспечения потока и обратной связи.....100

DevOps и DevOps-топологии..... 102

Топологии DevOps.....103

Успешные паттерны команд.....104

Функциональные команды требуют высокой инженерной зрелости и доверия.....105

Продуктовым командам требуется поддерживающая система...106

Облачные команды не должны создавать инфраструктуру приложений.....108

SRE имеет смысл при большом масштабе.....108

Факторы, учитываемые при выборе топологии..... 111

Техническая и культурная зрелость.....111

Размер организации, масштаб программного обеспечения и инженерная зрелость..... 112

Разделение ответственности для разрушения «силосов»..... 113

Зависимости и время ожидания между командами..... 114

Использование топологий DevOps для развития организации.....115

Итог: внедряйте и развивайте топологии команд, соответствующие текущему контексту.....118

Глава 5. Четыре фундаментальные топологии команд.....119

Команды, ориентированные на поток.....121

Компетенции внутри команды, ориентированной на поток..... 124

Почему команда, ориентированная на поток, а не «продуктовая» или «функциональная»?.....125

Ожидаемое поведение.....126

Поддерживающие команды..... 127

Ожидаемое поведение.....129

Поддерживающая команда или сообщество практиков?..... 133

Команды сложных подсистем ... 133
Ожидаемое поведение ... 134
Платформенные команды ... 135
Ожидаемое поведение ... 137
Формирование платформы из групп других фундаментальных команд ... 139
Как избежать командных «силосов» в потоке изменений ... 144
Хорошая платформа — «ровно настолько большая, насколько нужно» ... 145
Самая тонкая жизнеспособная платформа ... 146
Снижение когнитивной нагрузки и ускорение разработки продукта ... 146
Привлекательные, последовательные и разумно выбранные ограничения ... 147
Строительство на основе базовой платформы ... 148
Управление продуктом или сервисом в режиме реального времени ... 149
Преобразование обычных типов команд в базовые топологии команд ... 150
Переход преимущественно к командам, ориентированным на поток, для долговечности и гибкости ... 150
От инфраструктурных команд к платформенным ... 151
От компонентных команд к платформенным или другим типам команд ... 152
От инструментальных команд к поддерживающим или платформенным ... 153
Преобразование команд поддержки (support) ... 153
Преобразование архитектуры и архитекторов ... 155
Итог: используйте слабосвязанные модульные группы из четырех конкретных типов команд ... 156
Глава 6. Выбор границ, ориентированных на команду ... 159
Подход «сначала команда» к распределению ответственности и границ в программном обеспечении ... 160
Скрытые монолиты и связанность ... 161

Границы программного обеспечения, или «плоскости разлома» 163
Плоскость разлома: ограниченный контекст бизнес-домена 164
Плоскость разлома: соответствие нормативным требованиям 166
Плоскость разлома: частота изменений 167
Плоскость разлома: расположение команды 167
Плоскость разлома: риск 168
Плоскость разлома: изоляция производительности 169
Плоскость разлома: технологии 169
Плоскость разлома: пользовательские персонажи 170
Естественные «плоскости разлома» для конкретной организации или технологий 171
Реальный пример: производство 174
Итог: выбирайте границы программного обеспечения с учетом когнитивной нагрузки команд 177

Часть III. Эволюция взаимодействия команд для инноваций и быстрой поставки 180

Глава 7. Режимы взаимодействия команд 183
Четко определенные взаимодействия — ключ к эффективным командам 185
Три основных режима взаимодействия команд 186
Сотрудничество: драйвер инноваций и быстрого поиска решений, но с размытием границ 188
X-as-a-Service: четкое распределение ответственности с предсказуемой поставкой, но требует хорошего управления продуктом 191
Содействие: выявление и сокращение пробелов в возможностях 194
Поведение команд для каждого режима взаимодействия 195
Поведение команд в режиме сотрудничества: «Активное взаимодействие и взаимное уважение» 196
Поведение команд в режиме «X как сервис»: «Сосредоточенность на пользовательском опыте» 197
Поведение команд в режиме содействия: «Помогать и получать помощь» 198

Выбор подходящих режимов взаимодействия команд 198
Выбор базовой организации команд 200
Использование обратного маневра Конвея с режимами сотрудничества и содействия 202
Выявление эффективных программных интерфейсов между командами через осознанную эволюцию топологий команд 203
Выбор режимов взаимодействия команд для снижения неопределенности и улучшения потока 204
Использование режима сотрудничества для выявления жизнеспособных взаимодействий «X как сервис» 204
Временное изменение режима взаимодействия команд для развития опыта и эмпатии 205
Использование «неловкости» во взаимодействиях команд для выявления недостающих возможностей и неверных границ 205
Итог: три четко определенных режима взаимодействия команд 207

Глава 8. Эволюция структуры команд с помощью организационного восприятия 209

Какой уровень сотрудничества подходит для режимов взаимодействия команд? 210
Ускорение обучения и внедрения новых практик 212
Постоянная эволюция топологий команд 217
Комбинирование топологий команд для повышения эффективности 223
Триггеры для эволюции топологий команд 224
Триггер: программное обеспечение стало слишком большим для одной команды 224
Триггер: замедление темпа поставки 225
Триггер: несколько бизнес-сервисов зависят от большого набора базовых сервисов 227
Самонаправляемое проектирование и разработка 230
Рассматривайте команды и взаимодействия команд как органы чувств и сигналы 230
ИТ-эксплуатация как высокоценный сенсорный вход для разработки 232
Итог: эволюция топологий команд 236

Глава 9. ЗАКЛЮЧЕНИЕ. Цифровая операционная модель следующего поколения 237
Четыре типа команд и три режима взаимодействия 239
Мышление «команда прежде всего»: когнитивная нагрузка, программный интерфейс команды и архитектура размера команды 240
Стратегическое применение закона Конвея 241
Развивайте организационную структуру ради адаптивности и способности к восприятию 242
Одних «Топологий команд» недостаточно для эффективности ИТ 243
Следующие шаги: с чего начать работу с «Топологиями команд»? 244
1: Начните с команды 244
2: Определите подходящие потоки изменений 245
Словарь терминов 249
Рекомендуемая литература 253
Ключевые управленческие концепции и практики для надежного и быстрого потока 253
Ключевые практики и подходы для организаций, программного обеспечения и систем 253
Ключевые инженерные практики, обеспечивающие быстрый поток 253
Список литературы 255
Примечания 271
Предметный указатель 277
Благодарности 281
Об авторах 283

Предисловие

Поддержание систем небольшими и простыми — достойная цель, однако большинство успешных проектов ей не следуют. Законы эволюции программного обеспечения Лемана и, в частности, закон непрерывного роста отражают естественное стремление к добавлению новых функций по мере использования систем и выявления новых требований или возможностей. Способность справляться с этой возрастающей сложностью и даже использовать ее повышает значимость двух областей проектирования: проектирования систем и проектирования организации, которая создает и развивает эти системы. По первой области существует обширный фронт работ по проектированию систем и программного обеспечения, включая постоянно растущее число книг по предметно-ориентированному проектированию и архитектуре ПО. *Team Topologies* рассматривает проектирование организации разработки программного обеспечения с учетом закона Конвея.

> Основной тезис [. . . .] заключается в том, что организации, проектирующие системы [. . . .], вынуждены создавать проекты, являющиеся копиями коммуникационных структур этих организаций. Мы заметили, что этот факт имеет важные последствия для управления проектированием систем. Прежде всего, был найден критерий структурирования проектных организаций: проектная деятельность должна быть организована в соответствии с потребностью в коммуникации[1].

Приведенная выше цитата из заключения классической статьи Мела Конвея по организационному дизайну разработки программного обеспечения — наиболее подходящее начало для этой книги. *Team Topologies* описывает организационные паттерны структуры команд и режимов взаимодействия, рассматривая воздействие организации на систему как ключевой фактор проектирования.

По мере роста сложности системы, как правило, возрастают и когнитивные требования к организации, которая ее создает и разви-

вает. Управление когнитивной нагрузкой посредством команд с четкими зонами ответственности и границами является отличительной особенностью подхода к проектированию команд в *Team Topologies*. Для достижения правильно ограниченных областей ответственности ищется естественная — и относительно независимая — подструктура системы, с которой соотносятся команды. Этот подход учитывает закон Конвея и использует его для поддержания целостных структур с четкими границами и низкой связностью (так называемый обратный маневр Конвея, который описан в книге).

Если бы этим все ограничивалось, *Team Topologies* была бы полезным развитием идей статьи Конвея, помещенным в современный контекст. Разумеется, *Team Topologies* значительно шире. В частности, в книге выделяются четыре паттерна команд, описываются их результаты, форма и силы, на которые они реагируют и которыми формируются. Основной формой являются команды, ориентированные на поток (stream-aligned)*. Это команды, оптимизированные для потока и обладающие всем необходимым для непрерывной поставки ценности и полной восприимчивости к соответствующим циклам обратной связи. Это означает, что при проектировании системы стремятся не только к слабой связности, но и к такому разбиению, которое поддерживает поток и снижает зависимости и потребность в координации между stream-aligned командами.

Команды сложных подсистем и платформенные команды снижают нагрузку на такие команды, где последние выступают внутренними клиентами возможностей подсистемы или платформы первых (поддерживая все фазы разработки, поставки и эксплуатации для нескольких потоковых команд). Enabling-команды также обслуживают другие команды, но выступают как поставщики услуг, помогая stream-aligned-командам осваивать новые техники, исследовать новые технологии и т. д., позволяя им сохранять фокус и одновременно повышать эффективность.

Мэттью Скелтон и Мануэль Пайс применили свой обширный опыт, рассказав, что необходимо этим различным формам команд для успеха, а также выделив вариации контекста, определив вытекающие из этого последствия для проектирования и указав антипаттерны, которых следует избегать. Кроме того, они щедро вплетают идеи

* В дальнейшем будет использоваться либо термин «команды, ориентированные на поток», либо «ориентированные на поток команды» — в зависимости от сложности предложения. — *Пер.*

из смежных работ и дают ссылки на них. Все это, вместе с набором примеров, придает книге дополнительную глубину.

Team Topologies формирует и углубляет понимание организационной архитектуры через детальное представление ключевых структурных паттернов, режимов взаимодействия и аспектов эволюции. Благодаря ясности и фокусированности книга служит практическим руководством — как для формирования команд и создания условий для решения стоящих перед ними задач, так и для повышения эффективности существующих команд в области адаптивной поставки ценности.

Рут Малан (Ruth Malan),
консультант по Архитектуре в Bredemeyer Consulting

Вступительное слово

«[Современный] организационный дизайн … — это проектирование для технологий совместной работы, для потребностей клиента».

— ***Наоми Стэнфорд*** *(Naomi Stanford), Guide to Organization Design*

Команды всегда находятся в процессе становления, но именно они представляют собой лучший способ обеспечивать непрерывную и устойчивую поставку ценности, согласуясь с потребностями бизнеса. В идеале команды должны быть долгоживущими и автономными, с вовлеченными участниками. Однако команды не существуют изолированно. Необходимо понимать, как и когда взаимодействовать друг с другом. Эти взаимодействия должны эволюционировать со временем, поддерживая различные фазы исследования и исполнения, через которые проходят продукты и технологии в течение своего жизненного цикла. Иными словами, организациям недостаточно стремиться к автономности команд — необходимо постоянно осмыслять и развивать себя, чтобы быстро доставлять ценность клиентам.

В книге предлагается практическая пошаговая адаптивная модель организационного дизайна, которую авторы применяли и наблюдали в работе в компаниях с разным уровнем зрелости: Team Topologies.

При этом Team Topologies не является универсальной формулой успешного построения и эксплуатации программных систем. Существуют команды и организации, добивающиеся успеха при иной организационной структуре, отличной от описанной и рекомендуемой здесь (особенно в организациях с уже устоявшейся сильной культурой и передовыми практиками).

Team Topologies предназначена для того, чтобы предложить ясные паттерны, понятные и применимые для множества команд и организаций, а не для того, чтобы диктовать выдающимся специалистам, как им работать. Этот подход можно сравнить с нотными партиями для оркестра или биг-бэнда, а не с мелодией для выдающегося джа-

зового трубача. Печатные ноты помогают ансамблю добиться успеха, но не предписывают каждую деталь исполнения — многое остается на усмотрение исполнителей с учетом ситуации, площадки и состава. Аналогично огромную ценность представляет согласование общего словаря и способов совместной работы между командами для достижения качественной поставки программного обеспечения.

Подход Team Topologies помогает организациям, которые ищут способ оптимизировать структуру команд, а также тем, кто еще не осознает влияние дизайна команд на бизнес-результаты и особенно на программные системы. Team Topologies позволяет организациям добиваться успеха быстрее и более устойчиво, чем ранее.

Эта книга предназначена для всех, кому важна эффективность поставки и эксплуатации программных систем — для руководителей уровня C (включая CTO/CIO, CEO, CFO и т. д.), менеджеров, руководителей подразделений, архитекторов программных и системных решений, а также для всех, кто участвует в создании или эксплуатации программных систем и стремится повысить эффективность их поставки и работы.

Как появилась эта книга

В 2013 году, внедряя DevOps и непрерывную поставку в одной компании в Великобритании, Мэттью сформулировал первоначальные паттерны (и антипаттерны) DevOps Topologies в статье блога «What Team Structure Is Right for DevOps to Flourish?»[1]. В то время компания, с которой он работал как консультант, испытывала трудности с переходом на современные подходы к поставке программного обеспечения, и ранние паттерны топологий позволили ей рассмотреть различные варианты.

Мануэль взял интервью у Мэттью на конференции QCon London в 2015 году, где тот выступал с докладом о законе Конвея и ранних паттернах DevOps Topology. По итогам была опубликована статья «How Different Team Topologies Influence DevOps Culture» на InfoQ, переведенная на несколько языков[2]. Позже в том же году Мануэль помог расширить паттерны DevOps Topology, а к работе присоединилось сообщество.

С тех пор использование паттернов DevOps Topology стремительно возросло. Они неоднократно упоминались в докладах, статьях и

обсуждениях. Эти паттерны помогли организациям любого размера и из разных отраслей по всему миру осмыслить взаимосвязи между командами и то, как их взаимодействия влияют на организационную культуру и архитектуру программного обеспечения.

Со временем стало ясно, что исходные DevOps Topologies дают статическое представление взаимосвязей между командами, полезное для начальных обсуждений, но довольно ограниченное по охвату. Совместный опыт обучения и консалтинга организаций по всему миру показал, что одни команды работают лучше в относительной изоляции или автономии, тогда как другим требуется тесное сотрудничество. Возник вопрос — почему, и идеи продолжали развиваться на основе обратной связи клиентов.

В итоге это привело Team Topologies к тому виду, в котором они представлены в книге: динамическому и эволюционному подходу к организационному дизайну, основанному на реальных сценариях из разных регионов и отраслей.

Как использовать эту книгу

Team Topologies задумана как практическая книга. Цель ее — представить материал в интерактивной форме и обеспечить максимально возможный объем обучения. Для этого приняты определенные решения по структуре, помогающие ориентироваться в книге.

Во-первых, книга разделена на три части.

В первой части рассматривается закон Конвея — то, как организационные взаимосвязи ограничивают проектирование создаваемых систем и каким образом это свойство можно использовать в своих интересах. Затем дается определение команды и рассматриваются практические ограничения, влияющие на эффективную командную работу.

Во второй части исследуется набор статических паттернов команд, доказавших свою эффективность в индустрии, а также последствия выбора того или иного паттерна с учетом закона Конвея и организационного контекста. Этот раздел помогает определить топологии команд, в целом подходящие для конкретного организационного контекста. Также приводятся рекомендации по соотнесению команд с областями системы с учетом закона Конвея и базовых топологий команд.

Наконец, в третьей части рассматриваются способы эволюции организационного дизайна для формирования мощных возможностей инноваций и оперативной поставки в условиях быстро меняющегося операционного контекста. Объясняется, как использовать подход Team Topologies для создания «чувствующей» организации, реагирующей на требования рынка и пользователей, а также какие последствия это имеет для найма и развития навыков.

Каждая часть начинается с обзора ключевых выводов по главам. В тексте приводятся рисунки и выделенные блоки с информацией, которую важно знать и/или использовать как справочную. Также представлены узнаваемые сценарии, примеры и конкретные рекомендации для различных ситуаций.

Наконец, формы, цвета и шаблоны, встречающиеся на многих рисунках, также имеют значение на протяжении большей части книги. На рис. B1 приведено пояснение.

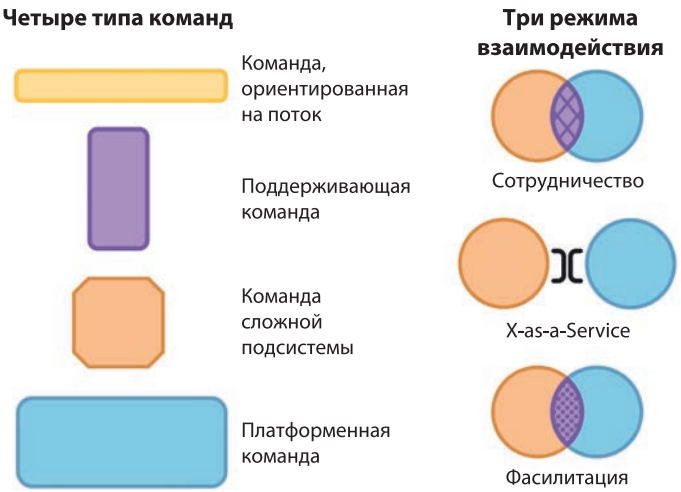


Рис. B1: Четыре типа команд и три режима взаимодействия

На страницах данного печатного издания иллюстрации монохромные, а архив с их цветными версиями выложен на сайте издательства. Его можно скачать по ссылке

https://files.alistbooks.kz/3211_TeamTop2.zip

Для наиболее полного понимания материал следует читать от начала до конца, поскольку тема развивается от главы к главе. Однако текст написан так, что каждый раздел в значительной степени самостоятелен.

Ниже приведены сценарии с соответствующими рекомендациями по чтению, которые могут соответствовать текущим потребностям читателя:

- Нужна бóльшая ясность по различным типам команд и тому, какие из них эффективны.
  - Изучите Главу 1 (обзор), затем Главу 4 (статические топологии), затем Главу 5 (базовые топологии).
- Нужно разделить большую монолитную программную систему.
  - Изучите Главу 6 (границы), затем Главу 3 (команда).
- Нужно улучшить архитектуру программной системы.
  - Изучите Главу 2 (закон Конвея), затем Главу 4 (статические топологии), затем Главу 6 (границы).
- Нужно повысить эффективность команд разработки ПО.
  - Изучите Главу 3 (команда), затем Главу 6 (границы), затем Главу 5 (базовые топологии).
- Нужно повысить моральный дух и эффективность внутри команд.
  - Изучите Главу 3 (команда), затем Главу 5 (базовые топологии).
- Нужно понять, куда вкладывать усилия с учетом планируемого роста.
  - Изучите Главу 1 (обзор), затем Главу 5 (базовые топологии), затем Главу 8 (эволюция топологий).
- Нужно понять, как эволюционировать топологии команд в ответ на меняющиеся бизнес-потребности.
  - Изучите Главу 7 (динамические аспекты), затем Главу 8 (эволюция топологий и организационное сенсирование).

Ключевые влияния, сформировавшие эту книгу

Помимо собственного опыта авторов, книга во многом опирается на ряд родственных подходов и направлений мышления. Во-первых, предполагается, что организация — это социотехническая систе-

ма или экосистема, формируемая взаимодействием людей и команд внутри нее. Иными словами, организация представляет собой взаимодействие между людьми и технологиями. В этом смысле книга перекликается с идеями из следующих областей: кибернетика (в частности, представление об организации как о «сенсорном механизме», восходящее как минимум к 1948 году, когда была опубликована книга Норберта Винера Cybernetics: Or Control and Communication in the Animal and the Machine), системное мышление (особенно работы W. Edwards Deming), фреймворк Cynefin для оценки сложности доменов (описан Dave Snowden и Mary Boone в статье A Leader's Framework for Decision Making в Harvard Business Review, 2007), теория адаптивной структурации (термин введен Gerardine DeSanctis и Marshall Scott Poole в статье Capturing the Complexity in Advanced Technology Use: Adaptive Structuration Theory), где подчеркивается, что влияние технологии не задано заранее и зависит от того, как группы и организации ее воспринимают.

Во-вторых, авторы исходят из того, что команда ведет себя иначе, чем просто набор отдельных людей, и что ее необходимо развивать и поддерживать в процессе эволюции и работы. Здесь используются идеи Bruce Tuckman (который предложил модель четырех стадий — forming, storming, norming, performing (рассматриваются и объясняются далее) — для развития команды в своей работе 1965 года «Developmental Sequence in Small Groups»), Russ Forrester и Allan Drexler (исследовавших эффективность командных организаций в статье 1999 года «A Model for Team-Based Organization Performance»), Pamela Knight (которая нашла доказательства того, что «бушевание» происходит на протяжении всего жизненного цикла команды, в своей работе 2007 года «Acquisition Community Team Dynamics: The Tuckman Model vs. the DAU Model»), Patrick Lecioni (исследующего общие проблемы взаимодействия в основополагающей книге «The Five Dysfunctions of a Team: A Leadership Fable») и аналогичные теории и исследования, ориентированные на работу в команде.

В-третьих, предполагается, что закон Конвея (или его вариант) является мощным фактором, определяющим форму программного продукта, и что организации выиграют от учета последствий этого закона. В связи с этим используются труды и идеи Мела Конвея; консультанта по архитектуре программного обеспечения и лауреата премий в области проектирования командных организаций Рут Малан; технического директора ThoughtWorks и одного из сторонников

«обратного маневра Конвея» Джеймса Льюиса, а также других авторов и практиков.

Наконец, авторы опираются на многочисленные источники, описывающие практические успехи в разработке и эксплуатации программных систем в масштабе всей организации, включая такие компании, как Adidas, Auto Trader, Ericsson, Netflix, Spotify, TransUnion и другие. Размер и скорость развития этих организаций позволили им увидеть ощутимые преимущества от изменений в организационной структуре и командном взаимодействии в течение периода от нескольких месяцев до нескольких лет.

По мере чтения книги авторы надеются вдохновить Вас на то, чтобы переосмыслить свои представления о командах, их структурах и принципах их функционирования.

ЧАСТЬ I

Команды как средство поставки

Ключевые выводы

Глава 1

- Закон Конвея указывает на значительные преимущества совместного проектирования архитектуры программного обеспечения и взаимодействий между командами, поскольку на них действуют схожие силы.
- Team Topologies проясняет назначение команд и их зоны ответственности, повышая эффективность их взаимосвязей.
- Team Topologies применяет гуманистический подход к разработке программных систем, одновременно подготавливая организации к стратегической адаптивности.

Глава 2

- Организации вынуждены создавать решения, отражающие пути коммуникации.
- Дизайн организации ограничивает «пространство поиска решений», сужая возможные варианты архитектуры программного обеспечения.
- Требование, чтобы каждый взаимодействовал с каждым, — верный путь к хаосу.
- Выбирайте архитектуры программного обеспечения, поддерживающие поток в границах команды.
- Ограничение путей коммуникации четко определенными взаимодействиями между командами приводит к модульным, слабо связанным системам.

Глава 3

- Команда, а не отдельные специалисты, — наиболее эффективное средство поставки программного обеспечения,.
- Ограничивайте размер групп из нескольких команд в организации, исходя из числа Данбара.
- Сужайте зоны ответственности команд в соответствии с максимально допустимой когнитивной нагрузкой.
- Устанавливайте четкие границы ответственности команд.
- Изменяйте рабочую атмосферу в коллективе, чтобы помочь командам добиться успеха.

1 Проблема организационных схем

Организации следует рассматривать как сложные и адаптивные организмы, а не как механистические и линейные системы.

— ***Наоми Стэнфорд***, *Guide to Organisation Design*

Специалисты по технологиям находятся в постоянном действии: создают и обновляют системы с невероятной скоростью и комбинируют различные типы технологий для формирования качественного пользовательского опыта. Для достижения требуемых бизнес-результатов мобильные приложения, облачные сервисы, веб-приложения, а также встраиваемые, носимые или промышленные IoT-устройства должны эффективно взаимодействовать.

Сегодня эти системы затрагивают почти все аспекты повседневной жизни людей, причём всё более существенно. Если программное обеспечение спроектировано плохо — или, что важнее, если нарушено соответствие между взаимодействием программного обеспечения, поставщика и клиента, — последствия затрагивают пользователей напрямую. Можно оказаться далеко от дома из-за сбоя приложения для вызова такси. Может не получиться оплатить аренду, если не работает интернет-банкинг или связанные с ним процессы. Возможна даже угроза жизни при отказе медицинского устройства. Никогда прежде явное социотехническое проектирование не было столь важным.

Создание и эксплуатация таких высокосложных взаимосвязанных программных систем — это командная деятельность, требующая совместных усилий специалистов с разными навыками на разных платформах. Кроме того, современные ИТ-организации должны быстро и безопасно поставлять и эксплуатировать программные системы, одновременно развиваясь и адаптируясь к изменениям и давлению со стороны бизнеса или регуляторной среды. Бизнес больше

не может выбирать между оптимизацией ради стабильности и оптимизацией ради скорости.

Однако, несмотря на эти риски и требования, многие организации по-прежнему структурируют людей и команды способами, которые противоречат современным практикам разработки и эксплуатации программного обеспечения. Организации, чрезмерно полагающиеся на оргсхемы и матричные структуры для разделения и контроля работы, часто не создают условий, необходимых для внедрения инноваций при сохранении высокой скорости поставки. Для успеха необходимы стабильные команды, а также эффективные модели и способы взаимодействия внутри команд. Требуются инвестиции в наделённые полномочиями и квалифицированные команды как основу гибкости и адаптивности. Чтобы выживать на всё более конкурентных рынках, организациям нужны команды и сотрудники, способные замечать изменения контекста и эволюционировать вместе с ними.

Хорошая новость заключается в том, что можно сочетать скорость и надежность при правильном мышлении и использовании подходящих инструментов. Такой подход подчёркивает адаптивность наряду с воспроизводимостью, ставя в центр внимания команды и людей. Как отмечают Марк Шварц и соавторы в работе *Thinking Environments* (2016), «организационная структура должна координировать зоны ответственности для поддержки целей поставки высококачественного программного обеспечения с ощутимым эффектом»[1].

Участникам технологических команд, управляющим этими взаимодействиями, необходимо перейти от восприятия команд как набора взаимозаменяемых специалистов, которые будут успешны при условии следования «правильному» процессу и использования «правильных» инструментов, к восприятию людей и технологий как единой социотехнической экосистемы человек/компьютер, углерод/кремний. При этом важно обеспечить внутреннюю мотивацию команд и предоставить им реальные условия для выполнения работы наилучшим образом в рамках такой системы.

В этой главе Team Topologies будет представлена как адаптивная модель организационного дизайна технологических структур, позволяющая бизнесу достигать скорости *и* стабильности. Но сначала необходимо рассмотреть, как реальные коммуникационные структуры в большинстве организаций сильно отличаются от того, что показывает оргсхема, и каковы последствия этого.