

Содержание



От издательства	12
Об авторе	14
Благодарности	15
Предисловие д-ра Баскара Гоша	17
Предисловие автора	19
Введение. Как мы к этому пришли	25
Принципы традиционного производства не так просто применить в ИТ-поставке	27
Понятия операционных издержек и объема работы выступают основой для смены подхода	30
Часть А. Создание подходящей экосистемы	35
Глава 1. Путь к изменениям	37
Явные ИТ-процессы.....	38
Создание первой схемы трансформации	38
Управление трансформацией	41
Видимые ИТ-услуги	48
Управление процессом поставки ИТ-услуг	50
Предоставление ИТ-услуг по принципу Lean	52
Глава 2. Принятие быстро меняющейся реальности	57
Анализируем портфолио приложений	59
Поиск минимального жизнеспособного кластера	61
Что делать с истинно устаревшими приложениями	63
Управление вашим портфолио и контрольными точками.....	64
Глава 3. Готовые программные пакеты и поставщики программных продуктов	71
Как выбрать подходящий продукт для вашей организации	73
Что же тогда делать с существующими устаревшими приложениями.....	76

Глава 4. Поиски подходящего партнера	83
Как добиться выгодных стратегических партнерских отношений с системным интегратором	84
Важно контролировать свой путь в ИТ	85
Смена парадигмы «Разработка – администрирование – внедрение»	86
Культурное взаимодействие в партнерстве.....	87
Контракты с партнерами	89
Партнерство со стороны SI	92
Оценка партнера	93
Заключение части А.....	97
Часть Б. Люди и организация процессов	99
Глава 5. Контекст во главе угла	101
Понимание бизнес-проблемы	102
Поиски жизнеспособного решения проблемы.....	105
Планирование стадии поставки и подготовка к ней	106
Глава 6. Структура, приносящая успех	115
Команда платформы	117
Agile-команды.....	120
Центр автоматизации тестирования	124
Команды по обслуживанию и обучению.....	125
Но как же проектные менеджеры?	126
Глава 7. Из тестировщиков в инженеры по качеству	129
Организация обеспечения качества.....	130
Процесс обеспечения качества.....	131
Пара слов об автоматизации функционального тестирования	136
Управление качеством и показатели качества.....	139
Глава 8. Управляйте людьми, а не «ресурсами»	143
Личные встречи	144
Обратная связь.....	145
Делегирование обязанностей	146
Создание «культуры без обвинений»	146
Оценка культуры вашей организации	147
Заключение части Б.....	149
Часть В. Технологические и архитектурные аспекты	151
Глава 9. Различные модели доставки продукта	153
Обзор моделей поставки продукта	153
Модель А: непрерывная доставка.....	154
Модель Б: облачная доставка.....	159
Модель В: доставка с поддержкой контейнеров	162
Оценка модели доставки: бессерверная доставка	166
Схема возможностей	166

Глава 10. Архитектура приложений и микросервисы	171
Хорошая архитектура дается нелегко	172
Совершенствование вашей архитектуры с течением времени.....	174
Знакомство с микросервисами.....	176
Глава 11. Эффективное управление приложениями и применение DevOps-инструментов	183
Современная эксплуатация приложений	184
Формирование DevOps-платформы и работа с ней.....	186
Работа с небольшими партиями изменений.....	189
Глава 12. Облако	193
Базовые принципы облачной экономики	194
Рассуждения об облачной архитектуре.....	194
Управление облаком.....	196
Проектирование надежности сайтов	198
Заключение. Осознанная работа	203
Тайм-менеджмент	205
Приложение. Аналогия с заводом: подробности	209
Фундаментальный принцип: процессы в производстве и креативные процессы в ИТ.....	209
Оценка продуктивности и качества на основе стандартизированных результатов	209
Функциональная специализация и набор навыков сотрудников.....	211
Предсказуемость процесса производства и управление им	212
Важность предварительного планирования и возможность рассчитывать на него.....	213
Управление доставкой.....	213
Автоматизация = продуктивность.....	214
Масштабирование усилий для доставки большей ценности	215
Централизация ресурсов.....	216
Офшоринг	216
Аутсорсинг.....	217
Материалы для самостоятельного изучения	219
Глоссарий	221
Список литературы и видеоресурсов	227

Впервые о DevOps заговорили в связи с переходом в эру цифровой экономики, когда скорость выпуска на рынок продуктов стала одним из ключевых конкурентных преимуществ. Технологиям, обеспечивающим стремительное развитие бизнеса, пришлось бежать со всех ног, чтобы только оставаться на месте, а для достижения дополнительных результатов – как минимум в два раза быстрее. Компаниям понадобились инструменты для быстрого и непрерывного улучшения качества существующих процессов разработки продуктов и их максимальной автоматизации, потому что хороший продукт стал равен хорошей ИТ.

Свой путь погружения в DevOps я начала несколько лет назад, когда возглавила отдел тестирования системы подготовки регулярной банковской отчетности Neoflex Reporting, которая отличалась большим количеством параллельных веток разработки и обилием ручных процессов. В ее разработку к этому моменту уже были вложены десятки тысяч человеко-часов.

Засучив рукава наша команда взялась за точечную автоматизацию этапов жизненного цикла продукта. В целом мы достигли неплохих результатов, но добиться слаженной и синхронной работы от всех участников процесса оказалось по-настоящему трудной задачей. Периодически возникающие «тут подкрутить», «там вручную запустить», «а это не на моей стороне», «я был на обеде», «исторически сложилось» тормозили ожидаемое от автоматизации ускорение.

Осознать, что же делать дальше, нам помогла книга, которую вы сейчас держите в руках. Мы прочитали ее всей командой и здорово переработали текущие процессы взаимодействия в парадигме слаженности, простоты и удобства. А процессы сборки, развертывания инфраструктуры, установки, тестирования и выдачи поставки объединили в непрерывный производственный конвейер.

«DevOps для современного предприятия» – книга об эффективной ИТ настоящего. Захватывающий и понятный путеводитель, способный обобщить, разложить по нужным полочкам существующий опыт и обогатить его ценными идеями.

В книге описаны основные шаги и принципы построения производственно-го взаимодействия, автоматизации процессов и развития культуры разработки ПО. Теория щедро сдобрена историями реальных людей и компаний, прошедших непростой, но интересный путь к DevOps.

Неоспоримая ценность этой книги в том, что она помогает вырваться из рутины бытия и взглянуть на текущие процессы совершенно другими глазами. Приходит осознание того, что на точечных «костылях» автоматизации далеко не уйти, появляется понимание того, как выглядит путь роста и развития, который подходит именно вашей компании, проекту, продукту.

Желаю вам приятного чтения, и пусть эта книга станет для вас источником неиссякаемого вдохновения!

Лина Чуднова,
руководитель практики DevOps компании «Неофлекс»



О компании «Неофлекс»

«Неофлекс» создает ИТ-платформы для цифровой трансформации бизнеса, помогая заказчикам получать устойчивые конкурентные преимущества в цифровую эпоху. Мы фокусируемся на заказной разработке программного обеспечения, используя передовые технологии и подходы.

Наш отраслевой опыт и технологическая экспертиза, усиленная собственными акселераторами разработки, позволяют решать бизнес-задачи любого уровня сложности. Среди наших заказчиков более половины российских банков, входящих в топ-100, а также компании из 18 стран Европы, Азии и Африки.

Телефон: +7 (495) 984 25 13

Сайт: www.neoflex.ru

Современный бизнес ожидает от ИТ все больше. Требования к производству программного обеспечения становятся жестче, планка качества – выше, времени и ресурсов – меньше. Бизнес хочет все сразу: Time2Market, снижение сбоев и отказов, и притом дешевле, а еще чтобы ИТ «прочитали мысли» и сделали правильно правильные вещи. Нужна магия, а время шаманов и энтузиастов ушло.

Необходим четкий процесс, который дает результат – когда это нужно заказчику или нужно быстрее. Все чаще информационные технологии не просто поддерживают бизнес, но становятся основным драйвером развития и трансформации.

На мой взгляд, DevOps не является революцией или универсальным лекарством. Это точно не магия, это лучше – качественный скачок, перевернувший принципы построения процессов производства ПО. Это то, что позволяет ИТ делать невозможное и при этом не ломаться от чрезмерного стресса и внутренних противоречий. Это вынужденная автоматизация рутинных операций. Это изменение требований к ИТ-специалистам, трансформация рынка труда ИТ и появление новой профессии «инженер DevOps».

Методики DevOps связывают ИТ в единый процесс, позволяя сделать его не только измеряемым и контролируемым, но и понятным и динамичным. Этот подход – если хотите, новая философия – связывает специалистов разных ИТ-областей в единую эффективную команду. Вы можете сосредоточиться на управлении, исходя из принципа достижения глобального оптимума, а не частных оптимумов на разных участках ИТ-производства. Узкие места процессов выходят из тени, а оптимизация становится неизбежной.

В цифровую эпоху скорость и возможность динамического изменения имеют первостепенное значение. Время – это не просто деньги. Зачастую это часть стратегии, основа конкурентной борьбы, а динамика изменений – это конкурентное преимущество, формирующее уникальность. В этой книге вы найдете советы и примеры, что позволит погрузиться в мир DevOps. Книга захватывает и вовлекает, так что вы сами не заметите, как DevOps станет частью вашей жизни и войдет в культуру производства.

Искренне желаю вам увлекательного чтения. Узнавайте, практикуйте и пробуйте подходы DevOps уже завтра. Поверьте, это полезно, и это пригодится!

Владимир Туровцев,
управляющий партнер, директор по развитию бизнеса Logrocon,
ведущий преподаватель программы MBA школы бизнеса МИРБИС,
победитель конкурса «Лидеры России» 2018–2019



О компании

Компания Logrocon – Microsoft Gold DevOps Partner. С 2012 года разрабатывает, тестирует и внедряет ПО. Организатор международной конференции DevOps Forum 2019, международных некоммерческих комьюнити-конференций DevOpsDays Moscow 2018 и 2017. Компания реализовала более 40 проектов по DevOps.

Телефон: +7 (495) 777-00-84

Сайт: <https://logrocon.ru/>

Соцсети:

<https://www.facebook.com/logrocon/>

<https://www.facebook.com/OrangeOceanIT/>

YouTube-канал: <http://www.youtube.com/c/Логрокон>

Сайт организованных в 2017–2019 гг. конференций: <https://devopsforum.ru/>

Сайт иммерсивной ИТ-площадки «Оранжевый океан»: <https://orange-ocean.ru/>



ОТ ИЗДАТЕЛЬСТВА

Отзывы и пожелания

Мы всегда рады отзывам наших читателей. Расскажите нам, что вы думаете об этой книге – что понравилось или, может быть, не понравилось. Отзывы важны для нас, чтобы выпускать книги, которые будут для вас максимально полезны.

Вы можете написать отзыв на нашем сайте www.dmkpress.com, зайдя на страницу книги и оставив комментарий в разделе «Отзывы и рецензии». Также можно послать письмо главному редактору по адресу dmkpress@gmail.com; при этом укажите название книги в теме письма.

Если вы являетесь экспертом в какой-либо области и заинтересованы в написании новой книги, заполните форму на нашем сайте по адресу http://dmkpress.com/authors/publish_book/ или напишите в издательство по адресу dmkpress@gmail.com.

Список опечаток

Хотя мы приняли все возможные меры для того, чтобы обеспечить высокое качество наших текстов, ошибки все равно случаются. Если вы найдете ошибку в одной из наших книг – возможно, ошибку в основном тексте или программном коде, – мы будем очень благодарны, если вы сообщите нам о ней. Сделав это, вы избавите других читателей от недопонимания и поможете нам улучшить последующие издания этой книги.

Если вы найдете какие-либо ошибки в коде, пожалуйста, сообщите о них главному редактору по адресу dmkpress@gmail.com, и мы исправим это в следующих тиражах.

Нарушение авторских прав

Пиратство в интернете по-прежнему остается насущной проблемой. Издательство «ДМК Пресс» очень серьезно относится к вопросам защиты авторских прав и лицензирования. Если вы столкнетесь в интернете с незаконной публикацией какой-либо из наших книг, пожалуйста, пришлите нам ссылку на интернет-ресурс, чтобы мы могли применить санкции.

Ссылку на подозрительные материалы можно прислать по адресу электронной почты dmkpress@gmail.com.

Мы высоко ценим любую помощь по защите наших авторов, благодаря которой мы можем предоставлять вам качественные материалы.



Об авторе



На протяжении десятков лет Мирко Херинг работал над ускорением доставки программных продуктов при помощи инновационных подходов (в современной терминологии – DevOps); десять лет назад он начал экспериментировать с Agile-методиками. Будучи руководителем по DevOps и Agile по Азиатско-Тихоокеанскому региону в компании Accenture, он оказывает поддержку крупным публичным и частным компаниям по всему миру, помогая им в поисках эффективных методов ИТ-доставки. Мирко ведет блог **NotAFactoryAnymore.com** и выступает на международных конференциях, делаясь своими знаниями и опытом. Вы также можете подписаться на его страницу в Twitter: **@MircoHering**.



Благодарности

То, что любят говорить о воспитании детей, справедливо и для написания книги – такую работу в одиночку не провернуть. И наверняка я упустил кого-то из тех, кто достоин упоминания на этих страницах. Когда мы встретимся на следующей конференции, с меня причитается угощение!

Прежде всего хочу поблагодарить фантастическую команду, которая поддерживала меня на протяжении всего процесса редактуры: Тодда, Джина, Лею и Карен – без вас мои мысли никогда не уложились бы в связный текст. Это был нелегкий труд, но работать с вами было мне в радость!

Еще хочу сказать спасибо за отзывы коллегам, которые посвятили этому свое время: Эрику, Енгу, Эджею и Эмили. Вы помогли отшлифовать конечный результат без ущерба для смысла.

Особую благодарность я хотел бы высказать трем ребятам, без которых это издание не состоялось бы. Эрик, Тодд и Джин помогли мне преодолеть установку «я никогда не напишу стоящей книги» и сфокусироваться на мысли «у меня и правда есть что поведать миру». Без вашей поддержки я не решился бы приступить к делу.

Также я хочу выразить благодарность руководству компании Accenture – Баскару Гошу, Адаму Бердену и Питеру Ваккасу – за поддержку этого проекта и предоставление мне гибкого рабочего графика. В результате я мог отвести достаточно времени на свое «хобби».

Спасибо Гэри Груверу, который поделился со мной советами по поводу написания книги и помог найти оптимальную схему работы.

В дополнение хочу поблагодарить всех сотрудников Accenture – практикующих DevOps и всех прочих – за то, что они помогли мне выработать свой подход к трансформированию компаний. Хочу сказать спасибо клиентам, с которыми я работал и у которых я постоянно учусь чему-то новому. Некоторые из ваших идей нашли свое отражение в этой книге.

И последнее, но самое важное: огромное спасибо моей жене Аньяли, которая здорово поддерживала меня – даже тогда, когда ее внимание было поглощено очаровательным младенцем, появившимся на свет как раз в дни написания этой книги. Аньяли, ты просто чудо!



Предисловие

д-ра Баскара Гоша

На протяжении многих лет работы в ИТ-сфере я наблюдал не один кризис, связанный с появлением новых технологий, бизнес-моделей, а также со сменой глобальных экономических циклов. На волне подобных перемен DevOps приобретает особую значимость. Он замечателен не только своими принципами, но и изменениями, к которым ведет.

Мне нравится говорить, что DevOps можно использовать без Agile, но Agile нельзя использовать без DevOps. Это один из многих процессов трансформации, которые были запущены на заре DevOps. Помогая разработчикам наращивать эффективность и, соответственно, добиваться лучших результатов, DevOps способствует творческому самовыражению, что, в свою очередь, приводит к постоянному улучшению процессов поставки систем.

И хотя я люблю вспоминать те времена, что я провел, управляя процессами инфраструктур в крупных организациях, дело существенно осложнял тот факт, что мне приходилось разделять обязанности между разработкой и управлением процессами. Конечно, это была эпоха монолитных систем, использующих методологию водопада в больших масштабах и периодическую поставку программных пакетов. Для такой среды разработки разделение обязанностей представляло практичную и эффективную модель, которая соответствовала темпам изменений в системе.

Однако в цифровую эпоху скорость приобретает первостепенное значение. Разделение обязанностей сегодня не содействует все более ускоряющейся смене подходов, которые на данный момент необходимы бизнесу.

В этой книге Мирко раскрывает больше, чем просто технику DevOps, – он делится своей страстью к улучшению процессов проектирования программных

продуктов. Используя понятные аналогии и рекомендации, Мирко показывает, как помочь компании внедрить DevOps. Не важно, начинаете ли вы свой путь в DevOps или уже неплохо его освоили и хотите узнать, как применить его принципы в больших масштабах, – в процессе чтения книги вы несомненно оцените DevOps и его преимущества.

Д-р Баскар Гош,
исполнительный директор Accenture Technology Services
Бангалор, Индия
Март 2018 года



Предисловие автора

Обучение – дело не обязательное, а добровольное.

Совершенствование – дело не обязательное, а добровольное.

Но чтобы выжить, мы должны учиться.

Уильям Эдвардс Демминг

Одним из самых благодарных занятий в моей карьере был поиск наиболее эффективных способов создания значимых проектов и передача этой информации как можно большему кругу людей. Когда мы работаем неэффективно, то тратим время на ненужные вещи, на повторяющиеся и скучные задания, что не приносит никакого удовлетворения. Я не вынашиваю цели изменить мир, побуждая людей развлекаться на работе, но, мне кажется, труд должен быть человеку в радость. Когда сотрудники занимаются любимым делом, хороший результат не заставит себя ждать.

С того времени, как я пришел работать консультантом в Accenture – а было это более десяти лет назад, – я успел поработать с десятками команд, увеличивая эффективность и темп их работы, а следовательно, и качество поставки продукции. Но и прежде, чем стать консультантом, я думал о том, как сделать процессы в ИТ более эффективными.

На заре моей трудовой деятельности, в конце 1990-х, в ИТ только начинали привлекать к работе сторонних сотрудников. Первые несколько лет я набирался опыта в исследовательских лабораториях IBM, работая в области телематики и над инструментами для разработчиков (например, разрабатывая языки для особых CPU и предоставляя соответствующие компиляторы и IDE-расширения). Когда я начинал, разработка пакетного программного обеспечения была развивающейся отраслью, но основной объем работы осуществлялся

для частной разработки и штатно. Единственным способом увеличения продуктивности было усовершенствование автоматизации, и во всех моих ранних проектах мы выстраивали креативные решения при помощи различных shell-скриптов, Perl-скриптов и других оригинальных инструментов, которые облегчают жизнь разработчикам и администраторам. Я был очень доволен, когда выстраивал все эти решения по автоматизации и видел, как всем участникам проекта становится легче и веселее работать.

Был любопытный случай. В течение пяти лет я был занят в двух больших проектах и работал над созданием таких инструментов для разработчиков, которые позволяли успешно поставлять продукты. Когда я закончил работу на этих проектах, то начал оценивать разные организации, и мне показалось, что того уровня автоматизации на данный момент недостаточно. Все-таки я всю свою профессиональную жизнь решал задачи по автоматизации для команд поставки. Когда я говорил об этом с коллегами, мне становилось ясно, что пакетное программное обеспечение и возможности нештатной поставки открывают путь к увеличению продуктивности и снижению стоимости, чем многие организации пользуются, вместо того чтобы инвестировать в хорошие практики разработки и инструменты¹. Я потратил следующие несколько лет, осваивая ту нишу на рынке, которая, как считалось, помогает организациям реализовывать инструменты для поставки. Но, по правде, организации не были в восторге от предложения вкладываться в это.

Предложение увеличить объем нештатных поставок и сократить среднюю стоимость рабочего дня разработчика (известную также как *средняя суточная ставка*) казалось более привлекательным для организаций, чем нечто более сложное и трудноизмеримое – разработка хорошей платформы поставки, которая помогает всем повысить производительность труда в ИТ-сфере. В конце концов, как вы будете измерять производительность в ИТ? Я солидарен с Рэнди Шупом и Эдрианом Кокрофтом, которые на конференциях признавались, что на протяжении всей своей карьеры пытались выявить достойный показатель производительности, но не смогли найти ничего стоящего. Я поднимал эту тему в блоге, рассказав о том, что производительность в ИТ очень сложно измерять; вместо этого стоит оценивать время выполнения циклов, неоправданные расходы и поставленную функциональность. Важно брать во внимание некоторые значимые метрики, иначе вы не сможете наблюдать за вашим развитием, поэтому получается, что в общем случае производительность в ИТ – достаточно расплывчатое понятие, и нам необходимо искать другие показатели, которые помогут нам судить о том, насколько наша работа эффективна.

В дальнейшем я провел несколько лет, работая над тем, чтобы понять, откуда растут проблемы в ИТ и как их решить. Мне повезло, так как мои исследования совпали со временем распространения технологий и методологий,

¹ Подробнее об этом рассказывается в моей статье на DevOps.com: *Why We Are Still Fighting with the Same Problems in DevOps as 15 Years Ago* («Почему мы не можем справиться с проблемами в DevOps, существовавшими еще 15 лет назад»).

которые впоследствии заложили основы нового способа реализации поставки в ИТ: Agile, DevOps и *облако*, среди прочего, значительно упростили реализацию тех решений, которые я построил за всю свою карьеру. Ниша, в которой я работал, становилась все более привлекательной, – сегодня сложно найти организацию, которая не говорит об Agile и DevOps.

Но, оглядываясь назад и оценивая, где сегодня находится ИТ-индустрия, мы понимаем, что ИТ-поставка все еще не там, где она должна быть. Мы склонны считать, что *непрерывная поставка* является прекрасной практикой создания современных архитектур приложений, но если мы поищем организации, которые в совершенстве ей овладели, то обнаружим, что их совсем немного; к тому же их непросто найти. Многие организации выстраивают рабочие процессы, которые долгое время развивались с оглядкой на методы традиционного производства. В конце концов, эти практики были хорошо систематизированы во многих учебных планах MBA и накопили сотни лет практики. Но данные подходы и идеи уже не актуальны.

Я сам не овладел этим в совершенстве и все еще упорно учусь каждый день, но хочу, чтобы мои упражнения были доступны как можно большему кругу людей. Как вы видите на рис. 1, я разработчик до мозга костей: в первую очередь разрабатываю технические решения, вместо того чтобы думать о заинтересованных людях. Я на своих ошибках понял, что одно лишь применение правильных методов и инструментов не способно магическим образом преобразовать организацию. Сменить культуру – самая сложная задача, но также и самая значительная. Мне понадобилось совершить множество ошибок и промахов, чтобы узнать то, что я собираюсь раскрыть в этой книге, и понять, что необходимо вести преобразования только в готовой к этому культуре, дабы достичь устойчивого эффекта.

Последние несколько лет я разрабатывал практикум, который провожу для СІО и других ИТ-руководителей, чтобы показать их проблемы и помочь определить возможные пути решения. Самое замечательное в том, что ты встречаешься с умными и прогрессивными лидерами, – это возможность каждый раз узнавать нечто новое. Я уже проводил данный практикум в течение некоторого времени и невероятно благодарен за опыт и идеи, которые передали мне СІО: они помогают мне совершенствовать мой практикум. (Материал этой книги частично отражает и знания, полученные в ходе таких сессий.)

DevOps для современных компаний ссылается на различные испытания, с которыми сталкиваются организации во время их преобразования в современные ИТ-организации. А такая трансформация сейчас характерна для многих организаций: будь то производители автомобилей или банки, их бизнес зависит от ИТ. Мы видим, что технологии развиваются все быстрее и быстрее. Но в то же время у нас есть *устаревшие* приложения, оставшиеся с давних времен. А новые приложения, создаваемые сегодня, устареют через несколько лет. Я даже согласен с идеей о том, что устаревшим можно считать код, который был написан вчера! Программное обеспечение и технологии меняют ландшафт бизнеса, подсказывая людям новые способы взаимодействия и об-

мена информацией. Более того, технологии позволили нам преодолевать препятствия и расстояния. В итоге мир становится более сложным, он меняется быстрее, и из-за этого новые типы потребления разрушают установившиеся бизнес-практики. Многие организации сталкиваются с фундаментальной необходимостью модернизации ИТ-инфраструктуры. Наши старые *ментальные модели* и методы больше не работают. Нам нужно по-новому подходить к необходимости поиска новых решений. Но путь к современным решениям и улучшению технологии все еще кажется туманным и непростым. Очень немногие организации старого образца сегодня прошли через это преобразование.

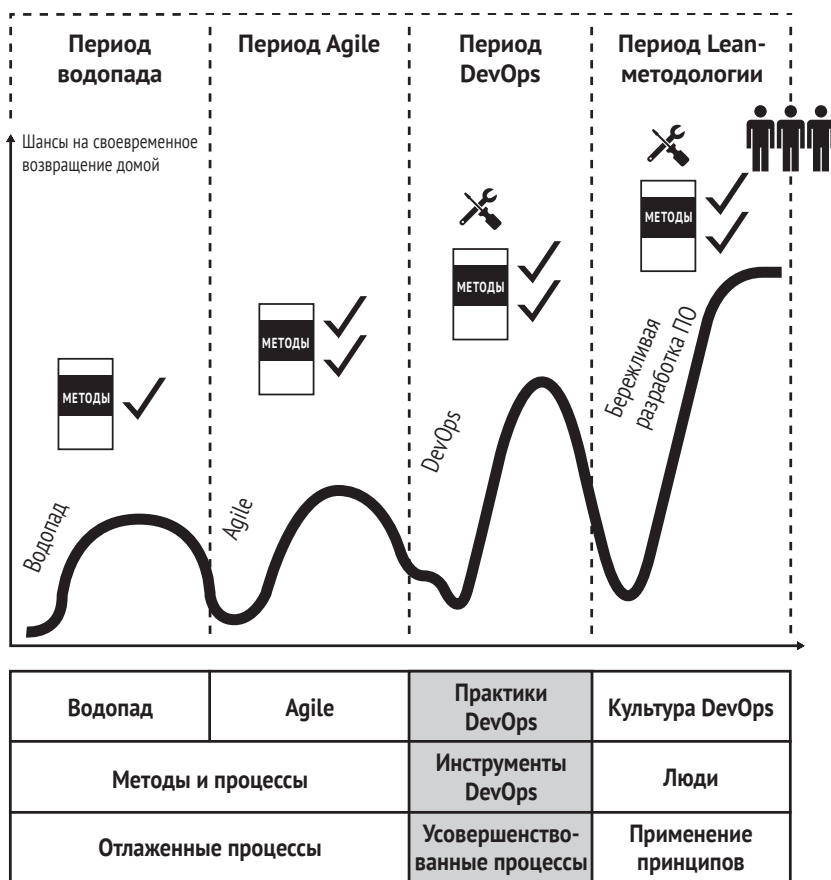


Рис. 1. Развитие представлений Мирко о переменах в организации

На протяжении своей карьеры я успел поработать с рядом крупнейших технологических организаций в каждой из вертикалей этой индустрии. И какие бы вы ни слышали оправдания, будь они связаны с технологией, сложностью или культурой, поверьте мне, я видал ситуации и похуже! Тем не менее эти

организации были способны к радикальному преобразованию и улучшению своих результатов. Хочу поделиться с вами некоторыми из открытий и достижений. ИТ не должна быть областью, в которой большая часть времени тратится на решение одних и тех же проблем.

Материал этой книги основан на принципах Agile, DevOps и Lean. На самом деле большинство разговоров с клиентами я начинаю с вопроса «Что эти принципы значат для вас?», ведь все эти методологии такие неоднозначные. Я часто применяю схему, показанную на рис. 2, чтобы ясно очертить для слушателей, как я пользуюсь этими принципами.

Выполняя роль консультанта, я в то же время поставил себе цель как можно больше дистанцироваться. Как только мои клиенты начнут успешно применять принципы Lean, DevOps и Agile, а также выходить на более эффективную работу, я смогу пойти дальше и посвятить свое время реализации существующих решений. Я не могу представить себе лучшей цели в жизни, чем предоставлять заказчику максимальную свободу и вовремя отступать, сосредоточиваясь на одном-двух новых проектах.

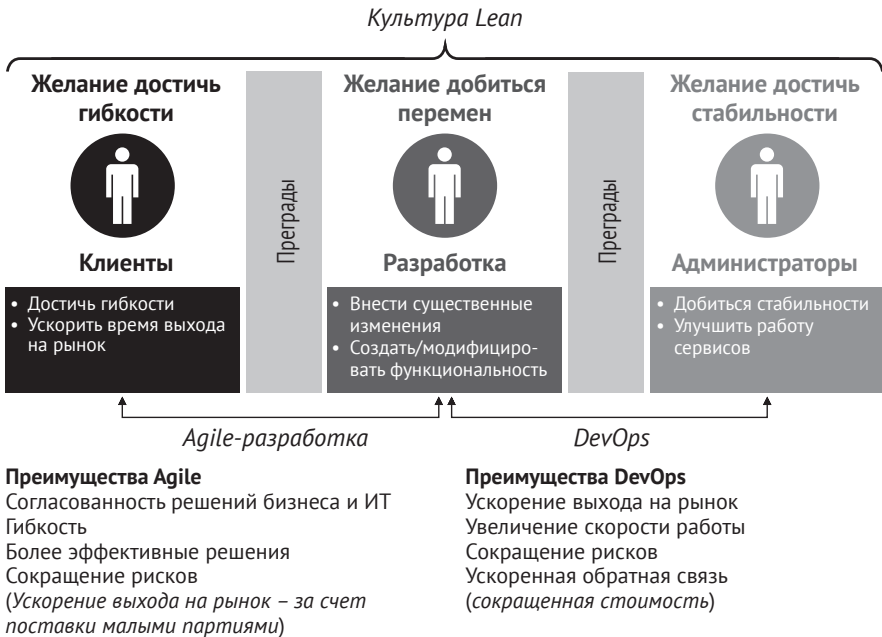


Рис. 2. Взаимосвязь Agile и DevOps:
как принципы Lean, Agile и DevOps связаны друг с другом

Но пока в сфере ИТ слишком многое все еще делается неправильно и людям приходится каждый день преодолевать проблемы, я буду принимать участие в совершенствовании мира ИТ. Я буду помогать организациям пережить преобразования при помощи этой книги, ведь моя команда не может быть вез-

десущей. У каждого из нас работы предостаточно. Честный ответ на вопрос о сложных проблемах состоит в том, что описанные в книге методы – только начало. Вы можете экспериментировать, добавлять что-то от себя и использовать мои советы частично. Каждое преобразование зависит от контекста, и не существует универсального рецепта успеха.

Надеюсь, что эта книга поможет вам с честью преодолеть неизбежные трудности. Я делюсь своим опытом и практическими упражнениями, и вы можете использовать их в вашей организации (будь она крупная или небольшая, старая или молодая), чтобы добиться улучшений. Я с нетерпением буду ждать встречи с вами на этом пути: на конференциях, где вы расскажете о своих успехах и провалах, на митапах или, возможно, на одной из моих консультационных встреч, где мы вместе будем искать решения ваших вопросов.

Я разработчик до глубины души и хочу создавать полезные приложения. Давайте сообща менять нашу индустрию, чтобы мы могли больше времени посвящать созидательной стороне ИТ.

Мирко Херинг

ВВЕДЕНИЕ



Как мы к этому пришли

Основной опасностью кризисных времен является не кризис, а то, что вы будете действовать согласно вчерашней логике.

Питер Ф. Дракер. *Managing in Turbulent Times*
(«Менеджмент во время кризиса»)

Прежде чем подобраться к самой сути, я хотел бы объяснить, почему важно менять подход к поставке ИТ-услуг. Мы явно не планировали ситуацию, которая сложилась на сегодняшний день, когда много организаций прикладывают массу усилий, чтобы поставлять продукты, приносящие пользу бизнесу. Услуги ИТ либо запаздывают, либо слишком дорого стоят, либо не способны обеспечить то качество, которого ожидают заинтересованные лица на стороне бизнеса, и это не потому, что мы сознательно принимаем ошибочные решения. Поразмыслив, вы поймете, что мы все примерно одинаково представляем себе, как выглядит хорошая работа, – примеры Netflix и Google встречаются в сотнях презентаций об ИТ. Почти каждая ИТ-организация говорит об Agile и гибкой поставке, автоматизации в жизненном цикле поставки программного обеспечения (англ. *Software Delivery Life Cycle*, сокр. SDLC) и применении современных паттернов архитектур, таких как *cloud-native приложение*, приложение двенадцати факторов и *микросервисы*.

Однако нам с трудом удастся отыскать организации, которые в совершенстве овладели этим новым методом работы в ИТ. Многие компании, которые могут похвастаться положительным опытом, относительно молоды и изначально развивались как облачная или интернет-платформа (компании, которые со времен основания делали ставку на интернет как основную платформу). Мы даже придумали для них название: DevOps-единороги. Потому что встретить их не проще, чем единорога, и их уровня слишком тяжело достичь средней организации с устаревшей ИТ-архитектурой.

Можно подумать, что основным испытанием для такой организации выступает трансформация старой архитектуры. В определенной степени это верно, но я склоняюсь к тому, что основная проблема – в менталитете. Многие лидеры в технологической отрасли используют идеи, заимствованные из области традиционного производства, несмотря на то что ИТ по своему характеру от него отличается¹, – они применяют ментальную модель производства для творческого, по своей сути, процесса. В традиционном производстве мы следуем предсказуемым процессам для получения одного и того же результата (продукта), раз за разом. В ИТ, наоборот, мы не занимаемся одним и тем же проектом дважды. Чтобы следовать созидательной природе ИТ и добиться изменений менталитета в компаниях, нам нужно обратиться к следующим трем областям: к организационной экосистеме, в которой мы находимся, к людям, с которыми работаем, и к технологиям, на которых держится наш бизнес. Этот трехчастный подход и определил структуру книги. Во введении я объясню, почему прежний стиль мышления более не актуален. Далее вы найдете рекомендации, которые помогут вам преобразовать свою организацию и свойственную ей линию поведения.

Структура книги такова:

1. Часть А «Создание подходящей экосистемы» (главы 1–4): как вы сформируете организацию, которая должна быть способна обслуживать ИТ? Как вам необходимо работать с другими организациями, такими как поставщики программных продуктов и компании – интеграторы решений, для того чтобы соответствовать новому способу ИТ-поставки? Первая часть книги дает ответы на эти вопросы и адресуется СІО и другим лидерам в ИТ, тем, кто задает стратегию организации.
2. Часть Б «Люди и организация процессов» (главы 5–8): в организации ничего не происходит без участия людей. Мы так мало задумываемся об этом, что называем сотрудников «ресурсом»! В части Б этой книги мы сосредоточимся на людях и на том, как воодушевить их на ежедневное совершенствование: ведь в этом, помимо прочего, залог успеха организации.
3. Часть В «Технологические и архитектурные аспекты» (главы 9–12): я уверен в том, что недавние достижения в области технологий существенно расширяют способы работы и принуждают лидеров индустрии переосмысливать методы поставки ИТ-решений. Я расскажу о некоторых ключевых тенденциях и о том, как наилучшим образом ими воспользоваться. Технологии развиваются с невероятной скоростью, поэтому я не стану упоминать ничего конкретного – инструменты, поставщиков или методы. Чтобы помочь вам сориентироваться в этом пространстве, я все же предоставлю список ресурсов, который, надеюсь, не утратит актуальности с течением времени.

¹ Некоторые книги о DevOps, такие как *The Phoenix Project*, описывают современные модели производства, которые отличаются от этой традиционной модели.

Каждая глава основного блока содержит обсуждение определенной проблемы и предлагает возможные решения и подходы. Я старался делать главы покороче, чтобы проще можно было усвоить материал.

В конце каждой главы приводятся упражнения в стиле «сделай сам»: я предлагаю вам некоторые практические шаги, позволяющие применить знания, почерпнутые из соответствующей главы. Вашему вниманию представлены шаблоны, некоторые вопросы, которые вы можете задать себе или вашей организации, или пошаговое руководство. Это не заменит сотрудничества с опытным консультантом, но поможет совершить первые шаги на пути глобального преобразования.

Я также привожу ссылки на некоторые из моих статей там, где широко раскрываю некую тему, но желающим изучить ее глубже могли бы понадобиться дополнительные подробности. Можете полюбопытствовать и прочитать эти статьи.

Также в книге вы встретите фрагменты, резюмирующие вышесказанное и предлагающие некоторые общие рекомендации в отношении ваших следующих действий. Я уже говорил о том, что был вынужден изыскивать собственные методы, позволяющие развивать скорость работы в ИТ-индустрии. По сути, информация, приведенная в книге, должна подготовить вас к началу преобразования вашей компании в современную ИТ-организацию.

И наконец, в книгу включено приложение, где описываются подробные аналогии с заводами (фабриками), о которых говорится на протяжении всей книги. Если это совершенно новая для вас тема, вам, вероятно, стоит сначала прочесть приложение, перед тем как подойти к основному содержанию книги. Тем, кто хорошо знаком с такими аналогиями, приложение поможет понять их глубже и, возможно, сделать полезные для себя выводы. А теперь давайте немного поговорим о том, почему схемы, применяемые в традиционном производстве, помогают достичь быстроты и надежности ИТ-поставки, к чему мы и стремимся, используя DevOps.

Принципы традиционного производства не так просто применить в ИТ-поставке

В последние годы методы современного традиционного производства вдохновляли новаторов на появление таких течений, как Lean, системное мышление и теория ограничений. Производственная система компании Toyota на самом деле послужила одним из источников идей, которые переросли в Agile и DevOps. Тем не менее многие менеджеры занимаются управлением, все еще черпая вдохновение в старых подходах, не упуская из виду образ модели производства, сформировавшейся еще во времена Генри Форда. Она основана на идее о том, что мы выстраиваем процесс производства, чтобы максимально рационализировать процесс управления, в результате чего можно нанять менее квалифицированных работников, но добиться наиболее качественных

результатов. Однако в сфере ИТ, напротив, для достижения наилучших результатов необходимы сотрудничество и креативность. Мне кажется, что старые принципы производства привели ко множеству проблем, с которыми мы сегодня сталкиваемся в ИТ: например, строгие процессы с административным стилем управления, раздутые и сгруппированные по функциональным навыкам организационные структуры, а также частое делегирование обязанностей угнетают рабочий поток и повышают риск технологических ошибок и сбоев.

Я назвал свой блог «больше никаких заводов» (notafactoryanymore.com), чтобы таким образом призвать к переменам в мышлении, которые необходимы исполнительным ИТ-директорам для успешного преобразования организаций. Слово «завод» (или «фабрика», *factory*) отсылает к тому типу производства, который когда-то предложил Генри Форд: массовое производство осуществляется специалистами, работающими на сборочной линии и выполняющими узкоспециальные задачи, отчего при производстве продукта очень редко происходят какие-либо изменения, если вообще происходят.

Устаревшие методы производства были основаны на изобретении Фордом сборочных линий и на работе, написанной Фредериком Тейлором, о принципах научного менеджмента¹. Запуск завода был дорогостоящим и длительным процессом. Сама работа на заводах оптимизировалась так, чтобы работники направляли все свои усилия на выполнение одной узкой задачи и могли научиться осуществлять ее невероятно эффективно. Для сокращения затрат производитель мог бы инвестировать в автоматизацию производства, приобретая новейшее оборудование; он мог бы сменить материал для изготовления продукта или переместить свой завод туда, где рабочему можно было бы платить меньше. Любые изменения в отношении самого продукта или процесса производства превращались в настоящее испытание, так как для этого приходилось приобретать новые машины, перенастраивать их, обучать всех работников новым процессам или менять схемы поставки материала для продукта.

Сам продукт практически не менялся (даже Генри Форд говорил о модели Т: «Вы можете получить “Форд-Т” любого цвета, при условии что этот цвет будет черным»), поэтому было достаточно просто сравнить результаты работы и стоимость продукта. Если вы измените процесс производства или начнете использовать другой материал для изготовления продукта, то можно с научной точки зрения оценить результаты (как раз эту концепцию предлагает работа Тейлора). Можно поставить под сомнение, что современное производство все еще позволяет вам измерять стоимость и продуктивность, так как все продукты на сегодняшний день производятся массово. Это тот тип производства, на котором основаны все экономические программы обучения и, соответственно, менеджмент. Я поступил в университет около пятнадцати лет назад, а недавно получил степень MBA, поэтому знаю не понаслышке, что обучение

¹ Фредерик Тейлор считается основателем научного менеджмента, который он реализовывал в стальной промышленности. Он опубликовал свою работу «Принципы научного менеджмента» в 1911 году.

ИТ-процессам все еще следует этой модели. Так или иначе, такую модель легко понимать и контролировать, что позволяет использовать научный подход к процессам.

В самом начале ИТ-отрасль имела много общего с производственным бизнесом. Давайте вернемся в 1990-е и зайдем в ИТ-отдел автомобильной компании. (Я предлагаю подобный сценарий, потому что в школьные годы работал в таком отделе по выходным и именно благодаря этому решил строить карьеру в ИТ.) Рабочие процессы в ИТ в целом организовывались аналогично другим производственным процессам и следовали тем же тенденциям. Создание новой ИТ-системы обходилось дорого и занимало много времени. Сама работа оптимизировалась таким образом, чтобы сотрудники могли стать хорошими специалистами по тестированию, Java-разработчиками, SAP-конфигураторами или инженерами по эксплуатации. Каждый из них мог сосредоточиться на одной специфичной задаче и выполнять ее невероятно эффективно в рамках процесса разработки по методологии водопада.

Для сокращения расходов ИТ-отдел мог предпринимать следующие шаги: можно было инвестировать в автоматизацию, приобретая улучшенные инструменты, сменить используемый тип программного обеспечения (например, перейти с частной системы на SAP, чтобы пользоваться уже готовой функциональностью, вместо того чтобы создавать ее самостоятельно), а позднее можно было бы перенести разработку туда, где сотрудникам можно меньше платить. В большинстве случаев внесение изменений в продукт сулило большие испытания, так как это подразумевало изменение условий контракта с поставщиками ПО или значительное преобразование существующей команды.

Сам продукт представлял собой нечто стандартное: *систему планирования ресурсов предприятия (ERP)* или *систему управления взаимоотношениями с клиентами (CRM)*, использующую наилучшие практики, основанные на готовых решениях. И хотя выпускаемые продукты не были идентичными, общие усилия для их создания были сопоставимы (например, реализация SAP в одной автомобильной компании была сравнима с реализацией SAP в какой-либо другой компании).

Как вы видите, несмотря на то что ИТ – особая сфера, она все равно пересекается в управленческой части с другими отраслями, поскольку в ней используются практики, взятые из традиционного производства. И они работают уже довольно продолжительное время. Но за это время сфера ИТ уже сильно изменилась, а мы все еще не пытаемся управлять ей как-нибудь по-новому.

Применение в ИТ-области «заводских» методов порождает следующее представление о рабочем процессе: сотрудники ИТ-отдела сидят за перегородками и колдуют каждый над своей задачей; один человек преобразует требования в проект, передает его другому человеку, который пишет некоторый код, а затем передает его еще одному человеку, который его протестирует, – и при всем этом никто друг с другом даже не разговаривает. Все делается механически, как на конвейере. (Здесь вспоминается Чарли Чаплин в фильме «Новые времена».) Мир меняется, и на сцену выходит человек, готовый нести перемены:

сокращать усилия и расходы, затрачиваемые в ИТ на операции и подготовительные мероприятия.

Понятия операционных издержек и объема работы выступают основой для смены подхода

Операционные издержки (их еще называют издержками на подготовку) – это расходы, необходимые для преобразования механизма производства, чтобы начать выпускать другой продукт или сменить способ изготовления продукта. Размер операционных издержек определяет оптимальный объем работ по продукту, что позволяет добиться определенных экономических результатов, в то время как на издержки на хранение воздействовать довольно сложно. Возможное влияние операционных издержек на оптимальный объем работ показано на рис. 3¹.

На определение оптимального объема работ влияют издержки на хранение и операционные расходы (чем больше издержки на хранение, тем меньше будет оптимальный объем работ, а чем больше операционные расходы, тем оптимальный объем работ будет больше). В ИТ издержки на хранение являют собой комбинацию возрастающей стоимости исправления дефектов по мере продвижения по жизненному циклу разработки и упущенной выгоды от того, что готовая функциональность простаивает, будучи все еще не выпущенной в среду эксплуатации. Эти два фактора не сильно изменились с приходом практик DevOps; чего коснулись изменения, так это операционных издержек.

По мере того как традиционное производство менялось с течением времени, начала усиливаться тенденция к сокращению операционных издержек, для того чтобы появилась возможность производить более разнообразные продукты. За этим последовала экспансия массового изготовления по индивидуальному заказу и 3D-печати. Методологии в сфере ИТ Agile и DevOps были направлены на достижение такого же эффекта.

Ранее операционные издержки были большими в ИТ-проектах. Была необходимость в приобретении оборудования и установке всякого рода межплатформенного ПО, а разработка оригинального продукта включала в себя множество задач, выполняемых вручную, таких как, например, развертывание изменений в коде, регрессионное тестирование данного решения, внесение изменений в окружение (так, для создания MVP тогда было необходимо потратить 6–12 месяцев вместо трех, а релиз новой функциональности занимал месяц вместо нескольких дней).

¹ Штефан Томке и Дональд Райнертсен, вероятно, лучше всех описали это в статье от мая 2012 года для Harvard Business Review под названием *Six Myths of Product Development* («Шесть мифов о разработке продуктов»). И хотя они говорили непосредственно о разработке продуктов, а не об ИТ, так или иначе ИТ-услуги подразумевают поставку ИТ-продукта [1].

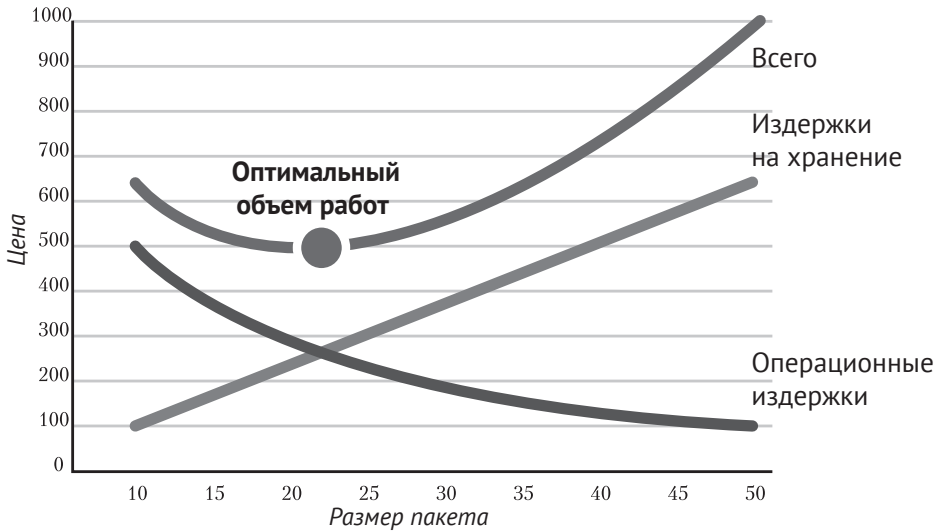
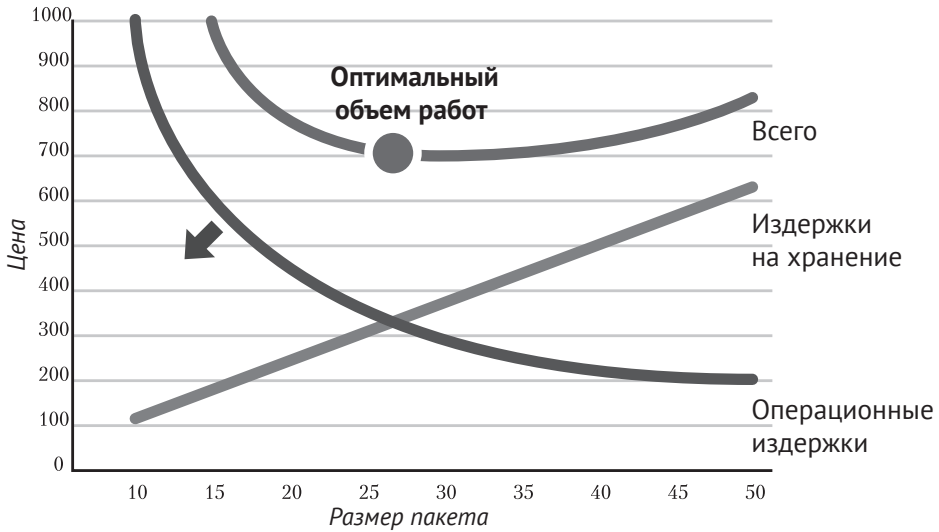


Рис. 3. Соотношение между операционными издержками и оптимальным объемом работ: уменьшение операционных издержек позволяет сократить объем работ

Сегодня это не так, и операционные издержки значительно снизились с началом применения облачных инфраструктур и Хаас и автоматизации процессов в жизненном цикле разработки, что позволило максимально сократить объем работ.

В этом малом оптимальном объеме работ таится настоящая выгода для бизнеса. Малый объем работ позволяет закладывать малые расходы на пробу бизнес-идеи и реже делать крупные ставки. Одну прекрасную аналогию этому по-

ложению высказал долгожитель в разработке продуктов, Дональд Райнертсен [2]: «Представьте себе лотерею, в которой вы получаете приз, если угадаете три цифры. Первый вариант ваших действий – дать мне пять долларов и попытаться угадать число из трех цифр, а затем я скажу вам, выиграли ли вы приз в \$1000. Второй вариант – дать мне два доллара и попытаться угадать первую цифру. Я в свою очередь могу сказать вам, что эта цифра верная, а вы сможете решить, давать ли мне еще два доллара ради того, чтобы попытаться угадать следующую цифру. Какой из вариантов выбрали бы вы? На мой взгляд, второй вариант очевидно более привлекателен. Обратная связь при малом объеме работ позволяет получать информацию, на которой можно основывать решения о том, стоит ли идея дальнейших вложений. Это именно то, что обратная связь от заказчика позволит вам, если вы будете вносить небольшие партии новой функциональности и иметь дело с малым объемом работ в ваших ИТ-проектах».

Малый объем работ привлекателен еще и по другим причинам. Он упрощает предоставление ИТ-услуг, так как изменения производятся пошагово и в малом объеме, так что приходится меньше беспокоиться при тестировании и выпуске в среду эксплуатации. В производстве можно легко увеличивать объемы работ, так как вы можете выполнять один и тот же процесс в увеличенных объемах, не рискуя сильно усложнить его, как это может произойти при масштабировании вашего ИТ-решения. Меньший объем работ в ИТ позволяет нам проверить, в том ли направлении движется продукт, и пользоваться обратной связью от заказчика, чтобы направлять созидательную природу ИТ-разработки¹. В производстве сложнее организовать циклы с проектированием, созданием и проверкой продукта: только представьте себе создание одного продукта на заводе (например, машины), его продажу и затем получение обратной связи, для того чтобы создать новый, и т. д. Задержки и расходы после запуска такого цикла вскоре начнут зашкаливать. Метод Lean-стартапа [3] Эрика Рая был адаптирован для продажи реальных товаров в технике Fast-Works, что позволило добиться замечательных результатов (уменьшение стоимости, увеличение скорости) [4] – по крайней мере, пока не стало нормой печатать все на 3D-принтерах и сокращать таким образом операционные и подготовительные расходы [5].

С малым объемом работ приходит совершенно другая экономическая модель для управления процессами с целью сокращения необходимых дополнительных издержек, требуемых от менеджеров. Другими словами, даже если технические операционные издержки были сокращены, это не значит, что автоматически снизятся и операционные расходы, связанные с менеджментом и обслуживанием архитектуры. А причина в том, что мы до сих пор следуем

¹ Так как в сфере ИТ свойственно совершать пробы и ошибки в процессе разработки продукта, меньший объем работ поможет сократить риски и ускорить получение обратной связи. Этим ИТ отличается от производства, в котором эффективность и продуктивность можно измерить.

принципам, перенятым из традиционного производства. Данная книга поможет вам отказаться от такого подхода.

В приложении я более детально рассмотрел некоторые идеи и принципы, которые перешли в сферу ИТ из производственной практики, и рассказываю о том, насколько они еще применимы сегодня.

Я надеюсь, что эта книга поможет вам создать предоставляющую ИТ-услуги организацию, которая не будет похожа на завод из фильма «Новые времена» с Чарли Чаплином: в ней люди будут работать вместе с заинтересованными сторонами, чтобы получить значимые для заказчиков решения. Как сказал мой коллега из Accenture Марк Рендел, «DevOps не стремится делать ИТ-услуги более эффективными. Речь идет об увеличении эффективности бизнеса с помощью ИТ» [6].

А теперь, осознав это, давайте приступим к преобразованиям.

ЧАСТЬ А

Создание подходящей экосистемы

Оказывая услуги консультанта, я часто говорю технологическим лидерам: представления о том, что такое хорошо, даже вкупе с благими намерениями, к сожалению, не всегда приносят наилучшие результаты. Экосистема, которую создает руководство компании, играет огромную роль в том, насколько успешными могут быть изменения. Скорее всего, экосистема, в которой вы работаете, была создана на фоне устаревших явлений – тех, которые были актуальны ранее, а также созданных ранее систем.

Понятие «устаревшие» применимо не только к вашим технологиям и приложениям. Как уже отмечалось во введении, существует консервативный тип мышления, вдохновленный командно-административной культурой, который желательно изменить. Итак, каким образом подготовить план, по которому будут проводиться изменения? Как выбрать правильные технологии? И кого же выбрать в надежные попутчики, чтобы уверенно двигаться от устаревших принципов к чему-то новому?

Владея инструментарием, предложенным в части А этой книги, вы сможете значительно приблизиться к созданию экосистемы, которая позволит ИТ процветать и преобразоваться. В этой части представлены высокоуровневые размышления о создании экосистемы, которая будет способствовать преобразованиям в современной ИТ-организации. Мы поговорим о пути преобразования, об ускорении работы в ИТ, о портфолио вашей компании, работе с устаревшими явлениями, выборе программного обеспечения и поиске подходящих партнеров, которые будут поставлять вам свои услуги. Также мы обсудим, как моя команда работает с клиентами – руководителями организации над изменением существующей экосистемы, стремясь к тому, чтобы это принесло пользу обеим нашим организациям; я покажу некоторые инструменты для анализа и действия, которые вы сможете создать самостоятельно. Иными словами, вы поймете, как работать с имеющимися в вашем распоряжении ресурсами так, чтобы они были для вас не помехой, а помощью.

ГЛАВА 1



Путь к изменениям

- Скажите, пожалуйста, куда мне отсюда идти?
- А куда ты хочешь попасть? – ответил Кот.
- Мне все равно... – сказала Алиса.
- Тогда все равно, куда и идти, – заметил Кот.

*Льюис Кэрролл, «Алиса в Стране чудес»
(пер. Н. Демуровой)*

Многие трудности в предоставлении ИТ-услуг связаны с узкими местами – такими «бутылочными горлышками», которые не позволяют вам быстро получить значимый отзыв от клиентов, который вы смогли бы использовать для улучшения ваших рабочих процессов и продуктов. Сделать работу видимой – это один из наиболее действенных способов для обнаружения узких мест, однако ИТ чаще всего сталкивается с неявной работой: в ИТ нет акций, стоимость которых показывала бы успехи команды или компании в процессе создания продукта, нет никакого склада, по наполнению которого видно, каков объем невостребованной продукции, и нет никакого явного процесса, за которым мы могли бы наблюдать, чтобы увидеть, как то, что поступает на вход, превращается в конечный продукт на выходе. Это ведет к интересной ситуации: в то время как большинство людей, работающих на традиционном производстве, хотя бы приблизительно понимают, как их продукция создается, в ИТ процессы не так очевидны. Я имею в виду реальные процессы, а не те, которые могут быть описаны на сайте компании или же в какой-нибудь методологии. Однако без четкого представления о работе действительно трудно улучшить рабочие процессы. Таким образом, принципиально важная задача руководства любой ИТ-организации – сделать процессы видимыми, в том числе четко обозначить текущий статус работы и такие показатели, как качество и скорость. В этой главе мы воспользуемся картами потоков ценностей, чтобы достичь этого результата, и положим начало пре-

образованиям при помощи начальной карты потоков ценностей и управленческого подхода.

Явные ИТ-процессы

Я предпочитаю начинать свою работу в качестве консультанта по DevOps с упражнения по составлению карты потока ценности. Причина достаточно проста: это упражнение наиболее эффективно поясняет, как все же выглядят процессы в ИТ. Вы, конечно, можете изучить описание методологий или взглянуть на какие-нибудь большие диаграммы в Visio, чтобы узнать что-то про предоставление ИТ-услуг, но зачастую реальность не вполне совпадает с процессами, описанными в документации.

В конце каждой главы я коротко описываю процессы, которые вы можете попробовать воспроизвести, – смело пользуйтесь этим. В двух словах: вы организуете собрание с участием представителей каждого подразделения вашей организации, чтобы составить такой план предоставления ИТ-услуг, который всех устроит, и, что даже более важно, найти те элементы системы, которые можно улучшить. Предполагаю, что на такую встречу надо пригласить помощника или кого-либо, кто может судить о деле беспристрастно.

В идеале мы хотим иметь возможность объективно измерять такие параметры ИТ-процессов, как производительность, длительность циклов и качество. К сожалению, зачастую это требует немалых усилий. Составление карты потока ценностей раз в три–шесть месяцев (в зависимости от того, насколько быстро происходят изменения и улучшения) позволит вам отслеживать процессы, затрачивая на анализ их состояния всего несколько часов в месяц. Это даст вам возможность сосредоточиться на текущем процессе, на продолжительности циклов и качестве. Позаботьтесь о том, чтобы результаты составления карты были очевидными для всех сотрудников; тогда люди обратят внимание на улучшение процессов. Подобная практика послужит явным напоминанием о том, что улучшение процесса – важная задача для организации.

Когда вы уже сформируете в компании хорошее понимание высокоуровневых ИТ-процессов и способность видеть области, нуждающиеся в улучшении, настанет время разрабатывать первую схему трансформации.

Создание первой схемы трансформации

Создание схем – это отчасти наука, отчасти искусство. Многие схемы на первый взгляд похожи, но при детализации можно заметить, что у двух разных людей никогда не получаются одинаковые схемы. Тем не менее стоит отметить, что не существует идеальных схем. Поступая строго согласно принципам Agile, очень важно понимать общее направление и определять для себя некоторые опорные точки, чтобы осуществлять оценку процессов и делать их видимыми. Многие вещи со временем изменятся, и вам придется это учиты-

вать по ходу дела. Приведу рекомендации о том, как создавать хорошие схемы трансформации вашего предприятия.

Основываясь на карте потоков ценностей ваших ИТ-услуг, вы сможете определять узкие места в процессах. Как нас учат *системное мышление, теория ограничений и теория очередей*, пока мы не справимся с «бутылочными горлышками» в наших процессах, ни одно из нововведений не приведет к их ускорению. Это очень важный момент, поскольку иногда мы тратим время на мелочи, упуская то, что может вызвать глобальный сдвиг. Один из прекрасных способов выявления «бутылочных горлышек» – проведение практикума с картами потока ценностей. Позвольте заинтересованным сторонам заострить внимание на значимых проблемах, к которым нужно пристально присмотреться, чтобы изменить тактику предоставления ИТ-услуг в целом. Коллективные размышления в большинстве случаев помогают выявить ряд узких мест.

Чтобы создать успешно работающую схему трансформации, необходимо еще уделить время установлению баланса между процессами и скоростью предоставления ИТ-услуг и их стоимостью и качеством. Уделять большое внимание процессам – это яркий пример тактики, основанной на системном мышлении, которая может поспособствовать устранению барьеров в вашей организации. Ранее «ответственный» за выполнение функции, например за работу на определенном участке – допустим, с центром тестирования (Testing Center of Excellence) или фабрикой разработки (Development factory), – вел инициативы по совершенствованию, чтобы воздействие на эту область и контроль над ней были более эффективными. Со временем это поспособствовало возникновению функций для обслуживания большего объема работ, хорошо оптимизированных, но в ущерб предоставлению ИТ-услуг в целом. Процессы улучшаются только при малых объемах работ.

Обычно для оценки ИТ-услуг пользуются тремя показателями: скоростью, стоимостью и качеством работы. Ранее было принято прилагать усилия к улучшению показателей стоимости или качества, что, в свою очередь, сокращало скорость поставки услуг. Если вы будете оценивать свои ИТ-услуги только с опорой на улучшение качества, то зачастую будете наблюдать расширение границ показателей качества, что увеличит стоимость работ и время, затрачиваемое на их анализ. Если же будете оценивать выполняемую задачу, основываясь на сокращении показателей стоимости, то наиболее распространенный подход – доверить больше работы менее опытным сотрудникам или не выполнять отдельные задачи в рамках процесса, что часто приводит к ухудшению качества и замедлению работы из-за необходимости ее переделывать. Принимать во внимание лишь стоимость или качество, не учитывая влияния процессов, по моему опыту, неразумно: это не принесет успеха ИТ.

И наоборот, если нам нужна скорость (особенно это касается «бутылочных горлышек», которые мешают быстро осуществлять поставку действительно малых партий продукта), то, обращая внимание на процессы, мы сможем ускорить поставку продукта – даже больших его партий, – что приведет к улучшению показателей качества и стоимости работ. Невозможно добиться скорой

поставки, если с качеством дела обстоят неважно, так как необходимость перерабатывать некачественно выполненную работу замедлит процесс. Единственное, что вы можете сделать для ускорения поставки, – автоматизировать процессы и избавиться от ряда неактуальных шагов. Если просто научиться быстро печатать на клавиатуре, это не ускорит выполнение процессов в целом. Между тем ИТ вынуждена стремиться к увеличению скорости поставки услуг. Я участвовал в процессах трансформации вместе с клиентами, которые поработали над сокращением показателя стоимости ИТ-услуг, но в целом впечатление заинтересованных лиц о процессах поставки оставалось неудовлетворительным. Я также был свидетелем множества инициатив по улучшению качества, которые сдерживали процесс поставки и даже почти полностью останавливали его. И я еще буду встречаться с инициативами по совершенствованию, связанными с ускорением процессов.

И еще несколько слов в предостережение тем, кто работает над ускорением процессов. Первое, о чем хочется сказать, – не такая уж большая проблема. Вы на самом деле можете обойти проблему проведения оценки быстроты поставки, уменьшая объем поставляемых партий продукта, которые, в свою очередь, можно будет поставлять быстрее. И хотя это не значит, что вы сможете добиться взаимозаменяемости партий разных размеров, данный подход все равно окажется выигрышным для организации, поскольку менее объемные партии так или иначе несут в себе меньше риска. Второе предостережение: люди могут предложить способы сократить путь, которые окажутся рискованными или ухудшат качество продукта. Чтобы этого избежать, вам нужно продолжать поиски способов оценки качества, помимо скорости, чтобы убедиться в том, что качество продукта не ухудшается при ускорении выполнения процессов. Для оценивания скорости процессов вам будет необходимо обратить внимание на работу, осуществляемую на протяжении всего цикла предоставления услуг, а также на изучение показателей, которые проинформируют вас о статусе этих работ. Полезными показателями для оценки быстроты процессов являются длительность цикла выполнения процессов (длительность цикла = период времени с утверждения рабочего процесса до его выполнения и выпуска результатов в среду эксплуатации) и объем работ, поставляемых за определенный период времени.

На рис. 1.1 показано, что в вашей схеме преобразования наверняка будут ключевые точки, связанные с основными функциями и возможностями (например, с автоматизированным регрессионным тестированием, упрощенным представлением бизнес-кейса), что логично. Тем не менее стоит обратить внимание и на нечто другое, а именно на схему покрытия областей применения и технологий. В следующей главе я расскажу, как создать портфолио областей применения, позволяющее вам определить ряд возможностей, которые вы будете стремиться развивать в процессе трансформации. Ваша схема должна содержать в себе приоритизированные списки (часто называемые волнами) областей применения, потому как ни одна крупная организация не сможет

развивать абсолютно все возможные направления. И не должна. Ведь некоторые области применения не стоят усилий и затрат.

И еще пару слов о схеме трансформации: для того чтобы добиться больших возможностей и перемен, потребуется много времени. К сожалению, организации не очень справляются с тем, чтобы терпеливо ждать завершения таких программ, поэтому вам нужно убедиться в том, что вы можете предоставить какой-то видимый результат на раннем этапе. Для этих ранних результатов можно не придерживаться всех правил. В них можно реализовывать области применения, которые не критичны для бизнеса, или те, которые не являются частью «бутылочных горлышек». Эти ранние результаты необходимы для того, чтобы сохранить поддержку и позволить организации увидеть первые успехи. Вам стоит рассматривать эту задачу как часть действий, необходимых для трансформации. Конечно же, в идеальной ситуации ранние результаты будут затрагивать одну из predetermined приоритетных областей.

Преобразование вашей организации займет время. На рис. 1.2 вы можете наблюдать шаблон схемы трансформации, который я часто использую в работе со своими клиентами. Паттерн, изображенный там, я довольно часто встречал. Он начинается с применения Agile, а затем приходит понимание того, что Agile без использования практик DevOps (таких как автоматизация тестирования, автоматизация развертывания и т. п.) не предоставит вам возможность ускорить процессы настолько, насколько вам это нужно. По мере внедрения DevOps ваша работа начнет ускоряться, вы начнете видеть ограничения организации и факторы, сдерживающие скорость работы, которые свойственны данной операционной модели. Для того чтобы преодолеть инерцию, потребуются значительные усилия организации и поддержка руководства. Не расстраивайтесь, если изменения не будут видны на следующее же утро.

Управление трансформацией

Как упоминалось ранее, схема – это очень важно, но без должного управления трансформацией вы не сможете добиться нужного результата. Довольно часто трансформация прекращается. Нет никакой возможности предвидеть все испытания, которые будут мешать продвижению, и без должного управления, которое поддерживало бы баланс между дисциплиной и гибкостью, процесс преобразований затормозится. Управление трансформацией позволяет сделать результаты видимыми, а также дает возможность влиять на них. Это отличается от обычного управления процессом предоставления ИТ-услуг, осуществляемого для ваших инициатив (например, в случае изменения плана технического наблюдения). На встрече совместно с рядом агентов и консультантов по трансформации на саммите DevOps Enterprise в 2015 году мы пытались определить, что необходимо для успешного внедрения DevOps. Все мы пользовались разными подходами и работали в разных организациях, но сходились в том, что отличает успешную организацию: способность постоянно совершенствоваться и успешно управлять этим процессом.

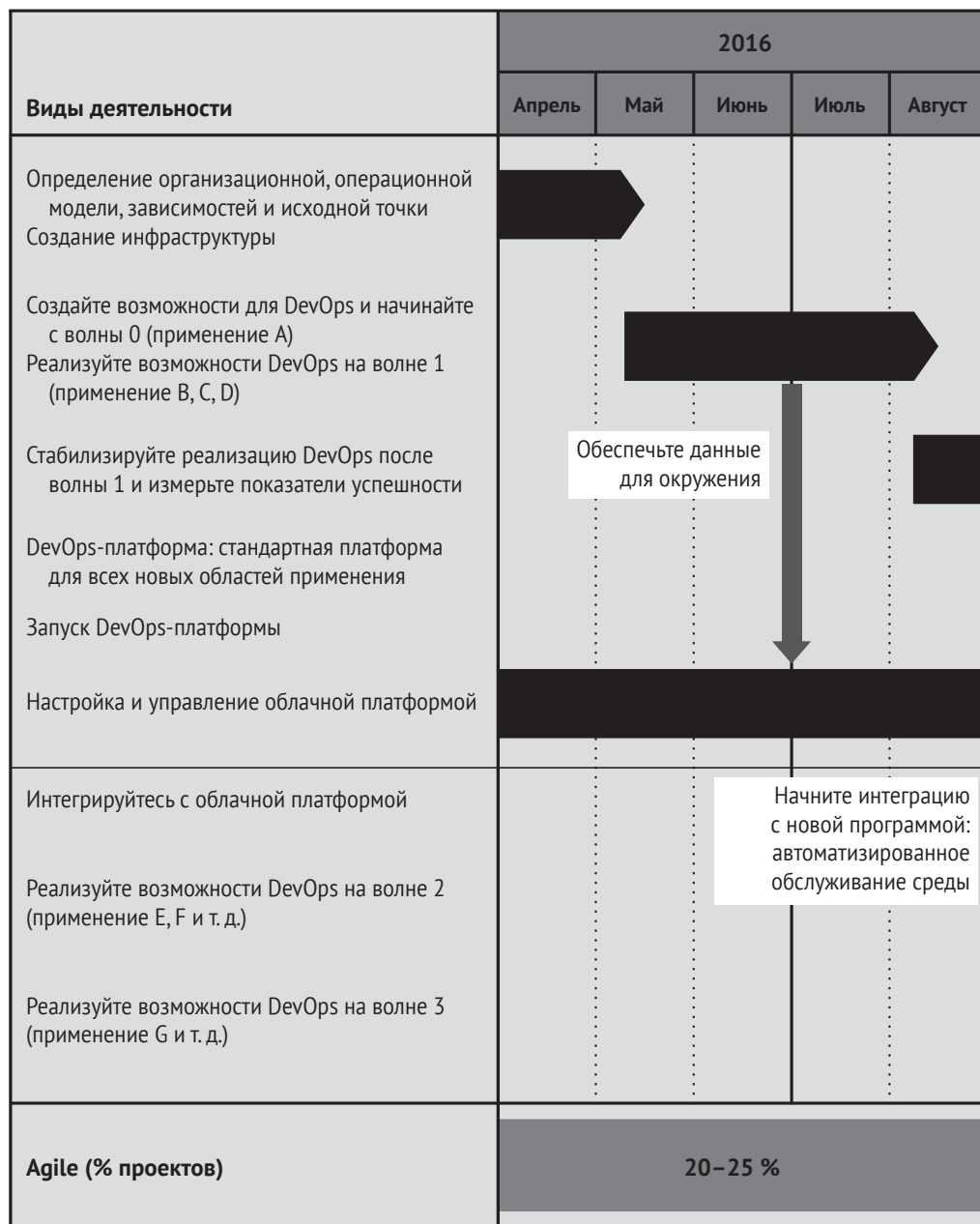


Рис. 1.1. Схема трансформации: пример, показывающий волны приложений и возможностей

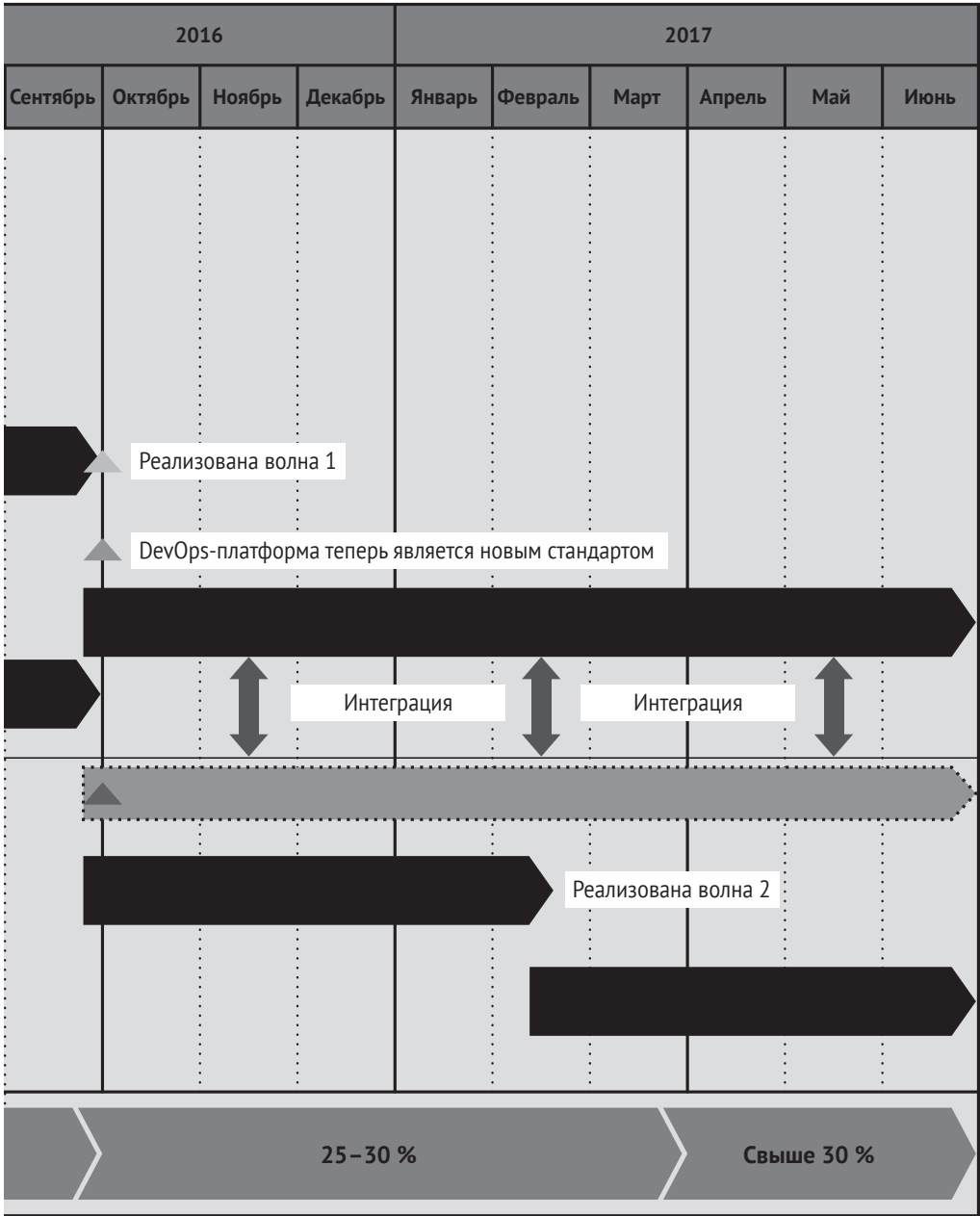


Рис. 1.1 (окончание)



Рис. 1.2. Распространенный шаблон схемы трансформации: изменение вашей организации займет время, пока вы будете пробовать различные методы



Рис. 1.2 (окончание)

Это непрерывное совершенствование и следование схеме трансформации вносит большой вклад в успешное преобразование вашей ИТ-организации. DevOps и Agile – не самоцель, поэтому не стоит стремиться к какому-либо конечному результату.

Как должно выглядеть успешное управление трансформацией? Управление трансформацией охватывает множество областей, поэтому важно, чтобы вы знали, с чем вам нужно сравнивать ваши успехи по мере реализации схемы. Это значит, что вам нужно определить точку отсчета перед началом преобразований, для того чтобы в дальнейшем оценивать показатели успешности. Слишком часто я встречался с инициативами по трансформации, когда шесть месяцев уходило на то, чтобы улучшить ситуацию, но затем нельзя было представить никаких доказательств тому – разве что анекдотичное «зато теперь у нас есть непрерывная интеграция с *Jenkins*». К сожалению, это не всегда убеждает руководство компании или другие заинтересованные стороны продолжать инвестировать в трансформацию. Даже если исполнительный директор будет за, компания не профинансирует дальнейшую работу по внедрению изменений: ведь никаких промежуточных результатов не достигнуто.

Тем не менее если вы сможете доказать, что с внедрением непрерывной интеграции вы смогли на 30 % снизить количество сбоев в работе запущенных экземпляров в окружении для сборки продукта, то у вас будут аргументы. Поэтому я настоятельно рекомендую выполнять упражнение по определению точки отправления в самом начале процесса трансформации. Подумайте обо всем, что вам важно, а также обо всем, что необходимо изменить; найдите подходящий способ определить текущее состояние. (Я привел несколько примеров в табл. 1.1.) Об использовании метрик я расскажу, когда мы немногим позже в этой главе будем говорить об управлении процессом предоставления ИТ-услуг.

Другой важный аспект управления трансформацией – поддержание гибкости и отслеживаемости работы. Для каждой инициативы по совершенствованию, согласно схеме, вам нужно применять научный подход:

- сформулируйте гипотезу, основанную на показателях успеха;
- составьте набор показателей;
- как только реализация будет готова, оцените результаты, используя гипотезу.

Что-то работает, что-то – нет; в процессе управления трансформацией вам понадобится тот и другой опыт. Не вините проектную команду за неверную гипотезу (все-таки на их месте должны быть мы, руководители, которые заварили кашу, – кого здесь винить?). Тревогу стоит бить только в случае, когда сотрудники не следуют процессу (например, показатели не были предоставлены или результаты были искажены), что, в свою очередь, мешает вам двигаться дальше.

Таблица 1.1. Показатели базовой метрики, которые доказали свою эффективность в управлении трансформацией

Показатель	Определение	Измерение
Длительность цикла релиза	Среднее время, затрачиваемое на объем работ от утверждения рабочего пакета (пользовательской истории, функциональности, набора требований) и до его релиза	Обычно показатель измеряется как временная разница между моментами приобретения рабочим пакетом какого-либо состояния в вашей трекинговой системе
Стоимость релиза	Объем работы, необходимый для выпуска новой функциональности, измеряемый как усилия, затрачиваемые на всю деятельность для релиза (или же может учитываться только работа, совершаемая в нерабочее время)	Обычно данные основаны на табелях рабочего времени
Длительность регрессии	Время, требуемое для подтверждения того, что при внесении изменений не появились дефекты	Период с момента развертывания до «зеленого света» после автоматизированной или ручной валидации
Доступность в среде эксплуатации	Доля времени, в течение которого среда эксплуатации доступна для должного предоставления услуг	Измеряется как доля времени, в течение которого среда эксплуатации доступна для использования, или как доля успешно проведенных операций
Среднее время восстановления	Время, которое требуется для исправления дефектов в среде эксплуатации	Измеряется как время с момента возникновения дефекта до момента полного его исправления
Долговечность команд	Средняя продолжительность времени, в течение которого команды не распадаются	Время (в месяцах) до того момента, когда команды начинают распадаться и перенаправляться на другие проекты

Пока вы развиваетесь, развивается и ряд жизнеспособных инициатив по совершенствованию. Критерии оценки для инициатив, которым вы хотите дать ход, должны быть основаны на:

- более раннем опыте;
- размере инициативы в рамках системы WSJF (Weighted Shortest Job First);
- уровне лояльности: насколько команда готова постоять за инициативу.

Не смущайте себя тем, как обстоят дела у крупного бизнеса, которому на начальном этапе необходимы большие вливания. Вам стоит поначалу ставить небольшие задачи, чтобы проверить, насколько реализуема идея. Понадобится приглядывать за общей схемой, чтобы видеть и понимать, что цели, обозначенные в ней, вполне достижимы. Если это не так, вы можете изменить количество инициатив по совершенствованию или обновить схему (если этого никак не избежать).

В процессе управления трансформацией вам необходимо иметь некоторое представление обо всех частях организации, чтобы следить за тем, что изме-

нения не дают крен (например, в сторону тестирования, разработки или администрирования). Встречи, на которых будет обсуждаться процесс управления трансформацией, должны проводиться хотя бы раз в месяц, и необходимой документации для таких встреч должно быть как можно меньше. Если команда будет тратить кучу времени на то, чтобы оценивать презентации PowerPoint для каждой встречи, это не принесет пользы процессу трансформации. В идеальном случае для поиска путей к совершенствованию я бы посоветовал вам наблюдать за данными в реальном времени, за картой потоков ценностей и за несложными бизнес-кейсами.

Видимые ИТ-услуги

Если говорить о том, что можно визуализировать вещи при помощи реальных данных, стоит упомянуть, что некоторые возможности DevOps могут весьма и весьма для этого пригодиться. Одним из лучших помощников в таком деле может оказаться конвейер развертывания¹. Конвейер развертывания –



Рис. 1.3. Пример конвейера развертывания:
DevOps-платформа Accenture предоставляет возможность наблюдать за процессом развертывания

¹ Гэри Грувер написал целую книгу, *Starting and Scaling DevOps in the Enterprise*, о конвейере развертывания как о средстве, способствующем трансформации.

это визуальное представление процесса, в результате которого программный продукт проходит путь от разработчика к среде эксплуатации с определенными промежуточными стадиями. Это визуальное представление показывает, что происходит с программой, а также любой положительный или отрицательный результат ее работы. На рис. 1.3 я оставил пример одного решения, с которым часто работаю. Вы можете видеть, как различные стадии жизненного цикла и связанных с ними задач представлены в виде конвейера [1]. Этот конвейер развертывания непосредственно предоставляет информацию о качестве вашего программного продукта в реальном времени. Вы можете также подумать о том, чтобы агрегировать дополнительную информацию на панели мониторинга (или дашборда) либо скомбинировать основные данные с дополнительными, но так или иначе конвейер развертывания предоставляет некоторый фундамент. Он несет в себе вместе с тем и действенную функцию, так как все его шаги имеют некоторое представление и обязательно исполняются, а результаты можно просмотреть непосредственно на панели мониторинга (или дашборда) – это снизит вероятность того, что люди начнут осуществлять неотслеживаемые действия. Любые улучшения и изменения процесса можно видеть в конвейере развертывания до тех пор, пока это единственный разрешенный способ поставки изменений. Если доступ к метрикам затруднен, вы можете добавить шаги на каждом этапе конвейера, для того чтобы логировать метрики, необходимые для дальнейшего анализа.



Рис. 1.3 (продолжение)

Сегодня важно иметь в своей компании аналитическое решение для создания панелей мониторинга за данными в реальном времени. Многие компании пользуются готовыми решениями для визуализации или аналитики либо же создают нечто на основе open-source вариантов (наподобие Graphite). Основной целью здесь является использование данных, генерируемых на протяжении всего жизненного цикла, чтобы наполнять панели мониторинга (или дашборды), которые можно будет использовать не только для целей управления трансформацией, но и в любом подходящем случае. Высокопроизводительные команды связали цепочку DevOps-инструментов с аналитическими панелями мониторинга, и это позволяет им видеть важную информацию в реальном времени. Например, мы можем наблюдать за качеством текущего релиза, за тем, как качество выпускаемого пакета связано с проблемами, возникающими после развертывания, а также за тем, насколько автоматизация тестирования улучшила показатели по дефектам на более поздних стадиях жизненного цикла продукта.

Управление процессом поставки ИТ-услуг

Управление ИТ-процессами, по-моему, недооценено как составляющая часть процесса трансформации. Правда в том, что большинство подходов к управлению имеют существенные недостатки и помогают достичь лишь малой доли запланированных результатов. На большинстве встреч по управлению, на которых я присутствовал или в которых принимал участие, использовались текущие отчеты, составленные по системе «светофора», которые я считаю довольно субъективными по своей природе, и они не совсем подходят для представления статуса работы. Более того, даже если критерии для цветовой гаммы где-либо прописаны, зачастую руководство предпочитает узнавать личное мнение проектного менеджера. Проектные менеджеры из *Института проектного менеджмента* (PMI) используют *показатель эффективности стоимости* (CPI) и *показатель эффективности рабочего графика* (SPI); они немного лучше, но все еще основываются на необходимости иметь подробный, правильно составленный проектный план, в соответствии с которым будет готовиться отчетность. Я придерживаюсь того мнения, что, поскольку большинство проектов со временем меняются, составлять подробный план всего проекта в целом – напрасная затея: вы наверняка обнаружите, что в какой-то момент отклонились от курса.

Кроме того, к тому времени, как текущий отчет будет представлен на встрече, он будет существовать уже как минимум пару часов. В худшем случае вы будете дезинформированы из-за того, что было упущено множество различных сообщений, а проектному менеджеру пришлось работать с не вполне корректными результатами. Слишком часто текущие отчеты пестрят зеленым цветом на протяжении многих недель и внезапно «краснеют», когда плохие новости уже невозможно игнорировать. Или, например, оцениваемое состояние под-

нимается по цепочке управления и все чаще и чаще маркируется зеленым цветом по мере продвижения вверх по списку, ибо все стараются показать, что владеют ситуацией. Помните: одна из наших целей – сделать работу видимой, и мы не сможем добиться значительных успехов, если представляемая нами информация будет далека от реальности.

Что имеет смысл использовать в управлении доставкой ИТ-услуг, так это объективные показатели, например: число работающих элементов функционала, время восстановления после инцидента, длительность циклов для доставляемой функциональности, а также истории/функциональность, поставляемые и принимаемые после каждой итерации. Эти показатели предоставляют достойные способы оценивать успехи и качество. Данную информацию, а также другие метрики не стоит собирать вручную. Вы должны иметь возможность получать информацию из панели мониторинга (или дашборда) в режиме реального времени. Некоторые метрики можно брать из вашего конвейера доставки, но для многих из них потребуются дополнительные источники данных (например, ваш инструмент для управления жизненным циклом Agile). «Цветной комментарий» предоставляется проектной командой, и его можно накладывать на презентацию, так как обсуждение или аннотированные скриншоты можно создавать для встречи, на которой будет обсуждаться управление (см. пример аннотированного графика выполнения работ на рис. 1.4).

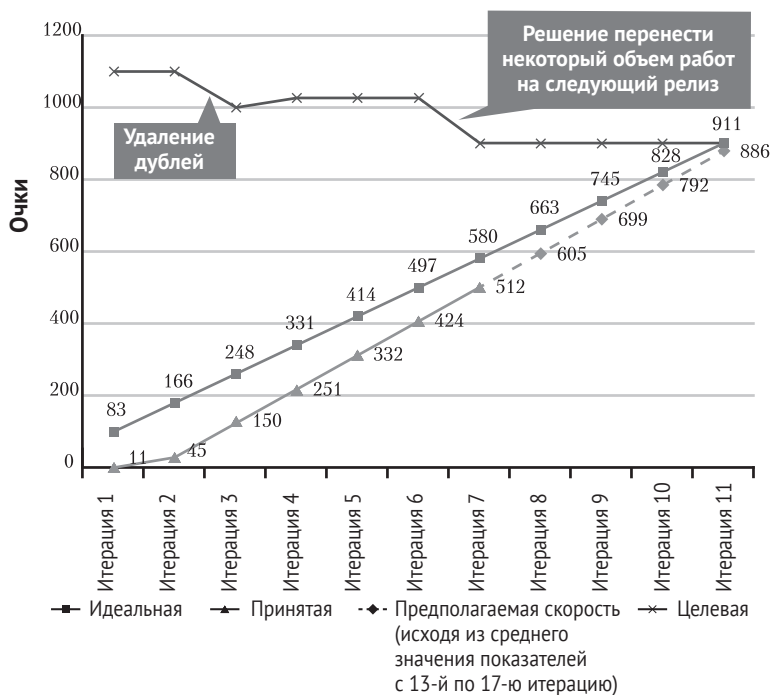


Рис. 1.4. Аннотированный график выполнения работ: на графике отображены аннотации статуса проекта

То же самое справедливо и для метрик, которые вы используете для управления ИТ-услугами. Нет весомых причин для того, чтобы не иметь таких данных по каждой стадии вашего процесса предоставления ИТ-услуг. Меня удивляет, что мы пользуемся ИТ для того, чтобы создавать замечательные аналитические решения для бизнеса, но сами же эти мощные решения не применяем в целях улучшения процессов организации. Ручной сбор данных для метрик неприемлем, если есть возможность внедрить в ваш процесс шаги по автоматическому логированию. В худшем случае вы немного автоматизируете каждый из шагов, который выводит данные в распространенном для логов формате. Мне часто приводилось такое делать, поскольку большинство используемых инструментов в жизненном цикле продукта не выводили извлеченные данные в формате, который можно было бы легко использовать. Использование большинства инструментов предполагает применение встроенной функциональности по составлению отчетов, что не всегда подходит. Вам нужно иметь возможность связывать данные из одного инструмента с другим (и, возможно, с инструментами различных поставщиков), поэтому вам придется создавать свои инструменты, чтобы убедиться, что данные в жизненном цикле вашего продукта остаются доступными для анализа. Подобное внимание к данным со временем значительно окупится.

Такой большой объем данных запросто может оказаться непосильным, и объем данных, генерируемых в процессе поставки ИТ-услуг и администрирования, чрезвычайно велик, так что подходы к работе с ними можно сравнить с подходами, применяемыми к большим данным. Самое главное здесь будет сосредоточиться на узких «бутылочных горлышках». Пробуйте создать метрики, которые смогут описывать их состояние, и следите за ними в процессе трансформации. Когда вы справитесь с основной проблемой, акценты сместятся на другие, и тогда вам уже понадобятся новые метрики. К счастью, если создать хороший фреймворк для анализа метрик, включая панели мониторинга (или дашборды) и подготовку данных, то у вас на руках будет вся необходимая информация. Панели мониторинга (или дашборды), как описывалось ранее, – прекрасный инструмент, позволяющий агрегировать информацию и сделать ее доступной для других инструментов.

Предоставление ИТ-услуг по принципу Lean

В процессе трансформации внимание часто уделяется техническим практикам, хотя многое можно улучшить, если применять принципы Lean к подходам по управлению ИТ-услугами. Под управлением ИТ-услугами я подразумеваю любой этап в ходе ИТ-поставки, когда необходимо подтвердить нечто перед тем, как продвигаться дальше. Это могут быть ключевые точки по финансированию проекта, принятие решения о развертывании в тестовой среде, смена панели управления и т. п. В жизненном цикле разработки продукта обычно бывает множество таких шагов, требующих подтверждения или проверки, ко-

торые отнимают довольно много времени и сил. А процессы управления со временем разрастаются. После возникновения проблемы мы совершаем проверку реализованного функционала и добавляем еще один шаг, для того чтобы данная проблема больше не возникала. В конце концов, никому не повредит лишний раз убедиться дважды, правда? Со временем это спровоцирует появление процесса управления, шаги которого обесмыслятся, и отследить их станет нереально. Я наблюдал процессы подтверждения развертывания, на которые уходило значительно больше времени, чем обычное развертывание, не подразумевающее добавления ценности и улучшения качества. Мне кажется, что некоторые шаги подтверждения превратились в этакие управленческие шаблоны, не несущие особого смысла, так как информация сейчас не оценивается должным образом. Следующие рассуждения помогут вам сократить ненужные шаги в ваших процессах.

Я хотел бы, чтобы вы внимательно оценили каждый этап в вашем процессе управления, чтобы понять: 1) насколько этот этап важен (например, в контексте отклонения некоторой заявки); 2) насколько рискованно его проходить и 3) сколько это стоит.

Давайте рассмотрим каждый из этих трех аспектов в отдельности.

1. Если обратить внимание на шаги рассмотрения заявок в жизненном цикле разработки ПО, как часто заявки отклоняются и как часто при рассмотрении обнаруживаются реальные проблемы, которые требуют решения? (Я имею в виду действительно важные проблемы, не просто отказы по формальным причинам вроде неправильного заполнения бланка заявки.) Чем реже обнаруживается, что процесс приносит должные результаты, тем более вероятно, что он не несет в себе ценности. То же самое можно сказать в случаях, когда заявки одобряются с вероятностью 90 %. Наверное, стоит обойтись уведомлением, вместо того чтобы заставлять людей ждать рассмотрения заявки, которая с высокой вероятностью будет одобрена. Или же вы можете совершенно избавиться от этого шага. Я работал с одним клиентом, у которого сотрудники команды разработки были вынуждены просить одобрения заявок перед каждым развертыванием после того, как все подготовительные шаги были осуществлены, что увеличивало этап подготовки перед развертыванием. Рассматривающий заявку не делал никакой полезной работы, если учесть, насколько быстро принималось решение. Это была сущая формальность. Я порекомендовал избавиться от этого шага и изменить процесс так, чтобы информация просто отправлялась на рассмотрение до и после развертывания; то же самое касалось и результатов тестирования. Подготовительное время значительно сократилось, у сотрудника уменьшился объем работы, и из-за того, что мы убрали шаг, выполняемый вручную, у нас появилась возможность автоматизировать процесс развертывания с самого начала.
2. Если мы и далее будем выполнять процесс без этого шага и что-то пойдет не так, насколько большими будут риски? Как много времени понадобится,

чтобы выявить проблему и исправить ее или отказаться от изменений? Если риск небольшой, то, опять же, шаг можно пропустить или ограничиться отправкой уведомлений.

3. Сколько усилий и времени требуется на выполнение этого шага? Как долго готовится документация на данном этапе? Сколько времени тратит на этот шаг каждая из заинтересованных сторон? Сколько времени в жизненном цикле уходит на ожидание одобрения?

Имея на руках такую информацию, вы сможете оценить, нужно ли продолжать использовать этот шаг или его стоит убрать либо как-то упростить. По моему опыту, около половины шагов, подразумевающих рассмотрение заявок, можно или автоматизировать (так как человек при этом следует повторяемым шагам), или свести их к отправке уведомления, в результате чего процесс перестанет замедляться. Я советую вам попробовать этот подход в вашей организации. Вы увидите, как легко на самом деле отказаться от всего ненужного и приблизиться к достижению минимального жизнеспособного процесса управления. Упражнение на такие случаи приведено в конце этой главы.

Первые шаги вашей организации

Здесь я представляю три упражнения, которые считаю невероятно полезными, так как они помогают достичь значительных результатов относительно малыми силами: 1) составление карты потока ценностей для вашего процесса поставки ИТ-услуг; 2) формирование основных метрик и 3) пересмотр подходов к управлению процессами. В результате вы сможете лучше понять ИТ-процессы и начать совершенствоваться.

Составление карты потоков ценностей для ИТ-процессов

Существует формальный процесс для составления карты потоков ценностей, однако здесь я представляю сокращенную версию, которая, по моему опыту, сравнительно неплохо помогает достичь поставленной нами цели: сделать процессы видимыми и справиться с некоторыми узкими местами¹. Вот этот краткий план по составлению карты потоков ценностей:

1. Соберите на переговоры всех заинтересованных участников цепочки выноса поставок (supply chain): например, представителей бизнеса, сотрудников отделов разработки, тестирования, менеджеров проекта, администраторов и бизнес-аналитиков).
2. Подготовьте высокоуровневую схему процесса. Стоит для наглядности упомянуть на ней такие понятия, как, например, «бизнес-идея», «старт проекта», «разработка», «тестирование/QA», «развертывание/релиз» и «создание ценности».

¹ Вам стоит обратиться к Value Stream Mapping Карен Мартин и Майка Остерлинга, если нужно в большей степени формализовать процесс.

3. Попросите всех за 15 минут совместно записать краткое пошаговое описание ИТ-процесса на карточках. Затем попросите приклеить эти карточки на доску, чтобы общими усилиями составить полноценную картину процесса. Внимание: вам, возможно, понадобится воодушевить людей на сотрудничество или самому вступить в обсуждение, если ситуация будет развиваться в нежелательном направлении.
4. Как только карта будет составлена, попросите нескольких человек заново описать весь процесс, а также спросите, стоит ли что-то добавлять.
5. Теперь, когда у вас есть осмысленное представление процесса, вы можете попытаться подробно обсудить его циклы, ключевые точки представлений заинтересованных сторон о качестве или каких-либо других аспектах, а также инструменты, которые будут помогать обслуживать процесс.
6. Попросите людей проголосовать за наиболее важные узкие места (например, дайте возможность каждому оставить лишь три отметки на доске напротив трех самых значимых в этом отношении шагов).

По моему опыту, это упражнение наилучшим образом помогает сделать ваш процесс видимым. Вы можете повторять данное упражнение каждые три–шесть месяцев, чтобы понимать, правильно ли вы обращаетесь с узкими местами, и видеть, как развивается процесс. Можете наглядно представить результаты этого процесса, вывесив их где-нибудь в офисе, чтобы показать всем приоритетные направления совершенствования. Узкие места, которым было уделено наибольшее внимание на встрече, могут стать ключевыми точками в вашей начальной схеме, поскольку это то, на что должны ссылаться ваши инициативы.

Формирование основных метрик

Так как наличие метрик невероятно важно для процесса управления трансформацией, я хочу, чтобы вы посвятили несколько минут заполнению табл. 1.2. Выделите для себя метрики, которые важны для вас сейчас и в будущем, и определите механизм, согласно которому вы будете их составлять. Существует несколько способов формирования основного ряда метрик. Они могут быть основаны на опросах, текущих исследованиях или, в идеальном случае, на существующих либо исторических данных. Если задача чересчур сложна, то вам стоит подумать об автоматизированном способе измерения метрик. Там, где это не сработает, вы можете проводить исследование вручную и оценивать процесс самостоятельно (например, во время текущих исследований), но такие метрики будут вызывать меньше доверия и их составление потребует больше времени.

Таблица 1.2. Пример формирования метрик: метрики должны иметь описание, механизмы составления и базовое значение

Метрика	Описание	Механизм составления	Основной (базовый) подход	Базовое значение
Длительность цикла	Среднее время, необходимое для того, чтобы продвинуться от готовой истории до развертывания в среде эксплуатации	Извлечение даты и времени из Agile-системы управления жизненным циклом	Анализ пользовательских историй, которые были успешно развернуты в среде эксплуатации за прошедшие шесть месяцев	168 дней

Пересмотр подходов к управлению процессами

Многое можно обсудить в части автоматизации, которая способна помочь усовершенствовать скорость и качество ИТ-процессов. Что люди часто недооценивают, так это то, каких результатов можно добиться за счет одной лишь коррекции процесса управления. Ниже я привожу небольшой список вопросов, который вы можете использовать для того, чтобы пересматривать ваш процесс. Задавайте эти вопросы, чтобы сосредоточиться на тех элементах, где управление ИТ-процессами действительно необходимо. Ваши ответы позволят оценить эффект и риски от сокращения шага в процессе, в идеальном случае даже с применением экономической модели, отражающей финансовое воздействие и сформированной вероятностью рисков.

Список вопросов об управлении процессом:

- Как часто кто-либо отклонял заявку на выполнение одной из задач, основываясь на причинах, не связанных с правилами соответствия процессу?
- Что может произойти с процессом, если будет совершен неверный выбор?
- Какую ценность принесет человек, одобряя эту заявку вручную вместо компьютера, который может осуществить этот шаг при помощи ряда параметров?
- Как много времени и средств будет затрачено на управление процессом (включая обычное время ожидания одобрения заявки)?
- Основан ли этот шаг на объективных показателях или на субъективных? Как вы это определили?