

Все о

Клод Обри

SCRUM

ПРОДАНО
25 000
ЭКЗЕМПЛЯРОВ
ВО ФРАНЦИИ

▶ изучение

▶ разработка

▶ интеграция

5-е издание

 **БОМБОРА**
ИЗДАТЕЛЬСТВО

Клод Обри
**Все о SCRUM. Изучение,
разработка, интеграция**
Серия «Библиотека
цифровой трансформации»

*http://www.litres.ru/pages/biblio_book/?art=66624184
Все о SCRUM. Изучение, разработка, интеграция:
ISBN 978-5-04-160195-9*

Аннотация

Эта книга написана для всех, кто хочет ознакомиться с методикой Scrum. Новички найдут здесь подробные примеры, а те, кто уже знаком с этой методикой, могут воспользоваться полезными рекомендациями. Книга будет интересна всем членам команды. Если вы только собираетесь внедрять эту методику, то в книге вы найдете полезные советы, которые сэкономят ваше время и сделают работу более эффективной.

В формате PDF А4 сохранен издательский макет книги.

Содержание

Предисловие	5
Пролог	12
1	18
1.1 Первая схватка со Scrum	20
1.2 Agile-движение	30
1.3 Скрам сегодня	40
1.4 Scrum – живой организм	44
2	49
2.1 Изменение парадигмы	50
2.2 Итеративный и инкрементальный подход	53
2.3 Спринт	59
2.4 Результат спринта	67
Конец ознакомительного фрагмента.	70

Клод Обри

Все о SCRUM. Изучение, разработка, интеграция

© Степанянц Е.Г., перевод на русский язык, 2020

© Оформление. ООО «Издательство «Эксмо», 2021

Предисловие

Scrum все-таки забавная штука. Мелочь, выросшая до гигантских масштабов. Несколько строк, которые перевернули наши методы работы, а то и жизни целиком. Всего несколько строк... Scrum в первую очередь – это метод реализации проектов в сложных, неопределенных условиях. И этой маленькой идее удалось пробиться: она быстро свергла имеющуюся методологическую аристократию (все PMI, CMMi, RUP, и т. д.). Вековые исполины пали. Ощущение, словно кто-то открыл окно в давно не проветриваемой комнате. Время здесь только сыграло на руку, эта современная и неопределенная эпоха растущей конкуренции. Кое-кто хорошо посмеялся: ведь Scrum рассекретил большие организации и обнажил все их маленькие обманы.

Забавно, как этот мальчик-с-пальчик превратился в великана. Scrum – будто черная дыра, поглощающая все вокруг. Клод в своей книге рассуждает не только о Scrum, но о всем том, что Scrum символизирует и предлагает для организаций и проектов в их стремлении к Agility¹. Это куда больше нескольких абзацев, что были в самом начале пути! Scrum позаимствовал все самое ценное, взял самое важное из нашего опыта и других Agile-методов. Экстремальное

¹ См. раздел 1.2.5 – *Прим. ред.*

программирование – это как сварливый и хмурый дядюшка, который ворчит из угла, что Scrum продался и позабыл все самое дорогое. Экстремальное программирование мнительно, но забывает, что и оно, и Scrum – близкие родственники и носят одну фамилию. Просто Scrum – хороший собеседник, и это повод для ревности и зависти. Scrum сегодня в самом деле блистает. Кому-то может показаться смешным, что на Scrum стали равняться.

Забавно, как люди ошибаются насчет Scrum. Они думают, что это легко и просто. Но в жизни все оказывается труднее. В этом парадокс наших методов: они сложны, как и организации, как проекты. Их нужно адаптировать, улучшать, нужно фокусироваться на ценности, пытаться, учиться на ошибках. Этот парадокс в Scrum повсюду. Крохотная идея стала чем-то большим, огромным, Давидом, Голиафом! Альтернативный подход стал образцовым. Люди цепляются за него: понимают, что это лучшее снаряжение для борьбы в современном мире. Но его легкость вводит и в заблуждение: клинок действительно острый, это наточенный инструмент для применения принципов и ценностей Agile. Многие ошибаются и... хватаются голой рукой за лезвие!

Успех этого метода может привести к тому, что он со временем станет слишком громоздким или быстро устареет. В этом книга Клода является своего рода противоядием. Если на то пошло, мир уже начинает путаться и теряться в Agility: вокруг него столько недопониманий, недоразумений – в ито-

ге сарафанное радио доносит до людей совершенно неверные суждения.

Именно это случилось с Lean в 1990-х. Еще один двоюродный дядя Scrum, но намного старше и напуганный нашей гибкостью (именно так, ведь именно она ознаменовала конец предыдущей системы). Сейчас Lean уже устарел. Мы помним, что это концепция «бережливого производства», и упускаем такие его компоненты, как уважение к людям и постоянное развитие. Lean – не для современных организаций или проектов. Это уже не время для Lean, стандартизации и устранения потерь за счет вовлечения и уважения к людям. Это время для адаптации, НЕ-стандартизации, НЕ-линейности, НЕ-повторяемости, вызванных сложностью, неоднозначностью нашей эпохи.

И вот парадокс: нет двух людей, которые прочитали бы книгу Клода и одинаково поняли Scrum. Говорим ли мы все об одном и том же? Да, но контекст и устремления определяют все остальное. Хочешь измениться? Надо попытаться стать пуристом – а попытки невозможны, пока не начнешь адаптироваться. Великолепный парадокс!

Смешно при виде того, как старая система идет и качается, вздыхая на ходу. Она пытается сопротивляться, но безуспешно: противостоять прогрессу не получается. Да здравствует Scrum (а также его ворчливый дядюшка XP и его амбициозный брат Kanban)!

Впрочем, Клод тоже забавный парень. А каким еще

можно быть, чтобы так подробно изучить, что произошло со Scrum (и не только) за последние пять лет? Эта книга не ограничивается сухим объяснением Scrum, она, скорее, о применении Scrum в жизни, об опыте, что накоплен за последние десять, двадцать лет. Здесь собраны все хорошие мысли и идеи, накопленные, реализованные, отвергнутые и принятые. Думаете, Клод простой парень, добродушный теоретик? Он задиристый, любознательный искатель правды – но он мягок и мудр. Как и сам подход, который Клод защищает, – сложный и разумный. Как и Scrum, он легкий на подъем и гостеприимен, но таит много сюрпризов. Здесь жажда знаний (её он распространяет – и потому он такой хороший учитель), и настоящее любопытство. Он наблюдает и бдит. Да, нашлось хорошее слово: он бдителен! Он бдит за порядком и логикой мыслей – не как хранитель прошлого, а как носитель будущего, новой перспективы, поэтому он и непрерывно изучает новые подходы и идеи. Перед вами уже пятое издание книги – как доказательство, что Клод изыскатель, пионер. Мне смешно, когда я вижу, как Клод набрасывается с кулаками на безыдейные речи или демонстративно разводит руками и оставляет журил и брюзг кряхтеть в их углу.

Прежде чем отпустить вас читать книгу и пожелать приятного времяпрепровождения, я напомним о важном, что стоит держать в голове при изучении Agile-движения. Это выход из зоны комфорта. Если у вас все будет легко получаться – скорее всего, вы в чем-то ошибаетесь. Agile предполагает ра-

дикальную смену парадигмы в отношении привычного видения компании, процесса создания ценности, динамики группы. Мы находимся на другом полюсе от французских обычаев последних пятидесяти лет: их картезианства, иерархии и – в противовес исключительной изощренности, которую я люблю – их культура совершенства.

Представьте, что на этой книге наклейка «Опасно!». И вы подвергаете себя этой опасности, чтобы полностью понять все тонкости Agility. Без риска и выхода из зоны комфорта вы завернете свои старые привычки в новые тряпки, вот и все. И Клод потерпит неудачу.

Попробуйте спровоцировать то самое «вау!», которое слышалось из всех американских домов в 50-е годы прошлого века, когда Элвис произвел фурор на телевидении со своим фееричным хитом «That's all right mama». Если вы не чувствуете в себе достаточно смелости, чтобы рассуждать и быть гибким, то дело скорее всего в том, что: а) на дворе 2030 год, и вы держите в руках очень старое издание книги Клода; б) вы – один из уникамов, которым гибкость присуща с рождения; в) вам пора взбодриться, выйти из зоны комфорта и рискнуть.

Попробуйте применить знания, полученные в этой книге, на практике. Сделайте это, чтобы увидеть, получить представление о пределах их действия, вынести реальные уроки. Практикуйтесь почти до абсурда – это разумнее, чем кажется. Будьте стремящимися к знаниям максималистами, а

не закостенелыми догматиками. Не замыкайтесь на крайностях, сфокусируйтесь на пользе и эффективности, на знании и понимании. Многие видят в Agile здравый смысл – и это по большей степени верно. По большей степени – и только: треть этого подхода не является ни интуитивной, ни несущей смысл. Эта его часть иногда забывается, тем самым нарушается общая логика. Чтобы исследовать таинственный сад, в нем надо прогуляться.

В наши дни все чаще можно услышать такие слова как «agile», «lean», «lean startup», «design thinking», «высокоэффективная компания». Каждое из этих слов – словно обертка общества, которому оно адресовано: agile – для ИТ-специалистов, lean – для методологов, lean startup – для бизнесменов, design thinking – для агентств и креативщиков, высокоэффективная компания – для предпринимателей. Но за всем этим стоит то же самое основное движение: это глубокая, революционная трансформация нашего мировосприятия, наших организаций, наших отношений. Будем ли мы только надеяться, что это преобразование осуществится, или трансформация действительно неизбежна (на что я рассчитываю)?

Как оценить, что подход работает? Что Scrum приносит плоды? Не считайте, каким образом ваша реализация Scrum повлияла на архаичную систему. Лучше просто расскажите друзьям и родственникам во время ужина о своей организации, о вашем использовании Scrum. Вот лучший показатель.

Черт, теперь вы думаете, что Scrum не такой уж и простой, подразумевает выход из зоны комфорта и риски, а пути к отступлению больше нет? Вы обескуражены? Отлично! Теперь все будет хорошо.

Пабло Перно

Агент-provokator, автор блога «Are you agile»

PS: У нас с Клодом есть еще один общий интерес: музыка, рок-н-ролл, а если сократить до размеров одной группы – Led Zeppelin, эти четверо парней, всколыхнувшие умы многих, ворвавшиеся дикой ордой на сцену и блистательно исполнившие непревзойденный рифф "Whole Lotta Love". Чтобы чтение этой книги стало еще приятнее, рекомендую вам включить Led Zeppelin и откупорить бутылочку шартреза или сливового.

Пролог

Почему пятое издание? Потому что Scrum развивается, Agility живет, постоянно черпая вдохновение из новых дисциплин. Потому что мир меняется, человечество не стоит на месте, и нужны новые способы разработки продуктов.

Я хочу объединить понятие системного подхода и идеи пермакультуры, чтобы закрепить этот живой опыт Agility.

Для кого это издание? Сферы использования Scrum расширяются все больше, выходя за изначально установленные границы информационных технологий. В этой книге я обращаюсь ко всем, кто – из любопытства или с целью углубить знания – ищет легкий и доступный подход к Scrum и последним достижениям в области Agility.

Scrum приобрел широкую известность. И через эту книгу я обращаюсь не только к новичкам, но и тем, кто уже знает и использует данную методику.

Именно для них в этом издании есть нововведения, которые помогут:

- ✓ понять, каким образом применять Scrum в той или иной ситуации. Здесь нет единых правил, и достичь результата можно самыми разными способами;

- ✓ изучить Scrum-паттерны и узнать, когда их необходимо использовать;

- ✓ избежать распространенных ошибок благодаря списку

антипаттернов, представленных в конце каждой главы;

✓ начать работать в Scrum-фреймворке благодаря новым главам, посвященным подготовке и разработке первоначального бэклога;

✓ привнести смысл в работу, действуя во имя благородных амбиций и поддержания командной этики.

• Путеводитель по книге

Книга состоит из семи больших частей:

✓ Первые две главы рассматривают положение Scrum в Agile-движении и помогают понять важность работы в маленьких итерациях.

✓ Главы **3–5** фокусируются на самом важном компоненте Scrum – людях, на их ролях и взаимодействии в команде и в экосистеме.

✓ Главы **6–8** посвящены бэклогу, его структуре и доработке – классическому артефакту Scrum, который обеспечивает команду делами и задачами.

✓ В главах **9–12** говорится о том, что задает ритм всему спринту: планировании, схватке, демо, ретро.

✓ Главы **13–15** представляют краеугольный камень успешного старта работы в формате Scrum: прелюдию.

✓ В главах **16–20** рассматриваются наиболее полезные дополнения к Scrum: процесс планирования, индикаторы, инструменты, инженерия.

✓ Две последние главы предлагают пути для устойчивого внедрения Scrum в масштабе организации.

Главы расположены в четкой последовательности, и иногда необходимо сначала ознакомиться с одной, прежде чем переходить к следующей.

• Что еще важно знать

✓ **Библиографические ссылки** теперь представлены в конце большинства глав, чтобы читатель сразу по прочтении мог расширить знания по теме. В списки включены только книги или статьи, которые я сам прочитал. Страница с дополнительными материалами есть и в моем блоге².

✓ **Стоит отметить.** Часть примеров содержат ссылки



на *Peetic*. Это вымышленный сайт встреч для животных и их хозяев. Мы придумали *Peetic* вместе с Пабло Перно во время образовательного проекта *Raids Agiles* в Севеннах.

✓ **Формат** глав пересмотрен, чтобы сперва ответить на вопросы *почему?* и *что?*, а потом уже *как?*. При ответе на последний мы рассмотрим как можно больше разных подходов и паттернов.

² Блог автора можно найти по адресу: <http://www.aubryconseil.com/pages/Livre-Scrum>. Все материалы на нем представлены на французском языке. – Прим. ред.

✓ **В пятом издании** вы заметите присутствие гендерно-нейтральных формулировок. Scrum, наиболее распространенный в сфере информационных технологий, является очень «мужским». Это очень грустно, и я искренне надеюсь, что гендерно-инклюзивная речь поможет восстановить баланс в языке, а затем и на практике в Scrum-командах.

✓ **В глоссарии** объясняется терминология Scrum и лексика, которую я использую.

• Благодарности

Мои рецензенты сделали многое для качества этой книги. На этот раз у меня появились новые помощники, такие разные и молодые, которых я от всего сердца благодарю:

- ✓ Алис Барралон, здравый смысл из Страны Басков.
- ✓ Жан-Франсуа Марронье, дух и ценности игры в регби.
- ✓ Потрясающий дуэт двух Жанов, а именно Жан Паласу-энос и Жан-Паскаль Буаньяр.
- ✓ Стив Эверс, непростой вандеец.
- ✓ Бертран Уриг, которого у меня не получилось ввести в заблуждение.

Спасибо Бенжамену Кабанну, Стефану Ланглуа, Фабрису Эметти, Максиму Гюйо и Дени Бенуа, которые рецензировали несколько глав моей книги, каждый в собственном, уникальном стиле.

Спасибо также Морису Понсе, Натаниэлю Ришану, Винсенту Баррье, Ромену Кутюрье.

Рисунки Патриса Кортиада с первого издания привносят свой колкий, великолепный юмор. Сейчас их около пятидесяти, и в этом выпуске добавлены новые. Патрис, большое тебе спасибо!

Кроме того, Рамюншо нарисовал маленьких человечков к главе о прелюдии, а Алис изобразила «ретрокаштан».

Я искренне благодарю всех людей, которых встретил во время моего обучения и участия в проектах, их отзывы и их поддержка были очень ценными для меня.

Я очень благодарен Пабло за его яркое предисловие к этому изданию.

Спасибо Рут, Жюльену и Лоре за их ободрение и поддержку.

Я закончил это издание в начале 2018 года. Я начал заниматься информационными технологиями в 1978 году – тогда в мои задачи входило написание встроенного программного обеспечения для самолетов и телефонов. Это было сложно. Но поскольку я был молодым инженером, специализирующимся на кибернетике, то не мог должным образом применить профильные знания на практике – ни тогда, ни позднее – в других областях.

К сожалению, в течение многих лет в сфере разработки программного обеспечения приходилось быть жертвой кар-

тезианского подхода вместо использования системного. Результатом стали громоздкие процессы, повлекшие за собой соответствующую реакцию в виде Agile-манифеста.

Теперь, когда «software is eating the world»³, системный подход, наконец, появляется (снова) и начинает восприниматься людьми как наилучший способ справиться с неопределенностью и сложностью.

В этом издании я больше говорю о системах и экосистемах. Уделяю больше внимания механизмам регулирования Scrum. Я рад, наконец, обратиться к кибернетике.

Со времени моего обучения системология здорово шагнула вперед. Появились новые дисциплины. Среди них я сразу выделил пермакультуру⁴ – дисциплину, которая появилась несколько лет назад и страшно меня заинтриговала.

Она приносит, на мой взгляд, очень важную идею постоянства культуры, дополняя Scrum и Agile и наделяя их бóльшим смыслом.

Клод ОБРИ

Кастане-Толозан, 30 января 2018

³ В переводе: «софт пожирает мир», известная цитата Марка Андрессена, американского изобретателя, инвестора, основателя Netscape Communications и венчурного фонда Andreessen Horowitz. – *Прим. ред.*

⁴ Пермакультура (от англ. permaculture – permanent agriculture – «Постоянное сельское хозяйство») – подход к проектированию окружающего пространства и система ведения сельского хозяйства, основанные на взаимосвязях из естественных экосистем. – *Прим. ред.*

1

Место Scrum в Agile-движении

В 1996 году я был консультантом по разработке программного обеспечения. Twitter еще не существовал. Чтобы не отставать от технического прогресса, я читал, пусть иногда месяцы спустя, материалы конференций того времени. Agility, *гибкость*, тогда не была в тренде, зато было ООП (объектно-ориентированное программирование). Одной из многочисленных конференций по теме была «OOPSLA» («Object-Oriented Programming, Systems, Languages & Applications»). Именно во время просмотра материалов 1995 года я впервые наткнулся на Scrum. Подписанная Кеном Швабером, статья представляла Scrum как эмпирический процесс разработки сложных продуктов.

Пока я читал, в голову пришла мысль: этот пришелец явно не с привычной мне планеты. Хотя в статье было много ссылок на объектно-ориентированное программирование (вероятно, просто чтобы ее включили в конференцию OOPSLA), она противоречила настроениям того времени.

Это все было слегка за пределами сферы моих интересов – архитектуры ПО и моделирования. И все же я оценил акцент на команде, а упоминание регби возбудило любопытство. Разработанные идеи заинтриговали меня, особенно эм-

пирический подход: они были противопоставлены индустриализации, модной на тот момент.

Я и представить себе не мог, что двадцать лет спустя Scrum станет настолько популярным, что будет вызывать критику из-за своего доминирующего положения среди других подходов.

Давайте коротко рассмотрим, что это вообще такое – Scrum.

1.1 Первая схватка со Scrum

1.1.1 Легкий фреймворк

В статье 1995 года Швабер взбудоражил читателей, рассказав о процессе и методологии. После этого Scrum чаще всего определялся как Agile-методология.

Затем Кен Швабер и Джефф Сазерленд, его со-основатель, установили, что Scrum – это процессный фреймворк (*process framework*).

Scrum не является законченным процессом (как и методом или методологией), это процессный фреймворк.

Процесс определяет способ работы, а фреймворк только определяет границы, *фреймы*. Это рамка, при помощи которой Scrum вводит несколько правил и принципов.

Классифицировать Scrum нелегко – проще объяснить механизм его реализации.

1.1.2 Scrum вкратце

Прежде чем перейти к ответу на вопрос *как?* – факт, на который стоит обратить внимание:

Scrum действительно помогает людям работать в команде.

Слово *команда* имеет фундаментальное значение!

Можно объяснить Scrum несколькими словами. Помнится, были челленджи, в которых надо было представить Scrum меньше, чем за пять минут. Удалось это далеко не всем. Большинство пытались прояснить, как, собственно, применять Scrum.

На данный момент моя версия ответа на этот вопрос звучит так:

- ✓ Люди работают в команде, следуя принципам Scrum.
- ✓ Ритм устанавливается при помощи серии итераций.
- ✓ Все, что необходимо сделать, включено в список задач.

Ориентируясь на этот список задач, команда непрерывно работает, условно разделяя весь процесс на события спринта:

- ✓ Первое событие в начале спринта – согласование цели и подготовка к работе.
- ✓ Второе событие – это ежедневная синхронизация команды для достижения общей цели.
- ✓ В конце спринта команда представляет результат и запрашивает обратную связь, а затем рассуждает о том, как прошел этот спринт, чтобы улучшить следующий.

1.1.3 Истоки Scrum

Scrum не аббревиатура. Это слово взято из игры в регби: в английском языке scrum означает *схватка*. Чтобы отличать одно от другого, к нашему Scrum добавили заглавную букву.

Вот почему мы не пишем SCRUM. И вот почему мы произносим именно так: скрам. Если хотите убедиться в своем правильном произношении, включите матч по регби с англофонным арбитром.



Рисунок 1.1 – Схватка

По правилам регби, схватка возобновляет игру после какого-либо нарушения правил. Правила постоянно меняются (и в этом очарование регби) – особенно те, что касаются самой схватки. Но те, которые регулируют ее присуждение, более менее постоянны:

- ✓ был пас вперед;

✓ мяч был выбит в аут.

Во время схватки команда может завладеть мячом, если ее игроки в каждой из трех линий действуют синхронно. Это момент, когда – чтобы вырваться вперед – необходима максимально сплоченная игра. Scrum использует данную метафору по отношению к команде.

В 2005, когда я впервые представлял Scrum на конференции, я много говорил о его связи с регби. Вот выдержка из моего выступления:

Scrum использует ценности и дух регби и адаптирует их для проектов по разработке новых продуктов. Как линия игроков в регби, разработчики объединяются и прилагают коллективные усилия для достижения конкретной, общей цели. Scrum-мастер, словно полузащитник, направляет членов команды и задает темп, чтобы обеспечить успех проекта.

• Почему именно регби?

Оба основателя Scrum – американцы. В США особо не играют в регби со схватками. Почему же именно этот вид спорта?

Отсылка к регби появилась в статье 1986 года, написанной японцами. Они любят эту игру вместе с ее схватками. Там было описано, как компании Honda, Canon, NEC и Fuji-Xerox демонстрируют совместный подход к разработке сво-

их продуктов, и подчеркивалась важность автономных и самоуправляемых команд. Авторы сравнили такой формат работы с игрой в регби:

Традиционный последовательный подход к разработке продукта, известный как «эстафета» (например, при поэтапном планировании программ), может противоречить целям максимальной скорости и гибкости. Вместо него, удовлетворяя требованиям конкурентоспособности, лучше подходит целостный подход – как при игре в регби, где команда прилагает совместные усилия и играет сплоченно, перемещаясь по полю и передавая мяч из рук в руки.

Джефф Сазерленд опирался на эту статью, когда придумывал Scrum.

1.1.4 В чем Scrum принципиально отличается?

В мире организаций, где центральное место все еще занимает процессный подход и индивидуализация целей, Scrum сильно выделяется.



Рисунок 1.2–6 свойств Scrum

1. Появление нового

Ты не можешь сразу все знать и уметь. Команда оказывается в условиях, благоприятных для появления новых отношений, связей, идей касательно продукта – для появления концепции и нового, лучшего способа совместной работы.

2. Приоритизация

Во время спринта команда сосредоточена на вещах, которые имеют наибольшую ценность, и старается не тратить время на то, что не приносит мгновенного и ощутимого

результата. Не нужно одновременно начинать много задач, лучше положиться на принцип «Прекращай начинать, начинай заканчивать».

3. Предприимчивость

Команда уполномочена самостоятельно организовывать свою работу в соответствии со своей целью. Такой подход позволяет вовлечь всех участников процесса и способствует их раскрытию в команде.

4. Прозрачность

Мониторинг прогресса осуществляется самой командой. Каждый участник в любой момент времени знает, на каком этапе команда находится. Хороший способ добиться прозрачности процесса и ее поддерживать – это визуальный менеджмент.

Я всегда практикую визуальное управление!



Рисунок 1.3 – Визуальный менеджмент

5. Практика

Изначально Scrum был создан для разработки сложных систем эмпирическим, опытным путем. Это может удивить тех, кто уверен: мол, Scrum не подходит для их «слишком сложной» сферы деятельности. Но Scrum нацелен именно на сложные задачи! В настоящее время, когда все вокруг представляется сложным, практический подход позволяет найти решение и продвигаться к нему через короткие циклы с частой обратной связью.

Эмпирический подход выражается в трех принципах: прозрачность, проверка и адаптация.

6. Постоянный ритм

Scrum использует фиксированные отрезки времени для создания ритма. Основа ритма Scrum – это спринт. Спринт завершается по расписанию, а не с окончанием работы. Отодвинуть завершение спринта по времени невозможно.

1.1.5 Где найти справочные материалы по Scrum?

На протяжении долгого времени не было книг о Scrum. Затем Кен Швабер опубликовал несколько работ, но ни одна из них не была безупречной и полностью понятной читателю.

В 2010 Джефф Сазерленд и Кен Швабер опубликовали небольшой документ – «Руководство по Scrum» («The Scrum Guide»).

Очевидно, что на 19 страницах текста авторы дают читателю лишь правила игры, рассказывая о Scrum в общих чертах. Но этого достаточно, чтобы получить представление о том, что это такое.

Это руководство – надежный источник информации о Scrum.

В начале 2017 года компания 3Back опубликовала статью объемом около двадцати страниц, в которой предлагалось развить концепцию Scrum. Авторы – Дэн Роастхорн и Дуг Шимп – ранее уже написали весьма содержательную кни-

гу «Исследуя Scrum»⁵. Их новая статья называлась «Обзор Scrum 3.0» (Scrum 3.0 overview ⁶)

В предыдущем издании этой книги я рассматривал некоторые паттерны, предложенные Роастхорном и Шимпом.

В этом пятом издании у меня следующая позиция: в Scrum 3.0 действительно есть интересные мысли.

Идеи, изложенные в этой книге, не противоречат вышеупомянутым источникам, но в поле моего зрения больше деталей, чем вы найдете в этих небольших документах. Кроме того, я опирался и на другие источники. И самое главное: я говорю не только о Scrum, в этой книге вы найдете то, что никак не связано со Scrum 3.0.

⁵ Ориг. назв.: Exploring Scrum, The Fundamentals. – Прим. ред.

⁶ 3Back, Scrum3.0, Web, 2017. <https://3back.com/scrum30/> – Прим. авт.

1.2 Agile-движение

1.2.1 Agile-манифест

Термин *Agile*, тесно связанный со Scrum, появился в сфере разработки ПО в документе под названием «Agile-манифест».

Манифест опубликован в начале 2001 года и дошел без изменений до наших дней. Он выражает позицию по отношению к громоздким и бюрократическим процессам, которые были в моде в то время (а иногда даже сегодня).

Эта позиция отражается в формулировке *не отрицающая важности того, что справа, мы больше ценим то, что слева*. основополагающие ценности манифеста следующие:

- ✓ Люди и взаимодействие важнее процессов и инструментов.
- ✓ Работающий продукт важнее исчерпывающей документации.
- ✓ Сотрудничество с заказчиком важнее согласования условий контракта.
- ✓ Готовность к изменениям важнее следования первоначальному плану.

Публикация Agile-манифеста вызвала громкую положи-

тельную реакцию и дала старт новой моде: упрощению процессов.

Помимо четырех основных ценностей, Манифест содержит двенадцать принципов. Ниже я цитирую первый, ключевой:

Наивысшим приоритетом для нас является удовлетворение потребностей заказчика, благодаря регулярной и ранней поставке ценного программного обеспечения.

Scrum стоит под знаменем Agile-манифеста. Два его основателя также входят в число 17 подписавших документ.

Манифест определяет универсальные ценности и принципы, но способ их реализации в проектах различается. В поисках необходимого для конкретной ситуации подхода понадобятся так называемые практики.

Scrum предлагает несколько четко определенных практик, с которыми мы познакомимся дальше. Но в семье Agile существует и много других практик, которые можно использовать со Scrum.

1.2.2 Agile-практики

Практика – это конкретный и проверенный подход, который позволяет решить одну или несколько общих задач или улучшить способ работы во время

разработки.

Практики, описываемые как *гибкие*, или *Agile-практики*, существовали и до Манифеста, а некоторые – задолго до него.

К примеру, еще в начале 1980-х сообщество разработчиков программного обеспечения рекомендовало совершать короткие циклы разработки (так называемая итеративная разработка).

Некоторые практики появились вместе с Agile-движением и со временем стали незаменимыми, многократно подтверждая свою эффективность. Среди них можно отметить ежедневные совещания (*схватки*, или собственно Scrum).

Практики эффективны как по отдельности, будучи внедренными в последовательную структуру Agile-подхода, так и дополняя друг друга. В частности, структура Scrum состоит из десятка разных практик. Именно поэтому Scrum часто называют *Agile-методом* (как можно прочитать в подзаголовках предыдущих изданий этой книги).

1.2.3 Agile-методика

Scrum и Экстремальное Программирование (XP) существовали и до Манифеста. После его подписания и публикации они стали считаться Agile-методами, в то время как другие попали в список отсталых и утратили популярность. С

недавнего времени к семье Agile присоединился Kanban ⁷.



Рисунок 1.4 – Быть гибким с Agile-методикой

⁷ Kanban появился среди Agile-методов совсем недавно; на самом же деле Lean Software, где зародился Kanban, опирался на фундамент, заложенный еще производственной системой заводов Toyota в 1950-х годах. – Прим. авт.

По статистике проводимых год за годом опросов, Scrum – самый популярный из трех. Scrum и Agile – слова, которые мы часто произносим в речи и складываем друг с другом, будь то предложение о работе или статья об Agile Scrum. *Но мы убедились, что Scrum на самом деле не метод* ⁸.

1.2.4 Agile-режим

Раньше крупные компании часто разрабатывали собственные методы работы. Сейчас это происходит реже, но некоторые утверждают, что во время разработок в формате Scrum переходят в Agile-режим, что означает их бóльшую гибкость.

Однако Scrum потерял смысл в этом Agile-режиме разработки программного обеспечения ⁹.

1.2.5 Agility

• За пределами информационных технологий

Agile-манифест объединил в себе Agile-методы и породил целое Agile-движение, сильно набравшее обороты за последние годы. Scrum теперь известен и распространен во многих

⁸ Kanban также не является Agile-методом, это скорее метод улучшения процесса или метод управления потоком. В конце концов, остается только Экстремальное Программирование, которое можно назвать методом. – *Прим. авт.*

⁹ Как и не стоит обращаться к Scrum, чтобы войти в Agile-режим. – *Прим. авт.*

организациях, пройдя путь от первых адептов до большинства крупных компаний, которые им заинтересовались.

Scrum берет начало в разработке ПО, но сейчас широко используется и за пределами ИТ:

- ✓ в маркетинге и торговле;
- ✓ в сфере цифровых технологий;
- ✓ при разработке оборудования (hardware) и систем, включающих программное обеспечение.

Марк Андрессен, основатель Netscape, произнес известную фразу: *софт пожирает мир*.

У меня сложилось впечатление, то же самое происходит с процессами: методы, ведущие начало от разработки ПО, проникают повсюду. Наличие программного обеспечения во всем, что окружает нас в повседневной жизни, только ускоряет распространение методов обработки данных во всех сферах жизнедеятельности.

Scrum не относится ни к какой конкретной области и может быть использован для разработки продуктов или услуг совершенно любого характера.

• Agile-менеджмент

Управление меняется. Новые подходы¹⁰ идут в ногу с Agile-принципами.

Тем не менее, все инициативы, претендующие на назва-

¹⁰ Steven Denning, Radical Management. – Прим. авт.

ние *Agile-менеджмент*, исходят не от движения, появившегося вслед за Манифестом. Сейчас самое время попытаться определить, что такое Agility.

• **Определение Agility**

Джим Хайсмит, один из подписантов Манифеста, так определяет Agility по отношению к изменениям:

Agility – это способность отвечать на изменения для того, чтобы процветать в непрерывно меняющейся экономической обстановке.

Мне кажется интересным добавить в это определение акцент на конечного потребителя, пользователя, на которого влияет результат проделанной работы. Вот определение, которое я использовал несколько лет назад:

Agility – это способность организации приносить максимум ценности, часто и своевременно, и адаптироваться к изменениям в своей среде.

Из этого определения ясно, что гибкость относится не только к команде, но и к организации.

• **Agile организация**

Принципы *освобожденных компаний* (liberated companies) также имеют много общего с Agile-движением с точки зрения ценностей и важности человеческого фактора.

Работа Фредерика Лалу¹¹, посвященная успешным новым организациям, выдвигает на первый план принципы *бирюзо-*

¹¹ Frederic Laloux, *Reinventing Organizations*. – *Прим. авт.*

вых организаций (teal organisations), идущих в ногу с Agile.

Scrum помогает на пути компаний к трансформации. Я знаю ряд организаций, которые начали с внедрения Scrum в рамках одной команды разработчиков и перешли к *освождению* (или становлению *бирюзовой*) всей компании.



Рисунок 1.5 – Agility – трамплин, чтобы выбраться с галеры на свободу

Попытки и устремления уже есть, но мы находимся только в начале смены всей парадигмы.

Для большинства организаций ценности и принципы Agility по-прежнему носят подрывной характер (хотя ценности при этом весьма быстро изымаются) и идут вразрез с до-

минирующей культурой.

Если организации, практикующие Scrum, не согласны с духом Agility, который несут Scrum-команды, несоответствие довольно скоро вызовет перебои. Необходим комплексный подход.

1.2.6 Вклад системологии

Agility рассматривается в масштабе всей организации, в то время как Scrum зачастую пытаются поместить в рамки одной команды.

Но Scrum не ограничивается командой: результат ее работы предназначен для людей за ее пределами, возможно, даже вне организации.

Это становится ясно на практике. Многие задачи невозможно решить внутри команды. Нужно обращаться к экосистеме, внутри которой развивается команда, в частности, к отношениям с заинтересованными сторонами.

В этом издании мы разработаем системный подход к Scrum.

Системология – способ мышления, направленный на исследование функционирования живых систем, их элементов и закономерностей.

Существует множество направлений системологии: холизм, кибернетика, теория систем и т. д.

Мы будем отдавать предпочтение пермакультуре в ее человеческом аспекте, потому что она придает большое значение устойчивости, нехватку чего мы наблюдаем в организациях. Внедрить Scrum в работу одной команды относительно легко, но дальнейшая работа может резко оборваться из-за того, что я бы назвал переворотом в экосистеме.

Пермакультура: философский системный подход к построению жизнеспособных экосистем. В нашем случае речь идет о человеческих экосистемах.

Соотнесение Scrum и понятий пермакультуры поможет нам построить экосистему, которая будет долгое время приносить пользу многим людям.

1.3 Скрам сегодня

Какова ситуация в отношении Scrum сегодня?

1.3.1 Scrum-мания

Повальное увлечение Scrum (Scrum-мания!) положило конец потенциальному соперничеству между Agile-методами. Изучение общественного мнения и веб-исследования показывают, что Scrum, без сомнения, является самым популярным.

В опросах три четверти респондентов утверждают, что используют Scrum или его варианты.

Использование Scrum стало, вероятно, наиболее распространенным явлением при разработке продуктов или цифровых услуг. Но говорить о его безупречной реализации еще рано.

Основная трудность для участников Scrum-сообщества состоит в том, чтобы убедить его пользователей не быть чрезмерно консервативными.

1.3.2 Критика Scrum

Вполне логично, что при нынешней популярности Scrum

написано много статей, критикующих его и предлагающих другие подходы.

Появляются разочарованные, которые не находят в Scrum обещанных преимуществ: довольных клиентов, отдачи от инвестиций, меньшее количество дефектов, быстрый выход на рынок, удовлетворение от работы в команде и т. д.

Другие понимают Scrum лишь частично:

✓ Пользователи, долгое время находившиеся под гнетом директора по информационным технологиям (CIO), которые обнаруживают, что Scrum приветствует изменения, и думают: теперь можно постоянно все менять.

✓ Менеджеры, которые уверены, что Agile-команда – на то и Agile, что может выполнять больше работы за меньшее время.

Реальность может их разочаровать. Мы постараемся дать ответ на эту критику и разочарования.

1.3.3 Scrum лично для вас

Scrum не является решением для всех типов разработки. Мы понимаем, что важна не столько область или используемые технологии, сколько сложность проектов и культура организаций.

✓ Если вы разрабатываете приложения из категории несложных по модели Synefin вероятно, Scrum для вас – не лучший выбор.

✓ Если ваша организация придерживается культуры контроля и вы не можете продвигать необходимые радикальные изменения, то Scrum не для вас.

✓ Если вы работаете в режиме непрекращающихся срочных задач, вы не сможете сосредоточить команду на спринте. Когда все работают в спешке, использовать Scrum не рекомендуется – в такой ситуации больше подходит Kanban. Но мы увидим позднее, что есть возможность использовать Scrum вместе с Kanban.

Scrum приветствует изменения, но не когда они постоянные. При строгом следовании принципам Scrum рабочая команда не должна отвлекаться, а состав не должен нарушаться. Изменения допустимы в конце спринта, но не во время.

Мы за изменения, но против постоянных прерываний процесса!

1.3.4 Контекстуализированный Scrum

Концепция Shu Na Ri была создана Алистером Кокберном (подписал Манифест и предложил использовать слово agile). Это модель развития, взятая из айкидо. Shu соответствует фазе обучения, когда ученик повторяет за учителем,

не пытаясь что-либо изменить.

✓ На – вторая фаза обучения. Ученик берет на себя инициативу и ответственность за то, что делает.

✓ Ri – последняя фаза обучения и овладение техникой. Ученик сам становится учителем и может создать новые, более высокие формы и принципы.

Тех, у кого проблемы со Scrum, можно, спросить: начали они с фазы Shu прежде, чем перейти к На? Сохранился при этом формат Scrum? Соблюдали они все принципы?

В противном случае существует риск исказить концепцию Scrum.

Контекстуализация без искажений – вот в чем загвоздка. Поскольку Scrum – это просто фреймворк, его нельзя использовать сам по себе, без адаптации к контексту.

Например, во время разработки ПО дополнительно пригодятся инженерные практики из Экстремального Программирования.

Мы поговорим об этом в главе «Контекстуализация Scrum».

1.4 Scrum – живой организм

1.4.1 Scrum меняется

Иногда люди думают начать работать в формате Scrum, но источники, к которым они обращаются, не особо достоверны. И новички сталкиваются с трудностями: они когда-то прочитали книгу или статью, посещали тренинг – но знания с тех пор не обновляли.

Некоторые практики, когда-то связанные со Scrum, уже устарели: Scrum постоянно развивается.

Scrum 1995 года, описанный в статье Швабера, не имеет практически ничего общего с Scrum 3.0. Если на то пошло, первое издание этой книги, написанное в 2009 году, сильно отличается от того, которое вы сейчас держите в руках.

Несмотря на простоту (или, возможно, из-за нее), Scrum по-прежнему часто неправильно понимают. Осенью 2016 года я выступил на нескольких конференциях с темой под названием «Scrum? Срам!» (см. главу 22) о борьбе с основными ошибками при применении Scrum. Среди пяти выявленных причин ошибок и недопонимания на первом месте стояло незнание того, чем на самом деле является Scrum. Все это приводит к его искажению.

1.4.2 Превосходство Scrum

Добившись успеха на уровне работы в командах, Scrum все больше начинает выходить на уровень организаций и крупных проектов.

Переход на следующий уровень, к *схватке схваток*, или крупномасштабному Scrum, поднимает много вопросов. Проявляются и точки пересечения с практиками, принятыми в организациях в других сферах деятельности.

Мы рассмотрим их в основном во второй части этой книги с точки зрения системологии.

1.4.3 Языковой аспект

Знакомство с новой культурой начинается с новых слов. И Scrum имеет свой язык.

Чтобы помочь разобраться в значении тех или иных терминов, я добавил глоссарий в конце книги. Много неологизмов пришли из английского языка. При распространении Scrum во Франции некоторые утвердились в английском варианте, другие же были переведены.

Словарь значительно расширился за десять лет и продолжает развиваться.

• Не переведенные термины

Такие слова, как, например, спринт, уже давно нашли место во французском словаре. И в английском, и во французском они означают одно и то же.

В этой книге я не переводил ряд английских терминов, связанных со Scrum/Agile. Это не значит, что не было попыток их перевода. По большей части перевод терминов был предложен, но не принят пользователями.

В этом списке следующие слова: бэклог, Scrum-мастер, эпик. Я объясняю, почему эти термины не переведены, в соответствующих главах ¹².

• Переведенные термины

Другие термины Scrum используются в переводе. Некоторым такое использование важно, но, к сожалению, как мне кажется, это никак не закреплено правилами. В повседневной речи, а иногда даже в письменном виде можно встретить: definition of done, sprint planning meeting, sprint review, impediment. Я трепетно защищаю использование французского языка, поэтому, соответственно: критерии завершенности, встреча по планированию спринта, обзор спринта, препятствие.

Иногда появляются смешанные термины, например, приемочный тестинг, а не приемочное тестирование.

¹² Тем не менее, калькированный перевод этих слов на французский есть в Квебеке – *Прим. авт.*

Словарь развивается с помощью терминов, которые приобретают все большее значение. Я имею в виду, в частности, такие слова как stakeholder или backlog refinement, для которых рекомендую использовать *заинтересованные стороны* и *доработка бэклога*. В этой редакции книги больше не говорю feature – теперь это *функциональность*.

Поскольку французский является живым языком, можно полагать, что некоторые слова из сферы Scrum, как, например, бэклог, войдут в состав языка. Люди задумаются, какого это слово рода, почему не женского? Среди таких неологизмов есть слово *аджайлист* для адепта Agile-методов.

Мы не используем ни скраммер, ни скрамист, когда говорим о приверженцах Scrum. Однако таких много.



Настраиваемся на Scrum!

Самое важное

Находясь под знаменем Agility и Agile-манифеста, Scrum предлагает простой способ разработки сложных продуктов. Однако за его кажущейся легкостью скрывается огромная мощь, способная во много крат увеличить эффективность используемых практик разработки.

И как мы еще увидим, если фреймворк простой, то его реализация требует больших усилий, а его влияние распространяется на всю культуру организации.

Какой вопрос надо себе задать?

Почему мы хотим использовать Scrum? Какие проблемы мы хотим решить с его помощью?

С чего начать?

С Shu.

К прочтению

«Руководство по Scrum» («The Scrum Guide»), 2017

Чтобы идти дальше

Книги ¹³

- Steven Denning, *Radical Management*, Jossey & Bass, 2010

¹³ Здесь и в последующих главах названия книг, не переведенных на русский язык, и имена их авторов приводятся на языке оригинала. – *Прим. ред.*

2

Разделение процесса на спринты

Я принял участие в десятках проектов в качестве разработчика или консультанта, и ни один из них не был похож на предыдущий, хотя используемые процессы и методы иногда совпадали.

Проекты по-разному развиваются с течением времени, и у каждого – свой собственный **цикл разработки** (или жизненный цикл). Цикл составляют **стадии и контрольные точки**. Стадии следуют друг за другом, а контрольные точки определяют переход к следующему этапу. Стадия преследует определенные цели. Контрольная точка служит для проверки того, что данный набор целей достигнут.

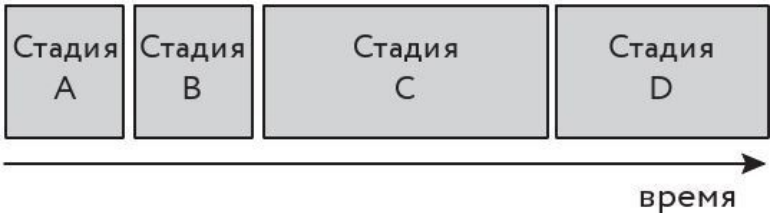


Рисунок 2.1 – В традиционном цикле стадии различаются

2.1 Изменение парадигмы

Течение цикла зависит от используемой **модели** (или процесса). Во Франции по-прежнему распространена **V-модель**, но компании, особенно крупные, обычно берут ее за основу для дальнейшей адаптации к их контексту и создания уже своей собственной модели.

В некоторых компаниях применение моделей является не просто рекомендуемым, а обязательным, в то время как в других организациях командам предоставляется больше возможностей.

И все же я часто отмечаю большой разрыв между моделью и ее реализацией в проектах, вне зависимости от того, была она рекомендована или навязана команде.

Этому есть причины:

✓ **Модель разработана методологами-теоретиками** и оказывается слишком удалена от реальности и неприменима на практике.

✓ **Контрольные точки оказываются неэффективными**, потому что для осуществления проверок и тестирования необходимо обратиться к огромному множеству документов, некоторые из которых – в сотни страниц.

✓ **Команда пропускает контрольные точки** – и накапливает работу, не выполненную на предыдущих этапах.

Это создает неудобства в дальнейшем.

Все это давно известно.

В начале 1980-х годов была предложена альтернатива итеративной и поэтапной разработке, чтобы избежать подобных ситуаций. Эта идея была реализована частично и без привязки к Agility: одни команды быстро производят прототипы, другие их реализуют.

В то же время возникла противоположная идея индустриализации процессов, и пришлось подождать, пока она провалится на практике.

Scrum и Agile-методы взяли предшественников за точку отсчета и пошли дальше с моделью цикла разработки, основанной на последовательном повторении одной стадии. В Scrum эта стадия называется *спринт*.

Спринт с точки зрения времени – повторяющаяся стадия фиксированной продолжительности.



Рисунок 2.2 – Повторение спринта

Все спринты протекают по одной схеме. В этом основное

отличие от традиционных подходов, где каждый этап предполагает работу разного характера.

Еще одно фундаментальное отличие заключается в том, что Scrum – это не более чем фреймворк. Он не определяет наполнение каждого спринта: за это отвечает команда.

2.2 Итеративный и инкрементальный подход

Scrum основывается на итеративном и инкрементальном подходе к разработке продукта. Давайте разберемся, что это значит.

2.2.1 Инкрементальная разработка

Инкрементальная, то есть поэтапная разработка – стратегия, при которой развитие продукта отмечается в конце каждого спринта. Инкрементальный подход позволяет создавать продукт по частям, где каждая новая часть добавляется к предыдущей.

Это контрастирует с традиционным подходом к проекту, где до завершения разработки ничего не готово. Хотя ничего – это тоже неправильно, так как появляются документы, созданные во время промежуточных контрольных точек. Но этого недостаточно, чтобы избежать туннельного эффекта.

Пока я писал эту книгу, использовал инкрементальный подход: сделал изначальный план (в виде карты мыслей), а затем писал главу за главой, не ориентируясь на порядок в плане.

Использование инкрементального подхода широко рас-

пространено в компаниях, которые занимаются разработкой продуктов. Часто можно слышать о поставках (то есть, частях системы, или *инкрементах*), которые значатся в контрактах с клиентами. Обычно такое деление на части носит технический характер, и ничего не работает до момента, когда все инкременты не состыкуются в единое целое.

На третьем спринте раздадим команде парашюты!



Рисунок 2.3 – Густав Эйфель, инженер, практиковавший инкрементальный подход

2.2.2 Итеративная разработка

Итерация – это процесс повторения. В математике и физике итерация представляет собой повторяющийся вычислительный процесс, который позволяет, например, решить уравнение путем последовательных приближений.

В сфере разработки ПО термин *итерация* используется для обозначения периода времени, в течение которого выполняются действия, которые будут повторяться в следующих итерациях. Этот термин часто соотносят с процессом.

Итеративный процесс позволяет проанализировать, что было сделано – с целью дальнейшего улучшения или завершения работы. Он основывается на идее, что трудно (если не невозможно) преуспеть в чем-то с первого раза. Обратная связь, собранная по результатам одной итерации, помогает вносить улучшения в следующую. Итерации прекращаются, когда достигнутый результат оценивается как удовлетворительный.

Пока я писал эту книгу, я также использовал итеративный подход: отправил первый черновик первому рецензенту. Благодаря обратной связи, я выпустил новую, улучшенную версию книги, которую отправил следующему рецензенту, и так далее.

2.2.3 Итеративный и инкрементальный подход

Scrum объединяет оба подхода с концепцией спринта:

- ✓ в конце спринта появляется инкремент продукта;
- ✓ обратная связь на этот инкремент позволяет скорректировать цель следующего спринта.

Другими словами, спринт – это итерация, в результате ко-

торой появляется новый контент (инкрементальный подход) и улучшается контент, добавленный в предыдущий инкремент (итеративный подход).

Пока я писал эту книгу, я использовал итеративный и инкрементальный подход: не рассылал черновик всей книги своим читателям, но делал это глава за главой. На протяжении какого-то времени я работал над первой версией одной главы одновременно с пересмотром другой главы после получения на нее обратной связи от одного или нескольких читателей.

2.2.4 Научный двухфазный подход

Ученые знают, как бороться с неопределенностью в сложных задачах при помощи подхода: сначала предлагается гипотеза, затем она проверяется, и после наблюдения за результатом решается, принять ее или отклонить.

По сравнению с простым итеративным подходом, когда запрашивается обратная связь, а затем принимается или отклоняется в соответствии с субъективным суждением, в научном методе все основывается на практической проверке. Если мы используем этот подход для разработки какой-либо функции продукта, наблюдение после эксперимента позволяет либо убедиться в ее ценности и продолжать над ней работать, либо понять, что она не эффективна.

Lean Startup популяризировал эту концепцию тестирова-

ния гипотез и коротких спринтов для получения информации о новом продукте. Мы все постоянно слышим термин MVP (Минимальный жизнеспособный продукт, *Minimum Viable Product*), который, на мой взгляд, плохо отражает его суть. Ведь речь идет не о продукте, пусть даже с минимальным количеством функций, но о результате, который позволяет проверить предположения, и, следовательно, узнать новую информацию.

Если гипотеза не подтверждается, команда переключается на другую функцию продукта.

Достижение MVP соответствует окончанию стадии исследования и началу стадии эксплуатации.

С точки зрения жизненного цикла продукта, Lean Startup используется на стадии исследования, а Scrum – на стадии эксплуатации.



Рисунок 2.4 – Две стадии разработки продукта

Концепция Lean Startup предполагает малозатратный по ресурсам этап проверки гипотезы несколькими конечными пользователями.

Представим, что эта книга была опубликована в Интернете, глава за главой. Я мог бы предположить, что эта глава интересна, ориентируясь на три покупки в течение недели после публикации ее плана. К концу недели я мог бы проверить свою гипотезу, прежде чем приступить к редактированию главы (а так как моя книга опубликована классическим путем, через издательство, я буду опираться на реакцию избранных рецензентов).

На стадии использования также можно использовать данный прием, проверяя на практике гипотезу о части продукта (см. главу о бэклоге).

2.3 Спринт

2.3.1 Четкий ритм

Помимо итеративного и инкрементального процесса, какие еще можно выделить характеристики Scrum-спринта?

- ✓ **Короткие итерации:** продолжительность спринта варьируется от недели до месяца.
- ✓ **Строгая последовательность:** спринты не перемешиваются между собой.
- ✓ **Четкий ритм:** спринты имеют одинаковую продолжительность.

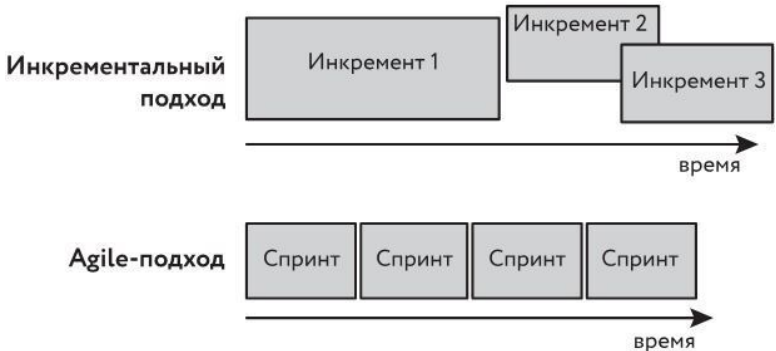


Рисунок 2.5 – Разница между инкрементальным и Agile-

Продолжительность спринтов всегда одинакова, что помогает команде держать ритм.

Внимание: надо держать ритм! Это позволяет избежать перегрузки. Необходимо помнить, что команда не сможет перерабатывать очень долго. Заставлять ее выходить за пределы своих возможностей означает снижать качество ее работы и подрывать здоровье ее участников: увеличивается количество ошибок, уменьшается мотивация, инженерные практики пренебрегаются.

2.3.2 Отрезок времени

Спринт соответствует определенному временному отрезку. После начала спринта команда не может менять дату его окончания, даже если она уже достигла поставленной цели!

Почему так? Такой подход позволяет избежать *почти законченного* синдрома (минутку, я почти закончил!), из-за которого дата окончания спринта может быть неоднократно отодвинута.

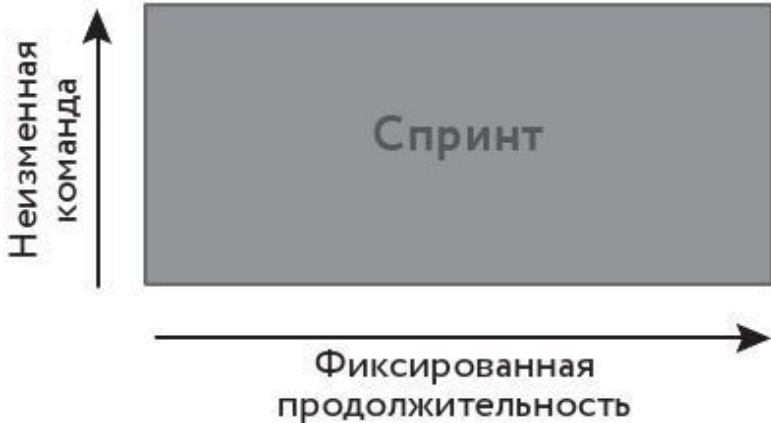


Рисунок 2.6 – Отрезок времени

Помню, как еще до появления Scrum команды договаривались о дате контрольной точки. Когда они решали отложить обзор на два-три дня, по истечении этого срока оказывалось, что им нужно еще немножко времени на работу...

Концепция временных отрезков устраняет любые отсрочки: в запланированную дату команда проводит объективный обзор ее прогресса, а затем корректирует содержание следующих спринтов.

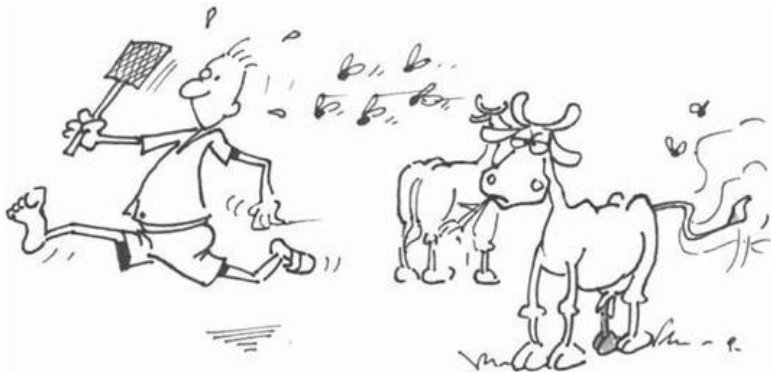


Рисунок 2.7 – Стая слепней помогает обрабатывать спринтерский бег на коротких дистанциях

2.3.3 Продолжительность спринта

Когда Scrum только появился, стандартной продолжительностью спринта был один месяц. Теперь видна тенденция сокращения спринтов до двух-трех недель. Если мы говорим о разработке ПО, то инженерные практики – непрерывная интеграция и развертывание – позволяют чаще выпускать новые версии продукта. Время, затраченное на развертывание версии, больше не является проблемой.

Продолжительность спринта кратна неделе: спринт не может длиться 13 или 17 дней. Так намного проще ориентироваться.

По части продолжительности. Здесь нет единого мнения,

все зависит от контекста. Но она должна быть четко зафиксирована.

Результаты опроса, проведенного в мае 2015 на странице моего блога, показывают, что чуть меньше половины (44 %) команд проводят спринты продолжительностью в две недели, и почти для четверти (24 %) команд спринты составляют три недели.

Чтобы определить продолжительность спринта, существует несколько критериев, на которые нужно обратить внимание:

- ✓ Производительность и результативность команды.
- ✓ Готовность заинтересованных сторон давать обратную связь.
- ✓ Простота в подготовке событий спринта – спринт предполагает дополнительную работу по организации событий.
- ✓ Частота изменений – спринт представляет собой период стабильности. Когда начинается спринт, команда не должна отвлекаться на другие вещи, а состав команды не должен меняться. Любые изменения переходят на следующий спринт.

Определение продолжительности спринта является одной из задач начинающей команды (см. главу 13).

2.3.4 События спринта

Цикл разработки – это последовательность стадий, каждая из которых отмечена определенными процессами. Для разработки программного обеспечения процессы обычно следующие: написание спецификации, архитектура (проектирование), написание кода, тестирование (интеграция и приемка). Для упрощения я буду использовать буквы С, А, К и Т соответственно для обозначения этих действий на схемах.

В Scrum каждый спринт заканчивается рабочим результатом. Все процессы проводятся во время одного спринта.



Рисунок 2.8 – Спринты и процессы в параллели

Внутри спринта процесс не последовательный (С, затем А, затем К, затем Т). Преобладающая идея – идея непрерывного потока, прерванного только в конце спринта.

Другой вариант – это так называемый последовательный

цикл, разворачивающий последовательные процессы, по одному процессу на стадию.



Рисунок 2.9 – Последовательные процессы

При последовательном процессе для каждой стадии определяется цель, сформулированная списком документов к созданию. Стадия длится несколько месяцев, ее продолжительность варьируется: команда останавливается, когда цели достигнуты. Результатом работы считается документация, так как готовый продукт получается только в конце.

Следовать этому подходу буквально означает:

- ✓ 100 % готовность спецификации до перехода к проектированию.
- ✓ 100 % готовность проекта до перехода к написанию кода.
- ✓ 100 % готовность кода до перехода к тестированию.

Это утопичная модель. В реальности команда всегда возвращается к предыдущим стадиям: на стадии тестирования команда возвращается к написанию спецификации – и так далее.

Такое различие между моделью и ситуацией в реальной жизни является одной из причин неудач проектов по V-модели и получаемых в их результате продуктов среднего качества.

Agile-подход более прагматичен: работа над спецификацией и проектированием непрерывна. Архитектура развивается с новыми функциями.

На другом полюсе последовательного подхода есть команды, которые не ищут легких путей: в их разработке нет стадии написания спецификации или проектирования. Они сразу приступают к написанию кода, затем следует длительный период тестирования и исправления ошибок.

Scrum в этом плане сильно отличается. Здесь качество играет важную роль, и тестирование является частью каждого спринта. Для разработки программного обеспечения крайне важно проводить тестирование параллельно с написанием кода, чтобы как можно скорее обнаружить аномалии. Это также позволяет понять, есть ли необходимость в упрощении, если тестирование кажется слишком сложным.

Опытная Agile-команда в каждом спринте проводит 100 % тестов до того, как будет написано 100 % кода, и достигает 100 % готовности кода до того, как на 100 % написана спецификация.

2.4 Результат спринта

2.4.1 Несколько дефиниций

Начнем с того, что определим смысл терминов, которые будем в дальнейшем использовать.

✓ **Ввод в эксплуатацию** – это создание ценности путем предоставления пользователям версии продукта ¹⁴.

✓ **Развертывание** – это установка версии с целью получения какой-либо информации. Для ввода продукта в эксплуатацию необходимо его развертывание, в то время как для развертывания можно не вводить в эксплуатацию.

✓ В конце каждого спринта команда получает **результат спринта**.

Руководство по Scrum называет результат спринта *потенциально готовым к поставке инкрементом*. Это выражение часто неправильно понимают: инкремент – это не всегда продукт. В Scrum 3.0 речь просто о результатах спринта, и я буду придерживаться того же мнения.

✓ Почему речь не о продукте? Как мы увидим в сле-

¹⁴ Также используется английское слово «release» – *Прим. авт.*

дующей главе, структурирующим элементом является скорее **команда**, чем продукт. Команда может работать над несколькими продуктами, последовательно или одновременно.

2.4.2 Спринт и ввод в эксплуатацию

Что насчет спринта? Включает ли последовательность процессов поставку конечным пользователям? Другими словами, всегда ли конец спринта соответствует вводу в эксплуатацию? Или для этого нужно ждать несколько спринтов?

Цель состоит в быстром предоставлении ценности. Решение о вводе в эксплуатацию основано на данной цели и не всегда совпадает с окончанием спринта. Это может быть во время спринта для одной команды или после нескольких спринтов для другой команды.

Ввод в эксплуатацию делается как можно раньше, чтобы поскорее предоставить ценность пользователям. Он не обязательно совпадает с концом спринта.

Течение и результат спринта, как правило, облегчают принятие решения. Например, в конце данного спринта принимается бизнес-решение о вводе в эксплуатацию в рамках следующего спринта.

Таким образом, ввод в эксплуатацию не связан со спринтом.

2.4.3 Спринт и развертывание

Что делать с результатом спринта, если не вводить его в эксплуатацию для конечных пользователей? Можно направить его заинтересованным сторонам для осуществления следующих целей (впрочем, эти варианты не единственные):

Конец ознакомительного фрагмента.

Текст предоставлен ООО «ЛитРес».

Прочитайте эту книгу целиком, [купив полную легальную версию](#) на ЛитРес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.