ГЛАВА 4

Scrum и самоорганизующиеся команды

Великие принципы, не рождающие действия, являются всего лишь эфемерной субстанцией. И наоборот, конкретные практики при отсутствии руководящих принципов часто используются неадекватно.

Джим Хайсмит*

Лозунг настольной игры «Отелло» — «Минута на обучение, вся жизнь на совершенствование». Это очень подходит команде, которая учится Scrum. Базовые практики и ценности Scrum просты и легки в применении. Но понять, как они превратятся в результат — лучшее программное обеспечение, — может оказаться непросто.

Правила Scrum просты и легки для понимания, что позволяет многим командам, применяющим agile-методологии, использовать его в качестве отправной точки. Вот основная схема scrum-проекта.

— В scrum-проекте существует три основные роли: **владелец продукта, scrum-мастер и член команды.** (Мы придаем особую важность понятиям «владелец продукта» и «scrum-мастер», когда речь идет о scrum-ролях.)

— Владелец продукта работает с остальной частью команды, чтобы поддерживать и определять приоритеты функций и требований продуктового бэклога, которые необходимо реализовать.

* Jim Highsmith. Agile Project Management: Creating Innovative Products (Upper Saddle River, NJ: Pearson Education, 2009).

Почитать описание, рецензии и купить на сайте МИФа

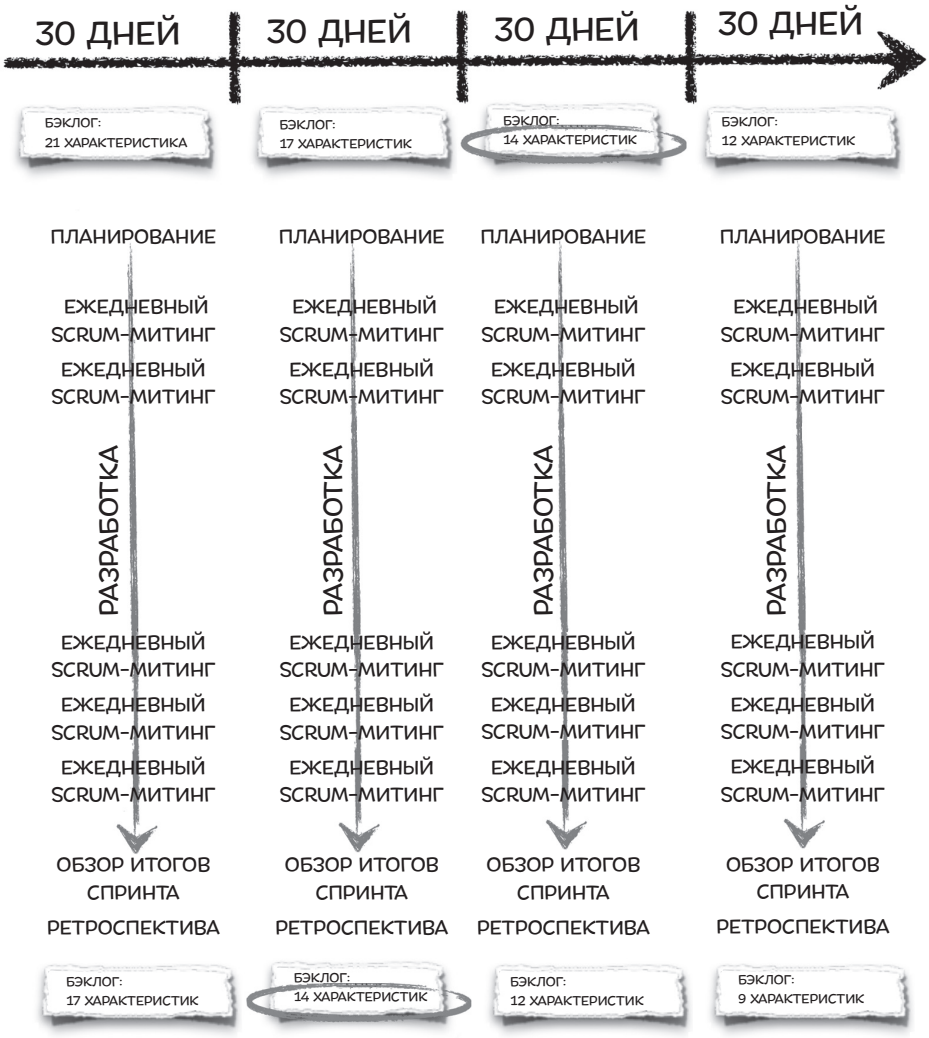

Рис. 4.1. Базовая схема Scrum

— Программное обеспечение строится с использованием ограниченных по времени итераций, называемых **спринтами.** В начале каждой такой итерации команда выполняет **планирование спринта**, чтобы определить, какие функции из бэклога они будут реализовывать. Это называется **бэклог спринта.** На протяжении спринта команда работает над созданием всех тех функций, которые в него вошли.

— Ежедневно все члены команды участвуют в короткой встрече (**Daily Scrum — ежедневный scrum-митинг)**, чтобы рассказать друг другу

Почитать описание, рецензии и купить на сайте МИФа

о достижениях и обсудить то, что препятствует дальнейшей работе. Каждый человек отвечает на три вопроса: что я сделал с момента последнего ежедневного совещания? Что буду делать вплоть до следующего ежедневного совещания? Какие препятствия есть на моем пути?

— Scrum-мастер поддерживает правильное направление работы над проектом, устраняет препятствия на пути команды и помогает ей, если есть просьбы о помощи. В конце спринта работающее ПО показывают владельцу продукта и стейкхолдерам проекта. Команда проводит одновременно **обзор итогов спринта** и **ретроспективный** обзор, чтобы выяснить ошибки. Таким образом удается улучшить сам процесс спринта и качество создания программного продукта в будущем.

Чтобы стать успешной, scrum-команде нужно делать больше, чем просто придерживаться базовой схемы Scrum. Эффективная scrum-команда должна быть **самоорганизующейся.** В это понятие мы вкладываем тот же смысл, что и Кен Швабер в своей книге Agile Project Management with Scrum (обратите внимание на слова, которые мы выделили курсивом).

> Чтобы работать по Scrum, команда должна глубоко и на интуитивном уровне понимать *коллективную ответственность и самоорганизацию.* Scrum-теорию, методы и правила легко понять. Но пока группа, состоящая из отдельных людей, не взяла на себя коллективное обязательство за создание чего-то материального в течение ограниченного промежутка времени, *эти индивидуумы, пожалуй, не овладеют Scrum*. Когда же члены команды перестают действовать как многие и принимают на себя обязательства за общие цели, команда становится способной к самоорганизации, может быстро пройти через сложности и на практике реализовать планы.

Цель этой главы — помочь вам «заполучить Scrum», опираясь на идеи из глав 2 и 3, то есть научить вас практикам и моделям этой методологии. Мы будем использовать эти практики, чтобы продемонстрировать идеи, стоящие за принципами коллективной ответственности и самоорганизации.

Правила Scrum

В книге Кена Швабера Agile Project Management with Scrum изложены правила Scrum, описывающие основной шаблон scrum-проекта. Вы можете бесплатно скачать правила Scrum на сайте www.scrum.org в формате PDF — это электронная книга The Scrum Guide, написанная Кеном Швабером

Почитать описание, рецензии и купить на сайте МИФа

и Джеффом Сазерлендом. Эти люди создали Scrum и помогли распространить его влияние на остальные направления программирования. Правила должны быть вам хорошо знакомы, потому что многие из них описаны в главах 2 и 3. Вот каким правилам следует типичный scrum-проект.

— Каждый спринт начинается с планирования, в котором участвуют scrum-мастер, владелец продукта и остальные члены команды. Встреча разделена на две части, по четыре часа каждая. Владелец продукта заранее выделяет приоритетные направления продуктового бэклога, который состоит из набора позиций, выбранных для разработки потребителями и заинтересованными сторонами. В первой части встречи владелец с командой выбирает элементы бэклога, которые будут поставлены в конце спринта, исходя из их значимости и оценки командой объема работ. Команда обязуется провести презентацию работающего программного обеспечения, включающего все эти элементы бэклога в конце спринта. На это тратится половина отведенного для планирования времени (для 30-дневного спринта это четыре часа, а для более коротких время пропорционально сокращается). В результате команда фиксирует элементы бэклога, одобренные командой на этапе планирования, и использует их в качестве бэклога спринта. Во второй части встречи члены команды (при помощи владельца продукта) знакомятся с индивидуальными заданиями, которые они будут выполнять для фактической реализации этих элементов.

Длина этой части встречи также зависит от длины спринта (но часто занимает меньше времени, чем другая). По окончании планирования выбранные участниками элементы становятся бэклогом спринта.

— Команда проводит scrum-митинг каждый день. Все члены команды (включая scrum-мастера и владельца продукта) обязаны* в них участвовать. Заинтересованные лица также могут присутствовать (но только в качестве наблюдателей). Встреча длится не больше 15 минут, поэтому все члены команды должны приходить вовремя. Каждый из них отвечает на три вопроса: что я сделал с момента прошлой встречи? Что буду делать с сегодняшнего дня и до следующей встречи? Какие препятствия есть на моем пути? Все члены команды должны быть кратки-

* Не кажется ли вашей команде, что ежедневные встречи — это нереально? Может быть, вы так думаете, потому что люди распределены между командами или имеются другие обязательства? Вы уже начали искать альтернативные варианты, например заменить ежедневные scrum-митинги на онлайн-собрания или вики-страницу? Это может быть показателем того, что вам необходимо изменить свой образ мыслей, чтобы получить наилучшие результаты от Scrum.

Почитать описание, рецензии и купить на сайте МИФа

ми. Если ответ требует обсуждения, то ответственные за составление графика немедленно организуют дополнительную встречу.

— Продолжительность любого спринта определяется на совместной встрече. Многие команды планируют спринт на 30 календарных дней, но возможны варианты — некоторые команды выбирают двухнедельные периоды. Во время спринта команда работает над задачами, вошедшими в бэклог спринта. Разработчики могут получить помощь от участников проекта, не входящих в команду, но они не имеют права указывать команде, как надо работать, и должны доверять ей. Если любой член команды в середине спринта обнаружит, что требуется добавочное время или дополнительные элементы бэклога, то, как только станет ясно, что спринт в опасности, нужно уведомить об этом владельца продукта. Владелец продукта — это член команды, который работает с пользователями и стейкхолдерами и оповещает их о ходе проекта. Он использует полученную информацию для коррекции спринта, чтобы он соответствовал реальным возможностям команды. Если же команда считает, что закончит работу до окончания спринта, то она может добавить в бэклог дополнительные позиции. Команда должна актуализировать состояние бэклога спринта и предоставлять к нему доступ остальным членам проектной группы. В экстренной ситуации владелец продукта может завершить спринт раньше и начать планирование нового спринта. Так бывает, если команда понимает, что не может предоставить работающее ПО (например, возникают серьезные технологические, организационные или кадровые проблемы). Но каждый должен знать: прекращение спринта случается редко и имеет крайне негативные последствия для производительности, а также с точки зрения доверия со стороны пользователей и стейкхолдеров.

— В конце спринта команда проводит его обзор, где демонстрирует работающее программное обеспечение для пользователей и заинтересованных сторон. Демоверсии могут содержать только те элементы бэклога, которые фактически были *сделаны** (то есть команда закончила все работы по созданию этих пунктов и владелец продукта принимает их как готовый продукт). Команда может представлять только функционирующее ПО, а не промежуточные элементы (схемы архитекту-

* То, что фактически сделанная работа — та, которая действительно выполнена полностью, и нет ничего, что нужно доделывать, интуитивно понятно. Но существует много нюансов. Это еще один пример упрощения, и мы будем возвращаться к этой концепции несколько раз на протяжении всей книги. Будьте внимательны, следите за ходом мысли! Теперь, когда вы видели некоторые из сделанных упрощений, мы не будем помечать новые упрощения сносками. Но мы все еще будем использовать этот прием в книге.

Почитать описание, рецензии и купить на сайте МИФа

ры, базы данных, функциональные спецификации и т. д.). Заинтересованные стороны имеют право задавать вопросы, на которые команда должна ответить. По окончании демонстрации стейкхолдеров просят высказать свое мнение. Если необходимо внести изменения, то это учитывается при планировании следующего спринта. Владелец продукта может добавить изменения в продуктовый бэклог, и если они должны быть выполнены немедленно, то их в итоге внесут в следующий бэклог спринта.

— По окончании спринта команда проводит ретроспективу (чтобы понять, как улучшить свою работу), где присутствует и scrum-мастер, и, при необходимости, владелец продукта. Каждый участник отвечает на два вопроса: что было сделано хорошо во время спринта? Что можно улучшить? Scrum-мастер отмечает любые улучшения, а также отдельные пункты (монтаж нового сервера сборки, внедрение нового метода программирования или перепланировка офиса и т. д.) и добавляет их в список продуктового бэклога как нефункциональные элементы.

Вот и все! Ничего сложного.

Но если это так просто, то почему все мы не используем Scrum?

Почему же такой значительной части нашей команды, работающей со Scrum и соблюдающей все правила, кажется, что получаемый ими результат — это всего лишь немногим лучше, чем ничего? Чего не хватает?

Описание: команда, разрабатывающая приложение для мобильного телефона в небольшой компании

Роджер — руководитель команды, пытающийся использовать Agile

Ави — владелец продукта

Эрик — scrum-мастер в другой команде

Акт I. Я могу использовать Scrum?

Hover Puppy Software — небольшая компания, создающая сайты и приложения для мобильных телефонов. Она пользуется популярностью. Шесть месяцев назад выпущенное ими приложение для мобильного телефона принесло огромную выручку, и CEO решил вложить деньги в запуск нового

Почитать описание, рецензии и купить на сайте МИФа

сайта Lolleaderz.com, который позволяет пользователям создавать рейтинги любимых видеороликов.

В начале проекта его руководитель Роджер хотел использовать Agile. Он был по-настоящему счастлив, когда узнал, что команда рада такой перспективе. Разработчики мобильных телефонов в другой компании уже использовали Scrum для создания очень популярного приложения, и идея получила там широкое распространение. Один из членов команды стал называть Роджера scrum-мастером, и эта роль закрепилась за ним.

Первым делом Роджер начал искать владельца продукта, но нашел его не сразу. К счастью, Hover Puppy — небольшая компания, где сотрудники называют директора по имени. Роджер просто подошел к нему и объяснил ситуацию, описывая владельца продукта как «короля заинтересованных сторон» проекта. Примерно в то же время один из менеджеров, Ави, только что закончил проект. CEO познакомил его с Роджером, объяснил, что полностью поддерживает Scrum, и оставил их вдвоем — выяснять, как лучше организовать работу над проектом.

Поначалу все шло хорошо. Роджер установил продолжительность каждого спринта в один месяц, а Ави придумал бэклог задач для разработки. Роджер организовал ежедневные scrum-митинги, Ави занес их в свой календарь, чтобы видеть их каждый день. Первый спринт прошел отлично: все двигались вперед вместе, и команда добилась определенных успехов. В конце спринта команда предоставила Ави демоверсию сайта с несколькими функциями, которые они вместе планировали. Казалось, что эксперимент со Scrum удался.

В течение следующих нескольких недель в проекте начали появляться трещины, но он все еще держался на плаву. Один менеджер по обслуживанию клиентов настраивал показ рекламы нового блокбастера своего клиента, кинокомпании, чтобы он происходил при каждом наведении мышки на сайт Hover Puppy. Команда пообещала Ави, что предоставит ему демоверсию этой функции в конце спринта. Но на раннем этапе разработки возникли технические проблемы, и пришлось перенести эту задачу на следующий спринт. Роджер объяснил, что scrum-команды всегда поставляют работающее ПО, а если создание функции еще не закончено, то задача переносится на следующий спринт. Но они не были уверены, что именно так должен работать Scrum.

Работоспособность команды снижалась с каждым спринтом. К концу третьего спринта Ави почувствовал, что в основном занимается командой, поэтому на клиентов у него остается совсем мало времени. В начале проекта ему казалось, что он контролирует работу команды. Теперь же он начал

Почитать описание, рецензии и купить на сайте МИФа

понимать, что поторопился, согласившись стать владельцем продукта в проекте Lolleaderz.com. Роджер расстроился, услышав жалобы Ави, что команде тяжело работать с другими аккаунт-менеджерами.

Но хуже всего было то, что команде на самом деле надоела вся эта scrum-возня. Прежде чем Роджер и Ави получили возможность разобраться в этой проблеме, случилась еще одна неприятность. Три разработчика высказали Роджеру свое недовольство, что им приходится ходить на ежедневные scrum-митинги. Один из них сказал: «У меня и так много работы, а эти совещания просто пожирают мое время. Почему я должен сидеть и смотреть, как вы раздаете всем задания на ближайшие дни? Разве нельзя отправить их нам по электронной почте?» Роджер пробормотал, что таковы правила Scrum, поэтому разработчики должны продолжать ходить на собрания. Такой невнятный ответ никого не удовлетворил, и Роджер уже начал задумываться, не стоит ли прислушаться к мнению разработчиков.

Крупный скандал произошел, когда Роджер планировал четвертый спринт. Ави настаивал на добавлении функции, позволяющей пользователям голосовать за понравившееся видео. Он считал, что если не сделать этого, то они начнут терять рекламодателей. Но разработка базы данных для этой функции была трудоемкой. Чтобы внести основные изменения в модель данных, требовалось привлечь высококвалифицированного администратора баз данных (DBA), а это означало, что у него не останется времени для написания хранимых процедур. Создалось впечатление, что на разработку этой функции потребуется еще неделя, но переносить ее на следующий спринт не представлялось возможным.

Миновало уже шесть месяцев, пять спринтов были пройдены. Роджер чувствовал, как Ави усложняет требования для команды разработчиков. Ави был расстроен, что не удается разработать сайт, который он мог бы продать своим клиентам. Команда предполагала закончить страницу видеотегов и социальных медиа двумя спринтами ранее, но до сих пор не сделала этого. На последней встрече аккаунт-менеджеров Ави обвинил команду в отставании от сроков. Роджер понял: проект находится под угрозой закрытия.

То, что начиналось так хорошо, превратилось в проект-монстр. Роджер чувствовал, что над ним нависла угроза, но не знал, как все исправить. Он перечитал книги и пересмотрел сайты о Scrum и пришел к выводу, что делал все правильно — по крайней мере, на бумаге. Проводились спринты, ежедневные scrum-митинги и ретроспективы. Он работал с владельцем продукта над приоритетами бэклога, для каждого спринта выбирал наиболее ценные функции, работал с командой над оценкой трудозатрат и назначал им разработчиков.

Почитать описание, рецензии и купить на сайте МИФа

Обращаясь к команде, Роджер в отчаянии сказал: «Меня просто затащили в кабинет к генеральному директору. Он недоволен отсутствием результатов, так же как и Ави. Послушайте, я буду защищать вас и возьму ответственность за случившееся на себя. Но для этого нужно поработать над нашими расчетами, потому что они совершенно неверные. И, кроме того, мне действительно необходимо, чтобы вы поработали по выходным и выправили ситуацию». За последние два месяца он просил об этом свою команду уже в третий раз. Этот проект так же, как и предыдущие, выбился из расписания.

Но что пошло не так? Можете ли вы определить, в чем проблема? Что бы вы сделали, если бы вас назначили scrum-мастером этого проекта? Вспомните об agile-ценностях и принципах. Есть ли способ их применения, который бы помог проекту?

Каждый член scrum-команды — владелец проекта

У каждого scrum-проекта есть владелец продукта, scrum-мастер и команда. Но не всякий проект можно назвать эффективным. Два владельца продукта, которые придерживаются разных методологий (Scrum и Waterfall), будут действовать по-разному. Scrum-мастер не делает того же, что командно-административный менеджер проекта или технический руководитель команды. Вот когда scrum-мастер, владелец продукта и команда начинают работать вместе, а не врозь, — это начинает походить на scrum-проект.

SCRUM-МАСТЕР УПРАВЛЯЕТ ПРОЦЕССОМ ПРИНЯТИЯ КОМАНДНЫХ РЕШЕНИЙ

То, как scrum-мастер делает свою работу, сильно отличается от командно-административной модели управления.

В командно-административном проекте его руководитель — владелец и хранитель расписания и плана работы. Он беседует со стейкхолдерами, получает требования, меняет план работы, получает от команды расчеты, распределяет задачи и строит график. Именно такой подход Роджер использовал в проекте Lolleaderz.com — он получил требования от Ави, смету от команды и распределил задачи между ее членами.

Участники командно-административной команды всеми силами спасают свою шкуру и не берут на себя ответственность, если в проекте возникают проблемы, вызванные планами каких-то других работ. Любому, кто имеет исключительное право собственности на график и план работы, остальные

Почитать описание, рецензии и купить на сайте МИФа

члены группы и владелец продукта с удовольствием позволяют принимать решения самостоятельно.

Это одна из причин, почему Scrum не отводит отдельной роли для человека, который бы единолично был владельцем плана. Если за точность выполнения плана в команде отвечает только тот, кто исполняет роль его владельца, то все остальные, столкнувшись с неприятностями, могут отмахнуться от них, сообщив: «Это проблемы того парня». Практически в каждом проекте рано или поздно возникают трудности, потому что все они обусловлены проблемами планирования. Поэтому scrum-мастер не имеет собственного плана. Он помогает команде создать его и, что еще важнее, правильно использовать Scrum и его практики. Он дает возможность каждому члену команды участвовать в планировании, а практики и ценности Scrum помогают им ощутить свою сопричастность.

ВЛАДЕЛЕЦ ПРОДУКТА ПОМОГАЕТ КОМАНДЕ ПОНЯТЬ ЦЕННОСТЬ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Представьте себе такой разговор между CEO и scrum-мастером. CEO спрашивает, что он получит в следующем году, инвестируя 2 миллиона долларов в программное обеспечение. Scrum-мастер отвечает: «Пока точно не знаю, предстоят ежемесячные обновления ПО, а в конце проекта получим программное обеспечение стоимостью не менее 2 миллионов долларов». Ни один здравомыслящий CEO не одобрит такой проект. Почему?

Причина в том, что он не основан на реальных **обязательствах.** Обязательства — это обещание сделать что-либо к определенному сроку. Настоящее обязательство тесно связано с дополнительной ответственностью, что подразумевает возможность информировать членов команды и искать решения, если обещание невозможно выполнить.

Большинство из нас присутствовали на собраниях, где члены команды утверждали, что не «подписывались» под сроками выполнения работ, поэтому не несут за них никакой ответственности. В этом вина некоторых неопытных руководителей. Они ошибочно полагают, что, как только задача записана в план, команда получает непреодолимое стремление ее выполнять.

Обязывают не планы, а наши обещания. План — это просто удобный способ их фиксации. Обязательства даются людьми и документируются в планах проекта. А это не просто бумажка. Это зафиксированная информация, и все держат ее в голове.

Почитать описание, рецензии и купить на сайте МИФа

В scrum-команде владелец продукта — это человек, который взял на себя выполнение обязательства. Он должен пообещать конкретные результаты, которые будут достигнуты в конце проекта. Владелец продукта — ключевое звено в реализации бизнес-цели, ради которой затевался проект.

Чем эффективнее на встрече с командой он сможет передать эти задачи и позволить ей осознать обязательства, тем лучше будет выполняться проект. Когда проект сталкивается с неизбежными трудностями (техническими проблемами, изменениями в бизнесе, увольнением сотрудников и т. д.), владелец продукта должен найти способ сохранить взаимопонимание в команде и поддерживать в ней чувство ответственности. Ежедневно он принимает решения, основанные на бизнес-изменениях, и встречается с командой, чтобы узнать, правильно ли она понимает бэклог и трансформировавшиеся задачи проекта.

Владелец продукта не ждет безучастно окончания спринта. Он обязан ориентироваться в происходящем. Его дело — расставлять приоритеты в бэклоге, чтобы быть голосом заказчика в команде разработки, помогать выявлять наиболее важные и ценные истории и задачи. Он должен быть уверен, что каждый член команды правильно понимает выражение *фактически сделано* в контексте бэклога задач. В процессе планирования спринта вся команда решает, какие функции из продуктового бэклога нужно переместить в спринт бэклога. Выбор делается на основании их значимости и расчетов трудозатрат на их выполнение. На этом его работа не заканчивается, затем владелец продукта руководит этим процессом.

Ежедневно владелец продукта принимает активное участие в проекте. Как и все успешные agile-команды, scrum-команды во многом зависят от личного общения, которое помогает правильно понимать, что они создают. Планирование спринта, проведенное в начале, дает каждому члену команды достаточно информации, чтобы начать работу, но владелец продукта не успевает сообщить всей команде подробности. Поэтому во время спринта владелец продукта каждый день занят. Он подробно отвечает на многочисленные вопросы членов команды, дает конкретные рекомендации, как вести разработку, и рассказывает, каким образом пользователи будут применять созданные функции, а также решает массу других вопросов, касающихся работы продукта.

Владелец продукта имеет право принимать такие решения. (Если он не занимается этим, значит, занимает чужое место. Scrum зависит от способности владельца продукта принимать решения от имени заказчика, в том числе выполненную работу.) Но он не располагает исчерпывающими сведениями. Обычно в проекте принимает участие много пользователей

Почитать описание, рецензии и купить на сайте МИФа

и заинтересованных сторон, владеющих ценной информацией, поэтому владелец продукта тратит много времени на общение с ними, чтобы получить ответы на вопросы разработчиков. Кроме того, он использует такое общение, чтобы отслеживать происходящие изменения и поддерживать актуальность продуктового бэклога, где отражаются последние изменения, в которых нуждается компания. Таким образом, если есть изменения в относительной ценности различных элементов продуктового бэклога, то владелец может изменить их очередность, подготавливая тем самым команду к следующей сессии по планированию спринта.

КАЖДЫЙ ВЫСТУПАЕТ В РОЛИ ВЛАДЕЛЬЦА ПРОЕКТА

В scrum-командах любят рассказывать басню про свинью и курицу, чтобы разобраться в том, как работает принятие на себя обязательств.

> Свинья и курица идут по дороге.
>
> Курица говорит свинье: «Давай откроем ресторан!»
>
> Свинья отвечает: «Хм, можно. А как мы его назовем?»
>
> Курица отвечает: «Почему бы не назвать его “Яичница с беконом”?»
>
> Свинья, немного подумав, говорит: «Нет, спасибо, ведь тогда мне придется посвятить себя проекту полностью, а ты будешь вовлечена лишь частично»*.

Так и в scrum-проекте: кто-то вовлечен частично, как курица, а кто-то полностью, как свинья. Чем это отличается от неэффективного водопадного подхода? Все сводится к тому, как действуют члены команды, руководитель проекта и владелец продукта**.

В scrum-командах часто говорят о ролях «свиней» и «куриц». Это упрощение введено для того, чтобы легче было определить человеку ту роль, которую он играет в проекте. Вспомните проекты, в которых вы участвовали: всегда ли вы действительно связывали собственные успехи или неудачи с успехом проекта?

---

* Из англоязычного варианта «Википедии» (The Chicken and the Pig, проверено 26 июля 2014 года).

** Разговор о «свиньях» и «курицах» выглядит странно, но scrum-команды действительно делают это. На самом деле некоторые старые версии scrum-руководства включают раздел, посвященный «свиньям» и «курицам»!

Почитать описание, рецензии и купить на сайте МИФа

Рис. 4.2. В истории про свинью и курицу роль первой значительно превышает вклад второй в готовый завтрак

Дело в том, что большинство людей считают себя «курами». Они готовы внести свой вклад в проект, но не хотят рисковать, ведь проверки, повышения, достижение будущих карьерных целей, сохранение рабочего места — все зависит от успеха проекта. Также легко забыть, что люди работают за деньги. Почему люди выполняют свою работу? Что на самом деле их мотивирует? Это не обязательно успех в текущем проекте.

Всех членов *эффективной* agile-команды можно считать «свиньями». Они искренне считают, что их личный успех зависит от успешности проекта. Но даже очень опытная scrum-команда легко возвращается к менталитету «курицы».

Бывало так, что вы как разработчик выбирали технологию для проекта лишь потому, что хотели изучить ее? Или в качестве руководителя проекта

Почитать описание, рецензии и купить на сайте МИФа

предпочитали agile-методологии, потому что они делают вас более востребованным? Скорее всего, да. Мы все хотя бы раз прошли через это. Каждый целеустремленный человек поступал подобным образом, и это важно признать. Но есть одно обстоятельство, предусмотренное правилами Scrum: выполнение задач спринта важнее всех прочих *профессиональных* целей*, которые у вас есть. Иными словами, пока ты в scrum-команде — оставайся «свиньей».

Когда все члены команды — «свиньи», значит, каждый взял на себя обязательства и будет делать все, что необходимо для их выполнения.

Вот пример, помогающий понять, что значит брать на себя обязательства. Допустим, вы CEO компании и у вас есть обязательства перед проектом. И выясняется, что члены проектной команды устали и нуждаются в кофе. Если все заняты и не могут прерваться прямо сейчас, то несмотря на занимаемую должность вы пойдете за кофе для команды, потому что действительно имеете обязательства и убеждены: именно на это нужно потратить свое время.

В то же время в каждом проекте программного обеспечения должны быть и «куры». Например, каждый пользователь — это потенциальная «курица». Сколько раз вы были разочарованы возможностями браузера, текстовым редактором или почтовым клиентом? Вы когда-нибудь оставляли свои отзывы на сайте или отправляли их по электронной почте команде поддержки? Это один из способов превратить себя в «курицу». Раз команда прислушивается к вашему мнению и решает проблему, значит, вы помогли повысить ценность продукта. И чем активнее «курица», тем более высокую ценность она может привнести.

Если в scrum-проекте вы играете роль «курицы», то ваше мнение важно. Вы заинтересованы в результатах, и команде интересны ваши мысли. Но ваша работа напрямую не связана с проектом, у вас другие цели (например, продажа продукта, его поддержка или управление компанией). Все это имеет значение, но не связано с разработкой конкретных функций.

Вот почему scrum-команды — и особенно владелец продукта — налаживают отношения со своими «курами». Один из самых эффективных способов, помогающий в этом, — обеспечить пользователей новыми версиями работающего ПО на регулярной основе и с предсказуемым графиком. Это поддерживает вовлеченность «кур» и помогает им видеть свое влияние на проект.

---

* Уточню: настоящая «свинья» заботится об успехе проекта больше, чем о чем бы то ни было в своей *профессиональной* жизни. Есть немало вещей в *личной* жизни — например, семья, — о которых следует заботиться больше, чем обо всем прочем. Если это не так, то у команды проблемы с образом мыслей и он будет мешать устойчивому темпу работ.

Почитать описание, рецензии и купить на сайте МИФа

Почитать описание, рецензии
и купить на сайте

Лучшие цитаты из книг, бесплатные главы и новинки:

Mifbooks Mifbooks Mifbooks

издательство
МАНН, ИВАНОВ И ФЕРБЕР
Максимально полезные книги