

План

- 1. Определения.
- 2. Роли в скрам-процессе.
- 3. Скрам митинги.
- 4. Особенности SCRUM методологии.
- 5. Вопросы коммуникации внутри команды.

1. Определения

Сегодня самой популярной гибкой методологией разработки ПО является Scrum.

Scrum (от англ. scrum «толкучка», схватка вокруг мяча (в регби)) - методология управления проектами, активно применяющаяся при разработке информационных систем для гибкой разработки программного обеспечения. Scrum чётко делает акцент на качественном контроле процесса разработки.

Скрам - это набор принципов, на которых строится процесс разработки, позволяющий в жёстко фиксированные и небольшие по времени итерации, называемые спринтами (sprints), предоставлять конечному пользователю работающее ПО с новыми возможностями, для которых определён наибольший приоритет. Возможности ПО к реализации в очередном спринте определяются в начале спринта на этапе планирования и не могут изменяться на всём его протяжении. При этом строго фиксированная небольшая длительность спринта придаёт процессу разработки предсказуемость и гибкость.

Спринт - итерация в скраме, в ходе которой создается функциональный рост программного обеспечения. Жёстко фиксирован по времени. Длительность одного спринта от 2 до 4 недель. В отдельных случаях, длительность спринта должна быть не более 6 недель. Считается, что чем короче спринт, тем более гибким является процесс разработки, релизы выходят чаще, быстрее поступают отзывы от потребителя, меньше времени тратится на работу в неправильном направлении. С другой стороны, при более длительных спринтах команда имеет больше времени на решение возникших в процессе проблем, а владелец проекта уменьшает издержки на совещания, демонстрации продукта. Разные команды подбирают длину спринта согласно специфике своей работы, составу команд и требований, часто методом проб и ошибок. Для оценки объёма работ в спринте можно использовать предварительную оценку, измеряемую в очках истории. Предварительная оценка фиксируется в бэклоге проекта. На протяжении спринта никто не имеет права менять список требований к работе, внесённых в бэклог спринта

Бэклог проекта - это список требований к функциональности, упорядоченный по их степени важности, подлежащих реализации. Элементы этого списка называются «пожеланиями пользователя» (user story) или элементами бэклога (backlog items). Бэклог проекта открыт для редактирования для всех участников скрам процесса.

Беклог продукта (Product Backlog) – приоритезированный список требований с оценкой трудозатрат. Обычно он состоит из бизнес-требований, которые приносят конкретную бизнес-ценность и называются элементы беклога.

Бэклог спринта - содержит функциональность, выбранную владельцем проекта из Бэклога проекта. Все функции разбиты по задачам, каждая из которых оценивается скрам-командой. Каждый день команда оценивает объём работы, который нужно проделать для завершения спринта.

Диаграмма сгорания задач (Burndown chart) - это диаграмма, показывающая количество сделанной и оставшейся работы. Обновляется ежедневно с тем, чтобы в простой форме показать подвижки в работе над спринтом. График должен быть общедоступен. Пример показан на рисунке 1. Диаграмма отображает завершенный спринт. Показывает оставшиеся нерешённые задачи и трудозатраты, необходимые для их завершения в расчёте на 21 рабочий день

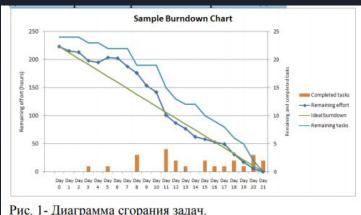


Рис. 1- Диаграмма сгорания задач.

Существуют разные виды диаграммы:

- диаграмма сгорания работ для спринта показывает, сколько уже задач сделано и сколько ещё остаётся сделать в текущем спринте.
- диаграмма сгорания работ для выпуска проекта — показывает, сколько уже задач сделано и сколько ещё остаётся сделать до выпуска продукта (обычно строится на базе

Доска задач - это наглядный инструмент мониторинга и управления внутри спринта. На стикерах указывается наименование истории пользователя, и они двигаются по соответствующим состояниям во время спринта. В начале спринта все истории пользователей находятся в первом столбце, отсортированные вертикально по важности. По мере того, как истории пользователей реализуются, члены команды меняют статусы у задач, и доска к середине спринта выглядит так, как на Рисунке 2.

Команда делает задачи по важности, начиная с самых верхних, доводя их до статуса «Готово». Доска, по завершению спринта,



2. Роли в скрам-процессе

По методике Scrum в производственном процессе есть определённые роли, разбитые на 2 группы «свиней» и «кур».

Свиньи создают продукт, тогда как куры заинтересованы, но не настолько — ведь им всё равно, будет ли проект удачным или нет, на них это мало отразится. Требования, пожелания, идеи и влияние кур принимаются во внимание, но им не разрешают непосредственно включаться в ход скрам-проекта. Основные роли в методологии скрам («Свиньи») «Свиньи» полностью включены в проект и в скрам-процесс.

- Скрам-мастер (Scrum Master) проводит совещания (Scrum meetings) следит за соблюдением всех принципов скрам, разрешает противоречия и защищает команду от отвлекающих факторов. Данная роль не предполагает ничего иного, кроме корректного ведения скрам-процесса. Руководитель проекта скорее относится к владельцу проекта и не должен фигурировать в качестве скрам-мастера.
- Владелец продукта (Product Owner) представляет интересы конечных пользователей и других, заинтересованных в продукте сторон.

- Скрам-команда (Scrum Team) кросс-функциональная команда разработчиков проекта, состоящая из специалистов разных профилей: тестировщиков, архитекторов, аналитиков, программистов и т. д. Размер команды в идеале составляет от 7 до 9 человек. Команда является единственным полностью вовлечённым участником разработки и отвечает за результат как единое целое. Никто кроме команды не может вмешиваться в процесс разработки на протяжении спринта. Дополнительные роли в методологии скрам («Куры»)
- Пользователи (Users)
- Клиенты, Продавцы (Stakeholders) лица, которые инициируют проект и для кого проект, будет приносить выгоду. Они вовлечены в скрам только во время обзорного совещания по спринту (Sprint Review).
- Управляющие (Managers) люди, которые управляют персоналом.
- Эксперты-консультанты (Consulting Experts)

3. Скрам митинг

Планирование спринта (Sprint Planning Meeting)

Происходит в начале новой итерации Спринта. Из бэклога проекта выбираются задачи, обязательства по выполнению которых за спринт принимает на себя команда; На основе выбранных задач создается бэклог спринта. Каждая задача оценивается в идеальных человеко-часах; Решение задачи не должно занимать более 12 часов или одного дня. При необходимости задача разбивается на подзадачи; Обсуждается и определяется, каким образом будет реализован этот объём работ; Продолжительность совещания ограничена сверху 4-8 часами в зависимости от продолжительности итерации, опыта команды. - (первая часть совещания) Участвует владелец проекта и скрам команда: выбирают задачи из бэклога продукта; - (вторая часть совещания) Участвует только команда: обсуждают технические детали реализации, наполняют бэклог спринта.

Ежедневное совещание (Daily Scrum meeting)

- начинается точно вовремя; все могут наблюдать, но только «свиньи» говорят; длится не более 15 минут; проводится в одном и том же месте в течение спринта. В течение совещания каждый член команды отвечает на 3 вопроса:
- Что я сделал с момента прошлой встречи для того, чтобы помочь Команде Разработки достигнуть Цели Спринта?
- Что я сделаю сегодня для того, чтобы помочь Команде Разработки достичь Цели Спринта?
- Вижу ли я препятствия для себя или Команды Разработки, которые могли бы затруднить достижение Цели Спринта?

(Над решением этих проблем работает скрам мастер. Обычно это решение проходит за рамками ежедневного совещания и в составе лиц, непосредственно затронутых данным препятствием.)

Скрам над скрамом (Scrum of Scrums)

Проводится после ежедневного скрам совещания. Позволяет нескольким скрам командам обсуждать работу, фокусируясь на общих областях и взаимной интеграции. Повестка та же, что и на ежедневном скрам совещании плюс следующие вопросы:

- Что каждая команда сделала с момента предыдущего ежедневного совещания?
- Что каждая команда сделает к следующему ежедневному совещанию
- Есть ли проблемы, мешающие или замедляющие работу каждой команды?
- Нужно ли другой команде сделать что-то из задач вашей команды?

Обзор итогов спринта (Sprint review meeting) Проводится в конце спринта.

- Команда демонстрирует прирост функциональности продукта всем заинтересованным лицам.
- Привлекается максимальное количество зрителей.
- Все члены команды участвуют в демонстрации (один человек на демонстрацию или каждый показывает, что сделал за спринт).
- Нельзя демонстрировать незавершенную функциональность.
- Ограничена четырьмя часами в зависимости от продолжительности итерации и прироста функциональности продукта.

Ретроспективное совещание (Retrospective meeting) Проводится в конце спринта.

- Члены команды высказывают своё мнение о прошедшем спринте.

- Отвечают на два основных вопроса:

Что было сделано хорошо в прошедшем спринте?

Что надо улучшить в следующем?

- Выполняют улучшение процесса разработки (решают вопросы и фиксируют удачные решения).
- Ограничена одним-тремя часами.

4.Особенности SCRUM методологии.

Распределённая команда.

Контроль качества программного продукта в Scrum возложен, в первую очередь, на владельца продукта. В случае если владелец продукта находится в другом городе или стране, данный контроль ослабевает и требуется применение дополнительных методик, например, использование уполномоченного владельца продукта. Но использование уполномоченного владельца продукта влечёт за собой ряд дополнительных трудностей (недостаточная глубина знаний о предметной области, недостаток взаимодействия с владельцем продукта и последующие расхождения в требуемом функционале), и не может полностью заменить владельца продукта. Рассматривается территориально распределенная команда, при этом владелец продукта находится рядом с одной частью команды. Распределённая команда- это когда, над одним проектом работает несколько человек, находящихся в разных офисах в разных городах в разных часовых поясах. Использование распределенной команды порождает проблемы коммуникации, о чём мы поговорим дальше.

Частичная занятость

Ответственность за контроль хода проекта и соблюдения правил методологии Scrum возложена на скрам-мастера. Согласно методологии, скрам-мастер должен, в обязательном порядке, находиться вместе с командой проекта, для того чтобы быть в курсе всех происходящих изменений.

В случае если команда территориально распределена, Scrum предлагает несколько вариантов решения:

- Создание отдельной скрам-команды для каждой территории, обладающей самостоятельностью (Scrum of Scrums). Такой подход возможен только если удаленная часть команды является функционально целостной, то есть состоит из всех необходимых для разработки членов команды.
- Назначение в удаленной части команды одного или нескольких человек, которые будут иметь представление о ходе проекта в рамках своей территории и выполнять роль помощника скрам-мастера. При таком подходе команда работает в рамках единого спринта, а скрам-мастеру необходимо взаимодействовать только с назначенными людьми.

Сложности контроля процесса разработки.

Scrum предполагает отсутствие контроля за процессом разработки по причине самоорганизованности команды, что усложняется по мере увеличения сложности проекта, численности проектной команды и территориальной распределенности части проектной команды. Скрам-мастер не может в полной мере контролировать процесс на уровне задач, только на уровне спринта, по причине низкой формализации процесса и отсутствие документации, что мешает выявлять проблемы и оптимизировать процесс в рамках спринта. Поэтому рекомендуется использовать небольшие команды сосоящие из 7-9 человек.

Масштабирование Scrum

Небольшие команды показывают чаще хорошие результаты, чем большие, поэтому необходимо по возможности вести разработку компактными командами. Часто бывает так, что размер проекта и сроки его реализации просто не позволяют вести разработку 7-9 людьми и приходиться задействовать несколько команд. Для организации разработки больших проектов или портфеля проектов необходимо масштабировать Scrum на следующий уровень. Со стороны команд разработки – это выливается в проведении Scrum of Scrum. На этот митинг собираются скрам-мастера (SM), в качестве представителей конкретных команд. Организует собрание руководитель программы (Program Manager – PM).

Нулевой спринт

Нулевой спринт - промежуток времени, за который мы проводим всю подготовительную работу для полноценного начала первого спринта. По длительности нулевой спринт не обязательно должен совпадать с длиной последующих спринтов. За это время нам необходимо попытаться из группы индивидуальностей создать команду, подготовленную к своему первому спринту, и дать ей в руки все необходимые инструменты.

Нам понадобится следующий минимум перед началом первого спринта:

- Знания о процессе.
- Знания о команде.
- Знания о продукте.

Нулевой спринт имеет еще одну важную характеристику, которая ставит его в особое положение по отношению к другим – особое участие бизнеса и Владельца Продукта. Их роли в нулевом спринте просто критичны. Новой Скрам команде для начала своего первого спринта необходимо иметь как минимум:

- Видение Продукта(Vision)
- Беклог Продукта (Product Backlog)

Видение продукта (vision) — это краткое описание сути будущего продукта. В этом документе вкратце описывается, что это за продукт, каковы цели и задачи его создания, кто его пользователи и каковы основные возможности будущей системы.

5.Вопросы коммуникации.

В Scrum для нас важны:

- люди и их взаимодействие, чем процессы и средства;
- работающее ПО, чем исчерпывающая документация;
- сотрудничество с заказчиком, чем обсуждение условий контракта;
- реагирование на изменения, чем следование плану.

То есть мы не ставим под сомнение важность пунктов справа, но в то же время для нас гораздо важнее записанное слева. Таким образом, Scrum не отрицает важности качественных процессов и инструментов для их обеспечения, различного рода документации, планирования работ и следования плану, просто для успешного проекта это не считается главным. На первый план выходят люди, участвующие в процессе разработки ПО, коммуникации между ними и сам результат — работающая ИТ-система, способная функционировать в условиях меняющихся требований.

To есть Scrum - это гуманистический подход к разработке, учитывающий человеческий и социальный факторы и предполагающий командную работу, опору на коллективный опыт и вовлеченность всех сотрудников в процесс разработки.

Внутри команды.

Методология Scrum полностью покрывает проблемы: инициации процесса коммуникации, низкой частоты коммуникации, различий во временных зонах, высоких затрат на коммуникацию. Упрощение процесса инициации взаимодействия между членами команды достигается с помощью создания общедоступного списка контактов, в котором дополнительно указывается пересечение в рабочих часах, в случае различия во временных зонах.

Все члены команды в течение рабочего дня должны быть доступны в онлайн-мессенджерах или скайпе, чтобы снизить процент использования почты. Перманентная доступность членов команды в онлайнмессенджерах и скайпе также позволяет уменьшить число вопросов, которые возникают в рамках ежедневных митингов, тем самым снизив затраты на коммуникацию.

Низкая частота коммуникации между членами команды является следствием увеличения длительности спринта, что свойственно территориально распределенным командам, поэтому на начальном этапе проекта длительность спринта не рекомендуется устанавливать больше 2 недель. Ежедневные митинги, в которых принимает участие вся проектная команда, также позволяют повысить частоту коммуникации. Недопонимание между членами команды может быть сведено к минимуму в результате применения таких методик, как совместная работа в рамках первых спринтов, общение вне работы (возможно виртуальное).

К вопросу формирования доверия в команде в Scrum подходят основательно, существует ряд техник и методик, которые помогают команде сплотиться и сформировать открытые дружеские отношения как внутри команды, так и с владельцем продукта. Популярной техникой сплочения команды является совместная работа в рамках первого спринта, а также дополнительные нерабочие мероприятия, на которых члены команды сближаются и начинают лучше понимать друг друга. Предлагается организовать первичную встречу всей команды с владельцем продукта вживую, для того, чтобы команда получила представление о бизнес-целях и задачах проекта, а также сформировала понимание об ожиданиях заказчика от команды разработчиков - это позволит поднять боевой дух команды и мотивировать сотрудников на реальные цели. Гибкие методологии также предлагают ряд игр для проектной команды, направленных на построение доверительных отношений. Но специалисты в области гибких методологий, заявляют, что важно не увлекаться мероприятиями по сплочению команды в начале проекта. В первую очередь сфокусироваться на получении первого рабочего образца продукции, а после проводить нерабочие мероприятия, чтобы позволить участникам проекта разбиться на подгруппы с учетом их умений, навыков и интересов.

Более сложным является процесс коммуникации в распределённой команде. Можно выделить три составляющие проблемы коммуникации в распределённой команде:

- Техническая:

если человека рядом нет, к нему нельзя просто взять и подойти, чтобы обсудить какие-то текущие проблемы. Решается эта проблема с помощью современных технологий. Мессенджеры, голосовая связь и даже камера в телефоне — все нам в помощь. Все члены команды есть в скайпе, есть общекомандный чат. Люди иногда создают подчатики на несколько человек - обсудить свою проблему. Для того чтобы сделать процесс коммуникации в команде наиболее эффективным, то есть требующим минимальных трудозатрат, предлагается использовать следующие каналы коммуникации - различные варианты видеоконференций. Для решения проблемы различия во временных зонах применяются различные подходы, в зависимости от количества часов. Хорошей практикой является формальное установления часов «пересечения», когда члены команды знают, что в это время следует максимально сфокусироваться на получении нужной информации от коллег из другого офиса, если таковая требуется. В случае если разница в часах существенная (больше 5 часов), то для планирования задач на спринт, совещание разбивается на несколько частей, которые проводят в разные дни.

- Мотивационная:

если у команды нет своей комнаты, где перед глазами есть доска со стикерами, списком проблем, фокус и приоритеты начинают теряться. Но, если людям не напоминать про цели, статус спринта и всяческие простые правила, команда расслабляется. Для успешного процесса нужны, так называемы информационные радиаторы. В качестве информационных радиаторов можно использовать: общее обсуждения того, что мы видим на виртуальном dashboard во время ежедневных митингов, и рассылка емкого короткого статус-письма, по факту — тот же самый радиатор. Например, начиная со второй половины итерации, QA каждое утро рассылать их видение происходящего. Список юзер-стори, статус каждой, кто отвечает за продвижение ее дальше. В том же письме напоминается список рисков и всего того, о чем стоит помнить прямо сейчас. Такое письмо рассылается раз в день после ежедневного митинга, оно должно быть максимально коротким и включать в себя только самое важное.

- Психологическая:

люди, которые не сидят рядом и не видятся каждое утро, обсуждая за кофе последние новости или успехи детей в школе, менее склонны прощать ошибки коллегам, особенно если про коллегу они знают только логин в системе контроля версий и e-mail. Может возникать концепция «мы и они» по отношению к коллегам из другого офиса и прочие неприятные ситуации. Если участники проекта ощущают себя одной командой, особой группой, они будут друг друга поддерживать, прощать, не желать подвести команду, и предлагать помощь коллегам. Самое больше препятствие для этого - чисто деловые отношения, когда члены команды воспринимают друг друга не как живых людей, а как непонятную картинку с аватара. Для налаживания контактов лучше при разговорах использовать камеры, чтобы мы видели живого человека, мимику, невербальную информацию; в общем чате и даже на ежедневных митингах приветствуется small talk, обсуждение какихто новостей, проблем. В аккуратных дозах это не мешает работе, но зато создает понимание и ощущение того, с кем ты работаешь. Вначале проекта, конечно, лучше всем собраться, если это возможно и познакомиться поближе, устроить тимбилдинг. Если все удалось, на несколько месяцев запала хватает, а, чтобы продолжалось подольше, нужно прилагать некие усилия по поддержанию человеческих отношений. Также стоит упомянуть практику face-to-face (one-to-one), которая тоже прекрасно срабатывает по скайпу, но она хороша и не для распределенных команд.

Этапы командообразования:

- Формирование На этапе формирования происходит создание команды и постановка целей, распределение и закрепление ролей (в том числе и социальных). Отдельные члены команды еще не очень понимают цель и задачи, которые перед ними поставлены:
- Бурление На этапе «бурления» участники осознают свои цели и определяют вектор движения. Обратите внимание, что эти векторы разнонаправленные между собой и с направлением, которое необходимо для достижения цели. Цель также стала более четкой и понимаемой для команды. На данном этапе особенно часто возможны конфликты и противостояние между членами команды, поэтому особенно возрастает роль скраммастера как модератора.
- Нормализация Следующим этапом идет нормализация, когда члены команды притираются к друг другу и начинают двигаться сонаправлено: Основной задачей скраммастера является фасилитация для перехода на следующий этап.

- Функционирование На этапе функционирования команда становится самоуправляемой и способной оптимизировать свою производительность, поэтому векторы, направленные к цели, удлиняются:
- Расформирование Когда цели, поставленные перед командой, достигнуты, наступает этап расформирования, и направление «движения» участников снова рассинхронизируется:

Чтобы работать эффективно, команда должна находиться на этапе «Функционирование». Главной задачей команды (и в частности скрам-мастера) является максимально быстрый переход между этапами. С заказчиками.

Заказчик проекта нацелен на то, чтобы в ограниченный срок и бюджет сделать самые важные функции продукта, без которых продукт не сможет выполнять свое назначение. Команда проекта получает обратную связь от заказчика по завершению каждой итерации. Это позволяет быстро исправлять те несоответствия, которые были допущены в итерации, и быстро улучшать продукт.

Поскольку гибкий подход подразумевает выстраивание по-настоящему партнерских отношений с заказчиком, необходимо выработать стратегию работы с «вредными» заказчиками. Ведь построить с ними партнерские взаимоотношения чрезвычайно сложно. Как дифференцировать клиентов по «вредности» и как (и стоит ли) работать с «вредными» клиентами? Сначала определимся, кого считать «вредным»

Лояльный клиент доставляет меньше всего проблем, обычно делает заказ на большую сумму, чем средний клиент. К тому же они часто делают повторные заказы. Таких клиентов надо холить и лелеять, для чего можно разработать программу повышения лояльности: от предоставления скидок до специальных условий работы. Значительную часть клиентов можно отнести к категории нормальных или средних. С ними не возникает особых проблем, но и чудес лояльности они не покажут.

Самым плохим вариантом являются нелояльные или проблемные клиенты. Для каждой организации нелояльность клиентов определяется по-разному. Во всех сферах деятельности к таким клиентам относят тех, кто не соблюдает в той или иной форме договоренности, и, прежде всего, это касается финансовой части. Если говорить про разработку, то здесь стоит выделить клиентов, которые сами срывают сроки: не предоставляют материалов, не успевают проверять сделанную работу и прочее. Сначала необходимо нелояльного клиента определить до появления проблем. Это можно сделать, наведя справки и в процессе непосредственного общения. Далее необходимо рассчитать возможные риски: проблемы с оплатой, различные задержки. Следует хорошо подумать: нужен ли вам этот клиент, ведь с ним не получиться работать на 100% по Scrum. В таком случае подход должен быть комплексным: во внимание нужно принять как стоимость и сроки проекта, так и возможные риски