

Скрам как методология разработки

Ранее мы уже познакомились с понятием **жизненного цикла проекта** и выделили последовательность этапов, через которые проходят проекты от инициации до завершения:

1. Определение концепции
2. Планирование и подготовка
3. Проектирование
4. Реализация и тестирование
5. Внедрение
6. Эксплуатация и развитие

Также мы говорили о подходах к выполнению этих шагов, другими словами, о **методологиях разработки**, и разбирали отличия гибких методологий от водопада.

Среди гибких методологий наиболее известные фреймворки: **Scrum**, **Kanban**, **LeSS** (Large-Scale Scrum), **SAFe** (Scaled Agile Framework). Фреймворк в этом случае – это набор правил, элементов, активностей, из которых строится процесс разработки.

Какой бы фреймворк (из гибких фреймворков) не был выбран для реализации проекта, необходимо в первую очередь понимать **основополагающие принципы, описанные в Agile-манифесте**:

1. Наивысшим приоритетом для нас является удовлетворение потребностей заказчика, благодаря регулярной и ранней поставке ценного программного обеспечения.
2. Изменение требований приветствуется, даже на поздних стадиях разработки. Agile-процессы позволяют использовать изменения для обеспечения заказчику конкурентного преимущества.
3. Работающий продукт следует выпускать как можно чаще, с периодичностью от пары недель до пары месяцев.
4. На протяжении всего проекта разработчики и представители бизнеса должны ежедневно работать вместе.
5. Над проектом должны работать мотивированные профессионалы. Чтобы работа была сделана, создайте условия, обеспечьте поддержку и полностью доверьтесь им.
6. Непосредственное общение является наиболее практичным и эффективным способом обмена информацией как с самой командой, так и внутри команды.
7. Работающий продукт — основной показатель прогресса.
8. Инвесторы, разработчики и пользователи должны иметь возможность поддерживать постоянный ритм бесконечно. Agile помогает наладить такой устойчивый процесс разработки.
9. Постоянное внимание к техническому совершенству и качеству проектирования повышает гибкость проекта.
10. Простота — искусство минимизации лишней работы — крайне необходима.

11. Самые лучшие требования, архитектурные и технические решения рождаются у самоорганизующихся команд.
12. Команда должна систематически анализировать возможные способы улучшения эффективности и соответственно корректировать стиль своей работы.

В Agile-манифесте также заявлены следующие ценности:

1. Люди и взаимодействие важнее процессов и инструментов.
2. Работающий продукт важнее исчерпывающей документации.
3. Сотрудничество с заказчиком важнее согласования условий контракта.
4. Готовность к изменениям важнее следования первоначальному плану.

Полный текст Agile-манифеста доступен по ссылке agilemanifesto.org

Мы остановимся на одном из самых популярных фреймворков - **Скраме (Scrum)**.

Основная особенность работы по Скраму – работа в итерациях, которые в Скраме называются **Спринтами**:

- Продолжительность Спринтов одинаковая, обычно 1-4 недели.
- В Спринте должна быть общая цель и план достижения.
- Синхронизация прогресса, выявление и решение препятствий происходит на короткой ежедневной встрече.
- Результат Спринта должен демонстрироваться стейкхолдерам в конце каждого Спринта.

Формулируя цель Спринта, важно фокусироваться на задачах, которые представляют собой **ценность для клиента**. Ошибкой будет являться планирование в Спринте этапов работы, то есть подготовка макета, описание сценария, реализация микросервиса и т.п. Результатом будет считаться только полная разработка функциональной возможности, даже если эта часть продукта не будет выведена в эксплуатацию.

Команды в Скраме небольшие (до 10 человек), но включают все необходимые компетенции. Состав такой команды должен сохраняться на протяжении всего периода разработки, без отвлечений на другие задачи.

В Скрам-команде выделяется три роли:

- Владелец продукта (Product Owner)
- Разработчики
- Скрам-мастер

Владелец продукта отвечает за суть и успех продукта, это также может быть представитель заказчика. Разработчиками в такой команде называют всех участников команды, кто занимается реализацией продукта. Скрам-мастер

помогает выстраивать эффективное взаимодействие команды, но не погружается в суть и разработку продукта.

Ни Владелец продукта, ни Скрам-мастер не являются менеджером проекта или руководителем команды. Особенность Скрама – **самоуправляемая команда**, которая достигает результата за счёт совместной работы, обмена опытом и мотивации.

Для внедрения Скрама недостаточно знания базовых принципов и понятий. Все участники должны быть вовлечены в процесс разработки, участвовать в обучении и адаптации лучших практик в своей команде. Большую пользу команде может принести Скрам-мастер, который может выступить проводником и помочь адаптировать фреймворк к конкретным условиям.

Артефакты в Скраме. Бэклог

Наполнение бэклога

Материальное представление результата работы или ценности называется **артефактом**. Ключевые артефакты в Скраме:

- **Бэклог Продукта (Product Backlog)**
- **Бэклог Спринта (Sprint Backlog)**
- **Элемент Бэклога (Backlog Item)**
- **Инкремент (Increment)**

Бэклог Продукта (Product Backlog) – это упорядоченный набор элементов, очередь, перечень всех функций и доработок, которые стейкхолдеры хотят получить от продукта.

Все требования, которые необходимо реализовать, называются **элементами Бэклога Продукта**, а тот объём, который попадает в Спринт, – **Бэклогом Спринта**.

Элементы Бэклога обычно представлены **пользовательскими историями** (User Stories).

Бэклог может содержать не только пользовательские истории, но также задачи и ошибки. Такое происходит в случаях, когда принимается решение о реализации какой-либо доработки, которая не может быть оформлена в целую историю, но соответствует целям и важна для развития продукта. По сути это изменение, которое планируется выполнить в следующих **Инкрементах** продукта (например, функции, требования, усовершенствования или информация по исправлению дефектов).

Инкрементом называют реализацию элементов Бэклога, сделанную в ходе конкретного Спринта.

Границы проекта

В процессе сбора данных и описания концепции, образа, видения продукта, его функций, важное внимание уделяется **полноте этого описания, анализу различных сторон продукта**.

Однако для реализации продукта нужно уметь оценивать возможности и адаптироваться к изменяющимся условиям. **В отличие от концепции, границы проекта, границы версий продукта более динамичны**. Ограничения сроков и ресурсов, изменения на рынке, возможности технологий и прочие факторы могут повлиять на скоуп проекта или версии.

Скоуп (Scope) – объём требований или работ, перечень задач, который необходимо выполнить для достижения целей.

Для наглядного представления планов по созданию и развитию продукта можно использовать **дорожную карту** (Roadmap). Это удобный инструмент, который легко поддерживать и который не требует глубокого погружения в детали, если необходимо держать в курсе не вовлечённых в разработку стейкхолдеров.

Чтобы проверить свои гипотезы, убедиться в том, что продукт нужен пользователям, **создаётся MVP**.

Минимально жизнеспособный продукт (Minimal Viable Product, MVP) – тестовая версия товара, услуги или сервиса с минимальным набором функций (иногда даже одной), которая несет ценность для конечного потребителя.

Для реализации MVP выделяется скоуп, который решает определённую задачу пользователя. Из всего скоупа продукта необходимо выбрать задачи **с наибольшим приоритетом**, без которых невозможно решить задачу пользователя.

!! В процессе выбора историй для реализации MVP есть соблазн выбрать функции-“бантики”, приятные элементы, улучшающие пользовательское взаимодействие и т.п. Однако необходимо помнить, что MVP нужно выпустить минимальными усилиями за минимальное время, чтобы как можно быстрее **получить реальную обратную связь от пользователей**.

Случается, что участники команды, предлагая улучшения, вовлекают остальных в обсуждение новой идеи, предлагают добавить “мелочь” уже в ближайшую итерацию, которая, кажется, не отнимет много времени и т.д.

!! Важно не допускать смещение внимания на новые, ложно срочные и важные задачи.

В таких случаях необходимо реально оценивать приоритеты – обязательно сравнивать приоритет новой задачи относительно текущих и откладывать подобные обсуждения до следующего мероприятия по уточнению бэклога.

Уточнение бэклога

Преимущество гибких методологий в возможности и даже в необходимости регулярно пересматривать требования и решения, адаптировать их в ответ на новые вводные.

!! Бэклог необходимо поддерживать в актуальном состоянии.

Для поддержания актуальности проводятся регулярные мероприятия с участием владельца/заказчика, команды разработки и других стейкхолдеров по **уточнению бэклога**:

1. Детализировать неясные элементы.

!! Пользовательские истории НЕ являются описанием конечных требований – это в первую очередь инструмент для высокоуровневой оценки и планирования.

История уточняется через описание деталей в **Acceptance Criteria (AC)**, критериях приёмки истории. Поскольку для подготовки детализации требуется время – по сути это описание детальных требований – заполняются критерии по мере проработки функции в рамках этапа анализа.

2. Добавить новые, удалить неактуальные элементы.

Необходимо также обсудить суть и причины изменений, проверить на соответствие целям.

К новым элементам также относятся ошибки, которые появляются в ходе этапа тестирования. В общем виде ошибка описывает проблему, для которой (аналогично требованиям в истории) должны быть проработаны решения и критерии готовности.

3. Декомпозировать крупные, сгруппировать мелкие элементы.

Декомпозиция - это процесс разделения крупных элементов на мелкие. Например, разбиение объёмной истории, которую невозможно оценить или которую невозможно реализовать в пределах одной итерации, на отдельные истории помельче.

Некорректно делить истории по архитектурным слоям (доработки фронтенда, бэкенда и т.п.) – необходимо отталкиваться от ценности для пользователя, формулировать историю, а к ней формулировать задачи для реализации. А если не удаётся сформулировать историю, то проверить эти задачи на соответствие целям и добавить в бэклог как самостоятельный элемент.

4. Оценить трудозатраты на реализацию.

На разных стадиях развития продукта можно и нужно использовать различные способы оценки трудозатрат.

5. Приоритизировать элементы, обсудить возможности, зависимости, риски.

Наиболее приоритетные истории и задачи должны быть в начале списка. При определении приоритетов должны учитываться различные факторы: соответствие целям, срочность, реализуемость, трудозатраты, риски, логические связи между функциями, зависимости от партнёров и других команд, мнение и рекомендации стейкхолдеров и пр. Можно применять модель приоритизации Weighted Short Job First или другие техники. А для проверки историй на соответствие исходным целям можно применить карту влияния (Impact Map).

6. Запланировать ближайшие итерации.

Регулярная работа с бэклогом должна в том числе приводить к оформлению более детальных планов. На старте разработки важно определить границы проекта, обозначить состав первой версии и, возможно, объявить последующие фазы.

!! Несмотря на участие широкого круга стейкхолдеров в процессе уточнения бэклога (включая команду разработки), **окончательное решение о приоритетах и планах принимает владелец/заказчик**, отвечающий за продукт целиком. Команда разработки, помимо обсуждения сути требований, со своей стороны должна предоставить максимально полезную и точную информацию о возможностях реализации, оценке, сложностях и рисках.

Для обеспечения качественной подготовки требований и других материалов, необходимых для старта реализации, в пользовательских историях и других элементах бэклога закрепляются **Definition of Ready (DoR)**, критерии/определение готовности к разработке. Это условия, выполнение которых обеспечивает возможность взять историю в разработку. Например, наличие АС, макета, диаграммы, комментариев директора, оценка и прочие условия, которые команда вырабатывает в процессе работы.

Условия для объявления пользовательской истории готовой с точки зрения разработки фиксируются в **Definition of Done (DoD)**, критерии/определение готовности разработки. Например, реализованы АС, успешно выполнено

нагрузочное тестирование, проведена демонстрация заказчику, обновлено руководство по эксплуатации и прочие.

Ритуалы в Скраме

Ритуалами называются активности, которые рекомендуется внедрять для организации эффективной работы команды. В Скраме можно выделить следующие ритуалы:

1. **Планирование Спринта** (Sprint Planning), где определяется, какие элементы Бэклога команда возьмёт в работу. Для качественного планирования обязательно стоит отвечать для себя на вопросы, почему конкретный Спринт ценен, что конкретно может быть сделано в этом Спринте, как конкретно будут реализовываться требования.
2. **Спринт** (Sprint) - непосредственно итерация разработки, объединяющая в том числе другие мероприятия.
3. **Ежедневный Скрам** (Daily Scrum meeting, или Standup meeting), который необходим для оценки прогресса, выявления препятствий и синхронизации действий команды. В таком мероприятии участвует вся команда, ежедневно, желательно в одном и том же месте в одно и то же время.
4. **Обзор Спринта / Демонстрация** (Sprint Review / Demo), когда происходит показ результатов спринта заинтересованным лицам (то есть своеобразная инспекция Инкремента).
5. **Ретроспектива Спринта** (Sprint Retrospective), когда команда обсуждает процесс и свои результаты, делится впечатлениями и пожеланиями, находит способы улучшить результаты и оптимизировать работу.

Ритуалы в Скраме. Планирование Спринта

Планирование Спринта (Sprint Planning) – мероприятие, которое начинает Спринт и на котором определяется, какие элементы Бэклога команда возьмёт в работу.

Спринт не может быть начат без Планирования Спринта, но непосредственно мероприятие может проводиться накануне (в период предыдущего Спринта).

Для качественного планирования обязательно стоит отвечать для себя на вопросы, почему конкретный Спринт ценен, что конкретно может быть сделано в этом Спринте, как конкретно будут реализовываться требования.

Участники Планирования

В Планировании участвует вся команда, то есть **Владелец продукта, Разработчики** (помните, что в Скраме разработчиками называют всех участников команды, независимо от специализации) и **Скрам-мастер**.

Также на Планирование **могут быть приглашены другие стейкхолдеры**, например, участники другой команды (у вас могут быть зависимые задачи), работники других подразделений (новые функции могут затрагивать их интересы), эксперты в предметной области (продукт может быть сложным или специфическим, вам может потребоваться консультация), технические эксперты (для проектирования решения), руководство компании (чтобы понимать процесс, прогресс и общее состояние работ) и так далее.

Роль Владельца продукта

Владелец продукта должен подготовить к обсуждению наиболее важные элементы Бэклога. Он должен убедиться, что все требования и задачи связаны с целями продукта/проекта, что есть необходимое описание для обсуждения.

Роль Владельца продукта иногда выполняет так называемый продуктовый менеджер или некоторый коллективный (из нескольких партнёров-участников) собирательный Владелец продукта, включающий компетенции из области менеджмента, анализа и маркетинга.

В проектной разработке менеджер или аналитик может выступать прокси-заказчиком, который понимает цели проекта, может определять приоритеты и раскрывать суть новых доработок.

Роль Скрам-мастера

Скрам-мастер, ответственный за само проведение мероприятия, то есть место, время, формат. Скрам-мастер помогает организовать всех участников, установить нужный контакт и вовлечение, а также преодолеть сложности, конфликты и недопонимание.

Скрам-мастер не является обязательным участником, его функции может выполнять любой участник команды, в том числе на постоянной основе.

Цель Спринта

Непосредственно на мероприятии определяется, почему ближайший Спринт будет ценен. Команда обсуждает, как повысить ценность доработок, как улучшить

продукт. В результате формируется цель Спринта, которая понятна всем участникам и которая объясняет, какой результат по итогам Спринта можно ожидать.

Элементы Бэклога

Владелец продукта обычно предлагает, какие элементы Бэклога хотелось бы включить в Спринт. К элементам Бэклога в первую очередь относятся пользовательские истории.

Команда изучает требования (представленные в формате критерием приёмки) и макеты, уточняет детали, обсуждает спорные моменты и пути решения задачи. Все должны одинаково понимать и ожидаемый результат работы, и способ решения задач.

Важно отметить, что подразумевается достаточное погружение в требования и реализацию всех специалистов, даже если не все участвуют в непосредственной разработке.

Пока не будет достигнуто понимание, невозможно будет оценить трудозатраты, а значит, и сроки разработки, следовательно, появляются большие риски в достижении цели Спринта. К слову, цель Спринта можно формировать и после выбора элементов Бэклога, но это не отменяет необходимость думать про итоговую ценность.

Неправильно оставлять пачку требований или задач на индивидуальное изучение одному сотруднику, тем более планировать таким образом его работу в Спринте. Прояснять подробности необходимо совместно и до старта разработки. А если команда осознаёт, что предлагаемый элемент Бэклога не готов к разработке, то может быть принято решение отложить разработку данного элемента и выбрать другой.

!! Обратите внимание, что в этот момент происходит переход от требований к задачам.

Задачи Спринта (план реализации)

Элемент Бэклога – это лишь общая формулировка желаемой доработки. Проектирование конкретной реализации и оценку трудозатрат невозможно сделать, опираясь только на заголовок пользовательской истории.

Распределение конкретных задач для конкретных исполнителей происходит именно на Планировании Спринта.

Чтобы началось рассмотрение элемента Бэклога, он должен соответствовать критериям/определению готовности к разработке – **Definition of Ready (DoR)**, то есть даже в формальном смысле быть полностью проработан.

Далее стоит опереться на понятие **Инкремента**, то есть конкретную реализацию элемента Бэклога. Конкретная реализация и задачи для её достижения определяются командой разработки. То есть разработчики определяют, как из элемента Бэклога получить Инкремент, соответствующий критериями приёмки **Acceptance Criteria (AC)**.

К моменту планирования очередного спринта верхние элементы Бэклога должны быть хорошо подготовлены, для чего в том числе проводятся мероприятия по уточнению Бэклога.

Бэклог Спринта

Цель Спринта, выбранные элементы Бэклога Продукта, а также план их реализации вместе называются Бэклогом Спринта.

Ритуалы в Скраме. Спринт

Спринт (Sprint) - непосредственно итерация разработки, объединяющая в том числе другие мероприятия.

Спринты представляют собой регулярные циклы продолжительностью до четырёх недель. Продолжительность Спринта определяется для всего периода разработки командой и должна быть оптимальной: с одной стороны, должен сохраняться фокус на цели и необходимая концентрация внимания, а с другой стороны, позволить выпустить Инкремент.

Простыми словами, если продукт достаточно простой, понятный и может быть разбит на относительно мелкие элементы, то можно двигаться короткими итерациями. Короткие итерации также полезны для знакомства с методологией и новой командой, так как можно на коротких дистанциях определить скорость работы, выявить недочёты процесса и скорректировать их в ближайшее время. Более длительные Спринты выбираются в случаях, когда невозможно декомпозировать элементы Бэклога без потери ценности. Тогда команда тщательно планирует более долгие итерации, но также достигает цели Спринта.

Инкремент Продукта

В результате Спринта должен появиться готовый к поставке Инкремент Продукта. Например, готовая страница сайта, раздел, новый фильтр или даже документ пользовательской документации (если это ценность для клиента).

Не нужно пытаться затолкать в Спринт целую версию продукта или огромный функциональный блок. Инкремент может быть полезен для пользователя, но выпускаться релизом с несколькими другими связанными доработками.

Некорректно также помещать в Спринт только сопроводительные или предварительные работы как развёртывание площадки, проработка дизайна или маркетинговой стратегии. Нужно помнить, что даже если большая часть задач потребует участия команды в других задачах, нужно попробовать реализовать нечто полезное и для клиента.

Объём Спринта

Не допускается продление Спринта или изменение объёма требований для Бэклога Спринта. При этом задачи на разработку по мере выполнения могут уточняться, дополняться, расширяться и закрываться с учётом новых вводных.

В команде должна вырабатываться привычка погружаться в интенсивную работу для достижения цели Спринта и не отвлекаться на прочие задачи. Исключением могут быть случаи, когда какие-либо поломки блокируют работу пользователя.

Если цель Спринта не достигается, то команда делает выводы относительно собственной производительности, выявляет препятствия, решает проблемы и корректирует подход к работе.

Задачи, которые не были выполнены, обсуждаются, пересматриваются и повторно оцениваются и планируются в следующих Спринтах.

Командная работа

Именно поэтому так важно, чтобы команда работала совместно и слаженно, чтобы конфликты не затягивались и чтобы не было частой смены сотрудников. А процесс улучшался за счёт достаточной мотивации и высокого профессионализма.

В таких командах компетенции и функции часто смешиваются и распределяются не только в соответствии с должностями, но и в соответствии с интересами, навыками и занятостью отдельных участников в рамках конкретного Спринта.

Ритуалы в Скраме. Ежедневный Скрам

Ежедневный Скрам (Daily Scrum meeting, или Standup meeting) – короткая ежедневная командная встреча, которая проводится для синхронизации прогресса.

В скрам-командах обычно можно услышать названия: «дэйли», «дэйлик», «стэндап», «статус», «синк», «созвон» и всякие похожие варианты сокращения англицизмов.

Участники встречи

Основные участники данной встречи - Разработчики команды. Это внутреннее мероприятие, которое не требует привлечения других стейкхолдеров. Владелец продукта участвует по возможности, а Скрам-мастер – в целом фигура опциональная и на первых встречах помогает соблюдать формат, который на самом деле должна поддерживать сама команда.

Формат встречи

Ранее большой акцент делался на встрече в конкретном месте офиса и только в положении стоя. Так это создавало некую ритуальность, а также подталкивало сотрудников проводить мероприятие быстрее, чем если бы они оставались на своём рабочем месте и могли отвлекаться на свои повседневные задачи.

Сегодня такие правила всё сложнее соблюдать и акцент вполне закономерно сместился на цели встречи, нежели на формат.

Тем не менее, прежними остались рекомендации участвовать полным составом, в одно и то же время, а также сохранять тайминг, то есть укладываться в 15-20 минут (важно уточнить, что рекомендуется формировать команды максимум из 8-10 человек).

Чтобы уложиться во временные границы встречи и действительно использовать это время с пользой, вместо свободного рассказа о своих задачах участники должны фокусироваться на нескольких конкретных вопросах:

- Что было сделано вчера? Как это помогло приблизить команду к цели Спринта?
- Что будет сделано сегодня? Как это поможет в достижении цели?
- Есть ли препятствия, вопросы, трудности? Нужна ли помощь?

На мероприятии Планирования Спринта команда распределяет конкретные задачи, даёт оценку и прогноз по выпуску доработок. Если кто-то из участников видит, что не справляется с исполнением плана, он должен сам в первую очередь обратить на это внимание команды, чтобы как можно раньше совместно найти способ исправить ситуацию.

Также формат требует наглядного представления Бэклога Спринта, поэтому в случае офиса рекомендовалось, например, использовать доску и стикеры. Но это лишь инструмент – вы можете актуализировать свои задачи в любой системе управления задачами (так называемых task-трекерах).

Цели встречи

Такая встреча **не** должна проводиться в качестве формального пребывания участников вместе или для демонстрации Владельцу продукта или руководству компании, как команда способна соблюдать правила.

Все участники должны синхронизировать свой прогресс и выявить возможные нежелательные отклонения от плана и скорректировать его. Речь не идёт о постоянном изменении Бэклога – это скорее возможность пересмотреть техническое решение, изменить исполнителя, помочь с конкретными трудностями друг другу и понимания статуса остальных участников.

Такое мероприятие не убирает необходимость других встреч или звонков, но помогает организовать эти последующие встречи.

В целом регулярные встречи улучшают коммуникацию и способствуют быстрому принятию решений, особенно в условиях удалённой работы.

Роли в Скраме. Скрам-мастер

В идеальных условиях команда самостоятельно может управлять процессами и инструментами, а каждый участник присоединяется к такой команде с достаточным опытом, знаниями и мотивацией. В реальности процесс становления самоорганизующейся команды занимает время, команда на собственном опыте проходит ошибки планирования и сложности коммуникации. Именно поэтому для поддержания желаемого качества процессов, организации всех мероприятий и настройки всех инструментов приглашается Скрам-мастер.

Скрам-мастер (Scrum Master) отвечает за применение методологии, соблюдение принципов Скрама и поддержание культуры Скрама в команде. Его роль состоит в некотором владении процессом и построении эффективной работы всей команды.

Командные мероприятия

Очевидно, что в обязанности Скрам-мастера входит проведение командных мероприятий, их организация, модерация и фасилитация.

Скрам-мастер лучше других ориентируется в ритуалах Скрама и может адаптировать формат под конкретные условия.

Встречи один на один

Однако Скрам-мастер также участвует и во встречах один на один (можно встретить написание «one-on-one», «one-to-one» и даже «1-2-1»). Важная работа происходит, когда Скрам-мастер индивидуально разговаривает с каждым участником команды. Это помогает прояснить роль участника, выявить скрытые проблемы, устранить недопонимания.

Внешнее взаимодействие

Скрам-мастер часто помогает справляться с внешними препятствиями и выстраивать эффективное взаимодействие команды с другими подразделениями. Это может быть общение со стейкхолдерами, которые иницируют работы или которые принимают результаты работы команды.

Работа Скрам-мастера тесно пересекается со многими задачами HR специалистов – действуя совместно, они стараются создать комфортные условия для работы сотрудников.

Администрирование инструментов

Настройка и администрирование инструментов **не являются прямой обязанностью** Скрам-мастера, однако это частая функция такого специалиста, о которой в компании или команде договариваются дополнительно.

Речь идёт не о профессиональных инструментах дизайна или разработки, а об инструментах для коммуникации, организации времени, управления задачами: это мессенджеры и программы для видеоконференций, таск-трекеры, почта, календари, списки задач. Скрам-мастер помогает согласовать правила их использования, администрирует группы, помогает оформить профили и даже шаблоны подписи.

Регламенты и совершенствование процесса

Все договорённости внутри команды и правила совместной работы обычно фиксируются в некоторые инструкции, которые можно назвать общим понятием регламента.

Регламенты полезны для погружения новых сотрудников в работу команды, а также в качестве помощи при определении ролей и обязанностей участников.

Нет смысла в излишне формальном описании, однако такие инструкции подсказывают, как действовать в той или иной ситуации, помогают сделать процесс понятным и прозрачным.

В задачи скрам-мастера входит также совершенствование процессов работы, что было бы невозможно без понимания текущего состояния процессов, анализа, отчётов и сбора обратной связи. Скрам-мастер может обновлять регламент, дополняя его новыми договорённостями, улучшающими процесс.

Настроение в команде

Направляя команду в продуктивное русло и акцентируя внимание на эффективности работы, Скрам-мастер также отслеживает общее моральное состояние команды.

Скрам-мастер помогает сделать сотрудничество позитивным и лёгким, поддерживает уважительное отношение в коллективе, обучает корректному взаимодействию в случае ошибок или проблем, поддерживает и мотивирует.

От требований к задачам. Бэклог Спринта

Планирование Спринта – это мероприятие, в ходе которого формируется Бэклог Спринта.

Бэклог Спринта не является однородной сущностью и объединяет:

- **Цель Спринта** (ценность для пользователя/клиента, которую планируется поставить в результате) → зачем делать?
- **Выбранные элементы Бэклога Продукта** (пользовательские истории) → что делать?
- **План реализации элементов или план разработки Инкремента** (задачи для реализации требований) → как делать?

Например, **цель спринта**: «Реализовать на странице сайта отображение списка пунктов приёма батареек с фильтром по городу»

Выбранный элемент из Бэклога Продукта (пользовательская история): «Я беспокоюсь за экологию и хочу знать, где находятся специализированные пункты в моём городе, чтобы сдавать отработанные батарейки»

Стоит обратить внимание, что уже для данной истории «напрашивается» декомпозиция: видеть список пунктов, видеть карту с метками пунктов, применить фильтр по городу, определить местоположение (то есть 4 отдельные истории).

В процессе перехода от требований к задачам обратите внимание на несколько рекомендаций:

1. На каждый пункт требований (или на каждый пункт критериев приёмки) должна быть сформулирована задача. По итогу несколько пунктов требований могут объединиться в одну задачу, но сам факт проговаривания задачи на каждое требование позволяет как проверить сами требования (а требования тоже могут быть некорректными), так и убедиться, что разработчик понимает, как его реализовать.
Критерий приёмки «При выборе в фильтре города обновляется список пунктов приёма батареек» преобразуется в **задачу**: «Реализовать GET запрос получения списка пунктов по городу».
2. Требование может затрагивать несколько архитектурных слоёв, например, базу данных, бэкенд и фронтенд. Для разных исполнителей указывайте отдельную задачу.
Критерий приёмки «Доступен выбор города из списка городов РФ. В списке отображается сначала 5 крупнейших городов, затем остальные по алфавиту – см. в приложении. По умолчанию – Москва» преобразуется в **задачи**: «Добавить справочник городов (отдельная таблица), обозначить признаком топ-5 городов и для каждого пункта приёма батареек указать идентификатор города (база данных)», «Передавать в ответе GET запроса отсортированный по алфавиту массив городов (бэкенд)», «Реализовать отображение списка всех городов в выпадающем списке с возможностью ввода названия города и фильтрации этого списка (фронтенд)». В спецификации должны быть указаны упомянутые города, например, в явном виде прописывается: «Список 5 крупнейших городов для быстрого выбора: Москва, Санкт-Петербург, Екатеринбург, Краснодар, Новосибирск».
3. Учитывайте зависимость задач друг от друга и необходимость планировать последовательность их выполнения.
Например, пока не будет конкретных данных о пунктах выдачи, которые можно загрузить в базу, список на странице сайта будет всегда пустым.
4. Используйте тестовые данные, в том числе для проверки пограничных случаев и их отображения на странице.

Например, можно придумать список пунктов приёма с очень длинными и очень короткими адресами, с фото и без фото.

5. Обсудите, как можно проверить результат работы.

Например, кто из других участников может выполнить ревью кода разработчика. Или проверить запрос в Postman, в том числе случаи ошибок. Или проверить корректность данных перебором всех городов РФ (а это более 1000 кликов), а может, сделать выгрузку из базы в таблицу и свериться с исходным списком в спецификации.

6. В задачах используйте формулировки с глаголами: разработать, реализовать, добавить, сформировать, обеспечить и так далее. Это помогает определить, что конкретно необходимо сделать исполнителю.

«Реализовать фильтр выбора города на странице списка пунктов», «Разработать карточку пункта приёма», «Добавить файл robots.txt».

Бэклог Спринта – это план работ, который в том числе должен иметь визуальное представление. Это может быть как доска со стикерами, так и многофункциональный таск-трекер.