Scrum — это спортивный термин, который пришел к нам из регби, и представляет собой фигуру, которую образуют игроки перед началом игры.

1. Почему появился Agile?

Теперь немного слов о том, как и зачем появился этот подход? История возникновения этого подхода стала ответом на запросы отрасли:

1. Заказчик не может сформировать четкие требования к ПО;

2. Новые технологии усилили конкуренцию и потребовали оперативного применения в бизнесе;

3. Заказчики и разработчики ПО не удовлетворены процессом взаимодействия.

#1 Заказчик не может сформировать четкие требования к ПО

В начальной фазе проекта заказчик не может сформулировать исчерпывающие требования к продукту. Этому есть несколько причин:

• у Заказчика существует только идея приложения и он не представляет всю его функциональность;

• у группы проекта есть разный взгляд на функциональность приложения;

В традиционных «водопадных» моделях руководитель проекта минимизирует изменения в проекте, используя для этого отдельные процессы — управление изменениями. Но если требования будут меняться раз в месяц, то управление изменениями становится трудоемким и замедляет ход проекта.

В Agile реакция на изменения важнее, чем следование плану. В Agile приветствуется, когда заказчик и пользователи вносят новые требования, чтобы сделать продукт более конкурентноспособным.

#2 Новые технологии усилили конкуренцию и потребовали оперативного применения в бизнесе

Подход Agile предоставил бизнесу главное преимущество – быстрые поставки новой функциональности. Это позволило каждый месяц выпускать продукт и оперативно получать обратную связь от пользователей.

#3 Заказчики и разработчики не удовлетворены процессом взаимодействия

При жестком сроке и в условиях постоянных изменений разработчики вынуждены формализовывать процессы взаимодействия с Заказчиком. Разработчики закладывают в бюджет работы по созданию детальных требований и спецификаций, а также риски на возможные их изменения. При этом Заказчик вынужден оплачивать документы, которые не несут реальной ценности для бизнеса.

Основная идея agile – сотрудничество с заказчиком важнее, чем контрактные обязательства. И поэтому agile-методы стремятся к уменьшению объема документации. Это позволяет Заказчику платить только за результат, имеющий ценность для бизнеса.

Как это работает:

1. Разработка идет отдельными этапами по принципу: выбрали на этап самые приоритетные функции → подготовили дизайн → сверстали → запрограммировали → запустили проект с минимальным функционалом → снова выбрали приоритетные функции… И так до последнего этапа.

2. В результате каждого этапа вы получаете полностью готовый, работоспособный проект, в котором реализованы самые важные с точки зрения вашего бизнеса функции. Который уже можно запускать.

3. Пока идет, например, первый этап разработки, вы можете пересмотреть свое видение проекта и что-то поменять в последующих. И эти изменения не будут дорогостоящими, как в «водопаде».

4. Каждый последующий этап будет наращивать функционал проекта, пока все функции не будут реализованы.

Артефакты в Scrum

В скрам используется всего четыре артефакта:

* Product Backlog
* Sprint Backlog
* Sprint Goal
* Sprint Burndown Chart.

Product backlog:

* Это список всех требований, которые нужно сделать по проекту. Когда в Backlog’e нет требований, проект считается завершенным.
* Все требования описаны по единому шаблону, который называют User Story (пользовательская история).
* Требования составлены так, что очевидно и понятно, какую ценность они представляют для пользователя
* Требования отсортированы по приоритетам, которые пересматриваются каждый спринт.

Sprint backlog:

* Это список всех требований, которые нужно сделать в ближайший спринт.
* В течение спринта, новые требования не могут появится в Sprint backlog.
* Все требования должны быть разделены на задачи и оценены.

Sprint Backlog — это обязательство команды: что они должны выполнить за ближайшие 2 недели. Каждое требование разделено на задачи, которые представлены на Kanban-доске.

Sprint Goal

* это краткое описание того, ради чего выполняется данный спринт.
* цель на спринт помогает команде принимать обоснованные решения.

Этот артефакт необходим для того, чтобы команда проекта могла самостоятельно принимать решение в случае появления альтернативных путей решения задачи. Чтобы решения команды были осознанными, Product Owner определяет цель спринта.

Sprint Burndown Chart

* дословно «диаграмма сгорания»
* в качестве «сгорающих» элементов выступают человеко-часы или идеальные единицы (Story Points).
* диаграмма обновляется каждый раз, когда завершается какая-либо задача.

На практике такая диаграмма очень наглядна: каждый день можно быстро узнать, насколько команда продвинулась вперед.

Р**оли в Scrum**

В скрам используется всего три роли:

* Product Owner
* Scrum Master
* Team.

### Роль Product Owner

* формулирует требования
* приоритезирует требования
* корректирует приоритеты на каждом спринте
* несет персональную ответственность за ценность требований для рынка/пользователей
* отвечает за взаимодействие с рынком
* только один человек

Product Owner — это представитель подразделения, которое владеет разрабатываемым продуктом. Эта роль требует сочетания следующих качеств:

* иметь личную вовлеченность в проект и его результаты;
* хорошо владеть навыком написания требований.

В некоторых случаях допустимо назначить более одного человека на роль Product Owner. Но в этом случае необходимо назначить среди них «главного», который будет авторизовать требования в Bcaklog’e и лично расставлять приоритеты.

### Роль Scrum Master

* следит за корректным применением принципов Agile и процессов (ритуалов) Scrum
* организует работу команды и обеспечивает её всем необходимым
* защищает команду, несёт ответственность за её эффективность
* только один человек.

Очень сложная роль. В классическом project management есть Руководитель проекта. В Scrum такая роль не предусмотрена. Лучшим синонимом роли Scrum Master будет «администратор». Скрам Мастер организовывает работу команды проекта, но не вмешивается в её работу.

* Скрам мастер не назначает людей на задачи — это делает сама команда;
* Мастер не заставляет людей делать работу — это ответственность команды;
* Мастер не указывает Product Owner какие требования он должен написать — это работа владельца продукта.

Тем не менее, если скрам-процесс проходит с нарушениями (кто-либо из команды опаздывает на daily-meeting), то мастер должен вмешаться и исправить ситуацию.

Функции Scrum Master’a существенно шире, но чтобы пояснить их все нужна отдельная статья. Пишите в комментариях, если таковая нужна.

### Team (команда проекта)

* кросс-функциональная
* взаимозаменяемая
* самоорганизующаяся
* с фиксированным составом (в ходе спринта)
* 4-10 человек.

Команда отвечает за разработку продукта итерациями (спринтами). Команда определяет самостоятельно:

* продолжительность спринта
* емкость (capacity) команды
* размер её фокус фактора (коэффициент слаженности)
* трудоемкость требований, которые будут реализованы в спринте
* очередность выполнения задач и много другое.

Команда НЕ принимает решений:

* какие требования являются приоритетными — это делает Product Owner.

### Ритуалы (процессы в Scrum)

В скрам есть несколько процессов, которые принято называть ритуалами. Каждый ритуал выполняется неукоснительно и в строгом соответствии с подходом. На практике такие процессы стараются немного адаптировать, но ключевые принципы не изменяют.

Ритуалы в скрам это:

* Sprint Planning Meeting
* Daily Meeting
* Sprint Review

### Sprint Planning Meeting (встреча по планированию спринта)

* выполняется всей командой перед началом спринта
* команда выбирает требования из Product Backlog и формирует Sprint Backlog
* если требуется учесть взаимосвязи между операциями, то это делается здесь
* команда декомпозирует требования на задачи (tasks)
* каждая задача проходит оценку в трудозатратах или универсальных единицах
* во время встречи Product Owner отвечает на вопросы команды.

стреча простая по сути, но крайне сложная по содержанию. В начале проекта может занимать 5-6 часов. И только после 3-4 спринта встреча становится более оперативной и длится 2-3 часа

### Daily Meeting (ежедневная встреча команды).

Из названия понятно, что встреча проводится ежедневно. Основные принципы:

* проходит ежедневно и только в одно и то же время;
* встреча проходит только стоя;
* поэтому длительность встречи не более 15 минут;
* чтобы успеть каждый должен ответить всего на 3 вопроса: что я делал вчера, чем я занимаюсь сегодня, какие есть проблемы?

Scrum Master следит за ходом встречи, побуждает участников высказываться полностью и слушать говорящего.

На ежедневной встрече команда обменивается опытом. Также становится понятно, кто и над какими задачами будет сегодня трудиться. Важно, чтобы команда делала этот ритуал самостоятельно. Я вообще рекомендую Scrum Masters не вмешиваться в ход встречи до тех пор, пока соблюдаются все требования к этому ритуалу.

### Sprint Review — сдача спринта Product Owner

По завершению каждого спринта команда обязана провести демонстрацию полученного результат.

Ценность Scrum для обычного заказчика во многом состоит в том, что результат работ (плохой или отличный, не важно) будет продемонстрирован в любом случае. Это знает и команда и Product Owner и другие заинтересованные лица. Если команда не проводит демонстрацию (иное название Sprint Review), то это дискредитирует все преимущества гибких процессов.

На встрече могут присутствовать любые сотрудники организации или просто заинтересованные лица. Важно, чтобы право голоса имели только участники Scrum процесса (Produt Owner, Team, Scrum Master).

## Преимущества методологии Scrum в создании сайтов и проектов

**Scrum** является адаптивной методологией, требующей вдумчивого применения, но в качестве ее безусловных преимуществ можно назвать:

* Возможность быстрого запуска проекта с наиболее приоритетными функциями и минимально возможным бюджетом;
* Ежедневный контроль над ходом работ, и более гибкий контроль над бюджетом проекта;
* Частые демонстрации проекта. Применение данной методологии предполагает регулярную демонстрацию разработок заказчику;
* Возможность вносить коррективы в техническое задание по ходу реализации проекта, что является несомненным преимуществом для заказчика.
* Гибкий бюджет. Он меняется в течение проекта, как в большую, так и в меньшую сторону.

## Недостатки методологии Scrum при разработке сайтов и проектов

Однако на сегодняшний день не существует системы полностью лишенной недостатков. К главным недостаткам методологии **Scrum** можно отнести:

* Сложности при заключении договоров. **Scrum** в принципе не подразумевает наличие фиксированного бюджета и фиксированного технического задания, что затрудняет юридическое оформление такого рода договоренностей. Также т.к. Scrum относится к семейству Agile, в Scrum не принято, к примеру, создание плана коммуникаций и реагирования на риски. [3] Таким образом, делая сложным или невозможным формальное (юридическое или административное) противодействие нарушениям правил Scrum.;
* Большое количество исключений. Специалисты в этой области считают данную методологию неприменимой для работы с государственными заказами, а также совершенно нерабочей при низкой квалификации команды, заниженных сроках работ или бюджете, некомпетентном менеджере проекта. В то время как другие методологии позволяют завершить проект при подобных условиях, хотя и на низком уровне;
* Другой слабой особенностью Scrum является упор на самоорганизующуюся, многофункциональную команду. При кажущемся снижении затрат на координацию команды, это приводит к повышению затрат на отбор персонала, его мотивацию, обучение. При определенных условиях рынка труда, формирование полноценной, эффективной Scrum команды может быть невозможным.