

[Scrum и Kanban](#) — это гибкие методологии создания продукта. По ним можно работать в любой отрасли, но особенно хорошо они подходят для ИТ. В основе обеих методологий лежат принципы Agile, о которых я писал [в предыдущей статье](#).

Чистый Scrum — более директивная методология — больше предписаний. Kanban — демократичнее, поэтому его проще внедрить.

Обе технологии близки, потому их инструменты можно комбинировать.

## Команды в Kanban и Scrum

Основа обеих методологий — [Agile](#), поэтому и в Scrum, и в Kanban работают небольшие автономные команды из 5—9 человек. В командах нет формального руководителя, и никто извне не диктует, как организовывать работу над продуктом.

Поскольку команда автономная и самоорганизующаяся, за успех или неудачу она отвечает как единое целое. Провал не свалить на ленивого разработчика или невнимательного тестировщика.

Обе методологии подразумевают, что команда располагается в едином пространстве. Лучше, если это не кубиклы, а общая комната. Главный принцип — свободное общение между специалистами и общие обсуждения.

А вот дальше в Kanban и Scrum начинаются различия.

**Kanban.** Над задачей может работать несколько узкопрофильных команд. К примеру, сначала работают аналитики, потом [дизайнеры рисуют прототип](#), а на третьем этапе включаются [разработчики](#).

При этом универсальные команды не запрещены.

В Kanban внутри команды нет ролей.

**Scrum.** Над проектом работает одна универсальная команда. В ней столько разноплановых специалистов, сколько нужно для решения любой задачи проекта.

Поскольку команда самоорганизуется, у специалистов scrum-команды нет формальной компетенции. Когда необходимо, тестировщик помогает дизайнеру, а аналитик — разработчику.

В scrum-команде помимо собственно специалистов есть две роли.

Scrum-мастер — человек, который организует работу. Это не управленческая должность, и он не раздает указания. Его задачи:

- вести собрания;
- устранять препятствия в работе (если команде мешает перфоратор в соседнем офисе, мастер ищет выход);
- замечать и вытаскивать на поверхность скрытые проблемы;
- отвечать за соблюдение методологии;
- следить за статусом задач.

В свободное от этих задач время скрам-мастер работает так же, как другие члены команды.

Владелец продукта — [product owner](#) — определяет ход проекта, он может представлять внешнего заказчика. Владелец знает все о рынке и целевой аудитории. Он ставит приоритеты задачам. Результат работы команда представляет владельцу продукта.

## Как составляют список задач

Для начала команда берет проект и делит его на десятки, а то и сотни задач поменьше. Это часть философии Agile, поэтому так делают и в Kanban, и в Scrum.

Все задачи проекта, которые предстоит выполнить, складывают в общий список — бэклог. Бэклог — это банк задач проекта. Каждая задача должна быть актуальна. Если потребуется, их можно добавлять в бэклог или удалять из него «на лету».

**Каждая задача имеет вес — обычно это время, которое нужно на решение. Команда сама оценивает вес всех задач, поэтому если проект не закончен в срок, виновата команда.**

Также у каждой задачи есть приоритет. В Kanban приоритеты расставляет команда, в Scrum — владелец продукта. Приоритеты можно и даже нужно пересматривать по ходу проекта — это один из столпов гибких методологий.

## Как работают над проектом

Когда начинается работа, Scrum становится понятно, почему его называют намного более директивной методологией.

Scrum-команда разбивает время работы над проектом на равные отрезки — спринты. Спринт может длиться и день, и месяц, а в последние годы стандартом стал спринт в 2 недели.

Поскольку все спринты одинаковы по длительности, в работе команды появляется ритм. Ритм — важный аспект методологии.

Спринт состоит из четырех последовательных этапов.

- **Планирование.** Команда проверяет задачи в бэклоге и выбирает самые приоритетные. На спринт берут столько задач, сколько успеют сделать.
- **Выполнение.** В идеальной команде специалисты работают параллельно: пока программист создает код, тестировщик пишет к нему тесты, а технический писатель — документацию.
- **Релиз.** Команда представляет результаты своей работы миру. К моменту каждого релиза продукт должен быть работоспособным, полезным для пользователя и более совершенным, чем до спринта. Больше про эти требования я написал [в статье про Agile](#).
- **Ретроспектива.** Команда обсуждает спринт и возникшие проблемы. Все вместе думают, как улучшить работу и сделать в следующем спринте больше.

В Scrum категорически нельзя добавлять задачи в текущий спринт, поэтому Scrum менее гибок. Даже если появилась срочная и важная задача, она пойдет в работу только со следующего спринта.

В конце спринта недоделанные задачи уходят обратно в бэклог. Нужно ли их доделывать и когда, определяют на этапе планирования следующего спринта.

## **По сравнению со Scrum-директивами Kanban — это оплот либерализма и хаос.**

Спринтов как таковых нет. Проект обычно делят на итерации, но они могут быть любой длины. Ритмичность Kanban не предписывает.

Вспомним этапы scrum-спринта:

- Планирование.
- Выполнение.
- Релиз.

- Ретроспектива.

В Kanban один этапы не завязаны на другие. Наступают они, когда решит команда. К примеру, релиз — по вторникам, планирование — когда закончились задачи, ретроспективы — каждый последний четверг месяца, а на фоне всего этого непрерывно идет разработка.

Поскольку выраженных спринтов нет, появляются особенности:

- новые задачи добавляют в любое время. Если нужно срочно что-то сделать, команда не ждет следующего спринта;
- задача остается в работе сколько угодно долго, пока команда не закончит ее или не отменит.

## Что за доски используют

Доска — это сердце kanban- и scrum-разработки, единственный визуальный атрибут методологий. Доски первым делом вешают на стену, когда хотят показать, что работают по Kanban или Scrum.

Доску используют, чтобы сделать проект прозрачнее, распланировать задачи и поставить ограничения.

Доску расчерчивают на столбцы. Каждый столбец — это состояние задачи («Разработка», «Тестирование», «Релиз»). Количество столбцов зависит от проекта, но чем их меньше, тем лучше.

Карточки — это задачи. На каждой описание, вес и приоритет. Когда задача проходит очередной этап, ее переклеивают в соответствующий столбец. При простом взгляде на доску понятно, как дела с проектом в целом и с каждой задачей.

Доски бывают физические и электронные.

Физическая доска. Часто доска — это действительно доска с клетками из малярного скотча. Задачи — липкие листочки, которые удобно двигать по доске.

Настоящая доска стоит в комнате и каждый в команде видит, как идут дела. Вокруг настоящей доски хорошо собираться во время совещаний.

Электронная доска. Электронная доска под рукой на пляже, на даче, в машине и в метро. Если у вас есть сотрудники, которые работают удаленно, электронная доска — единственный выход.

Я люблю настоящие доски, но без электронных иногда не обойтись.

## Какие ограничения соблюдают

В Scrum число задач, которые одновременно находятся в работе, ограничено их общим весом. Если известно, что команда делает за спринт 26 условных единиц, значит, общий вес задач на следующий спринт не может превышать 26.

**В Kanban число активных задач ограничено их весом по каждому статусу отдельно. При этом неважно, каков общий вес задач на доске.**

Посмотрите на канбан-доску. Цифра рядом с названием столбца — это и есть ограничение. Для простоты я предполагаю, что все задачи примерно равны по трудоемкости. Поэтому цифра обозначает, сколько задач может быть в колонке одновременно.

Колонки можно дробить на более мелкие, чтобы процесс работы был прозрачнее. При этом ограничение накладывают на родительскую колонку.

Ограничение колонки «Разработка» — не больше трех задач. Сейчас колонка заполнена, поэтому программисты не могут взять новую работу из бэклога. Казалось бы, бред: задача решена, карточки в «Готовом» а специалисты простаивают.

Теперь обратите внимание на соседнюю колонку. Интеграторы завалены работой и не могут вытянуть у разработчиков готовые задачи. Если разработчики возьмут новую работу, пробка станет еще плотнее. Вот почему в колонке «Разработка» ограничение для текущих и готовых задач общее.

Kanban предписывает, что в такой ситуации разработчики перемещаются к интеграторам и вместе с ними расчищают завал. Только когда в «Интеграции» появится место, разработчики протолкнут туда задачу из «Готово» и возьмут новую.

В результате команда добивается не максимального числа незавершенных задач, а максимальной скорости прохождения задачи по доске. Если говорить о продукте, скорость появления новых возможностей важнее числа возможностей в разработке.

## Настройка ограничений — важнейшая часть работы по Kanban.

Предположим, люди простаивают и не могут взять новые задачи, потому что колонка забита. В этом случае нужно смотреть, что происходит справа. Если справа заторов нет, значит, ограничение колонки слишком строгое.

Другой вариант — задачи простаивают, и среднее время выполнения снижается. Так бывает, когда на доске слишком много задач, и за них некому взяться. Если заметили симптом, делайте строже ограничения колонок.

Kanban предписывает экспериментировать. Поменяли ограничение на колонку — посмотрели, что получилось. И так все время.

## Какие показатели измеряют

В Scrum измеряют общий вес задач, выполненных за спринт. Разделив общий вес всех задач проекта на производительность за спринт, мы получим примерный срок окончания проекта. Задача команды в Scrum — повышение производительности.

В Kanban измеряют среднее время прохождения задачи по доске. Это время — показатель эффективности команды. Показатель дает интересный эффект: команда не концентрируется на выполнении конкретных задач. Она просто следит, чтобы среднее время выполнения было минимальным.

## Как внедряют

Перейти к Agile проще через Kanban. У этой методологии меньше ограничений, можно не формировать кросс-функциональные команды, а по-прежнему делиться на отделы. Scrum можно попробовать, когда почувствуешь суть гибкой методологии.

1. Найдите настольную игру getkanban и поиграйте в нее с коллегами. Официальная версия стоит \$450, но в интернете есть и другие варианты.

2. Если игра понравилась и вы прочувствовали методологию, возьмите любой проект и поделите его на мелкие задачи. Удобно, когда все задачи примерно равны по времени выполнения. Назначьте приоритет для каждой.
3. Подумайте, какие стадии проходит любая задача. В простейшем варианте это может быть «Запланирована → В работе → Завершена». Вариант посложнее: «Запланирована → Дизайн → Разработка → Интеграция → Готово». Статусы зависят от того, что именно вы делаете.
4. Проставьте ограничения на статус задач так, как подсказывает логика. Если у вас два дизайнера и три программиста, пусть в колонке дизайна будет максимум 2 задачи, а на доске «В разработке» — 3.
5. Проведите первое совещание. Обсудите, как организуете работу. Когда будете релизить продукт, а когда собираться на совещания.
6. Перенесите несколько задач из бэклога в «Запланировано» и запускайте процесс.
7. Следите, чтобы карточки путешествовали между колонками, когда задача перешла из одного статуса в другой.
8. Замеряйте среднее время выполнения задачи. Добавляя карточку на доску, пишите на ней время начала работы, а снимая — время завершения.
9. Экспериментируйте с ограничениями и рабочим процессом так, чтобы среднее время выполнения задачи уменьшалось.
10. Собирайтесь на ретроспективные собрания, где вы обсуждаете работу над проектом: что прошло хорошо, что получается плохо, как можно улучшить работу.

Мой опыт говорит, что сходу внедрить Kanban и Scrum не получится. Мы привыкли работать по каскадной методологии, поэтому сразу перейти на гибкую не выйдет. Тут важно не опускать рук, быть настойчивым и вдохновлять команду.

Помните, что доска на стене — это еще не Kanban или Scrum, на этом этапе нельзя останавливаться. Налаживайте общение между коллегами, делайте людей командой. На самом деле что для Kanban, что для Scrum взаимодействие важнее практик.

**Главное и в Kanban, и в Scrum — это люди.**