

# Практические занятия 9-10: План проекта и руководство по работе команды по методологии Scrum

|                        |  |
|------------------------|--|
| ДИСЦИПЛИНА             | Технологии управления командами разработчиков программного обеспечения |
| ИНСТИТУТ               | Институт перспективных технологий и индустриального программирования   |
| КАФЕДРА                | Кафедра индустриального программирования                               |
| ВИД УЧЕБНОГО МАТЕРИАЛА | Практические занятия   |
| ПРЕПОДАВАТЕЛЬ          | Зарипова Виктория Мадияровна   |
| СЕМЕСТР                | 3 семестр, 2025-2026 гг.   |

## Тема

Разработка и уточнение плана проекта, изучение принципов Agile и Scrum и применение их в руководстве командой разработчиков.

### Цель занятий

Освоение методологии Agile и Scrum.

### План занятий

- Составить план проекта и RACI
- Знакомство с методологией Scrum
- Выбор шаблона и заполнение его данными по своей команде и проекту
- Создание руководства для команды о работе по методологии Scrum

### Входные знания

Студентам нужно знать основы методологии Scrum и иметь навыки командной работы

### Итоги работы

Студенты изучат и научатся применять основные инструменты методологии Scrum и разьяснять их команде.

### Практические итоги работы и отчетные материалы

- План проекта в системе <https://app.clickup.com/>
- Матрица RACI
- Презентация для команды и заказчиков о принципах использования гибкой методологии Scrum на проекте. Презентация должна давать ответы на следующие вопросы:
  - Краткое описание проекта
  - Состав команды
  - WBS проекта
  - План проекта

- Почему в вашем проекте выбран именно Scrum
- Какие инструменты Scrum используете, в том числе какова периодичность спринта, как часто проводить ретроспективы и т.д. С обоснованием в зависимости от сложности и временных рамок вашего проекта.
- Шаблон описания задачи и жизненный цикл задачи. Заложить периоды на разработку и тестирование. Указать критерии готовности задачи
- Жизненный цикл спринта. Указать критерии готовности спринта. Указать что делать в случае, если время спринта вышло, но не все задачи выполнены и как использовать диаграмму сгорания работ в этом случае.
- Шаблон Release Note для заказчика
- RACI
- Презентация должна давать команде представление о том как они начинают работу в начале каждого спринта, кто и за что отвечает, как справляться с возникающими рисками и что делать в конце спринта, зачем нужна ретроспектива и как ее результаты будет использовать команда.
- Презентация должна давать заказчику представление о том кто и как и когда будет работать с командой заказчика и в каком формате, как и когда можно вносить новые требования, на основании чего принимаются решения о включении той или иной функциональности в спринт. Как методология удовлетворяет интересы заказчика.
- Студенты также должны быть готовы продемонстрировать описанные ими принципы на примере одного из шаблонов Coda.

**Справочный материал:**

- См. лекции и прилагаемый материал

# Что такое SCRUM

SCRUM – это проверенный и эффективный Agile фреймворк для успешного управления проектами. Используется как в крупных корпорациях, так и в стартапах.

Scrum разработали японские эксперты по менеджменту [Хиротака Такеучи](#) и [Икуджиро Нонака](#). В 1986 году они призвали менеджеров проектов перейти от традиционного последовательного метода разработки новых продуктов к более гибкому и быстрому подходу к разработке продуктов, в большей степени управляемому командой.

Название фреймворка, вдохновленное регби, напоминает о команде, которая объединяется, чтобы вместе решать сложные проблемы.

## Кратко о Scrum

Scrum (методология Agile) – это методология управления проектами, которая используется для разработки программного обеспечения и других продуктов. Она основана на принципах гибкой разработки и предлагает итеративный и инкрементальный подход к достижению целей проекта.

В Scrum проект разбивается на небольшие временные интервалы, называемые спринтами. Каждый спринт обычно длится от одной до четырех недель. В начале каждого спринта команда определяет, какие задачи они будут выполнять и какой результат они хотят достичь к концу спринта. Затем команда работает над выполнением этих задач, проводит ежедневные совещания для обсуждения прогресса и решения проблем, и в конце спринта представляет готовые результаты и проводит командную ретроспективу, чтобы оптимизировать процесс для перехода к следующему спринту.

Scrum отличается от других подходов, таких как Waterfall и Kanban, своей фокусировкой на гибкости и адаптивности. В Waterfall проект разбивается на последовательные этапы, а в Kanban используется система визуальных досок для отслеживания задач. В Scrum команда имеет большую свободу и ответственность в выборе и выполнении задач, что позволяет более эффективно реагировать на изменения и достигать поставленных целей.

В итоге, Scrum является эффективным инструментом для управления проектами, особенно в сфере разработки программного обеспечения. Он позволяет командам работать более гибко, быстро достигать результатов и улучшать их качество.

## Scrum побуждает команды:

- Учиться на основе накопленного опыта и постоянной обратной связи
- Разбивать массивные задачи на более мелкие выполнимые части
- Быть самостоятельными и иметь свободу подходить к проблеме предпочтительным для себя методом.
- Гибко переключаться между несколькими функциями
- Ставить приоритеты именно на задачи критического пути проекта и выполнять их

## Кто использует фреймворк Scrum?

Фреймворк Scrum широко распространен в командах разработки программного обеспечения. Однако он достаточно гибкая, чтобы применяться к любой форме командной работы.

Он используется в различных отраслях, таких как гостиничный сервис, строительство, консалтинг и финансовые услуги, и т.д..

## Agile vs Scrum: в чем разница?

Agile - это методология, представляющая из себя набор руководящих принципов, в то время как Scrum - это фреймворк или набор практических инструментов для управления проектами в рамках Agile.

Для образности: если Agile - это компас, указывающий команде правильное направление, то Scrum - это проводник по этому пути, предлагающий конкретные способы достижения цели. Оба они работают вместе для успешной реализации и завершения проекта.

Scrum - не единственный фреймворк сделанный по принципам Agile. Среди других популярных фреймворков Agile можно выделить Kanban.

Другие различия между Agile и Scrum:

- **Результат:** В Scrum результаты проекта предоставляются в конце каждого спринта. В Agile результаты предоставляются по окончании всего проекта.
- **Роли в команде:** В команде по проекту Scrum предусмотрены определенные роли, включая Scrum-мастера и владельца продукта. В Agile отсутствуют выделенные роли и любой член команды может выступить в роли ведущего, а роль владельца продукта размыта между PM, аналитиками и иногда лидами.

## Популярность SCRUM объясняется несколькими ключевыми причинами:

1. Scrum одинаково выгоден для всех участников процесса: стейкхолдеров (Stakeholders), конечных пользователей и команды. Благодаря частым итерациям команда всегда остается в контакте с заказчиком и быстро адаптирует продукт к требованиям заказчика, даже если они изменяются в процессе.
2. Scrum разбивает сложные проекты на атомарные задачи. Перед лицом большой цели мы часто чувствуем себя подавленно, и нас останавливает страх начать. В SCRUM это учитывается: проект разбивается на небольшие, управляемые блоки, что помогает команде оставаться сфокусированной и мотивированной и выполнять работу быстрее.
3. Scrum повышает моральный дух и поощряет самостоятельность. SCRUM опирается на внутренние мотиваторы человека, узнает потребность в достижении, положительной обратной связи и самостоятельном выполнении назначенной работы в рамках целостной командной среды. В SCRUM-команде нет традиционного начальника, которому бы нужно отчитываться или который бы говорил, что делать. Задачи внутри команды организуются самым эффективным образом.
4. Scrum повышает мораль команды, поддерживая потребность в достижении успеха, положительной обратной связи и самостоятельности. Роль PM сводится от тактической к стратегической и члены команды получают большую самостоятельность в выборе задач, а PM адекватную оценку ситуации, рисков и возможных решений от команды.

## Шесть ценностей и принципов Scrum

Успешность внедрения фреймворка Scrum часто определяется тем, насколько хорошо команда воплощает шесть ценностей и принципов:

1. Эмпирический процесс
2. Самоорганизация
3. Кросс-функциональное сотрудничество
4. Приоритизация
5. Установка временных рамок (Time-boxing)
6. Итеративный процесс

### 1. Эмпирический процесс

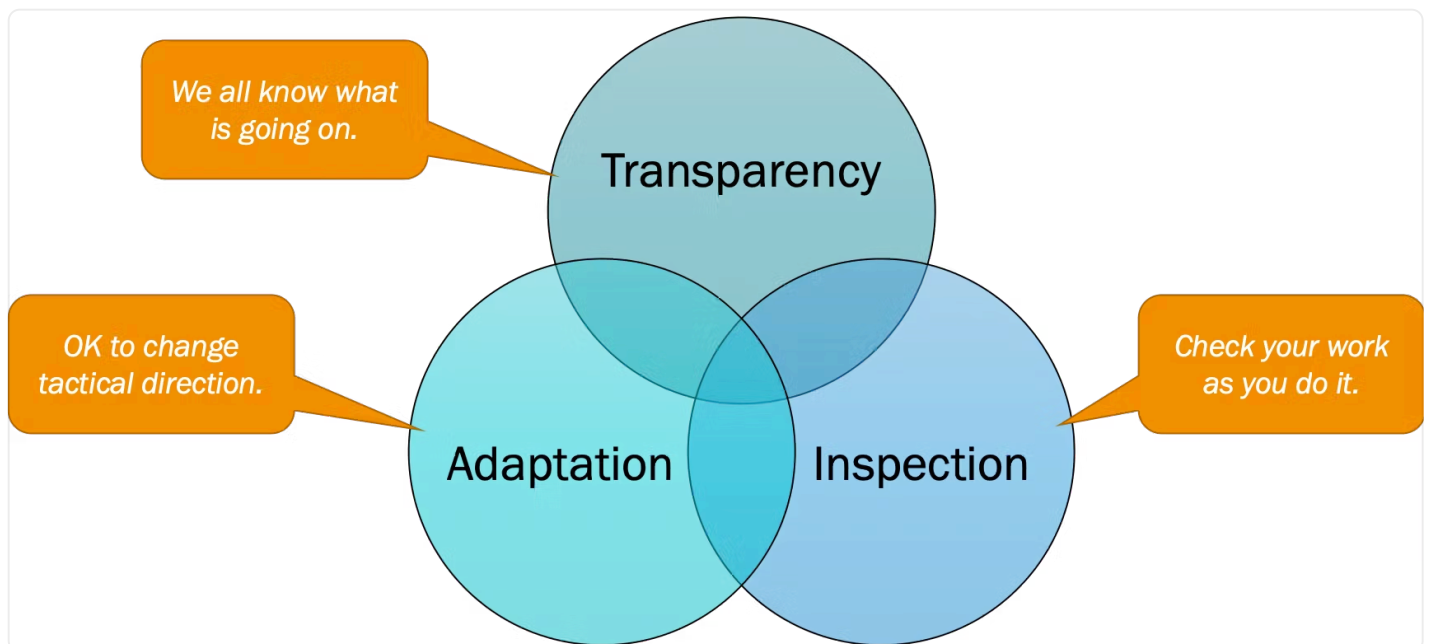
| Лишь опыт является источником знаний

Эмпиризм в Scrum означает фактографическую, основанную на опыте и доказательствах работу.

Прогресс в Scrum основывается на адаптации к реальности, а не на соблюдении плана. Фреймворк фокусируется на выполненной работе и адаптации к реальным условиям и задачам заказчика.

Эмпирический процесс обеспечивается тремя ключевыми принципами:

1. Прозрачность: Все участники, включая клиентов и команду, осведомлены обо всех аспектах проекта. Команда делится всей информацией о происходящем в проекте, как хорошей, так и плохой. Следует избегать скрытых планов или искажения фактов.
2. Сверка результатов: По окончании спринта команда Scrum демонстрирует достигнутые результаты и сверяет их с ожиданиями заказчика.
3. Адаптация: Команда Scrum постоянно адаптирует и улучшает свой продукт благодаря регулярной сверке результатов. Цель команды не выполнение плана, а непрерывное улучшение продукта, что в конечном итоге приводит к ощутимым деловым результатам, таким как ускорение времени выхода на рынок и повышение удовлетворенности клиентов.



Источник: [Scrum.org](https://www.scrum.org)

## 2. Самоорганизация

В Scrum команды работают независимо и организуют свою работу самостоятельно. Они определяют, кто, когда и как будет выполнять задачи, с минимальным внешним участием. Это означает, что команда берет на себя ответственность за достижение целей спринта и сама решает, как ей лучше работать. Все члены команды совместно несут ответственность за результаты.

Руководство по Scrum подчеркивает, что если вы хотите [добиться успехов в Scrum](#), не мешайте команде самой определять что, как, и когда делать.

## 3. Кросс-функциональное сотрудничество

В Scrum все члены команды работают вместе, объединяя разные навыки и знания для достижения общей цели. Предполагается что команда уже является самодостаточной и может работать эффективно без привлечения внешних специалистов.

Каждая команда Scrum состоит из:

- Владельца продукта, который определяет видение продукта
- Scrum-мастера, который обеспечивает соблюдение принципов и процессов Scrum
- Разработчиков, которые специализируются в различных областях, в зависимости от сложности проекта.

#### **4. Приоритезация**

В Scrum команда постоянно ранжирует задачи по приоритету таким образом, чтобы как можно быстрее и наилучшим способом достигнуть целей, поставленных бизнесом.

При ранжировании задач в бэклоге продукта, команда учитывает:

- Ценность, которую задача приносит бизнесу
- Риски или неопределенности, связанные с задачей
- Зависимости, которые могут возникнуть при выполнении задачи

Все это помогает команде эффективно управлять своим временем и ресурсами, работая над наиболее важными задачами.

#### **5. Установка временных рамок (Time-boxing)**

Установка временных рамок - это практика выделения фиксированного времени для выполнения задачи. Это помогает определить и ограничить время, выделяемое на выполнение открытых или неоднозначных задач для всей команды.

Практика установления временных рамок помогает командам:

- Четко определить, когда задача считается выполненной
- Создать временные ограничения для задач, стимулируя команду более эффективно выполнять работу. Каждый спринт должен приводить к результату, который можно продемонстрировать заказчику
- Определить объем для всех событий Scrum. Например, длительность спринта ограничена от 1 до 4 недель, в зависимости от организации. Сессия по планированию спринта не должна занимать более двух часов.

#### **6. Итеративный процесс**

Команды Scrum следуют итеративному процессу разработки своих продуктов. Они разбивают разработку большого проекта на меньшие, более управляемые части.

Каждая итерация строится на основе предыдущей, постепенно увеличивая сложность рабочего продукта до получения готового к использованию заказчиком приложения или результатов проекта.

Некоторые преимущества итеративного процесса:

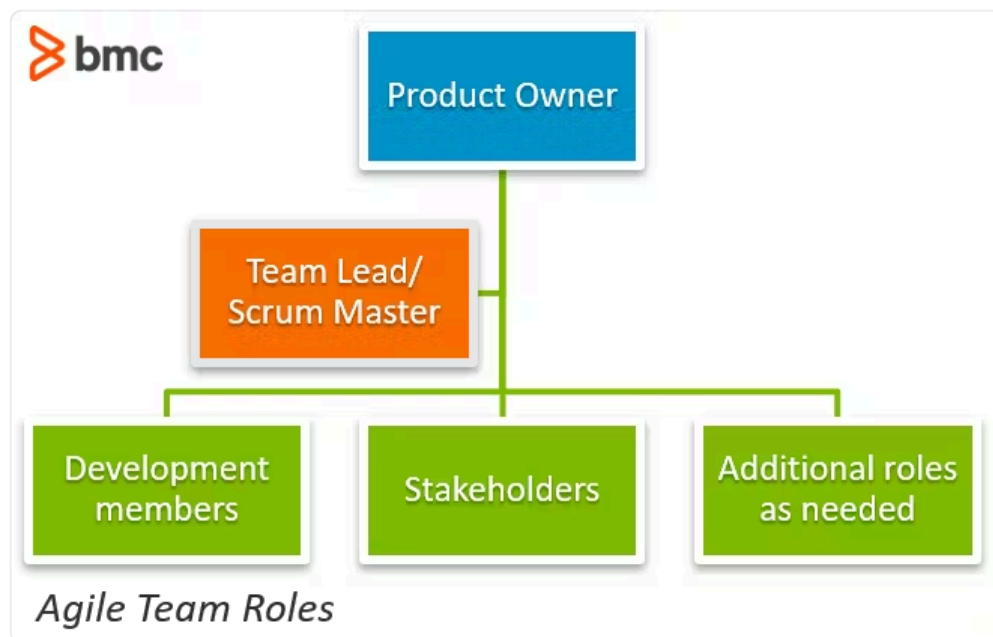
- Позволяет корректировать курс во время процесса: С каждым циклом спринта команда лучше понимает, что требуется, и может использовать эти знания в будущих спринтах.
- Повышает мотивацию команды во время работы: С каждой итерацией команда создает продукт, который лучше подходит для конечного заинтересованного лица или заказчика.

### **Какие ключевые роли в команде Scrum?**

Несмотря на то что команды Scrum могут адаптировать свои роли в соответствии со своими потребностями, в целом возможно пять ролей.

- Скрам-мастер или руководитель команды (Scrum master or Team Lead)
- Владелец продукта (Product owner)

- Команда Scrum (Scrum team)
- Заинтересованные стороны (Stakeholders)
- Другие члены команды (Other team members)



Source: [BMC](#)

## Владелец продукта

Владелец продукта выступает в роли 'CEO' продукта и определяет направление разработки и прогресса продукта. Он является связующим звеном между всей командой Scrum и заинтересованными сторонами бизнеса, учитывая потребности стейкхолдеров и долгосрочное видение бизнеса, и передает эти требования скрам-мастеру и команде Scrum.

Владельцы продуктов управляют бэклогом продукта и расставляют приоритеты элементов бэклога на основе бизнес-потребностей.

## Скрам-мастер

Скрам-мастер действует как наставник и менеджер проекта для команды Scrum. Он принимает указания от владельца продукта и назначает задачи команде для достижения правильных результатов. Он ориентирует команду на эффективное решение проблем и следит за соблюдением процессов Scrum, включая планирование спринта, обзор и ретроспективу спринта.

Скрам-мастер регулярно встречается с командой, чтобы обсудить прогресс, обучить членов команды, устранить препятствия, с которыми сталкивается команда Scrum, и поощрить их самостоятельность и ответственность. Он также помогает организовывать встречи с владельцем продукта и поддерживает актуальность информации о проекте.

## Команда Scrum

Основная роль команды Scrum - выполнять и завершать работу, назначенную на время спринта проекта скрам-мастером и владельцем продукта. Приветствуется самостоятельность и ответственность. Скрам-мастер также дает команде полномочия для решения возникающих проблем.

Команды Scrum состоят из специалистов различных направлений, от дизайнеров продуктов и специалистов по UX до специалистов по контролю качества и front-end/back-end разработчиков.

Включая владельца продукта и скрам-мастера, команда должна состоять не более чем из девяти человек и нести ответственность за свой результат как единая команда.

## Как работает методология Scrum?

В спринте задействован каждый участник команды. Типичный цикл спринта в методологии Scrum выглядит примерно так как на рисунке ниже.



Image source: [Atlassian Agile Coach](#)

### 1. Спринты

"Спринт - это короткий период времени, в течение которого команда Scrum работает над выполнением заданного объема работы." - Atlassian Agile Coach

Спринты являются центральной частью методологии Scrum и гибкого подхода. Подобно самому Scrum, они разбивают большие проекты на более маленькие и управляемые части работы. Спринты помогают командам выпускать готовый к рынку продукт быстрее и чаще, позволяя при этом гибко адаптироваться к изменениям.

Спринты могут длиться от 1 до 4 недель, но обычно они длятся две недели. Каждый спринт состоит из четырех событий:

1. Планирование спринта (Sprint Planning): Во время этого события команда определяет, что они могут достичь в течение следующего спринта, и создает план работы, набирая задачи из беклога продукта.
2. Работа над задачами и Ежедневные встречи (Daily Scrum): Команда работает над задачами из беклога спринта и встречается каждый день на короткой сессии, чтобы обсудить прогресс и планы на ближайший день. Не надо воспринимать эту встречу как отчет по плану. Основная цель таких встреч дать участникам возможность подать сигнал что что-то не так.
3. Обзор спринта (Sprint Review): В конце каждого спринта команда показывает то, что они создали в течение спринта.
4. Ретроспектива спринта (Sprint Retrospective): После обзора спринта команда встречается, чтобы обсудить, что прошло хорошо, что плохо, и что они могли бы сделать лучше в следующем спринте.

Вот несколько готовых шаблонов Coda для планирования проектов по Scrum:



- [Scrum project template](#): простой шаблон для управления маленькими и средними проектами
- [Sprint backlog template](#): шаблон с возможностью подсчета баллов
- [Scrum Sprint Planning Template](#): шаблон с возможностью группового обзора
- [Flexible Project Management Communication Plan Template](#): шаблон с планом коммуникации
- [Project Charter Template](#): шаблон для ведения нескольких проектов
- [Project Milestone Template](#): шаблон с возможностью учета майлстоунов
- [Critical Path Methodology Template](#): шаблон с задачами критического пути
- [Sprint Planning Gallery](#): галерея шаблонов Coda связанных с планированием спринта

## 2. Планирование спринта

Планирование спринта происходит в начале текущего спринта и проводится с участием всей команды с обсуждением того, что должно быть выполнено в текущем спринте и как это сделать. Вся сессия планирования спринта не должна занимать более двух часов.

Что происходит во время типичной сессии планирования спринта:

- Необходимо обновить элементы бэклога продукта (PBIs, Project Backlog Items). Владелец продукта должен прийти с результатами обзора прошлого спринта, обратной связью от стейкхолдеров и бизнеса и видением следующего спринта.
- Владелец продукта и разработчики обсуждают, какие элементы бэклога продукта (PBIs) включить в предстоящий спринт.
- Владелец продукта определяет общую цель спринта и приоритизирует элементы бэклога на основе требований стейкхолдеров и обстоятельств проекта для достижения данного плана.
- Разработчики обсуждают, возможно ли выполнение этих элементов, и, если необходимо, поднимают проблемы. Они учитывают доступное время и ресурсы и соглашаются с ожидаемыми результатами спринта.

**Идеальный результат:** команда может описать цель спринта и как именно эта цель будет закрыта задачами спринта. Цель или цели должны быть достижимыми и реалистичными..

## 3. Ежедневный стендап (Daily Scrum)

Daily Scrum - это встреча со всей командой (включая скрам-мастера и владельца продукта), где каждый дает отчет по текущему статусу своих задач и проверяет прогресс команды. Это короткая встреча, которая не должна занимать больше 15 минут и должна проводиться в удобное для всех время.

Цель этих встреч:

- своевременно предупредить команду о трудностях
- повысить мотивацию команды и члена команды, успех которого отмечают
- понять, если член команды застрял при решении какой-то из задач и коллективно придумать решение или быстро найти экспертов, внутри команды, которые могут помочь

Что происходит во время типичной встречи Scrum:

Каждый участник делится своими успехами, трудностями с которыми он столкнулся, и тем, над чем собирается работать сегодня. Команда также использует бэклог продукта для измерения и отслеживания своего прогресса.

Ранее руководства по Scrum содержали три вопроса:

- Над чем я работал вчера?
- Над чем я планирую работать сегодня?

- Есть ли у меня какие-либо проблемы или трудности, которые мешают прогрессу?

Однако помните что данные вопросы могут восприниматься вашей командой как слишком формальные. По факту с 2020 года они были исключены из рекомендаций SCRUM в основном потому что из-за них встреча напоминала отчет о статусе, а также они акцентировали внимание на отдельных участниках команды, а не на взаимодействии команды в целом.

Важно осознавать, что ежедневная встреча Scrum – это событие, предназначенное для координации планов разработчиков на ближайший день. Стоит рассмотреть возможность дать разработчикам самим формировать идеальный для них формат проведения такой встречи.

Вот альтернативный подход к проведению "ежедневников" Scrum:

1. Обзор итерационной цели

- Для команд, использующих Kanban: обзор краткосрочных целей

2. Пройдитесь по доске статусов справа налево – обсудите задачи, ближайшие к завершению

- Это позволяет сфокусироваться на работе и том, что вы можете сделать как команда, чтобы завершить что-то сегодня!

3. Поощряйте парную работу и "swarming" (когда больше одного человека работают над одной задачей), задавая вопросы вроде:

- "Кто может вмешаться и выполнить эту задачу сегодня или завтра?"
- "Кто может помочь им закончить это быстрее?"
- "Представляет ли этот пункт рабочего списка возможность для обучения кого-то еще?"

4. Убедитесь, что есть план действий для всех препятствий обозначенных командой и по этому плану назначен исполнитель. Иначе говоря, нужно построить механизм, при котором препятствия быстро и эффективно устраняются с учетом ресурсов команды.

5. Ближе к 15-минутной отметке, завершите встречу вопросами, типа:

- "Работает ли кто-то над задачами, не отображенными на нашей доске?"
- "Есть ли у нас хороший план действий на сегодня для достижения цели итерации?"
- "Мешает ли что-то нашей командной работе и продвижению к цели?"

Обратите внимание, что цель итерации фигурирует в начале и в конце встречи в предложенном новом формате.

**Желаемый результат:** Каждый член команды чувствует себя воодушевленным и мотивированным. У них есть четкое представление о том, над чем они работают и над чем работают их коллеги по команде. Если они сталкиваются с какими-либо проблемами, владелец продукта или скрам-мастер должны быть готовы помочь преодолеть препоны.

## 4. Обзор спринта

Обзор спринта обычно проводится в конце спринта. На обзоре спринта внимание сосредотачивается на команде Scrum, которая описывает и демонстрирует выполненную работу за прошедший спринт. Обычно на этих встречах присутствуют владелец продукта, скрам-мастер и вся команда Scrum. При необходимости могут присоединиться внешние заинтересованные лица.

Что происходит во время обычного обзора спринта:

- Участники команды показывают, что они сделали за спринт. Это может выглядеть как неформальные демонстрации продукта и проверка рабочих функций, созданных в течение спринта.
- Необходимо в обязательном порядке отметить успехи как всей команды, так и каждого ее участника. Это дополнительный мотивирующий фактор.

Обзоры спринта должны быть положительным событием для команды и способствовать поднятию ее боевого духа. Однако, если обзор спринта не привел к хорошему результату, скрам-мастер или владелец продукта должны это отметить. У каждой команды случаются слабые спринты, но задавайте вопросы, чтобы оценить и найти решение.

- Берет ли команда на себя слишком много работы?
- Меняет ли владелец продукта приоритеты во время итераций и тем самым путает команду?
- Можно ли улучшить рабочие практики команды?
- Борется ли команда с существующим техническим долгом? (Например, уделяется больше внимания скорости релиза, чем качеству кода)

Примечание: Обзор спринта отличается от ретроспективы спринта. В обзоре спринта фокусируются на выполненной в спринте работе. Ретроспектива спринта помогает посмотреть на спринт с более высокого уровня и обсуждает итоги спринта для улучшения будущих спринтов на процессуальном или стратегическом уровне.

Вот шаблон Coda для планирования ретро: [sprint retrospective](#).

## Артефакты Scrum

В разработке программного обеспечения термин «артефакт» относится к важной информации, необходимой во время разработки продукта. Scrum-команды используют эту информацию, чтобы реализовать на практике основные принципы Scrum: прозрачность, проверку и адаптацию. Это также позволяет команде понять, насколько хорошо на самом деле работает спринт.

The team uses Scrum artifacts to:

- Plan work and goals
- Create the daily tasks needed to achieve these goals
- Organize sprints according to priority or identify dependencies
- Execute daily tasks
- Review and analyze results from these tasks and readjust where needed

Команда использует артефакты Scrum для того чтобы:

- Планировать работу и цели
- Создавать ежедневные задачи, необходимые для достижения этих целей.
- Организовывать спринты в соответствии с приоритетом или определять зависимости.
- Выполнять ежедневные задания
- Просматривать и анализировать результаты выполнения этих задач.

### 1. Бэклог продукта

The product backlog functions as a comprehensive to-do list for the Scrum team for the entire project.

It lists desired new features, enhancements, bug fixes, and requirements to complete the project. Owned and maintained by the product owner between sprint cycles, the product owner should consult external sources and critical stakeholders to complete the list.

Журнал невыполненных работ по продукту представляет собой исчерпывающий список дел для Scrum-команды на весь проект.

В нем перечислены желаемые новые функции, улучшения, исправления ошибок и требования для завершения проекта. Бэклог принадлежит и поддерживается либо владельцем продукта, либо РМ и должен обновляться между циклами спринта на основании консультаций с командой заказчика и заинтересованными лицами.

Пример бэклога [Coda's product backlog template here](#).

### 2. Бэклог спринта

Журнал невыполненной работы спринта содержит список задач, над которыми Scrum-команда решила работать во время спринта. Для каждой задачи дается оценка усилий на выполнение, обычно выражаемая в story points (т.е. дословно оценка пользовательской истории, которую реализует условный программист в некоторых баллах сложности, при этом лид команды знает как конвертировать эти баллы в рабочее время в зависимости от компетенции членов команды).

Управляет и владеет бэклогом спринта команда, и каждый член команды может взять любую задачу из бэклога и отчитаться по ходу работ на ежедневной встрече. Если после спринта остаются незавершенные задачи, они добавляются обратно в бэклог проекта и решаются во время следующего спринта.

A good sprint backlog also includes:

- Time estimates for individual tasks
- What does the 'definition of done' looks like for each task

Хороший бэклог спринта также включает в себя:

- Оценку времени для отдельных задач
- Критерии оценки задачи, т.е. по каким критериям программист должен оценить и понять что он достиг поставленной в задаче цели

**Визуализируйте, используя Scrum board. Создайте собственную с помощью одного из шаблонов выше или выберите шаблон по вкусу из галлеии [Coda's template](#).**

### 3. Product increments

Also known as sprint goals, product increments are the usable end-product from a sprint. Items that meet the Scrum team's definition of achieving a milestone or completing a sprint goal become product increments.

Product increments are part of the continuous feedback ethos of the Scrum methodology. Investing in new product features in small amounts helps teams receive and adapt to any feedback regularly. They also have a concrete way to measure progress using this method.

### 3. Инкремент продукта

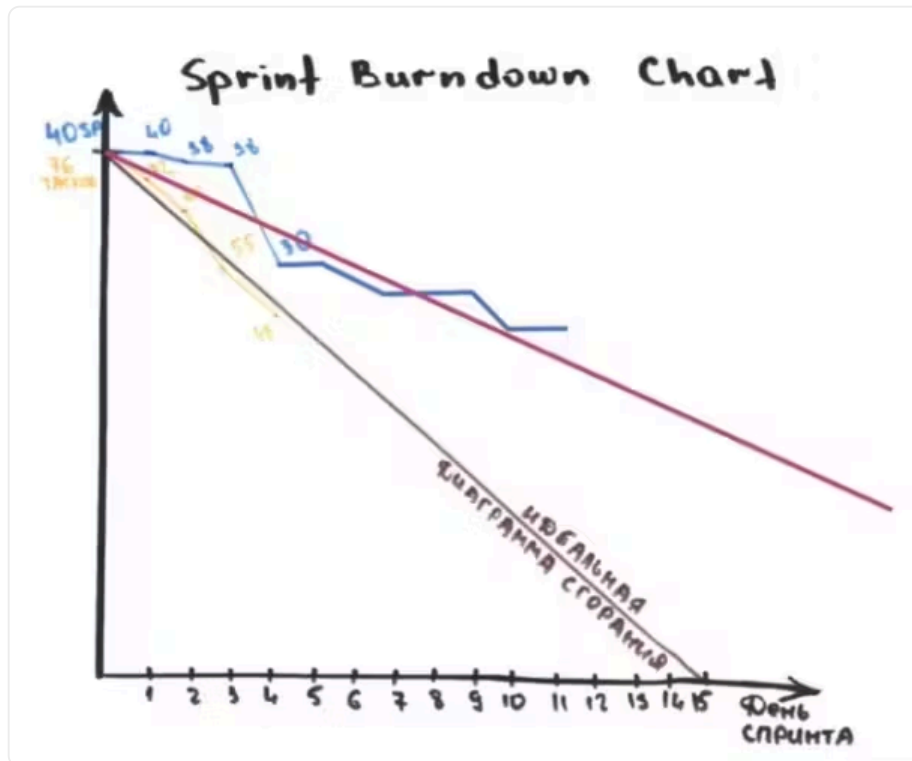
Инкременты продукта, также известные как цели спринта, представляют собой полезный конечный продукт спринта. По сути каждый инкремент добавляет бизнес-ценности конечному продукту и поэтому 80% задач спринта должны быть нацелены на создание инкремента.

Поскольку инкремент также представляется заказчику в ходе Sprint Review он должен быть законченным и действующим функциональным блоком. В рамках одного спринта можно создать несколько инкрементов.

### 4. Диаграмма Сгорания Работ Спринта (Sprint Burndown Chart)

Диаграмма сгорания легко показывает, сколько работы осталось выполнить Scrum-команде, и помогает оценить, когда работа будет завершена. Обычно вертикальная ось показывает количество работы, а горизонтальная время.

Этот инструмент эффективен за счет того, что прогресс оценивается не в контексте потраченного времени, а в контексте оставшейся работы. Рекомендуется, чтобы оставшаяся работа уменьшалась более-менее равномерно по ходу Спринта (см. идеальную прямую на рисунке ниже). Это ведет к уменьшению числа параллельных задач, стимулирует команду помогать друг другу ради скорейшего перехода задач в «Готово» и в итоге ведет к повышению эффективности Команды.



Преимущества использования диаграммы в методологии Scrum:

- \* Обеспечивает работу команды по графику.
- \* Сравнивает запланированную работу с тем, как продвигается команда.
- \* Обеспечивает мониторинг изменения объема проекта.

Если к середине Спринта график еще не пошел вниз, Команде необходимо срочно переходить на стратегию аварийной ситуации. Скрам-мастеру при этом важно помогать Команде действовать проактивно, не дожидаясь, когда Спринт закончится провалом.

[Смотрите пример в шаблоне Coda](#)

## Определение "Готово"

Что на самом деле означает «готово» для Scrum-команды? На первый взгляд ответ кажется простым, но он имеет решающее значение для вашего рабочего процесса по методологии Scrum.

Скрам-командам нужен способ определить, выполнена задача или нет. «Готово» может означать много разных вещей в зависимости от организации. Завершен ли проект, когда он запущен в производство? Или он завершен когда выпускается релиз и выполнены все задачи из лога изменений?

Инженер-программист Дэнни Смит предлагает один из способов определения готовности:

- История должна соответствовать критериям приемки, определенным владельцем продукта.
- Все, что указано в списке задач для этой истории, должно быть выполнено.
- Проверка кода завершена.
- Тестирование завершено.
- Код готов к немедленному развертыванию.

## Шесть вопросов, которые помогут вам решить, когда использовать Scrum-управление проектами

## **1. Вы четко знаете требования проекта?**

Гибкий и итеративный подход Scrum хорошо работает в проектах без четких требований. Применяйте его, когда ваш клиент или ключевая заинтересованная сторона имеет общее представление о том, что они хотят видеть, но не знают, как этого добиться.

## **2. Ожидаете ли вы изменений в процессе разработки?**

Если вы считаете, что существует высокая вероятность изменений, несмотря на четко определенные требования, используйте методологию Scrum. Итеративный процесс и регулярные циклы проверки облегчают вашей Scrum-команде работу с новыми условиями или изменениями в течение жизненного цикла проекта.

## **3. Вам нужно промежуточное тестирование или обратная связь?**

Рабочий процесс Scrum позволяет быстро обрабатывать отзывы о готовых рабочих продуктах и основывать будущие итерации на основе отзывов. Используйте эту методологию, когда планируете промежуточное тестирование ваших рабочих продуктов во время их создания.

## **4. Достаточно ли у вашей Scrum-команды опыта, чтобы самостоятельно управлять своим временем?**

Стиль работы Scrum основан на независимом принятии решений и самомотивации. Обычно лучше всего это работает с членами команды, обладающими определенной инициативой и стажем. Помимо Scrum-мастера, члены команды предоставлены сами себе при принятии решения о том, как выполнить работу.

## **5. Есть ли у вас люди, которые могут взять на себя роль владельца продукта и Scrum-мастера проекта?**

Это критически важные роли для успеха Scrum-команды.

## **6. Знакома ли ваша организация или клиент с процессом Scrum?**

У клиентов или заинтересованных сторон могут возникнуть вопросы о затратах или о том, сколько времени потребуется на разработку чего-либо. На этот вопрос сложно ответить, поскольку Scrum процветает благодаря изменениям без определенной конечной точки. Scrum работает лучше всего, когда существуют взаимные отношения доверия и подотчетности, и есть простой способ продемонстрировать прогресс или эффективность Scrum-команды с течением времени.