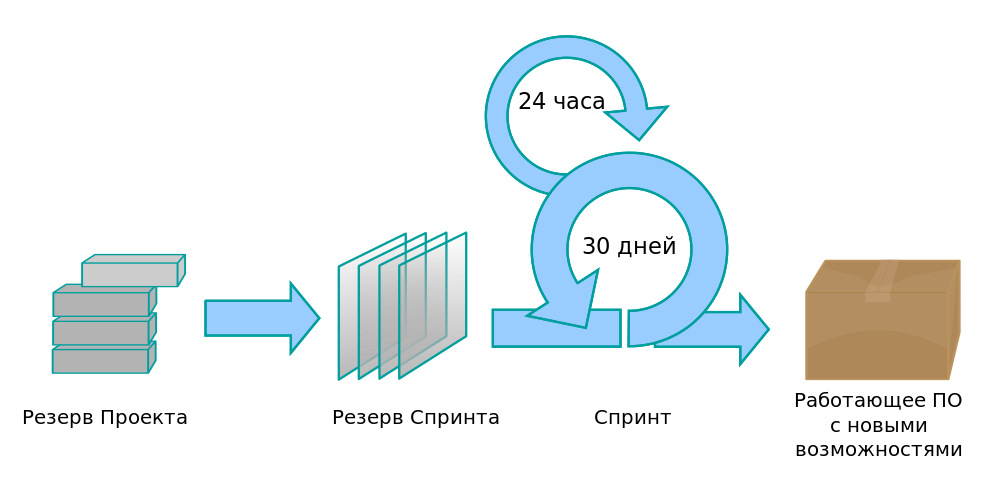
**Scrum** (от [англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *scrum* «толкучка») — методология управления проектами, активно применяющаяся при разработке информационных систем для [гибкой разработки программного обеспечения](https://ru.wikipedia.org/wiki/%D0%93%D0%B8%D0%B1%D0%BA%D0%B0%D1%8F_%D0%BC%D0%B5%D1%82%D0%BE%D0%B4%D0%BE%D0%BB%D0%BE%D0%B3%D0%B8%D1%8F_%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8). Scrum чётко делает акцент на качественном контроле процесса разработки. Кроме управления проектами по разработке ПО Scrum может также использоваться в работе команд поддержки программного обеспечения (*software support teams*), или как подход управления разработкой и сопровождением программ: *Scrum of Scrums*.

Скрам (Scrum) — 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**

Спринт — итерация в скраме, в ходе которой создаётся функциональный рост программного обеспечения. Жёстко фиксирован по времени. Длительность одного спринта от 2 до 4 недель. В отдельных случаях, к примеру согласно Scrum стандарту Nokia, длительность спринта должна быть не более 6 недель. Тем не менее, 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 и т. п. Разные команды подбирают длину спринта согласно специфике своей работы, составу команд и требований, часто методом проб и ошибок. Для оценки объема 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 (Project backlog)**

**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.

**Бэклог спринта (Sprint backlog)**

**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 объем работы, который нужно проделать для завершения спринта[[11]](https://ru.wikipedia.org/wiki/Scrum#cite_note-schwaber-11).

**Диаграмма сгорания задач (Burndown chart)**

Основная статья: [**Диаграмма сгорания задач**](https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0_%D1%81%D0%B3%D0%BE%D1%80%D0%B0%D0%BD%D0%B8%D1%8F_%D0%B7%D0%B0%D0%B4%D0%B0%D1%87)

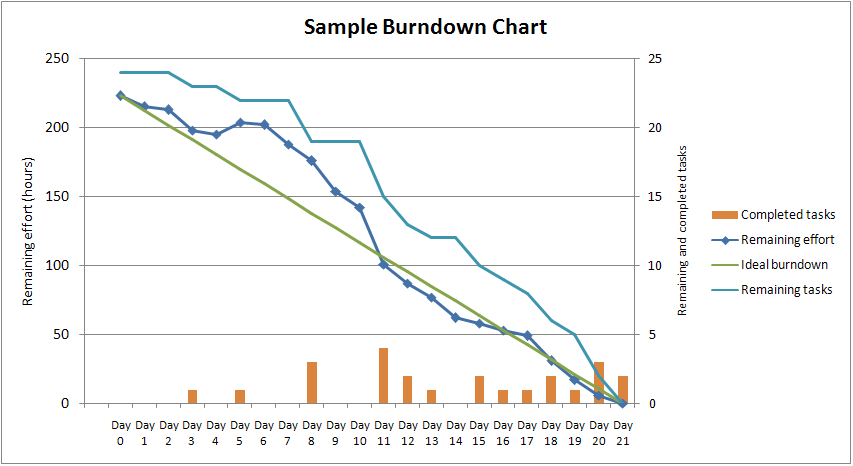


Диаграмма отображает завершенный спринт. Показывает оставшиеся нерешенные задачи и трудозатраты, необходимые для их завершения в расчете на 21 рабочий день.

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.

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 нескольких спринтов).

### История спринта (Sprint Story)

Требуемую функциональность, которую добавляют в бэклог, часто называют историей. Зачастую история имеет следующую структуру: «Будучи пользователем <тип пользователя> я хочу сделать <действие>, чтобы получить <результат>». Такая структура удобна тем, что понятна как разработчикам так и заказчикам.

### Остановка спринта (Abnormal Termination)

Остановка спринта может быть произведена раньше срока его планового окончания в исключительных ситуациях. Спринт может остановить команда, если понимает, что не может достичь цели спринта в отведённое время. Спринт может остановить **владелец проекта**, если исчезает необходимость в реализации цели спринта. После остановки спринта проводится совещание с командой, где обсуждаются причины остановки. После этого начинается новый спринт.

### Покер планирования (Planning Poker)

### Очки за пользовательскую историю (Story Points)

Абстрактная метрика оценки сложности истории, которая не учитывает затраты в человекочасах. Обычно используют одну из следующих шкал: [ряд Фибоначчи](https://ru.wikipedia.org/wiki/%D0%A0%D1%8F%D0%B4_%D0%A4%D0%B8%D0%B1%D0%BE%D0%BD%D0%B0%D1%87%D1%87%D0%B8) (1,2,3,5,8,13,21,34,55); линейную шкалу (1,2,3,4 … n); степень двойки (1,2,4,8 … 2n); размеры одежды (XS, S, M, L, XL).

### Задачи истории спринта (Sprint Story Tasks)

Добавляются к историям спринта. Выполнение каждой задачи оценивается в часах. Каждая задача не должна превышать 12 часов (зачастую команда настаивает, чтобы максимальная продолжительность задачи равнялась одному рабочему дню).

### Критерий готовности (Definition of Done (DoD))

Критерии, определяющие степень готовности элемента из *журнала пожеланий пользователя*.

### Скорость команды

Общее количество очков, набранных командой за предыдущий спринт. Данная метрика помогает команде понять, сколько историй она может сделать за один спринт.

## Роли в скрам-процессе

По методике Scrum в производственном процессе есть определённые роли, разбитые на 2 группы «свиней» и «кур». Эти названия были использованы из-за шутки[[12]](https://ru.wikipedia.org/wiki/Scrum#cite_note-autogenerated1-12)

Свинья идёт по дороге. Курица смотрит на неё и говорит: «А давай откроем ресторан!» Свинья смотрит на курицу и отвечает: «Хорошая идея, и как ты хочешь его назвать?» Курица думает и говорит: «Почему бы не назвать „Яичница с беконом“?». «Так не пойдёт, — отвечает свинья, — ведь тогда мне придётся полностью посвятить себя проекту, а ты будешь вовлечена только частично».

*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 (Core roles) 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 7±2 человека. 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 (Ancillary roles) 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*)**

## Артефакты

### Обязательные поля

* **ID** — уникальный идентификатор, порядковый номер, применяемый для идентификации историй в случае их переименования.
* **Название (Name)** — краткое описание истории. Оно должно быть однозначным, чтобы и разработчики, и **владелец проекта** могли понять, о чем идёт речь и отличить одну историю от другой.
* **Важность (Importance)** — степень важности данной истории, по мнению **владельца проекта**. Обычно представляет собой натуральное число, иногда для этой цели используются [числа Фибоначчи](https://ru.wikipedia.org/wiki/%D0%A7%D0%B8%D1%81%D0%BB%D0%B0_%D0%A4%D0%B8%D0%B1%D0%BE%D0%BD%D0%B0%D1%87%D1%87%D0%B8). Чем больше значение, тем выше приоритет.
* **Предварительная оценка (initial estimate)** — начальная оценка объёма работ, необходимого для реализации истории по сравнению с другими историями. Измеряется в story point’ах. Приблизительно соответствует числу «идеальных человеко-часов».
* **Как продемонстрировать (how to demo)** — краткое пояснение того, как завершённая задача будет продемонстрирована в конце спринта. Данное поле может представлять собой код автоматизированного теста для приёмо-сдаточного испытания.

### Дополнительные поля

Иногда, также, используются дополнительные поля в *бэклоге проекта*, в основном для того, чтобы помочь *владельцу проекта* определиться с его приоритетами.

* **Категория** (*track*). Например, «панель управления» или «оптимизация». При помощи этого поля **владелец проекта** может легко выбрать все пункты категории «оптимизация» и установить им низкий приоритет.
* **Компоненты** (*components*) — указывает, какие компоненты (например, база данных, сервер, клиент) будут затронуты при реализации истории. Данное поле состоит из группы checkbox’ов, которые отмечаются, если соответствующие компоненты требуют изменений.
* **Инициатор запроса** (*requestor*). *владелец продукта* может захотеть хранить информацию о всех заказчиках, заинтересованных в данной задаче. Это нужно для того, чтобы держать их в курсе дела о ходе выполнения работ.
* **ID в системе учёта дефектов** (*bug tracking ID*) — если вы используете отдельную [систему отслеживания ошибок](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D0%BE%D1%82%D1%81%D0%BB%D0%B5%D0%B6%D0%B8%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F_%D0%BE%D1%88%D0%B8%D0%B1%D0%BE%D0%BA), тогда в описании истории полезно хранить ссылки на все дефекты, которые к ней относятся.

## Встречи

### 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 и т. п.
  + (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 сделано с момента предыдущего ежедневного совещания?
* Что будет сделано с момента текущего совещания до следующего?
* Какие проблемы мешают достижению целей 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.