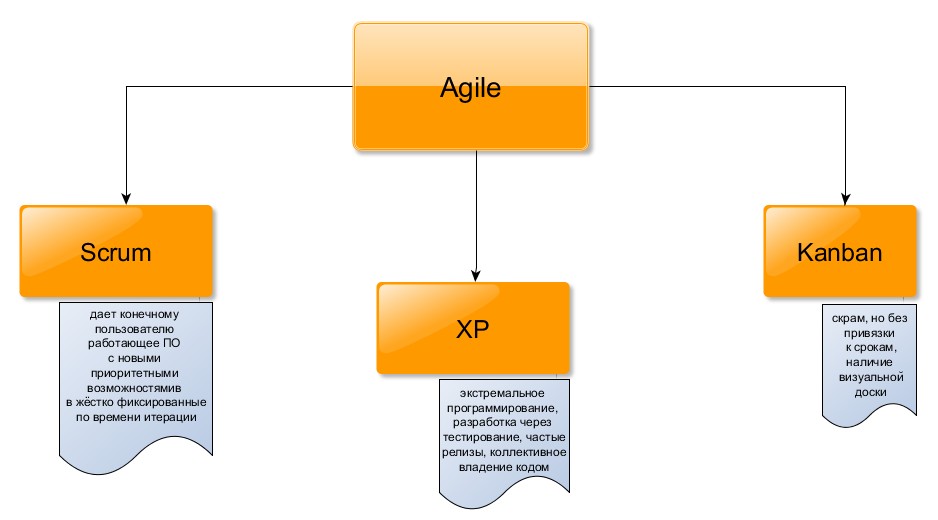
**4.1 Agile**

**Agile –** семейство гибких методологий разработки.



[**The Agile Manifesto**](http://www.agilealliance.org/the-alliance/the-agile-manifesto/)

 **Личности и их взаимодействие** важнее, чем процессы и средства

 **Работающее ПО** важнее, чем исчерпывающая документация

 **Сотрудничество с заказчиком** важнее, чем обязательства по контракту

 **Реагирование на изменения** важнее, чем строгое следование плану

**Ценности Agile**

 Гибкость и простота

 Частые релизы

 Самоорганизующаяся команда

 Больше общения

 удовлетворение клиента за счёт ранней и бесперебойной поставки ценного программного обеспечения;

 приветствие изменений требований даже в конце разработки (это может повысить конкурентоспособность полученного продукта);

 частая поставка рабочего программного обеспечения (каждый месяц или неделю или ещё чаще);

 тесное, ежедневное общение заказчика с разработчиками на протяжении всего проекта;

 проектом занимаются мотивированные личности, которые обеспечены нужными условиями работы, поддержкой и доверием;

 рекомендуемый метод передачи информации — личный разговор (лицом к лицу);

 работающее программное обеспечение — лучший измеритель прогресса;

 спонсоры, разработчики и пользователи должны иметь возможность поддерживать постоянный темп на неопределённый срок;

 постоянное внимание улучшению технического мастерства и удобному дизайну;

 простота — искусство не делать лишней работы;

 лучшие технические требования, дизайн и архитектура получаются у самоорганизованной команды;

 постоянная адаптация к изменяющимся обстоятельствам.

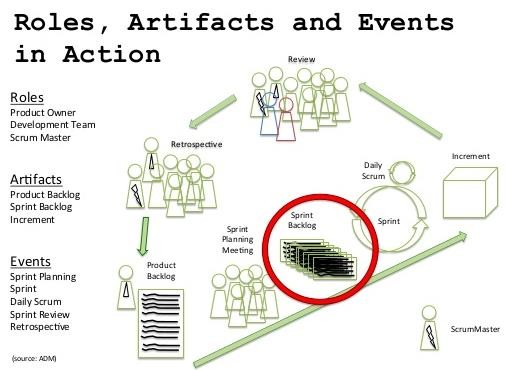
**4.2 Scrum**

Scrum - одна из самых популярных методологий гибкой разработки. Одна из причин ее популярности - простота. В Scrum всего три роли.

– Scrum Master

– Product Owner

– Team



Скрам Мастер (Scrum Master) - самая важная роль в методологии. Скрам Мастер отвечает за успех Scrum в проекте. По сути, Скрам Мастер является интерфейсом между менеджментом и командой. В Agile команда является самоорганизующейся и самоуправлямой.

Основные обязанности Скрам Мастера таковы:

– Создает атмосферу доверия,

– Участвует в митингах

– Устраняет препятствия

– Делает проблемы и открытые вопросы видимыми

– Отвечает за соблюдение практик и процесса в команде

Скрам Мастер ведет Daily Scrum Meeting и отслеживает прогресс команды при помощи Sprint Backlog, отмечая статус всех задач в спринте.

Product Owner - это человек, отвечающий за разработку продукта. Как правило, это product manager для продуктовой разработки, менеджер проекта для внутренней разработки и представитель заказчика для заказной разработки. Product Owner - это единая точка принятия окончательных решений для команды в проекте

Обязанности команды таковы:

– Отвечает за оценку элементов бэклога

– Принимает решение по дизайну и имплементации

– Разрабатывает софт и предоставляет его заказчику

– Отслеживает собственный прогресс (вместе со Скрам Мастером).

– Отвечает за результат перед Product Owner

Размер команды ограничивается размером группы людей, способных эффективно взаимодействовать лицом к лицу. Типичные размер команды - 7 плюс минус 2.

Артефакты

*Product Backlog* - это приоритезированный список имеющихся на данный момент бизнес-требований и технических требований к системе.

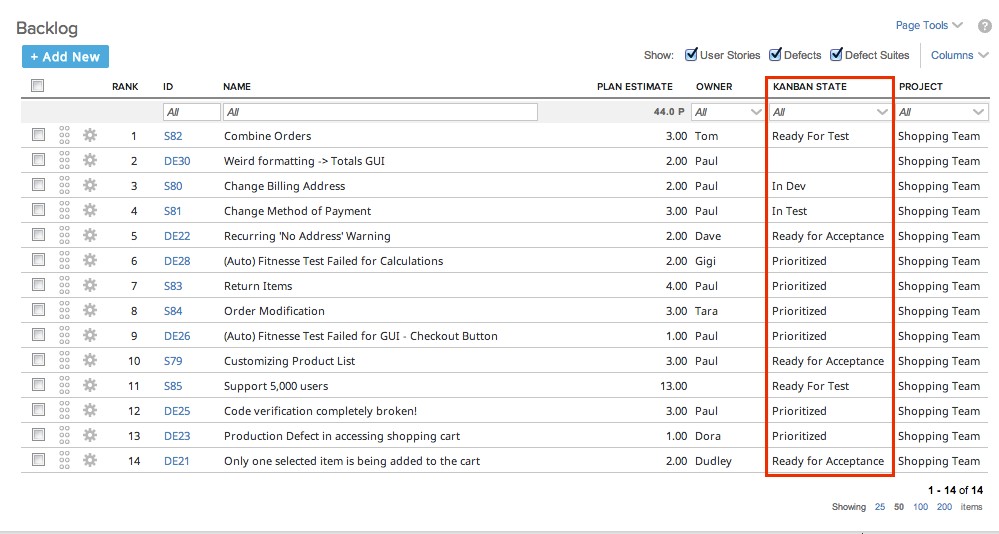
Product Backlog постоянно пересматривается и дополняется - в него включаются новые требования, удаляются ненужные, пересматриваются приоритеты. За Product Backlog отвечает Product Owner. Он также работает совместно с командой для того, чтобы получить приближенную оценку на выполнение элементов Product Backlog для того, чтобы более точно расставлять приоритеты в соответствии с необходимым временем на выполнение.

 Содержит список функциональных единиц системы (“user stories”), запланированных на след релиз

 Product backlog один на весь проект

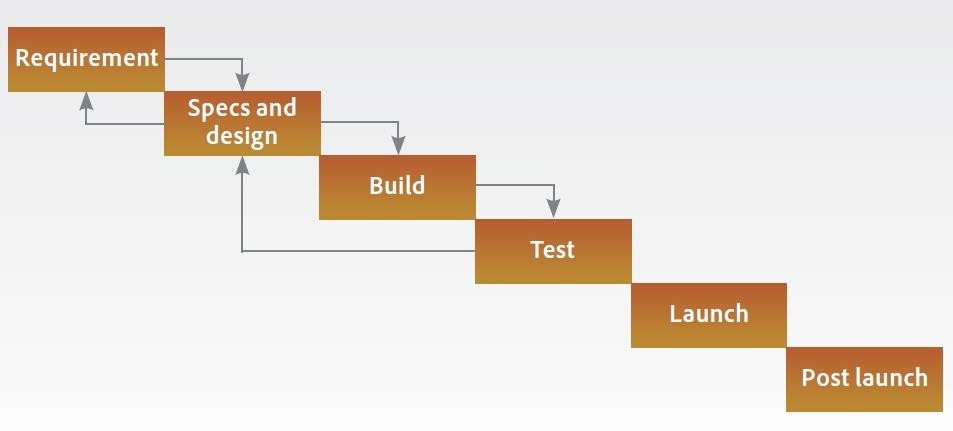
 Он не статичен – записи можно добавлять, удалять, менять им приоритет

 Общедоступен, но поддерживается одним человеком, владельцем продукта (“**product owner**”)



*Sprint Backlog* содержит функциональность, выбранную Product Owner из Product Backlog. Все функции разбиты по задачам, каждая из которых оценивается командой. Каждый день команда оценивает объем работы, который нужно проделать для завершения задач.

В Scrum итерация называется *Sprint*. Ее длительность составляет 3-6 недель.



Результатом Sprint является готовый продукт (build), который можно передавать (deliver) заказчику (по крайней мере, система должна быть готова к показу заказчику).

Короткие спринты обеспечивают быстрый feedback проектной команде от заказчика. Заказчик получает возможность гибко управлять scope системы, оценивая результат спринта и предлагая улучшения к созданной функциональности. Такие улучшения попадают в Product Backlog,

приоритезируются наравне с прочими требованиями и могут быть запланированы на следующий (или на один из следующих) спринтов.

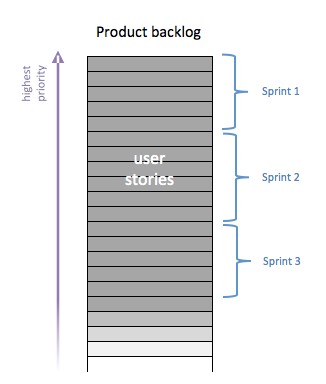
Каждый спринт представляет собой маленький "водопад". В течение спринта делаются все работы по сбору требований, дизайну, кодированию и тестированию продукта.

**Артефакты**

*Product Backlog*

*Sprint Backlog*

*BurnDown chart*



**Активности в Scrum**

 Работа над Бэклогом Продукта

 Планирование Спринта

 Ежедневное совещание

 Демонстрация

 Ретроспектива

Scope спринта должен быть фиксированным. Это позволяет команде давать обязательства на тот объем работ, который должен быть сделан в спринте. Это означает, что Sprint Backlog не может быть изменен никем, кроме команды.

Планирование спринта (Sprint Planning Meeting) происходит в начале новой итерации Спринта.

Из резерва проекта выбираются задачи, обязательства по выполнению которых за спринт принимает на себя команда;

На основе выбранных задач создается резерв спринта. Каждая задача оценивается в поинтах;

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.

**Sprint Planning**

 Участвует вся команда

 User stories разбиваются на задачи и оцениваются членами команды

 В результате команда подписывается на ту функциональность, на которую хватает времени спринта

 Focus Factor = Σ Идеальных часов выполненных задач / Capacity

**User Story** - результат процесса обсуждения команды и бизнеса, который фиксируется в легкой и неперегруженной форме. Эти обсуждения - невероятно важная вещь, потому что позволяют вытянуть на свет все реальные потребности, которые стоят за некоторыми требованиями и развеять все молчаливые предположения обоих сторон, которые останутся невысказанными и превратят разработку в катастрофу.

**User Story** – это нужды конкретного пользователя, выраженные в простой форме.

Как <тип пользователя>, я хочу <сделать>, и тем самым получить <выгоды>

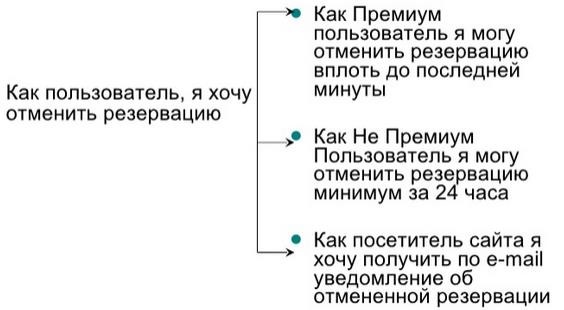
**Пример**,

*Как гость, я хочу зарезервировать номер,*

|  |  |  |  |
| --- | --- | --- | --- |
| *тем самым иметь* |  | | |
| *гарантию размещения* |
| *Как работник гостиницы, я хочу просматривать получать информацию о работе гостиницы* | *отчеты,* | *тем* | *самым* |
| Правило 3С для User Story |  |  |  |



User Story детализируется в Tasks.



**Planning Poker**

Покер планирование – техника оценки сложности предстоящей работы или ее относительного объема

Подготовка:

 Список функций либо пользовательские истории

 Каждому участнику - колода специальных карт

Проведение:

 Краткое описание каждого пункта

 Выбор карт каждым из участников

 Обсуждение самой высокой и низкой оценки

 Обсуждение до достижения консенсуса либо сигнала таймера

 Онлайн-инструмент: <http://www.planningpoker.com/>

Нужно понимать, что уровень детализации и формализации будет зависеть от отношений людей в проекте, их уровне профессионализма, опыта совместных и личных граблей.

Обычно:

 - начальные требования составляет Product Owner

 - команда дописывает и расширяет требования, разбивает на задачи и детализирует

 - дизайнеры рисуют mockup (layout)

Ежедневное совещание (Daily Scrum meeting) начинается точно вовремя;

длится не более 15 минут;

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?

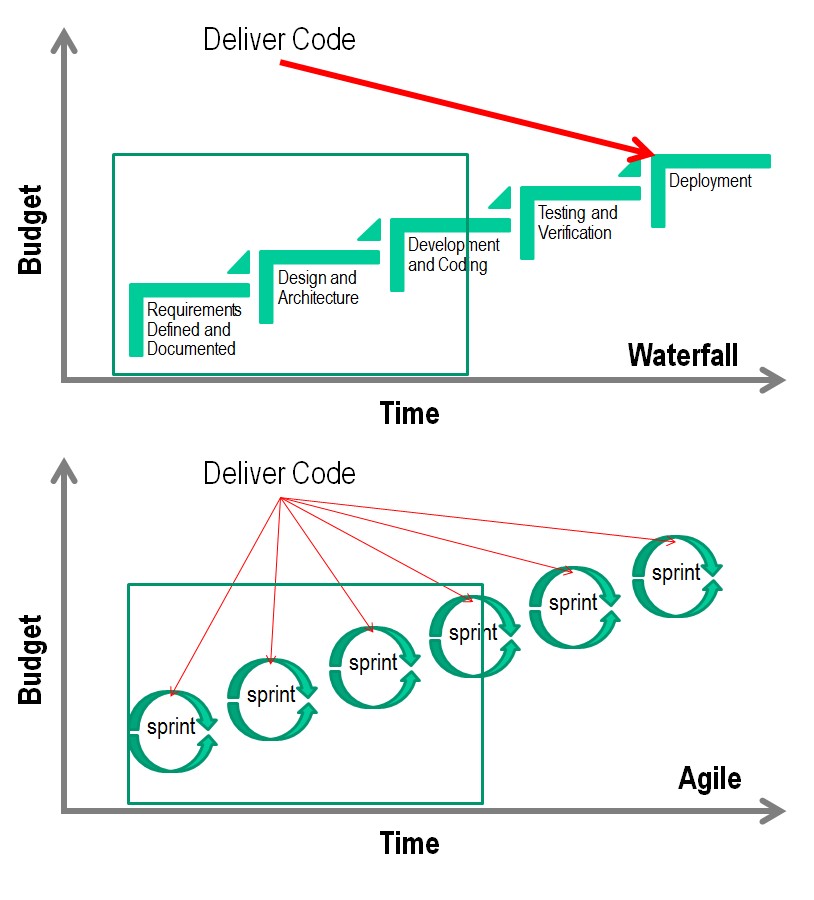
Ретроспективное совещание (Retrospective meeting) проводится после завершения спринта. Члены команды высказывают своё мнение о прошедшем спринте.

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.



Abnormal Termination

Остановка спринта раньше срока в исключительных ситуациях

Команда понимает, что не может достичь цели проекта в срок

Владелец проекта понимает, что исчезла необходимость в реализации цели спринта

Обсуждение причины остановки

Запуск нового спринта