Что было до аджайла

До гибкой разработки все диджитал и IT-компании создавали продукты целиком. Например, в студию обращался заказчик и говорил, что он хочет сайт. Студия отвечала, что не может начать работу без исчерпывающего технического задания. Заказчик вздыхал, садился за бумагу, писал ТЗ на сто страниц, пытаясь по пути не упустить ничего важного. Команда принимала техзадание, закрывалась у себя в студии, работала 2 месяца и показывала результат. После тестирования и правок сайт запускали.

Если в процессе у команды возникала новая идея, которая шла вразрез с техзаданием, ее либо игнорировали, либо возвращались на предыдущие этапы и все переделывали. В итоге растягивались сроки и бюджет, или получался не такой крутой продукт, как мог бы.

Также стало очевидно, что негибкая разработка не подходит для масштабных проектов: рынок постоянно меняется, технологии устаревают — и часто продукт выходит уже неактуальным.

Agile Manifesto

Со временем разработчики стали тестировать и изменять продукт во время работы. Так появились новые техники, которые назвали гибкими. В 2001 году 17 представителей профессии собрались в горной деревне штата Юта, покатались на лыжах и заодно обсудили свои подходы к разработке.

Итогом встречи стал «[Манифест Аджайла](http://agilemanifesto.org/)», в котором излагались основные принципы, выводы и целая философия гибкого программирования. Сейчас эту философию широко практикуют не только в диджитал-среде, но и в любой другой. Вот 4 основных ее положения:

Люди и взаимодействие важнее процессов и инструментов.

Работающий продукт важнее исчерпывающей документации.

Сотрудничество с заказчиком важнее согласования условий контракта.

Готовность к изменениям важнее следования первоначальному плану.

То есть разработчики отказались от формализации требований, безоговорочного следования жестким регламентам и планам — в пользу самого продукта, людей и продуктивного сотрудничества. Это была историческая веха.

Как команды работают по аджайлу

Возьмем ту же ситуацию. В студию приходит заказчик и просит сделать сайт. Вместе с командой они записывают все пожелания, идеи и представления о том, что должно получиться в результате. Потом решают, что самое приоритетное, и начинают работу.

Каждый день команда и заказчик обсуждают, что они сделали, какие проблемы возникли, что планируют делать дальше. Если в процессе появляются новые идеи, их рассматривают и внедряют — это может произойти прямо в процессе работы над этапом.

В конце условной недели у них получается каркас проекта с минимумом функций. Его запускают и тестируют на реальных пользователях, а сами продолжают работать дальше.

Следующий этап повторяет предыдущий с той только разницей, что теперь у команды есть обратная связь от пользователей. Таким образом появляется возможность сразу убрать лишнее и добавить недостающее — ну и параллельно работать над следующими по приоритетности пунктами.

Так повторяется несколько раз, и после тех же двух месяцев у заказчика есть готовый продукт, протестированный и актуальный.

# Скрам

Скрам — это способ организовать работу в духе Аджайла. Он помогает быстро выпускать и улучшать продукты, когда меняются требования рынка.

Многие называют Скрам методологией. Это не так, Скрам — фреймворк. Разница в правилах: методология даёт готовую инструкцию, как добиться результата, например, предписывает менеджеру анализировать работу отдела каждую пятницу.

Фреймворк не даёт инструкции на все случаи жизни. Сотрудники следуют правилам Скрама, но процесс работы меняют сами. Например, сами определяют, как часто нужно анализировать работу, в каком составе собираться, какие документы составлять.

**Процесс Scrum**  
  
Основой Scrum является Sprint, в течении которого выполняется работа над продуктом. По окончанию Sprint должна быть получена новая рабочая версия продукта. Sprint всегда ограничен по времени (1-4 недели) и имеет одинаковую продолжительность на протяжении все жизни продукта.   
  
Перед началом каждого Sprint производится Sprint Planning, на котором производится оценка содержимого Product Backlog и формирование Sprint Backlog, который содержит задачи (Story, Bugs, Tasks), которые должны быть выполнены в текущем спринте. Каждый спринт должен иметь цель, которая является мотивирующим фактором и достигается с помощью выполнения задач из Sprint Backlog.  
  
Каждый день производится Daily Scrum, на котором каждый член команды отвечает на вопросы «что я сделал вчера?», «что я планирую сделать сегодня?», «какие препятствия на своей работе я встретил?». Задача Daily Scrum — определение статуса и прогресса работы над Sprint, раннее обнаружение возникших препятствий, выработка решений по изменению **Процесс Scrum**  
  
Основой Scrum является Sprint, в течении которого выполняется работа над продуктом. По окончанию Sprint должна быть получена новая рабочая версия продукта. Sprint всегда ограничен по времени (1-4 недели) и имеет одинаковую продолжительность на протяжении все жизни продукта.   
  
Перед началом каждого Sprint производится Sprint Planning, на котором производится оценка содержимого Product Backlog и формирование Sprint Backlog, который содержит задачи (Story, Bugs, Tasks), которые должны быть выполнены в текущем спринте. Каждый спринт должен иметь цель, которая является мотивирующим фактором и достигается с помощью выполнения задач из Sprint Backlog.  
  
Каждый день производится Daily Scrum, на котором каждый член команды отвечает на вопросы «что я сделал вчера?», «что я планирую сделать сегодня?», «какие препятствия на своей работе я встретил?». Задача Daily Scrum — определение статуса и прогресса работы над Sprint, раннее обнаружение возникших препятствий, выработка решений по изменению стратегии, необходимых для достижения целей Sprint'а.  
  
По окончанию Sprint'а производятся Sprint Review и Sprint Retrospective, задача которых оценить эффективность (производительность) команды в прошедшем Sprint'е, спрогнозировать ожидаемую эффективность (производительность) в следующем спринте, выявлении имеющихся проблем, оценки вероятности завершения всех необходимых работ по продукту и другое.стратегии, необходимых для достижения целей Sprint'а.  
  
По окончанию Sprint'а производятся Sprint Review и Sprint Retrospective, задача которых оценить эффективность (производительность) команды в прошедшем Sprint'е, спрогнозировать ожидаемую эффективность (производительность) в следующем спринте, выявлении имеющихся проблем, оценки вероятности завершения всех необходимых работ по продукту и другое.

Agile принципы очень хорошо себя зарекомендовали, причем не только в сфере программирования. Поэтому очень важно быть ознакомленным хотя бы с базовыми ценностями Эджала