**Agile**

С английского переводится как «подвижный, быстрый, проворный»

● «Гибкие методологии»

● Agile-методология динамично организует разработку IT-продукта, постоянно адаптируя проект к новым обстоятельствам и требованиям

Как устроена разработка по Waterfall:

1. Выдвигаются требования к ПО, разрабатывается техническое задание (ТЗ).

2. Поставленные задачи воплощаются в коде.

3. Выполняется тестирование ПО.

4. Готовое ПО внедряется в работу.

Различия

Waterfall

● Конечный продукт должен идеально соответствовать требованиям, целям и задачам, которые были сформулированы до разработки

Agile

В приоритете не исходные установки, а актуальные потребности пользователя

● Нет предварительного генерального плана

● ИТ-продукт создается экспромтом

**Как устроена разработка по Agile**

1. Проходит через ряд циклов — итераций

2. Каждая итерация — это фактически отдельный проект

3. В рамках итерации разрабатывают фрагмент программы, улучшают функциональность, добавляют новые возможности

Пример по Agile

1. Коллектив разработчиков создаёт аудиоплеер.

2. Уже написан интерфейс и базовый функционал, воспроизводятся файлы формата MP3, WAV и OGG.

3. Пользователи предлагают добавить проигрывание CD-дисков и подключить горячие клавиши, чтобы быстро управлять плеером.

4. Программисты собираются на короткое совещание для реализации нового функционала. Один из разработчиков предлагает добавить воспроизведение онлайн-радио.

5. Разработка. Создается программный код, интегрируется в продукт, выполняется тестирование. 6. Новый функционал готов – выпускается новая версия продукта. Начинается новая итерация разработки. Итерация завершается и начинается новый виток разработки.

**Почему это важно?**

При коммерческой разработке у нас постоянно меняются условия:

✔ выходят новые законы

✔ изменяются бизнес-требования

✔ конкуренты вдруг выпускают что-то похожее на наш продукт или даже круче

*Мы должны соответствовать этим условиям, делать продукт лучше, чем конкуренты!*

Agile методология подразумевает

✔ После выпуска первого релиза вы снимаете метрики, что-то переделываете, делаете следующий релиз, и т.д.

✔ На каком-то этапе понимаете, что конкурент выпустил какую-то фичу, и вам нужно идти не туда ✔ В этот момент вы можете остановиться, принять другие требования и начать двигаться в новом направлении

Сравнение разработки

|  |  |
| --- | --- |
| 1-ая компания по разработке ПО | 2-ая компания по разработке ПО |
| Использует Waterfall  ✔ Пишет ТЗ  ✔ Проектирует систему  ✔ Рисует дизайн  ✔ Уходит несколько месяцев | Использует Agile  ✔ К этому времени уже запускает сайт, выпускает ПО  ✔ Начинает зарабатывать деньги  ✔ Захватывает рынок  ✔ Привлекает новых пользователей |

**4 центральных идеи Agile Manifesto**

1. Люди и взаимодействие важнее, чем процессы и инструменты.

2. Работающее ПО важнее, чем исчерпывающая документация.

3. Сотрудничество с заказчиком важнее, чем согласование условий контракта.

4. Готовность к изменениям важнее, чем следование первоначальному плану.

**12 принципов Agile Гибкие методологии**

1. **Наивысшая ценность** – это удовлетворение потребностей заказчика, благодаря регулярной и максимально ранней поставке ценного для него ПО.

2. Мы всегда готовы **изменять требования**, даже на поздних стадиях проекта, если узнаем что-то новое. Таким образом, мы создаем бизнесу или внешнему заказчику конкурентное преимущество.

3. При использовании Agile работающий продукт **выпускают** максимально **часто**. В манифесте прописаны сроки — от пары недель до пары месяцев.

4. Бизнес обязательно должен работать вместе с программистами, помогать им понять специфику данного рынка

5. **Команда** – один из краеугольных камней Agile. Наилучших результатов достигает команда мотивированных профессионалов.

6. Самый простой и эффективный способ узнать требования клиента, заказчика или пользователя – поговорить с ними.

7. Работающий продукт.

8. В Agile важен ритм, постоянные улучшения. Бизнес и программисты всегда должны располагать возможностью делать процесс устойчивым, постоянно его улучшать.

9. Agile вообще не будет работать, если вы написали кривой код. У вас должна быть хорошая гибкая архитектура, в которую можно добавлять разные элементы и, при необходимости, легко их изменять.

10. Простота. Она проявляется в технической составляющей, в дизайне. Простота очень важна также с точки зрения выпуска продукта: когда вы хотите «нарезать» того «слона», лучше начать с простой части.

11. Менеджер в команде меняет свою роль: он не столько занимается организацией процесса, сколько учит команду, поэтому команда должна быть самоорганизованной.

12. Команда должна регулярно анализировать реализуемые процессы: что получилось, как они этого добились, и постоянно улучшать организацию работы.

Часто используемые практики Agile

1. Стендапы. Каждый день в определенное время команда собирается и синхронизирует свою деятельность

2. Планирование итераций Команда планирует объем работ на ближайшую итерацию

3. Unit-тестирование Проверка на корректность отдельных единиц работы исходного кода программы. Минимальными единицами работы для нас являются процедуры и функции.

4. Планирование релизов Планирование большими кусками – что и когда будем выпускать в свет

**Agile методология**

✔ Scrum (самая большая и часто применяемая методология)

✔ Экстремальное программирование (XP)

✔ DSDM (Dynamic Systems Development Method)

✔ Crystal

✔ FDD (Feature Driven Development)

✔ Kanban

Преимущества и недостатки Agile

|  |  |
| --- | --- |
| Преимущества | Недостатки |
| ● ПО готово к использованию на самых ранних этапах его разработки, пусть и не с полной функциональностью  ● Разработчики постоянно в контакте с заказчиком  ● Нет жестких рамок = ПО постоянно изменяется и улучшается  ● Заказчик и пользователь партнёры и идейные вдохновители  ● Нет заранее и подробно сформулированного ТЗ = разработчик может решить задачу творчески | ● Не факт, что программа когда-нибудь будет завершена  ● Пользователь требует все и сразу  ● Работа над проектом требует не только профессионализма разработчика, но и сознательности пользователя  ● «Золотые пользователи»  ● Строительство без чертежей  ● Постоянная спешка |