**Waterfall**

Один из старейших способов разработки, но используется до сих пор. Принцип – разработка идёт только в одном направлении, так же, как и водный поток.

Этапы:

1. Анализ
2. Дизайн (бд, таблицы, минимальный прототип с базовым функционалом)
3. Разработка (Самый затратный по времени и сложности)
4. Тестирование
5. Релиз и поддержка

Особенности:

- Первая модель в истории

- Каждая фаза начинается после того как закончится предыдущая

- почти не используется сейчас

Минусы:

* Высокая цена фикса ошибок(поздно обнаруживается)
* Требуются годы на первую версию
* Сложно вводить фичи
* Много документации

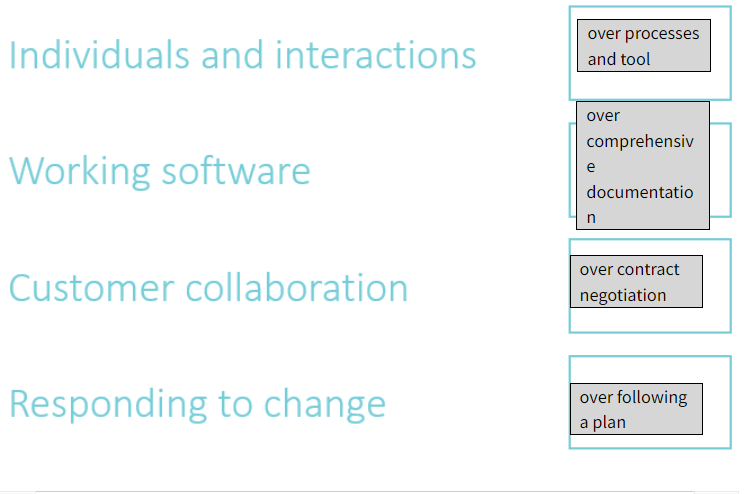
Когда применяется:

* Требования хорошо и ясно задокументированы
* Высокий уровень контроля (как в гос заказах)

**Agile**

Один из самых популярных. Не ставит чётких рамок. Важна коммуникация с людьми. Остаются только процессы, которые помогают людям делать работу и имеют большой выхлоп. Работа важнее, чем документация.

Особенности:

* Важна коммуникация между людьми и процессы, которые упрощают работу
* Работа важнее чем документация
* Контакт с заказчиком важнее, чем контракт
* Адаптация к обстоятельствам важнее, чем план
* 

Принципы:

- Удовлетворение клиента – самое важное

- Быстрый отзыв на изменение ТЗ

- Быстрая разработка ( недели – месяцы )

- Бизнесмены и разработчики должны работать вместе

- Давать работу замотивированным сотрудникам

- Отсутствие переработок

- Простота важна

- Команды должны быть самоорганизованными

**Scrum**

Основана на принципах Agile. Главный плюс – любая фича будет реализована в течении одного спринта (+-2 недели). На выходе получается большое количество версий продукта с разным функционалом внутри.

Роли:

* Product owner
* Scrum master
* Development Team

Ивенты:

- Sprint Grooming – обсуждение перед стартом

- Sprint Planning

- Daily scrum (Standup) – обсуждение что я сделал, что буду делать, что мне мешает

- Sprint Review

- Sprint Retrospective

Спринт длится 1-4 недели

**Kanban**

Метод управления разработкой, реализующий принцип “точно в срок” и способствующий равномерному распределению нагрузки между работниками. (Изобретён компанией TOYOTA)

Особенности:

* Визуализация задач на доске при помощи карточек
* За этапы реализации 1 таска могут отвечать несколько человек
* Над задачей трудятся столько времени, сколько надо для её решения
* Ограничьте число одновременно выполняемых задач на 1 сотрудника

Плюсы:

* Члены команды могут видеть текущее состояние задачи на любой момент времени
* Гибкость планирования
* Меньшая длительность итерации
* Быстрое выявление проблем

Минусы:

* Ограничение по размеру команды
* Краткосрочность планирования

**Extreme Programming**

Один из фреймворков Agile подхода, который позволяет создавать программное обеспечение высокого качества, при этом поддерживать высокое качество жизни у членов команды разработки.

Цель – справится с постоянно меняющимися требованиями к программному продукту и повысить качество разработки.

13 Практик экстремального программирования:

1. Вся команда
2. Игра в планирование
3. Частые релизы версий
4. Пользовательские тесты
5. Коллективное владение кодом
6. Непрерывная интеграция кода
7. Стандарты кодирования
8. Метафора системы
9. Устойчивый темп
10. Разработка, основанная на тестировании
11. Парное программирование
12. Простой дизайн
13. Рефакторинг

Преимущества:

* Работаем не больше 40 часов в неделю
* Быстрые изменения в коде и добавление новых функций
* Код постоянно тестируется
* Быстрый темп разработки
* Высокое качество кода
* Затраты на разработку ниже

Минусы:

* Успех проекта зависит от вовлечённости заказчика
* Трудно предугадать затраты времени на проект
* Успех ExtPrg зависит от уровня программистов
* Парное программирование
* Недостатки структуры и документации

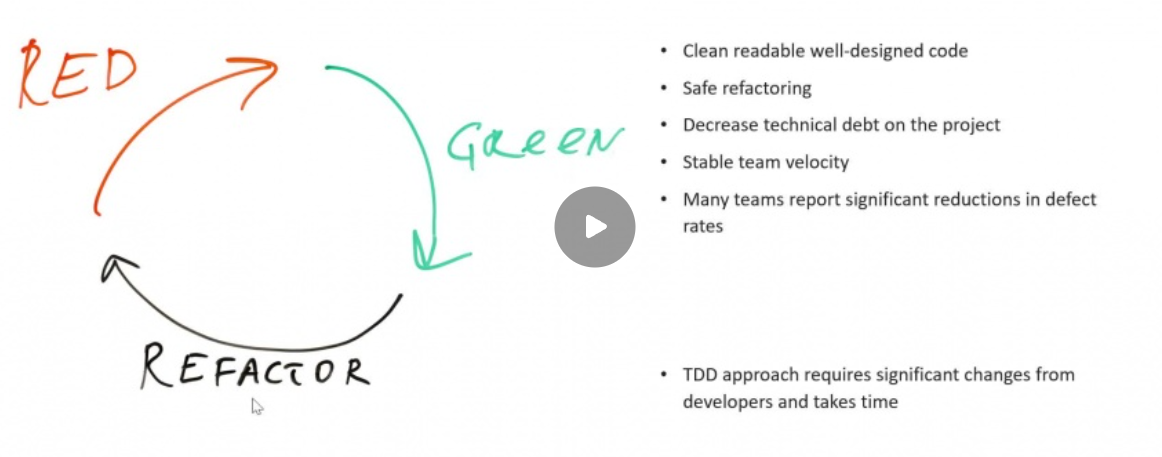
Принципы:

- Простота

- Коммуникация

- Обратная связь

**Test-Driven Development**



**Behaviour-Driven Development**

Процесс разработки, в котором приоритет отдаётся сотрудничеству как между техническими, так и нетехническими командами.

Тестовые случаи пишутся на естественном языке(псевдокод)

**Summary&Extras**