

Жизненный цикл разработки программного обеспечения



Жизненный цикл программного обеспечения



Идея

Разработка программного продукта начинается с идеи, как решить ту или иную проблему, возникающую у пользователя.

Цель: определение требований к системе



Требования

Результат:

- Формирование ТЗ, SRS
- Определение критериев приемки ПО

Создается “скелет” проекта основываясь на требованиях.

Определяются используемые технологии, инструменты, рабочие процессы, взаимосвязи между разными частями проекта, структуры баз данных, потоки данных и т.п.

Здесь разрабатывается **техническая архитектура**.

Разработка

Разработка ПО согласно сформированным
требованиям

Тестировщики, основываясь на требованиях (SQA, SRS, DDS) и готовом продукте производят проверку качества ПО.

Если находятся отклонения от требований / ошибки — они оформляются в виде отчетов о дефектах, исправляются и перепроверяются.

Тестирование

После успешного тестирования готовый продукт передается заказчику.

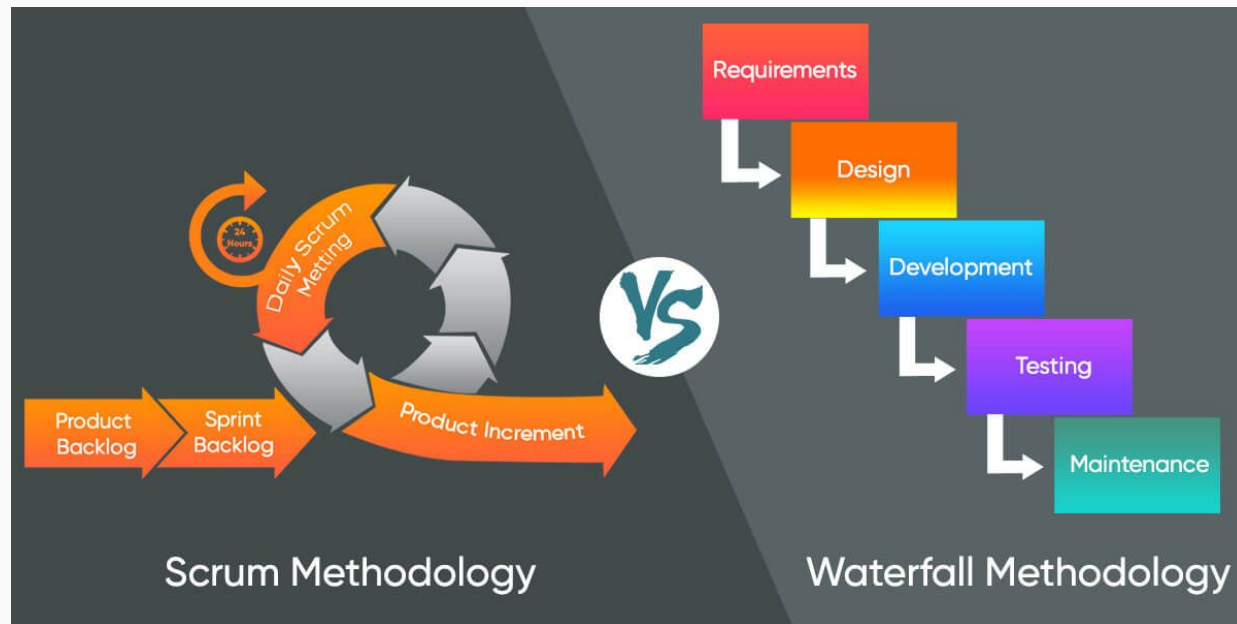
Кроме передачи может производиться настройка рабочих окружений, установка, конфигурация и запуск продукт.

Поддержка

Эта часть жизненного цикла является самым длительным и важным этапом разработки ПО

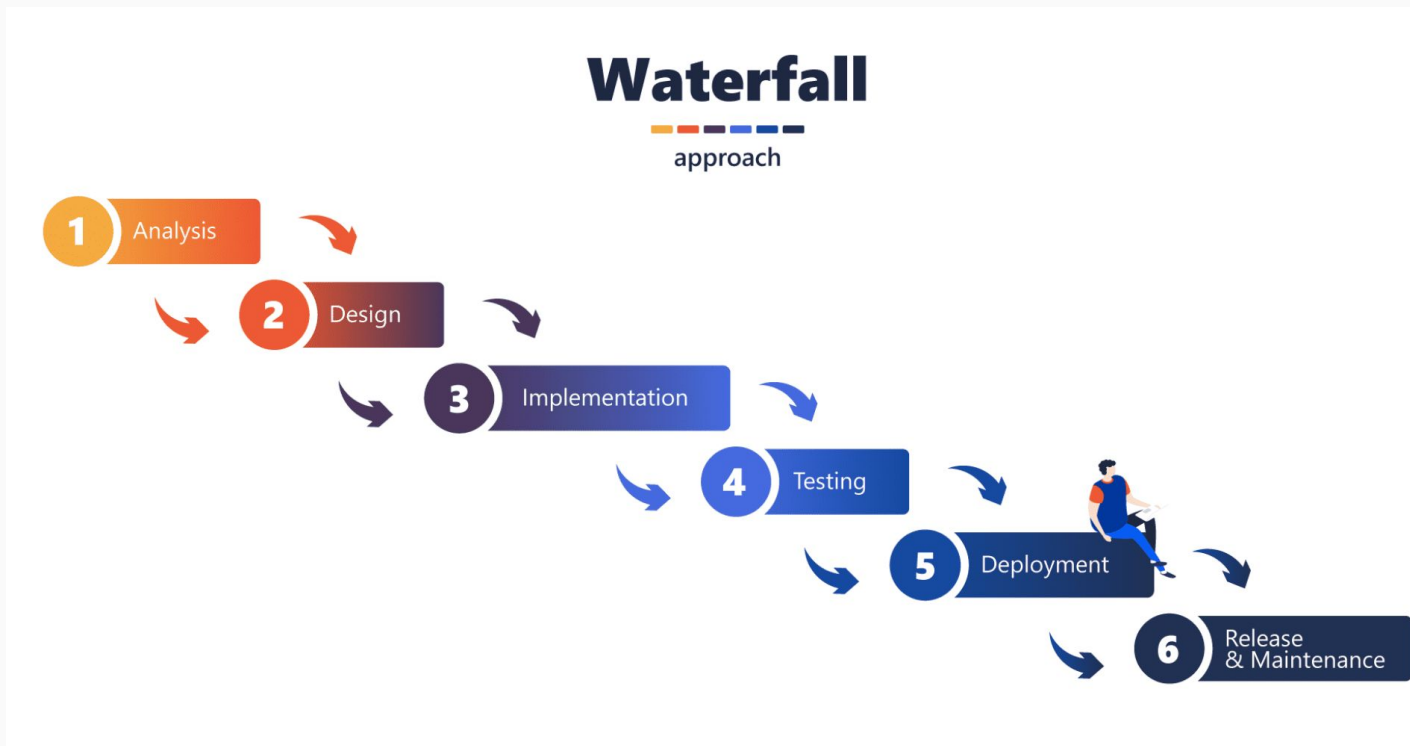
Закрытие

Последний этап жизни ПО: На данном этапе происходит вывод продукта из эксплуатации, его замена на современные аналоги, либо новые версии



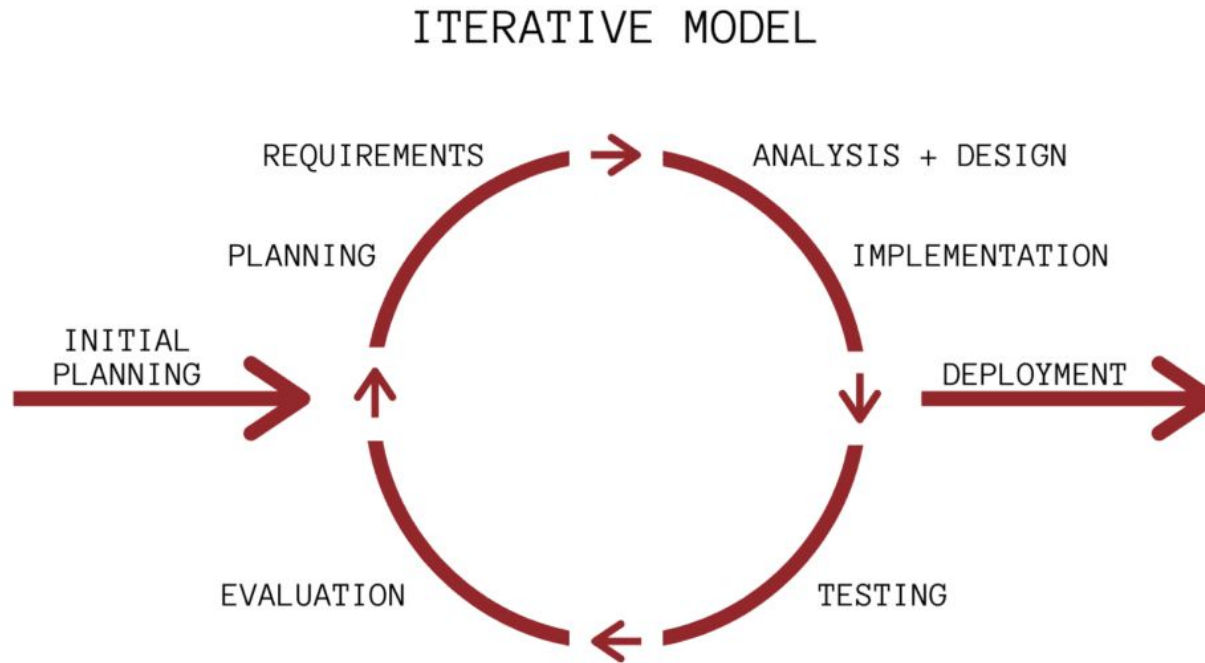
- Классическая - ориентированная на план
- Адаптивная - ориентированная на изменения

Waterfall



Waterfall — это работа по заранее написанному и согласованному ТЗ.

- В проектах с четко определенными требованиями, для которых не предусматривается их изменений в процессе разработки.
- В проектах, которые мигрируют с одной платформы на другую.
- Клиенту важно знать точные сроки выполнения всех работ над проектом. Исполнитель полностью несет ответственность за срыв сроков и незапланированное увеличение бюджета.



Итеративная модель

Преимущества итеративной модели:

- быстрый выпуск на рынок работающей версии продукта, пусть и с базовым набором функций;
- возможность оперативно получать обратную связь от пользователей и постоянно вносить изменения в продукт;
- быстрое устранение ошибок в каждом цикле, исключающее их накопление под конец разработки.

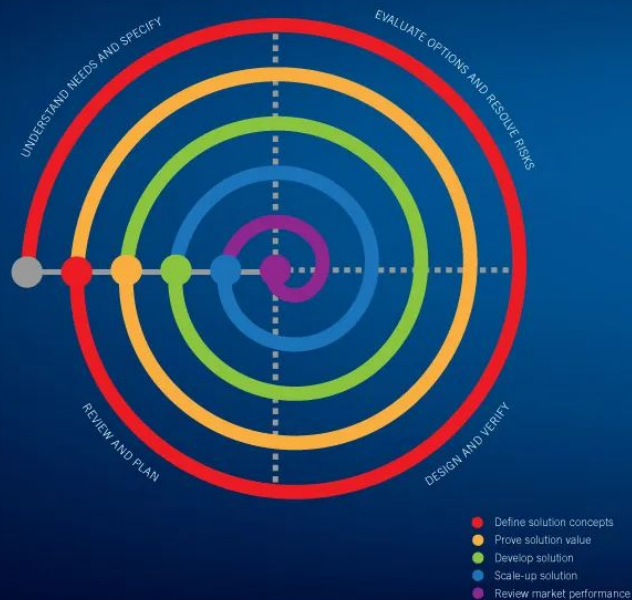
Недостатки итеративной модели:

- использование на начальном этапе разработки баз данных и серверов вызывает проблемы с масштабированием и стабильностью продукта;
- невозможность точно определить нужный бюджет и сроки разработки, так как заказчик сам не знает, когда продукт будет полностью готов.

Итеративная модель сегодня используется в больших проектах с нечеткими требованиями, а также при разработке инновационных продуктов с неопределённым и трудно прогнозируемым результатом.

Спиральная модель

SPIRAL MODEL OF INNOVATION



Спиральная модель

Основная идея спиральной модели состоит в том, что она использует циклический подход к разработке, в котором процесс разработки проходит через несколько итераций, каждая из которых завершается оценкой рисков и принятием решения о дальнейшей работе. Каждая итерация включает в себя анализ требований, проектирование, разработку, тестирование и оценку рисков.

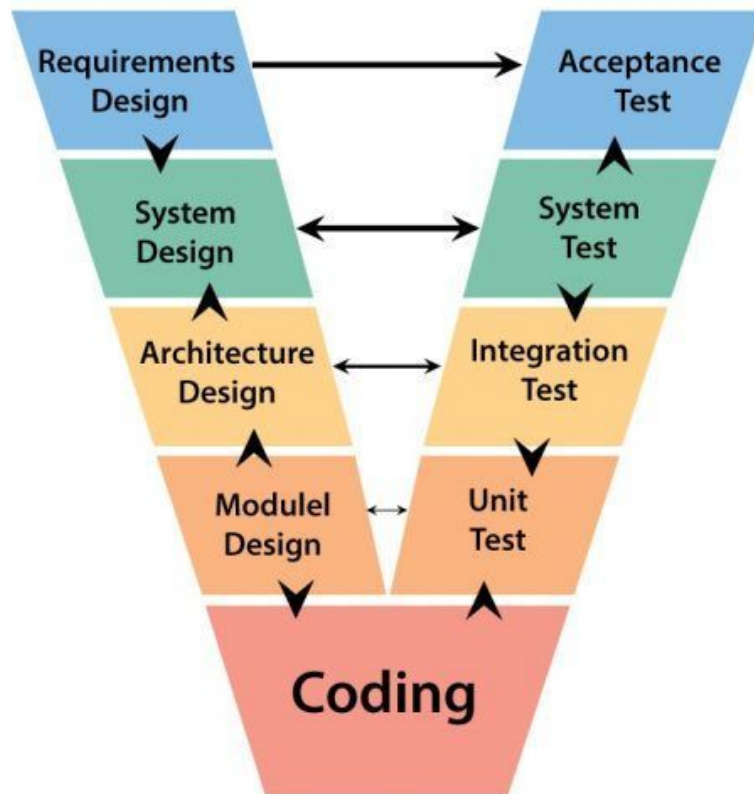
Основным преимуществом спиральной модели является ее гибкость и адаптивность к изменениям в процессе разработки. Однако, из-за более сложной структуры процесса разработки и управления рисками, она может быть более сложной в реализации, чем более традиционные модели разработки.

Спиральная модель

Спиральная модель разработки используется в тех случаях, когда проект имеет высокую степень неопределенности и значительные риски, которые нужно управлять на протяжении всего процесса разработки. Кроме того, спиральная модель может быть полезна в проектах, где:

- Требования к продукту не полностью определены или могут изменяться в процессе разработки;
- Проект имеет высокую степень сложности и требует поэтапного уточнения;
- Необходимо повысить уровень контроля качества;
- Существуют технические риски, которые необходимо оценить и управлять на ранних стадиях проекта;
- Проект требует высокой гибкости и адаптивности в процессе разработки.

V образная модель



Ключевое преимущество V-модели заключается в том, что она предоставляет четкую структуру, которая облегчает управление проектом и повышает его качество.

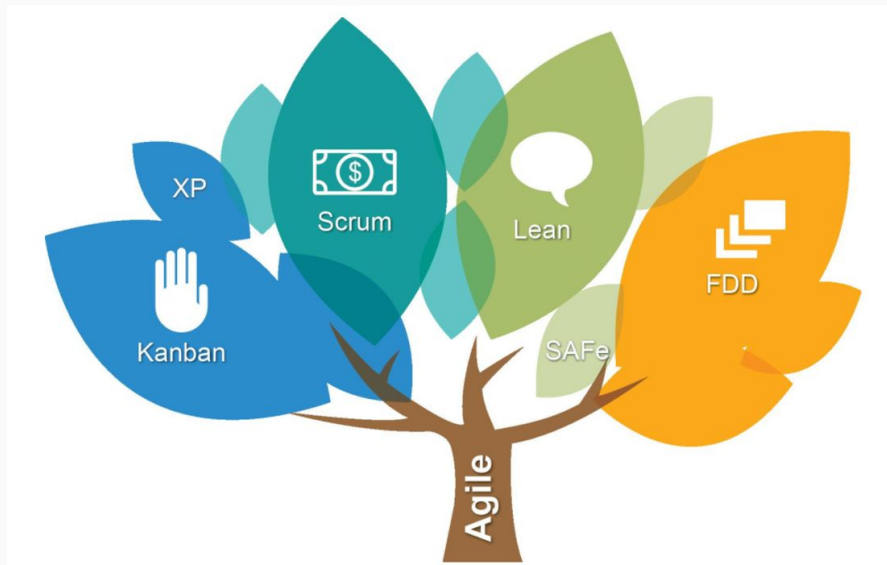
Недостатком является то, что V-модель может быть неудобной для использования в проектах, которые требуют частого изменения требований.

Модель V используется в проектах, которые имеют четко определенные требования и спецификации, и когда основной акцент делается на разработку качественного и надежного программного обеспечения. Она также может быть полезна в случаях, когда заказчик и разработчик имеют ясное понимание того, что должно быть разработано, и когда заказчику важно получить полный контроль над процессом разработки.

Гибкие методологии

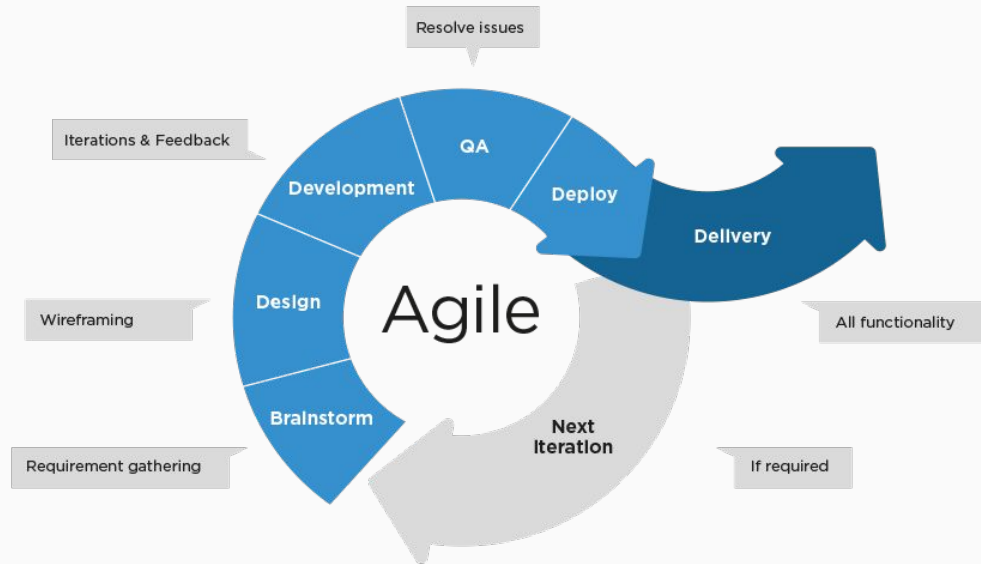
Направлены на быструю поставку бизнесу рабочего продукта за короткие итерации в ситуациях, когда четкое видение будущего решения не определено.

- Agile
- Scrum
- Kanban



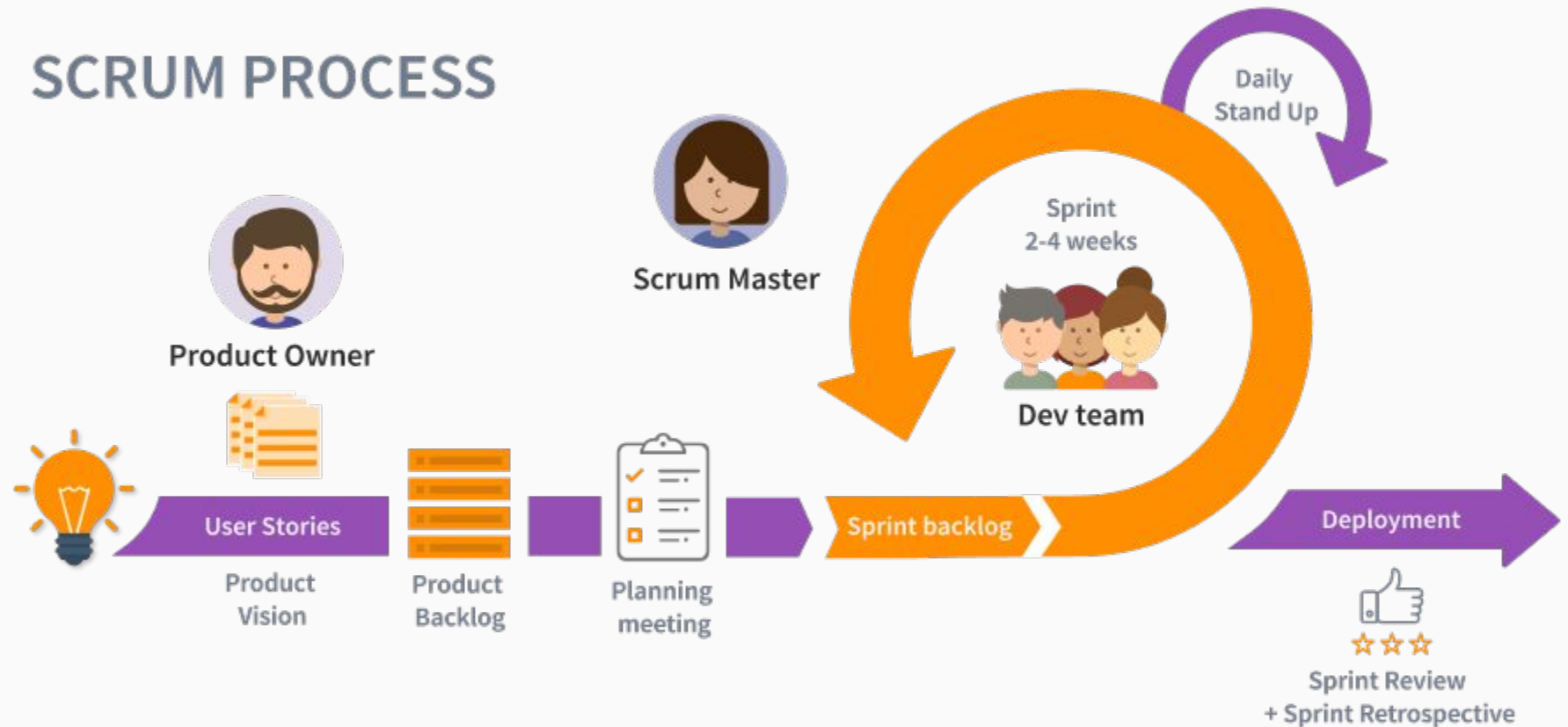
Agile

Agile – это название методики управления проектами, в которой проект разделен на несколько мелких частей, которые будут завершены к определенному сроку, вместо длительного выполнения всего проекта.





SCRUM PROCESS



Product Backlog - это список требований к продукту, который создает Product Owner в методологии Scrum. Он представляет собой живой документ, который постоянно обновляется и уточняется в течение жизненного цикла продукта.

Sprint Backlog - это список задач, которые команда разработки выбрала для выполнения в рамках текущего спринта в методологии Scrum.

Sprint - это период, в течение которого команда создает и совершенствует отдельную часть продукта, чтобы быть готовым к изменениям условий и исключить глобальную доработку.

Инкремент (Increment) в рамках методологии Scrum - это сумма всех элементов продукта, которые были завершены на конец спринта. Инкремент должен быть рабочим, тестируемым и готовым к выкатке на производство.

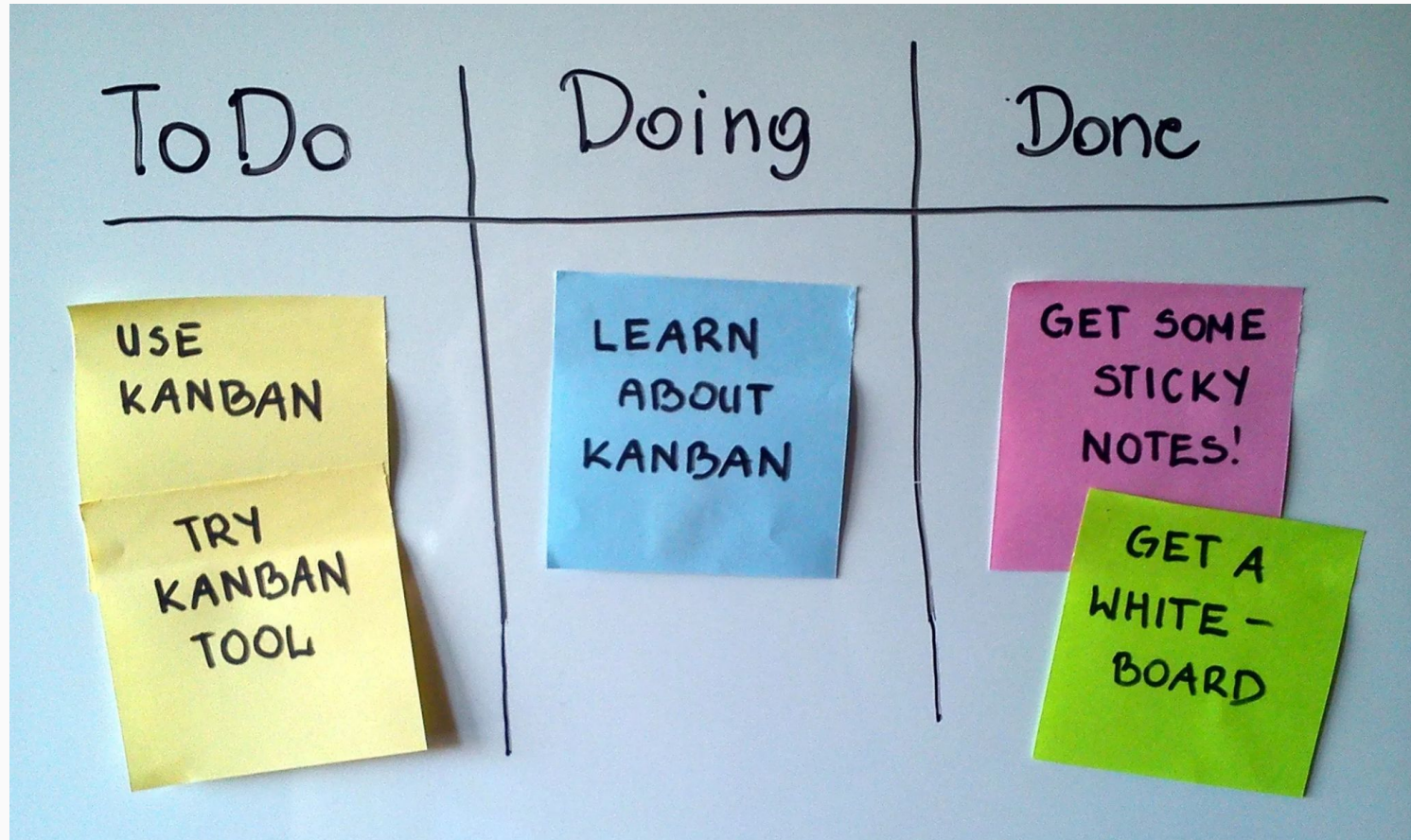
Планирование спринта - это важный этап в рамках методологии Scrum, который позволяет команде разработки определить, какие задачи из Product Backlog будут выполнены в течение следующего спринта и как они будут доставлены.

Daily Scrum - это ежедневное событие, которое проводится в рамках методологии Scrum, чтобы обеспечить коммуникацию и синхронизацию между членами команды разработки.

Backlog Refinement - это событие, которое проводится в рамках методологии Scrum, чтобы подготовить и уточнить задачи, которые могут быть включены в будущие спринты.

Sprint Review (Обзор спринта) - это событие в рамках методологии Scrum, которое проводится в конце каждого спринта. На этом событии команда разработки продемонстрирует свои достижения за прошедший спринт заказчику, Product Owner-у, заинтересованным сторонам (stakeholders) и другим участникам команд.

Sprint Retrospective (Ретроспектива спринта) - это событие в рамках методологии Scrum, которое проводится в конце каждого спринта. Цель ретроспективы состоит в том, чтобы обеспечить команде разработки возможность проанализировать свою работу за прошедший спринт и выработать улучшения для следующего.



- Начните с того, что есть сейчас (то есть возьмите текущий процесс в том виде в котором он есть)
- Придите к соглашению об эволюционном развитии (договоритесь что будете меняться)
- Поощряйте проявления лидерства на всех уровнях (поощряйте идеи и желание менять на всех уровнях)

Kanban Board

Backlog

To Do ②

Development ③

Testing ②

Deployment

Done

Explicit Policies

Cards must be prioritized and sized

Only product owner can and cards here

Any available developer can pull

Card is Done only when Code Review completed

Test cases must be documented before testing can start

Validation by PO

①

User Stories

Intangible

Defect

Fixed Date

F
H
I

G

Ongoing

D

Done

C

Ongoing

E

Done

B

A

плюсы:

- Возможность быстрого запуска проекта с наиболее приоритетными функциями и минимально возможным бюджетом
- Ежедневный контроль над ходом работ
- Частые демонстрации проекта
- Возможность вносить коррективы в техническое задание по ходу реализации проекта

минусы:

- Команда должна быть высококвалифицированной и опытной, ориентированной на клиента
- Сложности при заключении договоров
- Отсутствие необходимой документации