

1. Менеджер обсуждает с клиентом задачу по разработке нового функционала  
После чего составляется документация на проект, в которой прописываются что должно получиться в результате и в течение какого срока это должно быть выполнено

2. Менеджер коммуницирует с разработчиком, создает задачи в системе трекинга задач, определяет приоритет/важность этих задач, согласно требованиям клиента/заказчика

3. Разработчик выполняет задачи из системы трекинга задач согласно их приоритету  
Например, допиливает/разрабатывает какой-то текущий/новый функционал приложения/продукта

Результат такой работы разработчика является код самого приложения, а так же в идеале Unit-тесты, для проведения Unit-тестирования в момент непрерывной интеграции при сборке/компиляции кода

Этот код должен быть протестирован тестировщиком в ручном или/и автоматическом режиме

4. Здесь в работу вступает процесс CI(Continuous Integration)/CD(Continuous Delivery)/CD(Continuous Deployment)

Непрерывная интеграция(Continuous Integration) - сборка кода приложения, проверка его на синтаксический анализ, проведение unit-тестов

Непрерывная доставка (Continuous Delivery) - автоматическая выкатка кода на все типа окружений (Dev,QA,Stage), кроме окружения Production

Проведение интеграционных,функциональных, нагрузочных, приемочных тестов

Непрерывное развертывание (Continuous Deployment) автоматическая выкатка кода на все типа окружений включая Production

4.При использовании автоматического тестирования, тестировщик пишет такие тесты для покрытия ими кода проекта,а DevOps-инженер в свою очередь встраивает их в pipeline

Тестировщик проводит такие виды тестирования (вручную или автоматически)

- функциональное
- интеграционное
- нагрузочное
- приемочное

5. DevOps-инженер создает инфраструктуру и pipeline(конвейер для доставки кода с системы контроля версий до тестовых/stage-окружений (в случае Непрерывной доставки) или даже на прод (в случае Непрерывного развертывания) и запуском выше указанных тестов на QA/Stage-окружениях

т.е. по коммиту в ветку/открытию Pull/Merge Request или другому событию, в системе контроля версий

срабатывает Webhook, который запускает Pipeline

Либо такой pipeline может запускаться по кнопке, которую нажимает тестировщик в системе CI/CD

Если в результате тестов обнаруживаются ошибки(тесты падают по причине провала какого-либо вида тестирования)  
то задача отправляется на доработку разработчику

Если тесты проходят успешно, тогда, например, на Stage-окружении, или на каком-либо отдельном/специально выделенным окружении, менеджер предоставляет доступ клиенту/заказчику на ознакомление с результатом выполнения его требований/задачи, проверки клиентом функционирования и корректности работы того, что хотел получить клиент/заказчик в своих требованиях

6.После такого ознакомления

- в случае успешного ревью клиентом - функционал выливается на прод
- в случае замечаний/дополнений от клиента - отправляется на доработку к разработчику через менеджера проекта

**Классический Software Development Life Cycle имеет вид**

Планирование->Определение

требований->Дизайн/Прототипирование->Разработка->Тестирование->Доставка/Деплой->Обслуживание/Поддержка->Обратная связь от конечного потребителя->Снова планирование