**Модели Разработки ПО:**

**Водопадная модель:**

Водопадная модель — это та модель, где есть четкое последовательность действий. В подобной модели, все этапы идут строго друг за другом и на следующий этап мы можем перейти только после того, как сделаны все работы на предыдущем. Если мы завершили один этап, то вернуться к нему нельзя.

Пример: если мы что-то не учли на этапе написания спецификаций, то вернуться к этому этапу мы не сможем, в водопадной модели не принято так делать.

Минусы водопадной модели: если, к примеру, мы на стадии анализа допустили какой-то баг, то этот баг пробежит по следующим этапам и в конечном счете, этот баг станет большим, следовательно мы потерям время, а значит и деньги.

**Классическая структура водопадной модели:**

1. Разработки ПО в первую очередь начинается с идеи (у людей появляется проблема, и у другого человека появляется идея как эту проблему решить)

2. Происходит анализ идеи, на этом этапе в работу подключаются бизнес и системные аналитики, их задачей является выяснить все требования и выяснить, что заказчик хочет получить на выходе и уже на основании этих требований формируется тех. Задание.

3. Далее, мы переходим на стадию архитектуры и дизайна, где главным лицом является системный архитектор, который, собственно, проектирует проект на основании SRS и BRD, и так как мы уже понимаем бизнес составляющее, цели которые мы преследуем и знаем функциональные требования (какие вещи может делать наш продукт, а какие нет) мы можно принимает решение о способе реализации этого проекта, то есть, какие технологии мы будем использовать, на каком язык программирования и т.д. Что не мало важно, на этом этапе мы должны продумать из каких модулей будет состоять наша система и как эти модули будут взаимодействовать (какой-тип БД мы используем)

4. В работу вступаю разработчики, которые изучать SRC, BRD документы, знакомятся с архитектурой, после чего, они начнут писать приложение, чтобы в последующем передать первоначальную сборку тестировщикам.

5. Когда сборка готова, начинается тестирование, на этом этапе от тестировщика требуется сделать отчет о тестировании, в котором описано, что мы тестировали, а также список всех найденных ошибках, этот отчет мы передаем разработчика и начинается та стадия, где разработчики начинают исправлять баги. Спустя какое-то время разработчики выпустят новую версию сборки и главной задачей QA, проверить все ли в новой версии было отремонтировано это еще так же называют стадией верификацией и стадия верификация может проходить в неком цикле, до тех пор, пока не будут устранены все дефекты. Когда приложение станет стабильным мы получим RC Build.

После RC Build начинается приемочное тестирование, подобное тестирование дает нам понять, готов ли продукт к релизу или он еще весьма сырой подобно тестирование выполняется на основании набора типичных тестовых случаев и сценариев. Если говорить простым языком, то приемочное тестирование это когда мы берем RC Build и снова прогоняем все тесты по всему функционалу, которые мы делали раннее, мы берем все критические дефекты, которые ранее уже находились и снова все прогоняем.

Решение о проведении приемочного тестирования принимается, когда:

1. продукт достиг необходимого уровня качества;

2. заказчик ознакомлен с Планом Приемочных Работ (Product Acceptance Plan) или иным документом, где описан набор действий, связанных

с проведением приемочного тестирования, дата проведения, ответственные и т.д.

Фаза приемочного тестирования длится до тех пор, пока заказчик не выносит решение об отправлении приложения на доработку или выдаче приложения.

Acceptance тестирования - первичное тестирование.

Acceptance criteria - это критерия, с которой мы сравниваемся на стадии Acceptance тестирования, чтобы решить, готов ли продукт к релизу или нет.