**Жизненный цикл разработки ПО:**

При тестировании всегда нужно задавать вопросы, дабы уточнить какие-то детали (касается и собеседований тоже)

SDLS - жизненный цикл разработки ПО.

SRS – это наша спека, то бишь, набор всех требований, которые мы хотим получить на выходе. Ориентирован на технические детали

BRD - Документ бизнес-требования. Ориентирован на бизнес

RC Build - почти готовая сборка, где реализован либо весь функционал, либо большая его часть

Product owner - хозяин продукта. Тот, кто придумал идею(заказчик)

На стадии анализа мы стараемся узнать, что будем реализовывать, на стадии архитектуры мы стараемся узнать, как мы будем это реализовывать

Тестовое окружение — это среда, где тестировщик тестируем систему (сайт, приложение)

Жизненный цикл разработки ПО: ниже описаны стадии, которые продукт пройдет, начиная от идеи заканчивая релизом.

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, где бизнес и системные аналитики, выясняют все требования, а также, они стараются выяснить, что заказчик хочет получить на выходе и уже на основании этих требований формируется тех. Задание. На этом этапе составляются SRS, BRD документы

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), исходя из этого, мы можно принимать 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. Четвертый этап, это этапа разработки. На этом этапе в работу вступаю разработчики, которые изучать SRS, BRD документы, знакомятся с архитектурой, после чего, они начнут писать приложение, чтобы в последующем передать первоначальную сборку тестировщикам.

5. Пятый этап, этап тестирования. 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 разработчикам чтобы те начали фиксить 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.

6. Далее начинается приемочное тестирование, подобное тестирование дает нам понять, готов ли продукт к релизу или он еще весьма сырой. Подобно тестирование выполняется на основании 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, следуя которой, мы можем понять, готов ли наш продукт к релизу или нет.

Вопросы:

Все ли дефекты должны быть отремонтирован перед тем, как мы может отдать продукт пользователям? Нет! ремонт одного дефекта ведет к появлению новых дефектов