

**SDLC** - software development life cycle ← описывает весь жизненный цикл разработки П.О.

В тестировании тоже есть жизненный цикл(**STLC**) - software testing life cycle, оно в принципе соответствует этапам из SDLC, т.е. с точки зрения тестировщика он не ждёт пока что-то будет написано, оба и тестировщик и разработчик начинают работу в проекте одновременно одновременно.

Отдельно указано: мониторинг и контроль - непрерывное сравнение актуального прогресса с планом тестирования, речь идёт о том, что когда на этапе планирования мы спланировали что вот в таком то направлении мы будем двигаться, вот такие то шаги мы будем делать, это не значит что по этому плану мы будем жёстко следовать, это скорее план в общих чертах, работа которая должна быть выполнена, дак вот мониторинг и контроль это скорее про то что если что-то идёт не по плану то мы его конечно актуализируем (например разработчики не успевают по своим срокам, тогда тестировщики говорят/вносят поправку типа ок мы протестируем имеющийся материал более углубленно или если тестирование отстаёт тоже можно внести поправку что тестироваться на текущем этапе будет только жизненно важный функционал). Другими словами всё гибко но план должен быть чтобы мы понимали куда мы идём.

Итак 5 этапов в (STLC):

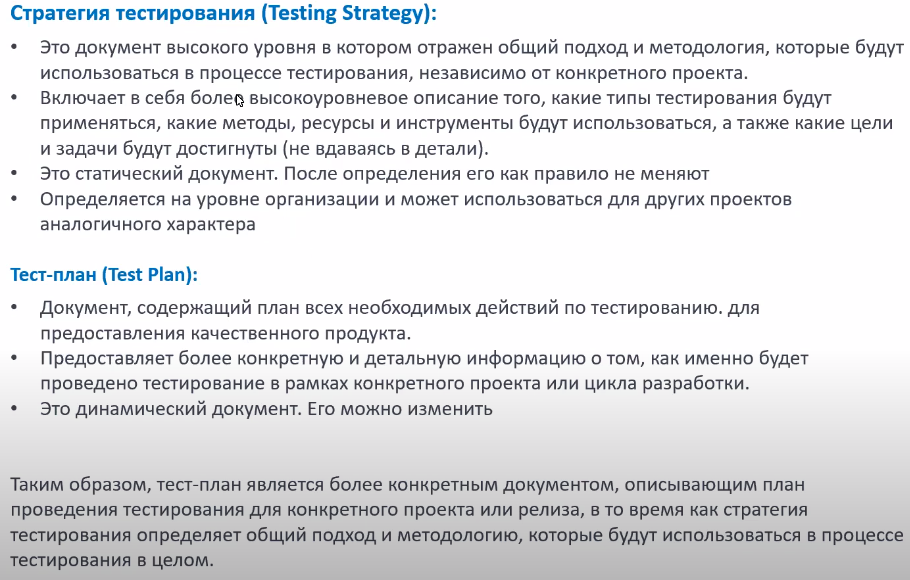
1)-Планирование

2)-Тест анализ

3)-Тест дизайн

4)-Выполнение тестов

5)-Анализ и отчёт о результатах тестирования

**1)-** **Планирование**

Стратегия тестирования - это некий общего характера документ (высокоуровневый, т.е. обобщённый).

Тест план отвечает на вопрос что будем тестировать, в какой последовательности, какими силами, т.е. это более конкретный документ.

**2) - Анализ**

**Анализ - это подготовка к написанию тест кейсов.** Нельзя сразу сесть и писать тест кейсы, сначала надо понять требования (разобратся), насколько они понятны и хорошо написаны, достаточно ли информации. Связь между требованиями(может быть так что на одном этапе длина логина не больше 8 символов, а в другом месте 10 символов) не должно быть противоречий. В каком то смысле мы тестируем и требования.

Разбиваем систему на некие модули, чтобы мы понимали как мы эту систему будем тестировать.

На этом этапе прописывается **тест сценарий**- в одну строку прописываем что мы будем тестировать. Высокоуровневое описание того, как будет происходить тестирование или проверка определённого функционала продукта. Описывает последовательность шагов, необходимых для достижения цели тестирования.

В рамках анализа тестирования нам нужно определить:

1)-Уровень детализации и качества наших требований.

2)-Связи между требованиями

3)-Уровни тестирования - на сколько глубоко нам надо протестировать каждое требование (требуется ли интеграционное, системное тестирование, и т.д.)

4)-Тестовая среда, уровень подготовки команды. Обычно делают 3 сервера: для разраб., для тестировщиков, продакшн. Команда достаточно компетентна для тестирования или надо её прокочать/обучить…

5)-Инструменты по управлению тестированием (Test management system), где мы будем создавать и хранить тесты (просто в XL) или специализированная система, типа jira или Trello.

6)-Доступность для консультации других участников проекта. Бывает что на проекте работаю разные команды не связанные друг с другом, и на границах интересов могут возникать вопросы требующие участия из других команд и на этапе анализа надо это предусмотреть, чтобы имелась возможность общения между командами, что то типа контактного лица в каждой из команд и как они будут связыватся.

Другими словами мы пока не знаем как функционал будет реализован но мы понимаем как мы будем тестировать.

Например: проверка процесса оформления заказа.

В текущем моменте у нас нету данной функции, мы не знаем как это будет реализовано, но мы знаем что эта функция будет и мы как бы обсуждаем как мы будем (как мы бы могли) его тестировать, т.е. мы проговариваем/планируем сценарий этого теста.

1)мы добавляем товар в корзину

2)мы переходим в корзину, добавленные товары там отображены.

3)пользователь оформляет способ доставки и заполняет необходимую для этого форму

4)пользователь выбирает способ оплаты, вводит необходимые данные

5)пользователь подтверждает заказ, ему должно прийти письмо с подтверждением.

**Цель данного этапа:**

**1) - Создание ясности благодаря определению точных целей тестов. Мы знаем что именно и на сколько глубоко нужно тестировать. (потому что изучили документацию, где описан функционал будущего П.О.)**

**2)Предотвращение дефектов: именно на этапе анализа проводится статическое тестирование**

Пояснения:

Когда система запущена и всё работает, то тестирование такой системы называется динамическое.

Статическое тестирование, почему так называется? Потому что у нас есть некий документ и всё, системы то у нас нет, тогда что мы тестируем? Мы тестируем требования

**3)-Тест дизайн**

Определяем как именно должно быть протестировано то, что было определено в рамках анализа. Другими словами это процесс перевода анализа тестирования в конкретные тесты (чек-листы, тест-кейсы и т.д.).

Чек лист - некий общий список того что мы будем тестировать

Тест кейс - детальная проработка каждого пункта чек листа где мы прописываем шаги последовательности тестирования(нажать кнопку, перейти туда-то, ввести то-то… конкретные шаги).

Если система маленькая возможно обойтись только чек листом, но если система большая и на проекте работает много тестировщиков то нужны чек листы, где конкретно описано как и что надо проверить и что для этого надо. Особенно когда есть текучка кадров и на проект приходит новый (возможно мало опытный)специалист.

**4)-Выполнение тестов**

Это не просто взял чек лист или тестовый сценарий и пошол по нему. В теории этот эьап существует из 2 шагов:

**Test implementation** - подготовка к началу тестирования.

1- Расстоновка приорететов активностей выполнения тестирования (прежде чем наченать надо определится в каком порядке мы вообще будем тестировать, тест кейсы надо отсортировать по приоритету важности)

2- Создание наборов тетст-кейсов (тест-сьютов) для последующего выполнения

3- возможно создание автоматизированных сценариев.

4- Настройка необходимого тестового окружения (это больше про начало работы и речь идёт о настройках компьютеров, наличия программ, подключений к серверу и т.д.)

5- Подготовка тестовых данных (например тестовый пользователь у него должен быть пароль, логин, если это товар то сначала его надо добавить в БД, потому, что когда система только разработана она пустая, а для тестирования нужны какие то стартовые данные).

К началу выполнения тестирования должно уже быть готово и настроено:

environments (тестовые среды),

Test management systems - где у вас заведены тест кейсы, т.е. не на бумаге а в системе типа Jira,

Bug tracker - система где фиксируются найденные ошибки

**Text execution** - собственно само выполнения тестов

1)- Выполнения тестов из выбранных наборов (manual / automation);

2)-Сравнения фактических и ожидаемых результатов

3)-Состовление отчётов о дефектах

4)-Проставление результатов выполнения тестов (пройден, не пройден, заблокирован и т.д.)

5)-В случае нахождения и исправления дефекта - Retest или регрессионное тестирование.

**5)-Анализ и отчёт о результатах тестирования**

Когда тестирование проведено мы составляем некий финальный отчёт - что мы протестировали, сколько тест кейсов проведено, какие из них успешные какие нет, сколько критичных (не критичных) ошибок было найдено, были они исправлены или нет. На основании этого отчёта команда (менеджмент) будут принимать какое то решение о качестве полученного продукта (версии) и можем ли мы отдовать проект в продакшн (конечным пользователям)