

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

Как альтернатива некоторые авторы выделяют следующие этапы **разработки** по (жизненный цикл по):

1-формирование требований (понять что вообще за продукт будет, как он должен работать, что делать)

2-создание тех анализа (более детальное и понятное команде описание как всё должно быть реализовано)

3-дизайн (дизайнер продумывает как продукт будет выглядеть)

4-разработка (кодинг)

5-Тестирование. В зависимости от задач пользователю могут отдать жизнеспособную версию в которой те или иные баги есть но основной функционал работает.

5-поддержка/улутшения

**STLC** - это процесс тестирования, который включает в себя определенную последовательность шагов, чтобы гарантировать достижение целей в области качества. В процессе STLC каждое действие выполняется планомерно и систематически. Каждый этап имеет разные цели и результаты. У разных организаций разные этапы STLC, однако основа остается прежней. 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.

На картинке сверху (и это только пример одного из авторов) 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

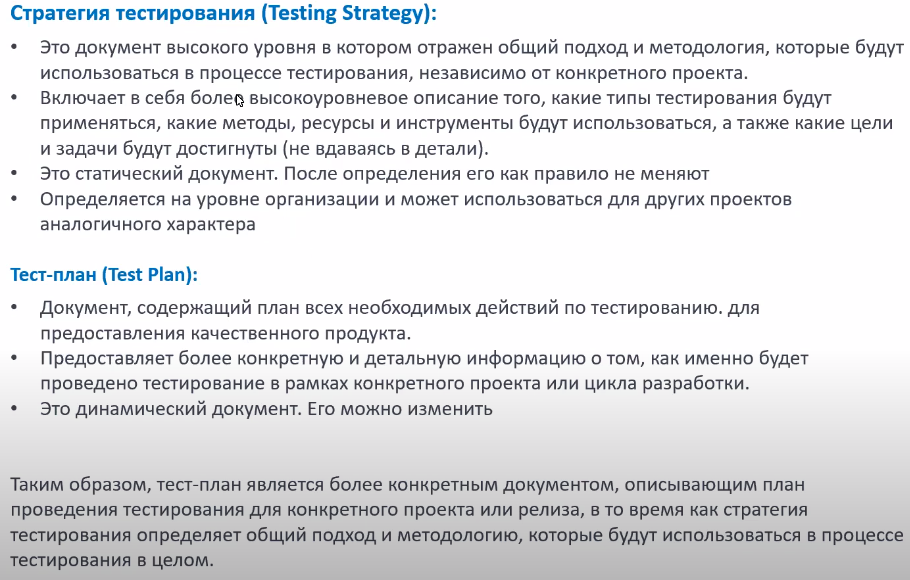
2)-Тест анализ

3)-Тест дизайн

4)-Выполнение тестов

5)-Анализ и отчёт о результатах тестирования

! У этого автора нет пункта “общее планирование и анализ требований” который у других идёт первым!

**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 этап 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)

ВНИМАНИЕ!

STLC по другому автору может иметь схожие нo немного другие этапы, например:

1Тестирование требований

2тестирование макетов - имеется в виду этап дизайн в sdlc какие то разрабатываемые макеты (например дизайнерские)и тестировщик при тестировании их замечает неточность/некорректность/баг или предлагает исправить тот или иной момент

3 проектирование тестов - думаем что и как будем проверять, какие данные будем использовать для проверок, какие сценарии надо бы в первую очередь какие во вторую (расставляем приоритет).

4 тестирование по - наслождаемся поиском багов до того как их нашол пользователь

5 разбор ошибок пользователя

по другому автору!

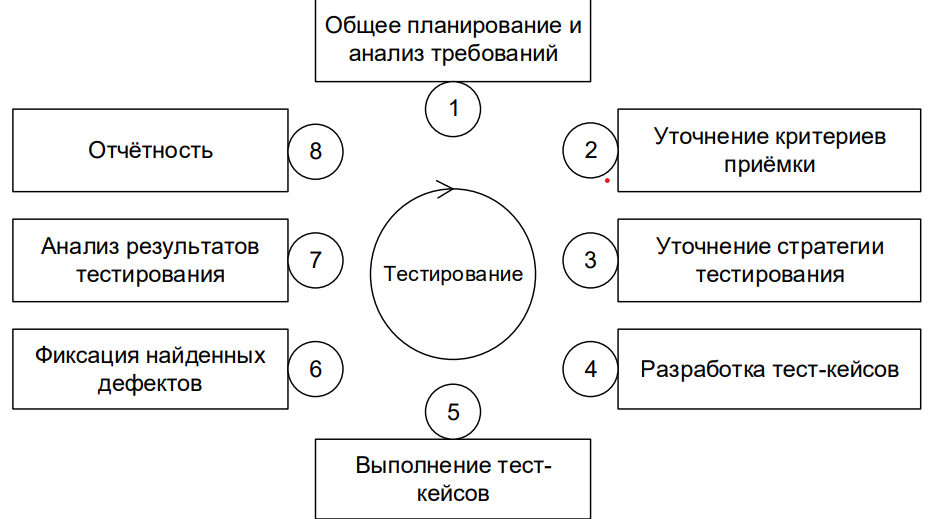
Следуя общей логике итеративности, превалирующей во всех современных

моделях разработки ПО, жизненный цикл тестирования также выражается замкнутой последовательностью действий

Важно понимать, что длина такой итерации (и, соответственно, степень подробности каждой стадии) может варьироваться в широчайшем диапазоне — от единиц часов до десятков месяцев. Как правило, если речь идёт о длительном промежутке времени, он разбивается на множество относительно коротких итераций,

но сам при этом «тяготеет» к той или иной стадии в каждый момент времени (например, в начале проекта больше планирования, в конце — больше отчётности).

Также ещё раз подчеркнём, что приведённая схема — не догма, и вы легко можете найти альтернативы, но общая суть и ключевые принципы остаются неизменными. Их и рассмотрим.



**Стадия 1** (общее планирование и анализ требований) объективно необходима как минимум для того, чтобы иметь ответ на такие вопросы, как: что нам предстоит тестировать; как много будет работы; какие есть сложности; всё ли необходимое у нас есть и т.п. Как правило, получить ответы на эти вопросы невозможно

без анализа требований, т.к. именно требования являются первичным источником ответов.

**Стадия 2** (уточнение критериев приёмки) позволяет сформулировать или

уточнить метрики и признаки возможности или необходимости начала тестирования (entry criteria44), приостановки (suspension criteria45) и возобновления (resumptioncriteria46) тестирования, завершения или прекращения тестирования (exit criteria47).

**Стадия** 3 (уточнение стратегии тестирования) представляет собой ещё одно

обращение к планированию, но уже на локальном уровне: рассматриваются и уточняются те части стратегии тестирования (test strategy48), которые актуальны для текущей итерации.

**Стадия** 4 (разработка тест-кейсов) посвящена разработке, пересмотру, уточнению, доработке, переработке и прочим действиям с тест-кейсами, наборами тесткейсов, тестовыми сценариями и иными артефактами, которые будут использоваться при непосредственном выполнении тестирования.

**Стадия** 5 (выполнение тест-кейсов) и стадия 6 (фиксация найденных дефектов) тесно связаны между собой и фактически выполняются параллельно: дефекты

фиксируются сразу по факту их обнаружения в процессе выполнения тест-кейсов.

Однако зачастую после выполнения всех тест-кейсов и написания всех отчётов о

найденных дефектах проводится явно выделенная стадия уточнения, на которой

все отчёты о дефектах рассматриваются повторно с целью формирования единого

понимания проблемы и уточнения таких характеристик дефекта, как важность и

срочность.

**Стадия** 7 (анализ результатов тестирования) и стадия 8 (отчётность) также

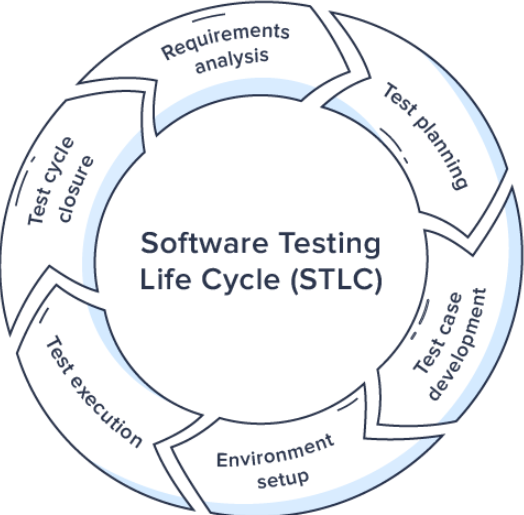
тесно связаны между собой и выполняются практически параллельно. Формулируемые на стадии анализа результатов выводы напрямую зависят от плана тестирования, критериев приёмки и уточнённой стратегии, полученных на стадиях 1, 2 и 3.

Полученные выводы оформляются на стадии 8 и служат основой для стадий 1, 2 и

3 следующей итерации тестирования. Таким образом, цикл замыкается.

или ещё по другому автору:

STLC имеет несколько взаимосвязанных фаз и в целом очень похож на SDLC. Эти фазы являются последовательными и называются:



**-) Анализ требований (Requirement Analysis)**: один из важнейших этапов, потому что именно на нем можно почти бесплатно исправить недостатки проекта. Этап анализа требований также определяет потенциальную потребность в автоматизированном тестировании и позволяет производить экономические расчеты затрат на рабочую силу на основе оценки проекта. На этом же этапе обсуждаются и документируются критерии начала и окончания тестирования

смотри -

1) <https://docs.google.com/document/d/1g6nNo7ri1-iHxJfROTaMnHoGvG3VKEPMKuR0_EvPF-w/edit>

2) <https://drive.google.com/drive/u/0/folders/1WYhNQLHWyyPGaEVY4_WZOVanA_RH7xRD>

**-) Планирование тестирования (Test Planning)**: на этом этапе формируется план тестирования, т.е. мы определяем действия и ресурсы, которые помогут достичь целей тестирования (участники и их роли, инструменты, окружение). Во время планирования мы также пытаемся определить метрики, метод сбора и отслеживания этих метрик. План составляют исходя из требований, тестовой стратегии и анализа рисков

**-) Разработка тест-кейсов (Test Case Development)**: подразумевает использование ручного и автоматизированного тестирования для достижения полного охвата функциональности программного обеспечения, при этом процесс основан на заранее установленных требованиях. Чаще всего тест-кейсы для автоматического тестирования пишутся отдельно, так как кейсы для ручного тестирования описаны в виде шпаргалок (cheat sheets)

**-) Настройка тестовой среды (Test Environment Setup)**: в плане тестирования четко указано, какую тестовую среду следует использовать. На этом этапе STLC настраиваются операционные системы и виртуальные машины, развертываются инструменты тестирования, такие как Selenium, Katalon Studio, а также тестовая среда и базы данных проекта. Мы также обращаемся с запросами к DevOps и администраторам, если требуется поддержка

**-) Выполнение тестов (Test Execution)**: тесты выполняются на основе готовой тестовой документации и правильно настроенной тестовой среды. Все результаты тестирования регистрируются в Системе управления тестированием. Отрицательно пройденные тесты, в которых фактический результат отличается от ожидаемого, регистрируются как ошибки и передаются команде разработчиков на доработку с последующей перепроверкой после исправления

**-) Завершение цикла испытаний (Test Cycle Closure):** окончательная генерация отчетов о тестировании для клиента. Они должны включать затраченное время, процент обнаруженных ошибок и положительных результатов тестирования, общее количество обнаруженных и исправленных ошибок. Что касается отдела тестирования, то это момент для анализа его работы, подведения итогов, анализа его продуктивности и возможности внести предложения по улучшению качества тестирования