



STLC_SberMarket

В этом документе создана схема STLC и описан жизненный цикл тестирования доработки функционала «Доставка» Сбермаркета.

STLC, или **жизненный цикл тестирования доработки функционала “Доставка”** — это последовательность действий, проводимых в процессе тестирования, с помощью которых гарантируется качество программного обеспечения и его соответствие требованиям. STLC включает действия по верификации и валидации. Тестирование состоит из серии действий, выполняемых по методике, с целью гарантирования качества продукта.

Этапы STLC-цикла

Цикл состоит из шести основных этапов:



1. Анализ требований

Тестирование на этапе оценки требований (или анализа требований). На этом этапе отдел QA оценивает требования с точки зрения тестирования, ищет требования к софту, которые нужно предварительно оценить. Для этого QA - команда может обращаться к представителям заказчика и аналитиком. Требования могут быть «функциональными» или «нефункциональными», то есть касаться или не касаться функциональной составляющей софта. Также на этом этапе проводится оценка возможности применения автоматизированного тестирования.

Действия на этапе оценки требований

- Определение типов тестирования
- Сбор информации о приоритетах в тестировании
- Подготовка матрицы отслеживания требований (RTM — **Requirement** Traceability Matrix)
- Определение тестового окружения
- Анализ возможности автоматизации тестирования (если нужно)

Результаты этапа оценки требований

- Матрица отслеживания требований (RTM)
- Отчет о возможности автоматизации (если нужно)

Виды тестирования

Тестирование требований

Тестовое окружение: Notion.

2. Планирование тестирования

На этапе планирования руководитель команды QA определяет стратегию тестирования и оценивает трудозатраты. Также оцениваются ресурсы, тестовое окружение, возможные ограничения и график тестирования. На этом же этапе готовится и финализируется план тестирования. QA-команда может обращаться к представителям команды аналитиков.

Действия на этапе планирования

- Подготовка стратегии (или плана тестирования)
- Выбор инструментов тестирования
- Оценка трудозатрат
- Планирование ресурсов, определение ролей и ответственности
- Дополнительное обучение команды (если нужно)

Результаты

- Документ со стратегией тестирования
- Документ с оценкой ресурсов

Тестовое окружение: Notion, Xmind

3. Создание тест-кейсов

На этом этапе происходит подготовка тестовых данных и создаются тест-кейсы.

Действия

- Создание тест-кейсов (и автотестов, если будет применяться автоматизация)
- Подготовка исходных данных для тестирования

Результаты

- Тест-кейсы и/или скрипты
- Тестовые данные

Виды тестирования

Функциональное тестирование, тестирование end-to-end, негативное тестирование, интеграционное тестирование.

Тестовое окружение

Notion, TestRail

4. Настройка тестового окружения

Это **настройка** харда и софта, в которых будет осуществляться процесс тестирования. Это один из критически важных аспектов процесса, он может проходить параллельно этапу создания тест-кейсов. QA-команда может и не включаться в этот процесс, если тестовое окружение ей обеспечит команда разработки. QA-команда должна будет проверить работоспособность окружения (хотя бы smoke-тестом).

Действия

- Понять нужную архитектуру, настройки окружения и подготовить список требований к харду и софту
- Настроить тестовое окружение и тестовые данные
- Провести smoke-тест окружения

Результаты

- Настроенное окружение для проведения тестирования
- Результаты smoke-тестирования окружения

Тестовое окружение

Notion, Windows 10 Корпоративная LTSC, Android 11 RKQ1.201004.002

5. Выполнение тестирования

На **этапе выполнения тестов** QA проводит тестирование, выполняя подготовленные тест-кейсы. Процесс состоит из выполнения тестовых скриптов (при необходимости эти скрипты могут корректироваться). Далее идет создание баг-репортов. Если найдены баги, информация о них передается команде разработки для исправления и повторного тестирования QA-командой.

Действия

- Выполнение тестирования в соответствии с планом
- Получение результаты тестирования
-

Обновление RTM-матрицы (тест-кейсы из RTM-матрицы связываются с найденными багами)

- Повторное тестирование исправленных багов

Результаты

- Завершенная RTM-матрица
- Обновленные тест-кейсы
- Найденные и описанные баги

Виды тестирования

Функциональное тестирование, тестирование end-to-end, негативное тестирование, интеграционное тестирование.

Тестовое окружение

Windows 10 Корпоративная LTSC, Android 11 RKQ1.201004.002, Fiddler Classic, DevTools

6. Завершение цикла тестирования

На этапе **завершения тестирования** создается отчет о результатах тестирования. QA-команда обсуждает и анализирует баги, делает выводы из возникших проблем, чтобы избежать подобных проблем в будущем.

Действия

- Оценка критериев завершения цикла (основывается на времени, трудозатратах, покрытии тестами)
- Подготовка документа с выводами, сделанными во время тестирования
- Подготовка отчета о завершении тестирования
- Подготовка отчета для клиента с количественными и качественными характеристиками тестируемой системы
- Анализ результатов тестирования

Результаты

- Отчет о завершении тестирования

Тестовое окружение

Notion, TestRail, RedMine

Тестовая документация

Требования, тест-план и тест стратегия, тест-кейсы, тестовые данные, багрепорты, RTM-матрица, отчет о результатах тестирования.

Тестовая среда для этапа анализ требований

Команда QA анализирует предварительные условия и детали среды, в которой предполагается проведение тестирования.

Среда разработка (Development Env) – это среда, в которой работают программисты. Здесь они занимаются написанием и отладкой кода, а также выполняют модульное тестирование

Среда тестирования (Test Env) – это окружение, в котором работает команда QA. Здесь выполняется проверка функциональности и регресс с использованием тестовых данных. Как правило эта среда не связана или частично связана с внешними системами (нет полноценной интеграционной схемы)

Превью (интеграционная) среда (Preview Env) – это среда с настроенной интеграционной схемой между системами и продуктами, а также со структурой данных, приближенной к продуктивной

Продуктивная среда (Production Env) – это окружение, в котором развернуто ПО, где продукт доступен пользователям.