

# STLC\_SberMarket

## Функционал «Доставка»

**STLC**, или **жизненный цикл тестирования** — 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:

### 1. Анализ требований к тестированию.

На этом этапе группа тестирования изучает требования к программному обеспечению с целью определения требований к тестированию.

Команда QA свяжется с различными заинтересованными сторонами (клиент, бизнес-аналитик, технические руководители, системные архитекторы и т. Д.), Чтобы детально понять требования.

Требования могут быть функциональными (определяющими, что должно делать программное обеспечение) или нефункциональными (определяющими производительность и безопасность системы).

Мы также определяем возможность автоматизации для данного тестового проекта на этом этапе.

#### Деятельность:

- Определим типы тестов, которые мы должны выполнить.
- Соберём информацию о приоритетах тестирования и фокусе.
- Подготовим все требования и создадим матрицу покрытия.
- Определим подробности среды тестирования в месте, где мы должны выполнять тесты.
- Анализ осуществимости автоматизации (при необходимости).

#### Виды тестирования:

Тестирование требований

**Результаты этапа:**

- Матрица покрытия.
- ТЭО автоматизации (если применимо).

**Окружение:**

Устройство с ОС Windows 10, MacOS 12.5, браузер Google Chrome версия 106.0.5249.119 , Notion 2.18

## **2. Планирование тестирования в рамках STLC**

На этом этапе старший менеджер по обеспечению качества обычно устанавливает усилия и смету затрат на проект и завершает план тестирования. В настоящее время мы также определяем стратегию тестирования.

**Деятельность:**

- Подготовка плана тестирования / стратегического документа для различных типов тестирования.
- Подбор тестовых инструментов.
- Оценка результатов испытаний.
- Планирование ресурсов и определение ролей и обязанностей.
- Требование к обучению.

**Виды тестирования:**

Тестирование концепции.

**Результаты этапа:**

- План / стратегия тестирования.
- Оценка усилий.

**Окружение:**

Устройство с ОС Windows 10, MacOS 12.5, браузер Google Chrome 106.0.5249.119 , Notion 2.18, Mindmeister

## **3. Разработка тестового набора.**

На этом этапе тестовые примеры и тестовые сценарии создаются, проверяются и дорабатываются. Мы идентифицируем, создаем и оцениваем тестовые

данные, а затем снова их редактируем.

**Деятельность:**

- Скрипты автоматизации (если применимо).
- Подготовка базовых тестовых примеров и сценариев.
- Создать тестовые данные (если доступна тестовая среда).

**Виды тестирования:**

Модульное тестирование

**Результаты этапа:**

- Тест-кейсы.
- Тестовые данные.

**Окружение:**

Windows 10, MacOS 12.5, Google Chrome 106.0.5249.119 ,  
Notion 2.18, TestRail

## **4. Настройка среды тестирования.**

Тестовая среда определяет программные и аппаратные условия, в которых мы тестируем. Конфигурация тестовой среды - один из критических аспектов процесса тестирования. Этот тест может выполняться одновременно с этапом разработки тестового примера, если это возможно. Эта деятельность несет системы как правило, из которых тестовая группа не участвует. Поэтому испытательная группа должна будет выполнить тест на готовность (тест на дым) тестовой среды, чтобы определить, является ли она удовлетворительной.

**Деятельность:**

- Понять необходимую архитектуру, среду, а также составить список требований к оборудованию и программному обеспечению для тестовой среды.
- Настройка теста среды и тестовых данных.
- Выполнить smoke-тестирование на билде.

**Виды тестирования:**

smoke-тестирование

**Результаты этапа:**

- Настройка среды и тестовые данные готовы.
- Результаты испытаний smoke-тестирования.

#### **Окружение:**

Windows 10, MacOS 12.5, Google Chrome 106.0.5249.119 ,  
Notion 2.18, TestRail

## **5. Выполнение тестов.**

На этом этапе тестировщики будут выполнять тесты на основе планов тестирования и подготовленных тестовых примеров. Мы сообщаем об ошибках команде разработчиков для исправления и запускаем тест снова после исправления.

#### **Деятельность:**

- Выполнить тесты в соответствии с планом.
- Документирование результатов тестирования и регистрация ошибок.
- Проверить исправления.
- Следовать за дефектами до закрытия.

#### **Виды тестирования:**

Интеграционное, E2E, функциональное, юзабилити, регрессионное, нагрузочное, тестирование безопасности.

#### **Результаты этапа:**

- Матрица покрытия с завершенным статусом выполнения.
- Отчеты о дефектах.
- Составить отставание для технического обслуживания.

#### **Окружение:**

Windows 10, MacOS 12.5, Google Chrome 106.0.5249.119 ,  
Notion 2.18, TestRail, Postman, Redmine, Fiddler, iPhone 12 (iOS 16.0)

## **6. Закрытие цикла испытаний.**

Команда тестирования обсуждает и анализирует цикл тестирования, чтобы улучшить используемые стратегии тестирования. Мы должны извлечь уроки из

текущего цикла испытаний. Таким образом, идея состоит в том, чтобы устранить узкие места в процессе для будущих циклов тестирования. Мы также собираем Лучшие практики для подобных проектов в будущем.

#### **Деятельность:**

- Оценка критериев для завершения цикла, основываясь на времени, охвате тестирования, затратах, программном обеспечении, критических бизнес-целях и качестве.
- Подготовка статистики теста на основе вышеуказанных параметров.
- Документация деталей проекта.
- Подготовить отчет к завершению теста.
- Качественный и количественный отчет о качестве объекта испытаний клиенту.
- Анализ результатов испытаний для определения распределения дефектов по типу и серьезности.

#### **Виды тестирования:**

Приёмочное тестирование, тестирование пользовательской и сопроводительной документации

#### **Результаты этапа:**

- Заключительный отчет
- Тест-статистика
- Пользовательская и сопроводительная документация

#### **Окружение:**

Windows 10, MacOS 12.5, Google Chrome 106.0.5249.119 ,  
Notion 2.18, TestRail, Postman, Redmine, Fiddler, iPhone 12 (iOS 16.0)