

# QA Manual (Практика)

## Для кого этот документ

Junior QA / Manual QA, включая ситуацию «один QA на проекте». Материал прикладной, с примерами из веб-приложений.

---

## 1. STLC — Software Testing Life Cycle

**STLC** — это жизненный цикл тестирования ПО. Он описывает **что, когда и зачем делает QA** на протяжении разработки.

Важно: STLC **не живёт отдельно** от SDLC, он идёт **параллельно**.

### 1.1 Этапы STLC (классическая модель — 6 этапов)

#### 1. Анализ требований (Requirement Analysis)

**Цель:** понять, что и как нужно тестировать.

**Входные артефакты:** - BRD / SRS / User Stories - Макеты (Figma) - Бизнес-правила

**Что делает QA:** - анализирует требования на тестируемость - выявляет противоречия и пробелы - формирует список вопросов к ВА / РО - определяет типы тестирования

**Пример (e-commerce):**

В требованиях написано: «Пользователь может оформить заказ без регистрации». QA уточняет: можно ли сохранить историю заказов? нужен ли email?

---

#### 2. Планирование тестирования (Test Planning)

**Цель:** определить **как именно** будет проходить тестирование.

**Результат:** Test Plan (или упрощённая версия).

**QA определяет:** - scope тестирования - типы тестирования (functional, regression и т.д.) - критерии входа / выхода - риски - инструменты

**Пример:**

Регрессионное тестирование перед релизом — обязательно, нагрузочное — вне scope.

---

### **3. Разработка тестовой документации (Test Design)**

**Цель:** подготовить сценарии проверки.

**Артефакты:** - Test cases - Checklists - Test scenarios

**Пример (позитивный сценарий):**

Пользователь добавляет товар в корзину → оформляет заказ → получает confirmation email.

---

### **4. Подготовка тестовой среды (Test Environment Setup)**

**Цель:** обеспечить условия для тестирования.

**QA проверяет:** - доступность стенда - тестовые данные - версии браузеров - зависимости (API, платежи, email)

**Пример:**

Тестовая платёжная система (sandbox), тестовые карты.

---

### **5. Выполнение тестирования (Test Execution)**

**Цель:** выполнить тесты и зафиксировать дефекты.

**QA делает:** - выполняет тест-кейсы - заводит баг-репорты - проводит re-test и regression

**Артефакты:** - Bug reports - Test execution results

---

### **6. Завершение тестирования (Test Closure)**

**Цель:** подвести итоги.

**QA формирует:** - Test Report - список известных дефектов - вывод о качестве сборки

---

## 2. Verification & Validation (V&V)

Классическая и **очень любимая на собеседованиях** тема.

### 2.1 Verification — Верификация

**Определение:**

Проверяем, **правильно ли мы разрабатываем продукт**.

**Фокус:** соответствие спецификациям.

**Отвечает на вопрос:**

«Мы делаем продукт правильно?»

**Когда:** - до выполнения кода - на этапе требований и дизайна

**Примеры verification:** - review требований - review дизайна - review тест-кейсов

**Пример:**

В ТЗ указано поле **email** — QA проверяет, что оно есть в макете.

---

### 2.2 Validation — Валидация

**Определение:**

Проверяем, **тот ли продукт мы делаем**.

**Фокус:** соответствие ожиданиям пользователя.

**Отвечает на вопрос:**

«Мы делаем правильный продукт?»

**Когда:** - после реализации - при выполнении тестов

**Примеры validation:** - функциональное тестирование - UAT - exploratory testing

**Пример:**

Пользователь ожидает, что кнопка «Купить» сразу ведёт к оплате, а не к регистрации.

---

## **Короткая формула для запоминания**

**Verification** — build the product right

**Validation** — build the right product

---

# **3. Ошибки, дефекты и отказы (Error, Defect, Failure)**

Очень важнаяterminология.

## **3.1 Error (Ошибка)**

**Определение:**

Человеческая ошибка — неправильное действие или решение.

**Кто допускает:** - аналитик - разработчик - дизайнер - QA

**Пример:**

Аналитик забыл указать обязательность поля `phone`.

---

## **3.2 Defect (Bug)**

**Определение:**

Реализация, не соответствующая требованиям.

**Где живёт:** - в коде - в конфигурации

**Пример:**

Поле `email` принимает строку без `@`.

---

## **3.3 Failure (Сбой)**

**Определение:**

Наблюдаемое некорректное поведение системы во время работы.

**Проявляется:** - на проде - на тестовом стенде

**Пример:**

Пользователь нажал «Оплатить» → страница упала с 500 ошибкой.

---

### Связь терминов

Error → Defect → Failure

Или проще:

человек ошибся → в коде баг → пользователь увидел проблему

---

## 4. Краткий практический пример (e-commerce)

**Ситуация:** оформление заказа.

- Error: разработчик неверно понял бизнес-логику скидок
- Defect: скидка применяется дважды
- Failure: итоговая цена отрицательная

QA фиксирует **defect**, пользователь сталкивается с **failure**, корень — **error**.

---

### Итог

Этот мануал — практическая база: - STLC — как QA работает по этапам - V&V — что и зачем мы проверяем - Error / Defect / Failure — как правильно говорить о проблемах

Дальше логично продолжить: - STLC vs Agile (Scrum) - STLC в реальном проекте без документации - как из этого делать тест-кейсы и баг-репорты

Готов продолжать 👉