

# Manual QA: руководство для Junior QA (Web / E-commerce)

---

## 1. Теория качества: ISO/IEC 25010 (кратко и по делу)

**ISO/IEC 25010** — международный стандарт модели качества ПО. Для Junior QA важно понимать не формулировки, а что именно проверять руками.

### 1.1 Основные характеристики качества

#### 1. Функциональная пригодность (Functional suitability)

Работает ли функциональность так, как описано в требованиях.

#### 2. Производительность (Performance efficiency)

Скорость загрузки страниц, время отклика.

#### 3. Совместимость (Compatibility)

Работа в разных браузерах, отсутствие конфликтов.

#### 4. Удобство использования (Usability)

Понятный интерфейс, читаемые тексты, логичные действия.

#### 5. Надёжность (Reliability)

Система не падает, корректно обрабатывает ошибки.

#### 6. Безопасность (Security)

Валидация данных, защита форм, доступы.

#### 7. Сопровождаемость (Maintainability)

Косвенно: читаемые ошибки, предсказуемое поведение.

#### 8. Переносимость (Portability)

Работа на разных устройствах и ОС.



Junior QA не обязан проверять всё глубоко, но должен думать в этих категориях.

---

## 2. Что такое ручное тестирование

**Ручное тестирование** — это проверка веб-приложения без автотестов, глазами и логикой пользователя.

Цель: - найти ошибки до пользователя - проверить сценарии использования - убедиться, что система ведёт себя ожидаемо

---

## 3. Сценарии тестирования

### 3.1 Позитивные сценарии

Проверяют корректную работу при **валидных данных**.

Пример: - добавить товар в корзину - оформить заказ с правильными данными

### 3.2 Негативные сценарии

Проверяют поведение системы при **ошибочных данных**.

Пример: - пустые обязательные поля - неверный email - недоступный товар

Ожидание: - понятная ошибка - отсутствие падений - данные не сохраняются

---

## 4. Пример тестирования e-commerce (практика)

### 4.1 Позитивный сценарий: оформление заказа

**Название:** Checkout — успешное оформление заказа

**Предусловия:** Товар есть в наличии

**Шаги:** 1. Открыть каталог 2. Добавить товар в корзину 3. Перейти в checkout 4. Заполнить все обязательные поля корректно 5. Подтвердить заказ

**Ожидаемый результат:** - заказ успешно создан - пользователь видит страницу подтверждения

---

### 4.2 Негативный сценарий: обязательное поле

**Название:** Checkout — пустое обязательное поле

**Шаги:** 1. Перейти в checkout 2. Оставить поле Email пустым 3. Нажать «Оформить заказ»

**Ожидаемый результат:** - поле подсвеченено - показано сообщение об ошибке - заказ не создан

---

## 5. Тест-план (Test Plan)

**Тест-план** — это краткое описание *что, как и в каком объёме мы тестируем*. Для Junior QA тест-план может быть простым, но он помогает структурировать работу и ожидания.

### 5.1 Назначение тест-плана

Тест-план отвечает на вопросы: - Что именно тестируем? - Что не входит в тестирование? - В каком окружении проводится тестирование? - Какие типы тестов используются?

---

### 5.2 Пример простого тест-плана (Junior уровень)

Проект: E-commerce store

Цель тестирования:

Проверить базовую работоспособность пользовательского оформления заказа.

Область тестирования:

- Каталог товаров
- Корзина
- Checkout

Не входит в тестирование:

- Платёжные системы (mock / sandbox)
- Административная панель

Типы тестирования:

- Ручное функциональное тестирование
- Позитивные и негативные сценарии
- UI / Usability проверки

Окружение:

- Browser: Chrome (latest)
- OS: Windows / macOS
- Environment: Staging

Критерии завершения:

- Все критические сценарии выполнены
- Блокирующих багов нет

---

## 6. Протокол тестирования (Test Report)

После тестирования Junior QA обычно оформляет **отчёт / протокол**.

## 5.1 Пример протокола

Проект: E-commerce store

Дата: 2026-01-XX

Тестировал: Junior QA

Проверенные модули:

- Каталог
- Корзина
- Checkout

Результат:

- Позитивные сценарии: OK
- Негативные сценарии: найдено 2 бага

Ссылки на баги:

- BUG-101
- BUG-102

## 6. Как правильно оформлять баг

**Summary:** Коротко, что не работает

**Environment:** Браузер, ОС

**Steps to reproduce:** Пошагово

**Actual Result:** Что происходит

**Expected Result:** Что должно быть

## 7. Бесплатные и условно бесплатные инструменты QA

### 7.1 Браузеры и DevTools

- Google Chrome
- Firefox
- DevTools: Console, Network, Elements, Mobile view

### 7.2 Документация и тест-кейсы

- Google Sheets
- Notion (Free)
- Markdown (Git / Confluence)

### **7.3 Баг-трекинг**

- Jira (Free для маленьких команд)
  - GitHub / GitLab Issues
- 

### **7.4 API и backend (базово)**

- Postman (Free)
  - Insomnia
- 

### **7.5 Responsive и визуальная проверка**

- Chrome DevTools (Device Mode)
  - PerfectPixel (pixel-perfect)
- 

### **7.6 Запись экрана / браузера (для багов)**

- **OBS Studio** — бесплатно, видео экрана
- **Loom (Free)** — быстрая запись браузера
- **Встроенная запись экрана** (Windows / macOS)

Видео багов сильно упрощает жизнь разработчикам.

---

## **8. Советы Junior QA**

- Тестируй как пользователь
  - Проверяй крайние случаи
  - Не бойся задавать вопросы
  - Лучше простой баг-репорт, чем умный, но непонятный
- 

## **9. Итог**

Этот документ можно использовать как: - учебный материал для Junior QA - внутренний QA-гайд - основу для PDF / DOC

Хороший QA — это не про инструменты, а про внимание к деталям.