Автоматизированные тест-кейсы для BugBuster - Маркетплейс

# 1. НАСТРОЙКИ ОКРУЖЕНИЯ И КОНФИГУРАЦИЯ

# 1.1 Конфигурация BugBuster

БАЗОВЫЕ НАСТРОЙКИ:

* Base URL: https://marketplace.test.ru
* Browser: Chrome 119+
* Viewport: 1920x1080 (desktop), 360x640 (mobile)
* Timeout: 30 секунд
* Retry attempts: 3
* Screenshot on failure: true
* Video recording: true для критических тестов

ТЕСТОВЫЕ ДАННЫЕ:

* Test user email: testuser001@yandex.ru
* Test user password: MyPassword123!
* Seller email: seller@test.ru
* Admin email: admin@test.ru
* Valid phone: +7 (905) 123-45-67

СЕЛЕКТОРЫ ЭЛЕМЕНТОВ:

* Email input: input[data-testid="email-input"]
* Password input: input[data-testid="password-input"]
* Submit button: button[data-testid="submit-btn"]
* Search input: input[data-testid="search-input"]
* Cart counter: span[data-testid="cart-counter"]
* Add to cart btn: button[data-testid="add-to-cart"]

# 1.2 Окружения тестирования

БРАУЗЕРЫ ДЛЯ АВТОМАТИЗАЦИИ:

* Chrome 119+ (основной)
* Firefox 119+ (кроссбраузерное тестирование)
* Safari 17+ (мобильное тестирование)

РАЗРЕШЕНИЯ ЭКРАНА:

* Desktop: 1920x1080, 1366x768
* Mobile: 360x640, 414x896
* Tablet: 768x1024

# 2. АВТОМАТИЗИРОВАННЫЕ ТЕСТЫ - ПОКУПАТЕЛЬ

# 2.1 Регистрация и авторизация

### TEST\_AUTO\_REG\_001: Успешная регистрация нового покупателя

**ОПИСАНИЕ:** Автоматизированный тест регистрации с валидными данными

**ПРИОРИТЕТ:** Critical

**ОКРУЖЕНИЕ:** Chrome, 1920x1080

**ДАННЫЕ:**

* email: "testuser\_" + timestamp + "@yandex.ru"
* password: "MyPassword123!"
* firstName: "Алексей"
* lastName: "Петров"
* phone: "+7 (905) 123-45-67"

ШАГИ АВТОМАТИЗАЦИИ:

1. Перейти на страницу: "/registration"

2. Дождаться появления элемента: input[data-testid="email-input"]

3. Заполнить поле: input[data-testid="email-input"] значением ${email}

4. Заполнить поле: input[data-testid="password-input"] значением ${password}

5. Заполнить поле: input[data-testid="password-confirm"] значением ${password}

6. Заполнить поле: input[data-testid="first-name"] значением ${firstName}

7. Заполнить поле: input[data-testid="last-name"] значением ${lastName}

8. Заполнить поле: input[data-testid="phone-input"] значением ${phone}

9. Установить чекбокс: input[data-testid="privacy-consent"]

10. Нажать кнопку: button[data-testid="register-btn"]

11. Дождаться появления элемента: div[data-testid="success-message"]

ПРОВЕРКИ (ASSERTIONS):

* Проверить что текст содержит: div[data-testid="success-message"] "письмо для подтверждения"
* Проверить что URL содержит: "/email-confirmation"
* Проверить что элемент видим: div[data-testid="confirmation-info"]
* Проверка в БД: Пользователь создан со статусом "unconfirmed"

### TEST\_AUTO\_AUTH\_001: Авторизация зарегистрированного пользователя

**ОПИСАНИЕ:** Автоматизированный тест входа в систему

**ПРИОРИТЕТ:** Critical

**ОКРУЖЕНИЕ:** Chrome, 1920x1080

**ДАННЫЕ:**

* email: "buyer@test.ru"
* password: "Password123!"

ШАГИ АВТОМАТИЗАЦИИ:

1. Перейти на страницу: "/login"

2. Заполнить поле: input[data-testid="email-input"] значением ${email}

3. Заполнить поле: input[data-testid="password-input"] значением ${password}

4. Нажать кнопку: button[data-testid="login-btn"]

5. Дождаться перехода на страницу: "/"

ПРОВЕРКИ (ASSERTIONS):

* Проверить что URL равен: "/"
* Проверить что элемент видим: a[data-testid="user-profile-link"]
* Проверить что текст содержит: span[data-testid="user-name"] "Покупатель"
* Проверить что cookie существует: "auth\_token"

### TEST\_AUTO\_AUTH\_002: Авторизация с неверными данными

**ОПИСАНИЕ:** Тест обработки ошибок при неправильных учетных данных

**ПРИОРИТЕТ:** High

**ДАННЫЕ:**

* email: "buyer@test.ru"
* password: "WrongPassword"

ШАГИ АВТОМАТИЗАЦИИ:

1. Перейти на страницу: "/login"

2. Заполнить поле: input[data-testid="email-input"] значением ${email}

3. Заполнить поле: input[data-testid="password-input"] значением ${password}

4. Нажать кнопку: button[data-testid="login-btn"]

5. Дождаться появления элемента: div[data-testid="error-message"]

ПРОВЕРКИ (ASSERTIONS):

* Проверить что текст содержит: div[data-testid="error-message"] "Неверный email или пароль"
* Проверить что URL равен: "/login"
* Проверить значение поля: input[data-testid="email-input"] равно ${email}
* Проверить что cookie не существует: "auth\_token"

# 2.2 Поиск и фильтрация товаров

### TEST\_AUTO\_SEARCH\_001: Поиск товара по названию

**ОПИСАНИЕ:** Автоматизированный тест поиска товаров

**ПРИОРИТЕТ:** Critical

**ДАННЫЕ:**

* searchQuery: "iPhone 15"

ШАГИ АВТОМАТИЗАЦИИ:

1. Перейти на страницу: "/"

2. Заполнить поле: input[data-testid="search-input"] значением ${searchQuery}

3. Нажать клавишу: "Enter" или нажать кнопку: button[data-testid="search-btn"]

4. Дождаться появления элемента: div[data-testid="search-results"]

5. Дождаться появления элементов: div[data-testid="product-card"]

ПРОВЕРКИ (ASSERTIONS):

* Проверить что URL содержит: "/search?q=iPhone%2015"
* Проверить что количество элементов > 0: div[data-testid="product-card"]
* Проверить что текст содержит: h1[data-testid="results-title"] "iPhone 15"
* Проверить что каждый элемент содержит текст: div[data-testid="product-title"] "iPhone"

### TEST\_AUTO\_FILTER\_001: Фильтрация товаров по цене

**ОПИСАНИЕ:** Тест применения ценового фильтра

**ПРИОРИТЕТ:** High

**ДАННЫЕ:**

* minPrice: "50000"
* maxPrice: "100000"

ШАГИ АВТОМАТИЗАЦИИ:

1. Перейти на страницу: "/category/electronics"

2. Дождаться появления элемента: div[data-testid="price-filter"]

3. Заполнить поле: input[data-testid="price-min"] значением ${minPrice}

4. Заполнить поле: input[data-testid="price-max"] значением ${maxPrice}

5. Нажать кнопку: button[data-testid="apply-filters"]

6. Дождаться обновления элемента: div[data-testid="product-list"]

ПРОВЕРКИ (ASSERTIONS):

* Проверить что URL содержит: "price\_min=50000&price\_max=100000"
* Проверить что все цены в диапазоне: div[data-testid="product-price"] между 50000 и 100000
* Проверить что элемент видим: div[data-testid="active-filters"]
* Проверить что текст содержит: span[data-testid="filter-chip"] "50 000 - 100 000 ₽"

# 2.3 Работа с корзиной

### TEST\_AUTO\_CART\_001: Добавление товара в корзину

**ОПИСАНИЕ:** Автоматизированный тест добавления товара в корзину

**ПРИОРИТЕТ:** Critical

**ПРЕДВАРИТЕЛЬНЫЕ УСЛОВИЯ:**

* Пользователь авторизован как buyer@test.ru

ШАГИ АВТОМАТИЗАЦИИ:

1. Перейти на страницу: "/product/iphone-15-pro-max"

2. Дождаться появления элемента: div[data-testid="product-details"]

3. Выбрать опцию: select[data-testid="color-select"] значение="black"

4. Выбрать опцию: select[data-testid="memory-select"] значение="256gb"

5. Установить количество: input[data-testid="quantity-input"] значение="1"

6. Нажать кнопку: button[data-testid="add-to-cart"]

7. Дождаться появления элемента: div[data-testid="cart-notification"]

ПРОВЕРКИ (ASSERTIONS):

* Проверить что текст содержит: div[data-testid="cart-notification"] "добавлен в корзину"
* Проверить что текст равен: span[data-testid="cart-counter"] "1"
* Проверить что текст кнопки изменился: button[data-testid="add-to-cart"] "Добавлено"
* Подождать 2 секунды и проверить текст кнопки: "Добавить в корзину"

### TEST\_AUTO\_CART\_002: Изменение количества товара в корзине

**ОПИСАНИЕ:** Тест изменения количества товара в корзине

**ПРИОРИТЕТ:** High

**ПРЕДВАРИТЕЛЬНЫЕ УСЛОВИЯ:**

* Пользователь авторизован
* В корзине есть товар с количеством 1

ШАГИ АВТОМАТИЗАЦИИ:

1. Перейти на страницу: "/cart"

2. Дождаться появления элемента: div[data-testid="cart-items"]

3. Очистить и заполнить поле: input[data-testid="quantity-input-0"] значением "3"

4. Нажать клавишу: "Enter" или нажать кнопку: button[data-testid="update-quantity"]

5. Дождаться обновления элемента: span[data-testid="item-total-0"]

ПРОВЕРКИ (ASSERTIONS):

* Проверить значение поля: input[data-testid="quantity-input-0"] "3"
* Проверить расчет цены: span[data-testid="item-total-0"] = цена \* 3
* Проверить что итог обновился: span[data-testid="cart-total"]
* Проверить счетчик корзины: span[data-testid="cart-counter"] "3"

### TEST\_AUTO\_CART\_003: Применение промокода

**ОПИСАНИЕ:** Тест применения промокода со скидкой

**ПРИОРИТЕТ:** Medium

**ПРЕДВАРИТЕЛЬНЫЕ УСЛОВИЯ:**

* Общая стоимость корзины: 10000 рублей
* Действующий промокод: "SALE10" (скидка 10%)

ДАННЫЕ:

* promoCode: "SALE10"
* expectedDiscount: 1000

ШАГИ АВТОМАТИЗАЦИИ:

1. Перейти на страницу: "/cart"

2. Заполнить поле: input[data-testid="promo-input"] значением ${promoCode}

3. Нажать кнопку: button[data-testid="apply-promo"]

4. Дождаться появления элемента: div[data-testid="promo-success"]

ПРОВЕРКИ (ASSERTIONS):

* Проверить что текст содержит: div[data-testid="promo-success"] "Промокод применен"
* Проверить что элемент видим: div[data-testid="discount-amount"]
* Проверить что текст содержит: span[data-testid="discount-value"] "-1 000 ₽"
* Проверить расчет итога: span[data-testid="final-total"] "9 000 ₽"

# 2.4 Оформление заказа

### TEST\_AUTO\_ORDER\_001: Успешное оформление заказа

**ОПИСАНИЕ:** Полный процесс оформления заказа

**ПРИОРИТЕТ:** Critical

**ПРЕДВАРИТЕЛЬНЫЕ УСЛОВИЯ:**

* Пользователь авторизован как buyer@test.ru
* В корзине товары на сумму 5000 рублей

ДАННЫЕ:

* city: "Москва"
* street: "ул. Тверская, д. 10, кв. 25"
* zipCode: "101000"

ШАГИ АВТОМАТИЗАЦИИ:

1. Перейти на страницу: "/cart"

2. Нажать кнопку: button[data-testid="checkout-btn"]

3. Дождаться загрузки страницы: "/checkout"

4. Проверить предзаполненное поле: input[data-testid="email"] "buyer@test.ru"

5. Заполнить поле: input[data-testid="city"] значением ${city}

6. Заполнить поле: input[data-testid="street"] значением ${street}

7. Заполнить поле: input[data-testid="zip"] значением ${zipCode}

8. Выбрать радиокнопку: input[data-testid="delivery-courier"]

9. Выбрать радиокнопку: input[data-testid="payment-card"]

10. Установить чекбокс: input[data-testid="terms-agree"]

11. Нажать кнопку: button[data-testid="place-order"]

12. Дождаться перехода на страницу: "/payment"

ПРОВЕРКИ (ASSERTIONS):

* Проверить что URL содержит: "/payment"
* Проверить что элемент видим: div[data-testid="order-summary"]
* Проверить что текст соответствует регулярному выражению: span[data-testid="order-number"] "#\d{6,}"
* Проверка в БД: Заказ создан со статусом "new"

# 3. АВТОМАТИЗИРОВАННЫЕ ТЕСТЫ - ПРОДАВЕЦ

# 3.1 Управление товарами

### TEST\_AUTO\_SELLER\_PROD\_001: Добавление нового товара

**ОПИСАНИЕ:** Автоматизированный тест добавления товара продавцом

**ПРИОРИТЕТ:** Critical

**ПРЕДВАРИТЕЛЬНЫЕ УСЛОВИЯ:**

* Пользователь авторизован как seller@test.ru

ДАННЫЕ:

* productName: "Кроссовки Nike Air Max 270 мужские"
* sku: "NIKE-AM270-BLK-42-" + timestamp
* category: "Обувь > Мужская обувь > Кроссовки"
* brand: "Nike"
* price: "12990"
* quantity: "15"
* description: "Стильные мужские кроссовки Nike Air Max 270"

ШАГИ АВТОМАТИЗАЦИИ:

1. Перейти на страницу: "/seller/products"

2. Нажать кнопку: button[data-testid="add-product-btn"]

3. Дождаться загрузки страницы: "/seller/products/add"

4. Заполнить поле: input[data-testid="product-name"] значением ${productName}

5. Заполнить поле: input[data-testid="product-sku"] значением ${sku}

6. Выбрать из выпадающего списка: select[data-testid="category"] значение="footwear-men-sneakers"

7. Выбрать из выпадающего списка: select[data-testid="brand"] значение="nike"

8. Заполнить поле: input[data-testid="price"] значением ${price}

9. Заполнить поле: input[data-testid="quantity"] значением ${quantity}

10. Загрузить файл: input[data-testid="main-image"] файл="test-product-image.jpg"

11. Заполнить текстовое поле: textarea[data-testid="description"] значением ${description}

12. Выбрать опцию: select[data-testid="size"] значение="42"

13. Выбрать опцию: select[data-testid="color"] значение="black"

14. Нажать кнопку: button[data-testid="save-and-publish"]

15. Дождаться появления элемента: div[data-testid="success-message"]

ПРОВЕРКИ (ASSERTIONS):

* Проверить что текст содержит: div[data-testid="success-message"] "успешно создан"
* Проверить редирект на: "/seller/products"
* Проверить элемент в таблице: tr[data-sku="${sku}"]
* Проверить бейдж статуса: span[data-testid="status-badge"] "На модерации"
* Проверка в БД: Товар создан с корректными данными

### TEST\_AUTO\_SELLER\_PROD\_002: Редактирование товара

**ОПИСАНИЕ:** Тест изменения цены и количества товара

**ПРИОРИТЕТ:** High

**ПРЕДВАРИТЕЛЬНЫЕ УСЛОВИЯ:**

* У продавца есть опубликованный товар с SKU "TEST-PRODUCT-001"

ДАННЫЕ:

* newPrice: "11990"
* newQuantity: "20"

ШАГИ АВТОМАТИЗАЦИИ:

1. Перейти на страницу: "/seller/products"

2. Найти строку: tr[data-sku="TEST-PRODUCT-001"]

3. Нажать кнопку: button[data-testid="edit-product"]

4. Дождаться загрузки страницы

5. Очистить и заполнить поле: input[data-testid="price"] значением ${newPrice}

6. Очистить и заполнить поле: input[data-testid="quantity"] значением ${newQuantity}

7. Нажать кнопку: button[data-testid="save-changes"]

8. Дождаться появления элемента: div[data-testid="success-message"]

ПРОВЕРКИ (ASSERTIONS):

* Проверить что текст содержит: div[data-testid="success-message"] "изменения сохранены"
* Проверить значение поля: input[data-testid="price"] ${newPrice}
* Проверить значение поля: input[data-testid="quantity"] ${newQuantity}
* Проверка в БД: Товар обновлен с новыми значениями

# 3.2 Управление заказами

### TEST\_AUTO\_SELLER\_ORDER\_001: Обработка нового заказа

**ОПИСАНИЕ:** Тест подтверждения заказа продавцом

**ПРИОРИТЕТ:** Critical

**ПРЕДВАРИТЕЛЬНЫЕ УСЛОВИЯ:**

* Существует новый заказ с ID "12345" и статусом "new"
* Продавец авторизован

ДАННЫЕ:

* orderId: "12345"
* comment: "Заказ принят в обработку, отправка завтра"

ШАГИ АВТОМАТИЗАЦИИ:

1. Перейти на страницу: "/seller/orders"

2. Найти строку: tr[data-order-id="${orderId}"]

3. Нажать кнопку: button[data-testid="view-order"]

4. Дождаться загрузки страницы

5. Выбрать из выпадающего списка: select[data-testid="status-select"] значение="confirmed"

6. Заполнить текстовое поле: textarea[data-testid="comment"] значением ${comment}

7. Нажать кнопку: button[data-testid="update-status"]

8. Дождаться появления элемента: div[data-testid="success-message"]

ПРОВЕРКИ (ASSERTIONS):

* Проверить что текст содержит: div[data-testid="success-message"] "Статус заказа обновлен"
* Проверить бейдж статуса: span[data-testid="status-badge"] "Подтвержден"
* Проверить что комментарий сохранен: div[data-testid="order-history"] содержит ${comment}
* Проверка в БД: Статус заказа изменен на "confirmed"

# 4. АВТОМАТИЗИРОВАННЫЕ ТЕСТЫ - ПРОИЗВОДИТЕЛЬНОСТЬ

# 4.1 Нагрузочное тестирование

### TEST\_AUTO\_PERF\_001: Нагрузочный тест поиска

**ОПИСАНИЕ:** Тест производительности поисковой системы

**ПРИОРИТЕТ:** High

**КОНФИГУРАЦИЯ НАГРУЗКИ:**

* Виртуальные пользователи: 100
* Продолжительность: 5 минут
* Время нарастания нагрузки: 30 секунд

СЦЕНАРИЙ НАГРУЗКИ:

1. Перейти на страницу: "/"

2. Ждать случайное время: 1-3 секунды

3. Искать случайный термин из списка: ["iPhone", "Samsung", "Nike", "Adidas"]

4. Дождаться результатов: div[data-testid="search-results"]

5. Измерить время отклика

6. Повторить

МЕТРИКИ:

* Среднее время отклика: < 1 секунды
* 95-й перцентиль: < 2 секунд
* Процент ошибок: < 1%
* Пропускная способность: > 50 запросов/секунду

### TEST\_AUTO\_PERF\_002: Тест производительности корзины

**ОПИСАНИЕ:** Нагрузочный тест операций с корзиной

**ПРИОРИТЕТ:** High

**КОНФИГУРАЦИЯ НАГРУЗКИ:**

* Виртуальные пользователи: 50
* Продолжительность: 3 минуты

СЦЕНАРИЙ НАГРУЗКИ:

1. Авторизоваться как тестовый пользователь

2. Добавить случайный товар в корзину

3. Обновить количество

4. Удалить из корзины

5. Измерить каждую операцию

МЕТРИКИ:

* Добавление в корзину: < 500мс
* Обновление количества: < 300мс
* Удаление из корзины: < 200мс
* Процент ошибок: < 0.5%

# 5. КРОССБРАУЗЕРНЫЕ АВТОМАТИЗИРОВАННЫЕ ТЕСТЫ

# 5.1 Тестирование в разных браузерах

### TEST\_AUTO\_CROSS\_001: Регистрация в Chrome

**ОПИСАНИЕ:** Базовый тест регистрации в Chrome

**ОКРУЖЕНИЕ:** Chrome 119+, 1920x1080

**ШАГИ:** [Наследуется от TEST\_AUTO\_REG\_001]

### TEST\_AUTO\_CROSS\_002: Регистрация в Firefox

**ОПИСАНИЕ:** Базовый тест регистрации в Firefox

**ОКРУЖЕНИЕ:** Firefox 119+, 1920x1080

**ШАГИ:** [Наследуется от TEST\_AUTO\_REG\_001]

### TEST\_AUTO\_CROSS\_003: Поиск товаров на мобильном

**ОПИСАНИЕ:** Тест поиска на мобильном устройстве

**ОКРУЖЕНИЕ:** Chrome Mobile, 360x640

**ДОПОЛНИТЕЛЬНЫЕ ШАГИ:**

* Включить touch события
* Проверить мобильные элементы: button[data-testid="mobile-menu"]

# 5.2 Адаптивность интерфейса

### TEST\_AUTO\_RESPONSIVE\_001: Переключение размеров экрана

**ОПИСАНИЕ:** Тест адаптивности при изменении размера окна

**ШАГИ АВТОМАТИЗАЦИИ:**

1. Установить размер окна: 1920x1080

2. Перейти на страницу: "/"

3. Проверить что элемент видим: nav[data-testid="desktop-nav"]

4. Установить размер окна: 768x1024

5. Проверить что элемент видим: nav[data-testid="tablet-nav"]

6. Установить размер окна: 360x640

7. Проверить что элемент видим: button[data-testid="mobile-menu"]

# 6. КОНФИГУРАЦИЯ АВТОМАТИЗАЦИИ

# 6.1 Настройки запуска тестов

СУТОЧНЫЕ АВТОМАТИЧЕСКИЕ ЗАПУСКИ:

* Smoke тесты: каждые 4 часа
* Regression тесты: ежедневно в 02:00
* Performance тесты: еженедельно в выходные
* Cross-browser тесты: еженедельно

ОКРУЖЕНИЯ ДЛЯ ЗАПУСКА:

* Test окружение: https://test.marketplace.ru
* Staging окружение: https://staging.marketplace.ru
* Pre-production: https://preprod.marketplace.ru

УВЕДОМЛЕНИЯ:

* Slack канал: #qa-automation
* Email: qa-team@marketplace.ru
* Dashboard: BugBuster UI

# 6.2 Отчетность

МЕТРИКИ АВТОМАТИЗАЦИИ:

* Время выполнения тестов
* Процент успешных/неуспешных тестов
* Обнаружение нестабильных тестов
* Метрики покрытия
* Бенчмарки производительности

АРТЕФАКТЫ:

* Скриншоты при ошибках
* Видеозаписи для критических тестов
* Логи браузерной консоли
* Логи сетевого трафика
* Снимки состояния БД

ИНТЕГРАЦИИ:

* Jira: автоматическое создание багов
* Confluence: обновление документации
* Jenkins/GitHub Actions: CI/CD pipeline
* Allure: детальные отчеты

# 7. ПОДДЕРЖКА И ОБСЛУЖИВАНИЕ

# 7.1 Стабильность тестов

СТРАТЕГИИ ПОВЫШЕНИЯ СТАБИЛЬНОСТИ:

* Явные ожидания вместо sleep
* Механизмы повтора для нестабильных тестов
* Паттерн Page Object Model
* Изоляция тестовых данных
* Очистка после каждого теста

МОНИТОРИНГ СТАБИЛЬНОСТИ:

* Порог нестабильных тестов: 5%
* Максимальное время выполнения: 30 минут
* Мониторинг использования памяти
* Обнаружение крашей браузера

# 7.2 Масштабирование

ПАРАЛЛЕЛЬНОЕ ВЫПОЛНЕНИЕ:

* Максимум 10 параллельных потоков
* Распределение по типам браузеров
* Балансировка нагрузки между серверами
* Docker контейнеры для изоляции

ОБЛАЧНОЕ ВЫПОЛНЕНИЕ:

* Интеграция с BrowserStack
* Поддержка Sauce Labs
* AWS Device Farm
* Локальная grid как fallback

ЗАКЛЮЧЕНИЕ

Данный набор автоматизированных тест-кейсов обеспечивает комплексное покрытие функциональности маркетплейса через BugBuster. Тесты структурированы для максимальной эффективности и поддерживаемости.

Ключевые преимущества автоматизации:

* Быстрое выполнение регресс-тестирования
* Стабильные результаты
* Интеграция в CI/CD pipeline
* Детальная отчетность
* Поддержка множественных браузеров и устройств

Рекомендуется начать с автоматизации критических тест-кейсов, затем постепенно расширять покрытие.