

QA-стратегия и план тестирования интернет-магазина

Период тестирования: 3 месяца разработки + 1 месяц поддержки

1. Общий подход и цели

- Обеспечить высокое качество продукта на каждом этапе SDLC
- Обнаружить баги на ранних стадиях разработки
- Обеспечить покрытие требований тестами на всех уровнях
- Обеспечить прозрачность QA-процесса через тестовую документацию и метрики
- Уменьшить регрессионные риски за счет автоматизации

2. Пирамида тестирования:

1. Unit-тесты
2. Integration-тесты
3. Системное тестирование
4. Приемочное тестирование

3. Этапы работ по неделям

Этап 1: Подготовительный (1 неделя)

Цель: понять проект, собрать требования, подготовить тестовую стратегию и окружение

Действия:

- Анализ требований
- Определение рисков и зависимостей
- Составление тест плана
- Планирование этапов тестирования
- Выбор и настройка тестовых инструментов
- Подготовка тестовых данных и тестового окружения

Этап 2: Альфа-версия (2-3 недели)

Цель: тестирование базовой версии приложения

Действия:

- Smoke и sanity тестирование при каждом билде
- UI и функциональное тестирование (просмотр каталога, добавление товаров в корзину, оформление заказа)
- Тест-дизайн: эквивалентное разбиение, граничные значения, сценарии использования
- API тестирование: проверка эндпоинтов каталога, корзины, оформление заказа
- Занесение багов в баг-трекер
- Начало покрытия автотестами smoke-сценариев

Документация:

- Чек-листы
- Тест-кейсы
- Баг-репорты

Метрики:

- Кол-во найденных багов
- Покрытие тестами

Этап 3: Разработка функционала (4-9 недели)

Цель: инкрементальное тестирование новых функций и начало регрессии

Действия:

- Интеграционное тестирование: добавление доставки, оплаты
- Тестирование бизнес-логики (цен, скидок, доступности)
- Тестирование разных сценариев оформления заказа (с доставкой, без, отмена)
- Интеграция с платежным шлюзом (стаб, мок, sandbox)
- Автоматизация основных regression-сценариев
- Кроссбраузерное тестирование

Методы:

- Регрессионное, интеграционное, системное тестирование
- Комбинированное открытие по путям/ состояниям

Этап 4: Подготовка к релизу (10-12 неделя)

Цель: провести полный цикл тестирования и зафиксировать качество перед релизом

Действия:

- Полный regression тест (ручной + авто)
- Нагрузочное тестирование
- Проверка безопасности
- Юзабилити и A/B тесты
- Smoke-тесты на прод окружении
- Сбор и анализ финальных QA-метрик
- Подготовка QA-отчета о готовности продукта

Метрики:

- Кол-во критичных/блокирующих багов
- Code coverage
- Показатели performance: TTFB, load time, error rate

Этап 5: Поддержка (13-16 неделя)

Цель: быстрая реакция на баги, мониторинг стабильности

Действия:

- Мониторинг ошибок (логирование, Sentry)
- Анализ отзывов пользователей
- Тестирование фиксов
- Поддержка автотестов, запуск по расписанию
- Актуализация документации

4. Управление рисками

- Неполные требования - вероятность средняя - влияние высокое
- Проблемы с платежной интеграцией – вероятность высокая – влияние критичное
- Регрессия при релизах – вероятность высокая – влияние среднее
- Перегрузка на релизе – вероятность средняя – влияние среднее

5. Документация

- План тестирования
- Тест-кейсы/чек-листы
- Bug reports
- Отчеты о тестировании
- Release checklist