

## ОТЧЕТ О СТРАТЕГИИ ТЕСТИРОВАНИЯ WEB ПРИЛОЖЕНИЯ

Данный отчет охватывает период от разработки первой альфа-версии web интернет-магазина до истечения срока поддержки проекта после релиза (1 месяц).

Отметим ключевые функции интернет-магазина, а также данные по срокам разработки.

*Через данный сайт посетитель может:*

- Просматривать каталог товаров;
- Добавлять заинтересовавшие товары в корзину;
- Оформить заказ;
- Выбрать способ доставки;
- Оплатить заказ онлайн через сторонний сервис.

*На данный момент разрабатывается первая альфа-версия, которая будет готова через две недели. Само приложение должно быть выпущено через 3 месяца. После выпуска приложения выделяется 1 месяц на поддержку.*

Обозначим следующие этапы разработки, подразумевающие вовлечение QA инженеров в процесс:

1. Разработка альфа-версии;
2. Тестирование альфа-версии;
3. Исправление ошибок в альфа-версии;
4. Регрессионное тестирование;
5. Выпуск приложения и поддержка после релиза.

Далее указанные этапы будут рассмотрены подробнее.

### **1 Разработка альфа-версии**

На данном этапе QA инженер занимается тестированием спецификации и составлением тест-кейсов согласно спецификации на web приложение.

Первая неделя работы посвящена тестированию спецификации. Приступивший к работе над проектом QA инженер внимательно изучает спецификацию на приложение, в случае некорректных формулировок или неточностей заводятся репорты, которые направляются команде

разработчиков на уточнение. Спецификация дополняется необходимой информацией или исправляется.

По итогам тестирования спецификации составляется диаграмма состояний и переходов (ДСП), упрощающая написание тест-кейсов.

Вторая неделя посвящена написанию тестовых сценариев на приложение на основе протестированной ранее спецификации и в соответствии с ДСП. Тестовые сценарии должны отвечать следующим требованиям:

- Атомарность – проверка только одной конкретной функции;
- Однозначность – человек, проводящий тестирование, должен четко понимать, что от него требуется выполнить в сценарии;
- Воспроизводимость – возможность многократного повторения тестового сценария;
- Независимость – каждый тест-кейс должен быть самостоятельным и не зависеть от результатов других тестовых сценариев (или зависеть минимально);
- Конкретность – шаги должны быть описаны максимально точно и подробно.

Сценарии покрывают весь функционал приложения согласно спецификации.

Каждый тест-кейс включает в себя индивидуальный номер, заголовок (как правило емкий и точно описывающий суть тест-кейса), предусловие (некоторые сценарии подразумевают предварительное выполнение некоторых действий, например, авторизацию пользователя), шаги воспроизведения (при этом некоторые очевидные вещи, как, например, щелчок по левой кнопке мыши при взаимодействии с кнопками на сайте, опускаются, достаточно использования формулировки «нажать кнопку»), ожидаемый результат выполнения тест-кейса. При необходимости указывается окружение (браузер, версия ОС и тд).

Дополнительно создается графа для оценки трудозатрат на выполнение тестовых сценариев, проводится предварительный подсчет времени, который потребуется на выполнение составленных тест-кейсов.

## **2 Тестирование альфа-версии**

Через 2 недели после начала работы QA инженера в проекте будет готова первая альфа-версия приложения. К этому моменту протестирована спецификация и составлены тест-кейсы.

Тестирование включает в себя несколько стадий:

1. Функциональное тестирование. Проверка соответствия продукта спецификации. Подразумевает:

- Тестирование переходов между карточками товаров, возможности выбора и добавления товаров в корзину. Если при поиске товаров предусмотрены фильтры, проверить корректность их работы;

- Тестирование функционала корзины: изменение количества товаров в корзине, удаление товаров из корзины;

- Тестирование формы указания личных данных пользователя при оформлении заказа. Важно проверить валидацию всех полей (ограничение по количеству символов в поле email + соответствие формату почтового адреса, невозможность введения букв в поля индекса и мобильного телефона);

- *(в случае, если все же будет предусмотрен личный кабинет и авторизация пользователя) Проверка форм регистрации, авторизации и восстановления пароля.*

2. Для тестирования функционала «Остаток товара на складе» и тестирования хранения личных данных на бэкенде используется метод серого ящика с использованием Postman. Тестирование через API позволяет проверить корректность отображения остатка товара на складе после выполнения очередного заказа.

3. Тестирование безопасности. Если в интернет-магазине предусмотрены промокоды на скидку, важно проверить, что нет возможности внедрения инъекций и оформления «ложной» скидки. Также необходимо

проверить защищенность личных данных пользователей, возможно ли их просматривать, *будучи не авторизованным. Соответствие реальных состояний после авторизации или выхода из личного кабинета ожидаемым.*

4. Интеграционное тестирование. Проверка совместимости нашего приложения с сервисом для оплаты (обеспечивается ли корректный переход на страницу этого сервиса, правильная ли сумма для оплаты там указывается).

5. UI тестирование. Задача – поиск несоответствий между альфа-версией и макетом сайта, представленным графическими дизайнерами, например, в Figma («найти n отличий»). Может быть несоответствие цветов кнопок, расположения карточек товаров на странице, очередность полей в форме оформления заказа и тд. В этом разделе проводится также тестирование глобализации (тестирование приложения для различных местоположений, форматов дат и чисел).

6. Кросс-платформенное тестирование. Проводится для проверки совместимости приложения с другими браузерами, ОС (Windows, Linux, Mac), на мобильных устройствах (предпочтительно, большая часть интернет-заказов в настоящее время делается через телефон).

Данный этап является самым продолжительным, поскольку подразумевает тщательную проверку работы основного функционала приложения. Важно выявить ошибки, если такие имеются, и отправить приложение на доработку. На тестирование имеет смысл закладывать две недели, при условии, что в команде имеется как минимум два QA инженера (работающие, например, в разных ОС или разных браузерах). Баг-репорты оформляются в баг-трекере (например, Jira или YouGile) и отправляются разработчикам.

Основные составляющие баг-репорта:

1. Заголовок;
2. Описание бага;
3. Окружение, при котором возникает баг;
4. Шаги воспроизведения;

5. Ожидаемый результат (переносится из тест-кейса);
6. Фактический результат;
7. *Скриншот (при необходимости).*

Багам в баг-трекере присваивается степень серьезности и приоритет (от критичных до незначительных).

Тестовые сценарии дополняются графой «фактические временные затраты», в которую вносится информация о фактически затраченном времени на выполнение каждого тест-кейса. Данные суммируются и сравниваются с оценочными временными затратами.

### **3 Исправление ошибок в альфа-версии**

После проведения первого тестирования, приложение отправляется на доработку. Процесс может занять от нескольких дней, до нескольких недель (в зависимости от количества недостатков и багов, и их трудности в техническом исправлении). На данном этапе QA инженер уточняет сценарии для последующего регрессионного тестирования.

### **4 Регрессионное тестирование**

Когда разработчик заканчивает работу над исправлением багов, обнаруженных в ходе предыдущего тестирования, проводится повторное тестирование (регресс-тестирование). В первую очередь проводятся тест-кейсы, по результатам которых были обнаружены баги и результат выполнения которых не совпал с ожидаемым. Повторное тестирование по успешным тестовым сценариям может как проводиться, так и не проводиться, это зависит от того, были ли обнаружены баги в определенном разделе приложения, или нет. Если, например, функционал добавления товаров в корзину работал корректно, а все баги были обнаружены в разделе с оформлением заказа, проведение тест-кейсов с карточками товаров можно опустить. Новые баг-репорты (в случае обнаружения багов) оформляются в баг-трекере и направляются команде разработки.

Данный этап занимает от недели до двух. Если обнаружены новые баги, приложение направляется на повторную доработку. После нее также

проводится регрессионное тестирование по тем сценариям, где не совпали фактический и ожидаемый результаты выполнения.

Параллельно командой QA инженеров разрабатываются сценарии для приемочного тестирования.

## **5 Приемочное тестирование, выпуск приложения и поддержка после релиза**

Приемочные тесты пишутся для заказчика, в них включены все функциональные требования, отраженные в аналитике. Цель – показать заказчику, что весь функционал сайта работает. Оформление тестовых сценариев аналогично тому, которое используется для тестовых сценариев при тестировании альфа-версии, дополнительно добавляется лишь графа «компонент», в которой уточняется, в каком разделе веб приложения реализована данная функция. Если на данном этапе окажется, что в приложении все еще имеются ошибки, его следует отправить обратно на доработку. Это последняя стадия перед представлением проекта приложения заказчику и его релизом. Считаем, что она должна быть завершена за две недели до релиза. Оставшееся время отводится на тестирование приложения фокус-группой (примерно 10-15 человек), которые оставляют свои отзывы о работе приложения, удобно ли оно для использования.

Когда релиз приложения произведен, выделяется один месяц на поддержку. За это время собирается информация от первых пользователей (анализ данных от фокус-группы, либо мониторинг обратной связи от пользователей (отзывы, жалобы)) и в случае выявления ошибок в работе приложения составляются отчеты о багах, которые направляются разработчикам. Ошибки исправляются, проводится регресс, по итогам которого приложение обновляется до новой версии с исправлением выявленных ошибок.

Периодизация процесса разработки и тестирования веб приложения онлайн-магазина с привлечением QA инженера представлена в таблице 1.

Таблица 1 – Периодизация процесса разработки и тестирования веб приложения

№ недели	Статус работы над приложением	Задачи QA инженера	Привлекаемые ресурсы
1	Разработка альфа-версии	Тестирование спецификации; Составление ДСП	Команда QA инженеров
2		Написание тестовых сценариев	Команда QA инженеров
3	Тестирование альфа-версии	Функциональное тестирование; Тестирование безопасности; Интеграционное тестирование;	Команда QA инженеров
4		UI тестирование; Кросс-платформенное тестирование	Команда QA инженеров
5	Доработка альфа-версии	Написание сценариев для регресса	Команда QA инженеров
6			Команда QA инженеров
7	Регресс-тестирование	Регрессионное тестирование	Команда QA инженеров
8		Регрессионное тестирование; Написание сценариев для приемочного тестирования	Команда QA инженеров
9	Исправление ошибок после регресса	Написание сценариев для приемочного тестирования	Команда QA инженеров
10	Приемочное тестирование	Приемочное тестирование	Команда QA инженеров
11	Подготовка к релизу	Тестирование приложения с привлечением фокус-группы; Финальное согласование с заказчиком	Команда QA инженеров; Фокус-группа
12			Команда QA инженеров; Фокус-группа
13	Пост-релиз	Поддержка приложения	Команда QA инженеров; Фокус-группа
14			Команда QA инженеров; Фокус-группа
15			Команда QA инженеров
16			Команда QA инженеров