План тестирования

Основная информация:

Заказчик ООО «Ростелеком»

Создан (автор) Пятницкий М.А.

Стенд https://b2c.passport.rt.ru

Дата 16.02.2023

Версия 1.001

Статус В стадии согласования

Оглавление

1. Введение

1.1. Основная информация

1.2. Цель

2. Область тестирования проекта

2.1. Область тестирования веб-сайта

3. План работы

4. Тест-план и стратегия тестирования

4.1. Функциональное тестирование

4.2. Процедура тестирования

4.3. Отчеты об ошибках

5. Приложение

5.1. Инструменты

5.2. Список браузеров

6. Критерии качества

7. Срок начала и окончания тестирования

8. Результаты

**1. Введение**

**1.1. Основная информация**

Документ описывает методы и подходы к тестированию, которые будут использоваться тестировщиками отдела тестирования, для тестирования сайта. План тестирования может использоваться как тестировщиками, так и менеджерами, разработчиками. Объект тестирования — это деятельность, направленная на проверку работоспособности функций сайта в браузерах Chrome, Firefox, Internet Explorer 11, Opera.

**1.2. Цель**

Тест-план проекта преследует следующие цели:

● Определить существующую информацию о проекте и программных компонентах, подлежащих тестированию.

● Описать стратегии тестирования, которые будут использоваться.

● Определить необходимые ресурсы для проведения работ по тестированию.

● Привести результаты тестирования. Результаты будут отправлены заказчику в виде отчетов. Все найденные ошибки будут отслеживаться с помощью баг-трекинговой системы.

**2. Рамки проекта**

**2.1. Область тестирования сайта**

В объем работ по тестированию сайта входит тестирование следующих компонентов и функций:

1. Регистрация.

2. Авторизация.

3. Восстановление логина и пароля.

**3. План работы**

Стороны договорились следовать следующему плану работы:

1. Подготовка плана тестирования.

2. Согласование тест-плана.

3. Функциональное тестирование и отчеты об ошибках.

4. Подготовка ежедневных отчетов.

5. Подготовка финального отчета.

**4. Тест-план и стратегия тестирования**

**4.1. Функциональное тестирование**

Цель функционального тестирования состоит в том, чтобы убедиться, что весь программный продукт работает в соответствии с требованиями, и в приложении не появляется существенных ошибок. Функциональное тестирование является наиболее существенной частью тестирования программного обеспечения, включающее в себя проверку различных аспектов системы. Программный продукт должен пройти все запланированные тесты. Только в этом случае можно быть уверенным в его качестве.

Цель тестирования:

Обеспечение надлежащего качества целевой функциональности

Выполните каждый сценарий, используя допустимые и недопустимые данные:

● Ожидаемые результаты возникают при использовании достоверных данных.

● Соответствующие сообщения об ошибках или предупреждения отображаются, когда используются неверные данные.

● Каждый пункт проверен.

Входные критерии:

● Подготовлено тестовое окружение, приложение готово к тестированию на тестовой площадке.

● Все дымовые тесты пройдены.

Критерий приемки:

● Нет блокирующих багов.

● Все баги с высоким приоритетом поправлены.

● Тесты пройдены.

**4.2. Процедура тестирования**

Процедура тестирования предполагает следующие пункты:

● Сообщения об обнаруженных ошибках программного обеспечения.

Необходимо проверить различные аспекты тестируемого программного обеспечения, для этого требуется выполнение различных типов тестирования.

Основные типы тестирования, которые будут выполнены:

● Функциональное тестирование.

● Тестирование пользовательского интерфейса.

● Регрессионное тестирование.

Также будет проверено, как программный продукт запускается в браузерах, которые должны его поддерживать.

**4.3. Отчеты об ошибках**

Отчеты об ошибках создаются для того, чтобы предоставить команде разработчиков и руководителю проекта исчерпывающую информацию об обнаруженных ошибках. Они должны быть полезны при определении причин ошибок и их исправлении.

Степень серьезности ошибок можно разделить на четыре категории:

● Критические (блокирующие) дефекты - это сбой всей программной системы или критической подсистемы, тестирование не может быть выполнено после возникновения дефекта. Это также относится к сбоям при потере данных и процессам, которые оставляют несогласованные данные, хранящиеся в базе данных.

● Мажорные дефекты — ошибки, которые также приводят к выходу из строя всей или части системы, но существуют некоторые альтернативы обработки, которые позволяют продолжить работу системы. Это также относится к сбою или прерыванию работы системы во время нормальной работы некритического потока.

● Незначительные дефекты не приводят к сбою, но приводят к тому, что система показывает неправильные, неполные или противоречивые результаты.

● Тривиальные дефекты - это небольшие ошибки, которые не влияют на функциональность: опечатки, грамматические ошибки, неправильная терминология и т.д.

Информация, указанная в каждом отчете об ошибке:

● Название тестируемого продукта.

● браузер, в котором проводились тесты.

Каждый баг-репорт содержит следующую информацию о дефекте:

● Краткое описание, представляющее собой краткое описание проблемы.

● Местонахождение дефекта в программном продукте.

● Шаги для воспроизведения ошибки;

● Тяжесть дефекта.

● Ожидаемый результат

● Фактический результат

Стороннее программное обеспечение будет использоваться для отчетности и поддержки обнаруженных ошибок. Команда тестирования зарегистрирует в этом программном обеспечении все дефекты, которые будут обнаружены в процессе тестирования.

**5. Ресурсы**

**5.1. Инструменты**

Будут использованы следующие инструменты:

1. Баг-трекинговая система - Jira

Jira - инструмент управления проектами и отслеживания ошибок. Также имеет доски Scrum и Kanban. Имеет ряд плюсов:

● Пользователи могут создавать собственные фильтры и настраивать рабочие процессы

● Есть удобная система тикетов, позволяющая легко следить за ходом выполнения задач

● Доступ к отчетам с полезной информацией в режиме реального времени

● Интеграция с более чем 3000 приложений обеспечивает прозрачность конвейера разработки

● Jira идеально подходит для больших и удаленных команд

● Есть мобильное приложение, позволяющее получить доступ к системе в любое время

Вышеперечисленные плюсы подтверждают выбор баг-трекинговой системы и ее эффективность в использовании.

1. Для написания тест-кейсов будет использована Zephyr for Jira, которая имеет интеграцию с баг-трекинговой системой Jira, что позволяет отслеживать весь процесс в комплексе.
2. Для захвата изображения или видео рекомендуется использовать Share X, программа проста в использовании и имеет большой функционал.

**5.2. Список браузеров**

● Chrome (последняя официальная сборка)

● Firefox (последняя официальная сборка)

● IE11 (последняя официальная сборка)

● Opera (последняя официальная сборка)

**6. Критерии качества**

Продукт должен работать в соответствии с требованиями. Продукт не должен содержать критических и блокирующих дефектов в окончательной версии проекта.

**7. Срок начала и окончания тестирования**

Начало: 20 февраля 2023 года

Окончание: 16 марта 2023 года

**8. Результаты**

1. Создание тест-плана

2. Создание тест-кейсов по требованию

3. Создание чек-листа

4. Описание обнаруженных багов в баг-трекинговой системе.

5. Отчет о количестве выполненных проверок и соотношение успешных и работающих с ошибкой функций.