**ТЕСТ ПЛАН**

ОГЛАВЛЕНИЕ

1. ВВЕДЕНИЕ
   1. Цели тестирования
   2. Подготовка к тестированию
2. ОПИСАНИЕ ТЕСТИРУЕМОЙ СИСТЕМЫ (объём работы)
3. ТЕСТОВАЯ СТРАТЕГИЯ
   1. Критерии начала и завершения тестирования
   2. Уровни тестирования
   3. Виды тестирования
   4. Инструменты и тестовая среда
4. ЦИКЛ ТЕСТИРОВАНИЯ
   1. Ежедневная активность
   2. Срочная активность
   3. Тестирование релиза
   4. Отчетность
   5. Ответственные лица

Приложение А Рабочий процесс

Приложение Б Шаблон тест-кейса

1. **Введение.**

Целью составления данного Тест Плана является описание процесса тестирования сайта shop.bugred.ru (полный адрес [http://shop.bugred.ru/](https://www.google.com/url?q=https://www.google.com/url?q%3Dhttp://shop.bugred.ru/%26amp;sa%3DD%26amp;source%3Deditors%26amp;ust%3D1628519035508000%26amp;usg%3DAOvVaw3pcYSbkfvNrrHfVjF5rDvZ&sa=D&source=editors&ust=1628519035547000&usg=AOvVaw0L2cGOTrmkuspRvINXq8pF)). Документ позволяет получить представление о плановых работах по тестированию проекта.

Настоящий план тестирования определяет и описывает стратегию и принципы, применимые для тестирования на проекте shop.bugred.ru. Документ определяет распределение обязанностей при тестировании и описывает тесты, намеченные к выполнению.

Тестовый план разработан для решения следующих задач:

- собрать и систематизировать необходимую информацию о тестируемом

проекте;

- определить требования к тестированию;

- определить последовательность действий при проведении тестирования.

Тестовым планом будут пользоваться исполнители работ для получения представления о тестировании на проекте.

* 1. **Цели тестирования**

Основными целями тестирования на проекте shop.bugred.ru являются:

- обеспечение надлежащего уровня качества, заявленного к разрабатываемому

продукту;

- повышение доверия заказчика к разрабатываемому продукту и команде;

- обеспечение работоспособности продукта;

- обеспечение соответствия разрабатываемого продукта заявленным

требованиям;

- своевременное предоставление информации о качестве продукта и стадии

разработки заинтересованным лицам;

- предоставление информации о статусе готовности заявленной

функциональности и/или продукта в целом;

- обеспечение пользователя наиболее удобным программным средством для

решения поставленных задач.

* 1. **Подготовка к тестированию**

Для получения наиболее полного представления о проекте, тестировщику необходимо ознакомиться со всей имеющейся документацией, составить список возникших вопросов и довести его менеджеру проекта. Все вопросы должны быть уточнены и устранены.

На проекте shop.bugred.ru имеется достаточно полная спецификация требований, а так же карта проекта. Вся имеющаяся документация по проекту shop.bugred.ru находится в wiki: [https://testbase.atlassian.net/wiki/spaces/SHOP/overview?homepageId=1411056054](https://www.google.com/url?q=https://www.google.com/url?q%3Dhttps://testbase.atlassian.net/wiki/spaces/SHOP/overview?homepageId%253D1411056054%26amp;sa%3DD%26amp;source%3Deditors%26amp;ust%3D1628516436554000%26amp;usg%3DAOvVaw3C0NKELH4XKdiCU7mWaZLF&sa=D&source=editors&ust=1628516436579000&usg=AOvVaw3H0DjA9Xs8pSZz7xoRp1ON)

Так же перед началом работы на проекте необходимо ознакомиться с особенностями трекинга рабочего времени, принятого на проекте:

- тестировщики, о которых заказчик знает, трекают рабочее время в Jira.

В день должно выходить 8 часов. Переработки/отгулы/отпуска должны быть согласованы с менеджером проекта. Активны могут быть не более двух задач. Подробные схемы флоу размещены в приложении А данного тест-плана.

На проекте приняты следующие принципы построения дерева тест-кейсов:

- разделение проверяемого приложения на модули;

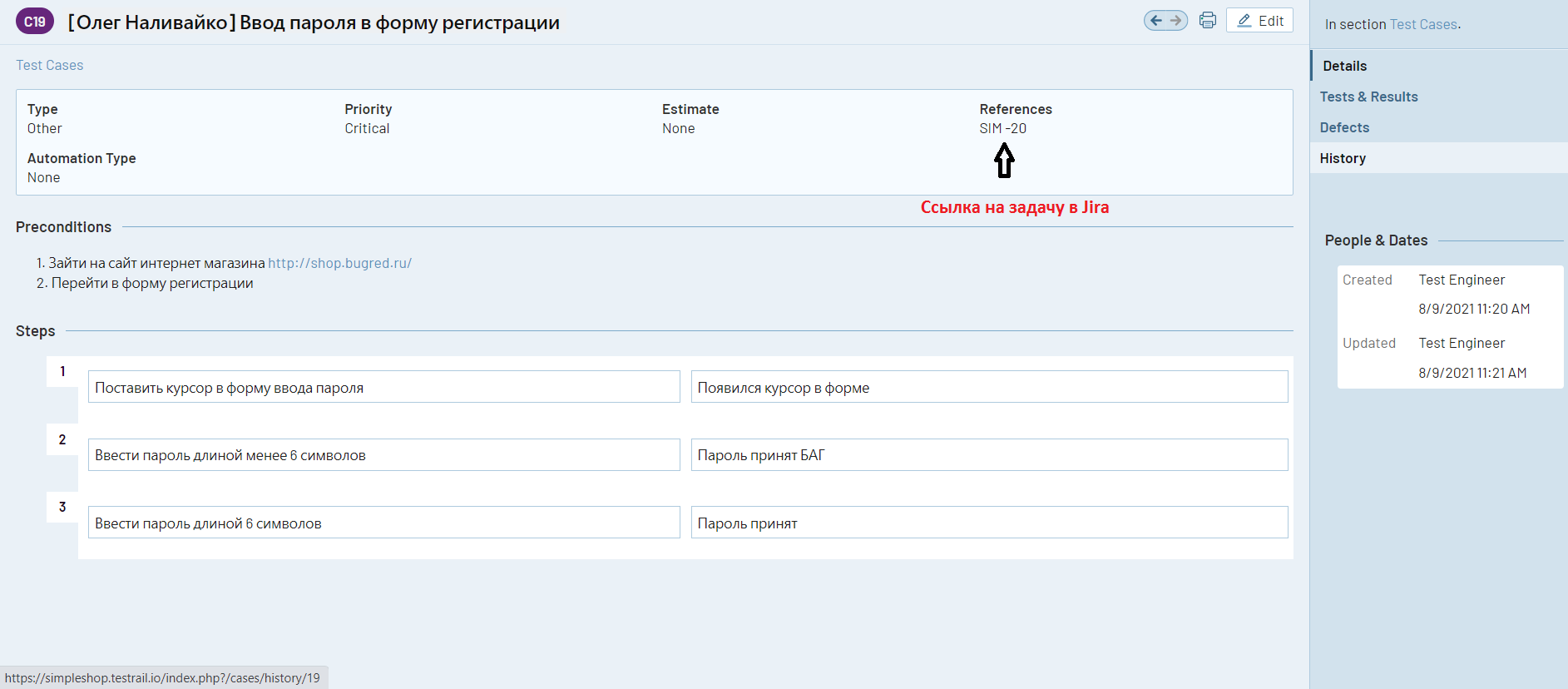
- создание отдельных папок для каждого модуля;

- проверки записываются отдельными тест-кейсами по каждому модулю.

Тестировщики тест-кейсы пишут в TestRail.

Тест-раны составляются по функциональностям проверяемого приложения с выделением отдельных ранов (сьютов) по видам тестирования (смоук, регресс).

Трекинг времени на создание и рефакторинг тест-кейсов производится непосредственно в задачу, к которой относятся тест-кейсы. При этом, тестировщики, работающие в Jira, линкуют тест-кейсы с задачей.



С шаблоном тест-кейса можно ознакомиться в приложении В.

1. **Описание тестируемой системы**

Проект shop.bugred.ru представляет собой интернет магазин по продаже одежды. Сайт позволяет пользователю узнать больше о товарах, просматривать их, делать заказы, узнавать новости о скидках, пользоваться обратной связью.

Компоненты и функции, которые должны быть протестированы:

- авторизация пользователя;

- создание и корректирование заказа;

- формирование и редактирование данных заказа;

- выгрузка данных одобренного заказа;

- формирование и отправка email-уведомления для пользователей;

- пользовательские интерфейсы;

- базы данных.

**3. Тестовая стратегия**

**3.1 Критерии начала и завершения тестирования**

Разработка ведется по методологии Scrum с итерациями 2 недели. В течение итерации команда разработчиков ежедневно выпускает для команды тестирования новую сборку. После этого команда тестирования запускает тестовые ситуации в тестовой среде.

Тестирование следует начинать, когда:

- готов тестовый стенд;

- закончена разработка требуемого функционала и/или компонента системы;

- имеется необходимая техническая документация

- все тестировщики ознакомлены с целями проекта и архитектурой ПО.

Тестирование можно считать завершенным в случае, если:

- критические функции покрыты тестами на 100%;

- покрытие всех функций стремится к 80%;

- закрыты все критические дефекты;

- успешно пройдены все регрессионные тесты;

- количество незакрытых дефектов не превышает 15%.

**3.2 Уровни тестирования**

На проекте shop.bugred.ru определены уровни тестирования:

- Интеграционное тестирование - после разработки тестов на отдельные классы необходимо проверить, как они будут работать вместе в рамках одного исполняемого процесса.

- Системное тестирование - проверка как функциональных, так и не функциональных требований в системе в целом. При этом выявляются дефекты, такие как неверное использование ресурсов системы, непредусмотренные комбинации данных пользовательского уровня, несовместимость с окружением, непредусмотренные сценарии использования, отсутствующая или неверная функциональность, неудобство использования и т.д. На проекте shop.bugred.ru применяется системное тестирование на базе требований.

- Приемочное тестирование - финальный этап тестирования продукта перед его релизом. Целью приемочного тестирования является определение готовности продукта, что достигается путем прохода тестовых сценариев и случаев, которые построены на основе спецификации требований к разрабатываемому ПО.

**3.3 Виды тестирования**

Для достижения целей и решения задач тестирования разрабатываемого ПО на проекте shop.bugred.ru используются следующие виды тестирования:

- Тестирование требований - сравнение разработанных для проекта требований со свойствами качественных требований, с целью выявления в них недочетов, противоречий и логических ошибок, способных повлечь возникновение серьезных дефектов в процессе разработки.

-Тестирование пользовательского интерфейса - проверка удобства использования интерфейсов разрабатываемого ПО на возможность выполнения с его помощью пользовательских сценариев. Тестирование должно проверять корректность работы интерфейсной части приложения при любых возможных настройках экрана (различное разрешение, масштаб, шрифт), при изменениях фокуса, при работе с мышью и клавиатурой.

- Тестирование удобства использования интерфейса - проверка удобства использования, быстроте освоения, наглядности пользовательских интерфейсов системы.

- Кроссбраузерное тестирование - проверка верстки и функциональности, корректности расположения элементов на странице, правильность отображения шрифтов, таблиц и цвета, и др.

- Функциональное тестирование - проверка соответствия всех функций разрабатываемого ПО функциональным требованиям, заявленным заказчиком. Проверяем, корректно ли работают функции разрабатываемого ПО при различных входных данных, их комбинации и количества, а также корректно ли выполняются операции с точки зрения сценариев использования.

- Дымовое тестирование - простейший вид тестирования, основанный на определении успешности сборки системы из ветви исходного кода, находящейся в разработке. Обычно проводится один раз в день.

- Верификация - проверка успешности исправления разработчиком ошибки, проведенная тестировщиком в тестовом окружении.

- Тестирование релиза - тестирование, проводимое как последняя стадия разработки.

**3.4 Инструменты и тестовая среда**

Тестовое окружение:

- браузер Google Chrome версии 92.0.4515.131

- браузер Opera версии 77.0.4054.277

1366x768, 1920x1080, 1280х1024

- тестовый стенд:

https://shop.bugred.ru.project-qa/login

- продуктовый стенд:

https:/shop.bugred.ru.project-prod/login

Для занесения и отслеживания задач, багов, тест-кейсов и баг-репортов используется система управления версиями Jira:

https://belhard-qa.atlassian.net/jira/software/c/projects/

Для занесения и отслеживания тест кейсов используется система TestRail:

(https://simpleshop.testrail.io/)

**4. Цикл тестирования**

В данной главе описан процесс тестирования, который состоит из следующих видов активности в порядке приоритета: срочная внеплановая активность, тестирование релиза, ежедневная плановая активность, разработка новых тестов, отчетная активность.

**4.1 Ежедневная активность**

Ежедневная активность тестирования заключается в проведении запланированных тестовых активностей на тестовом стенде. В результате формируется отчет, который рассылается всем заинтересованным лицам. В данный вид активности также включается проведение верификации некритичных ошибок и доработок. Тестировщик контролирует ход тестирования и получает ежедневный отчет о ходе тестирования. Он управляет включением тестов, настраивает тестовый стенд, конфигурирует необходимые окружения для проведения тестов. Тестировщик постоянно анализирует спецификацию требований к системе, технические задания и регулярно выдает рекомендации по улучшению тестирования, а также сообщает разработчикам о расхождениях функциональности описанной в задании и реальных характеристик системы, если таковые появляются. Тестировщик осуществляет также дополнительные ручные тесты. В процессе усовершенствования системы некоторые контрольные примеры могут перестать выполняться успешно, например, при изменении пользовательского интерфейса. Поэтому ежедневной задачей тестирования является анализ причин неудач, обновление и исправление контрольных примеров. В процессе тестирования тестировщик открывает новые инциденты по результатам тестирования, назначая их разработчикам.

**4.2 Срочная активность**

Заключается в выполнении тестировщиком срочных поручений менеджера проекта, большая часть из которых является критическими и блокирует ход разработки. В данный вид активности также включается проведение верификации критических ошибок и доработок. Тестировщику следует внимательно подходить к верификации, т.к. внесение изменений оказывает влияние на другие участки кода и может вносить дополнительные ошибки. Необходимо тщательно проверять предполагаемые зависимые участки кода таким образом, чтобы верификация приближалась к регрессионному тестированию по каждому инциденту.

**4.3 Тестирование релиза**

Данная процедура носит временный характер, стартуя в момент начала финального тестирования и заканчиваясь после принятия релиза. Обычно целью тестирования релиза является сдача продукта с определенными характеристиками к определенному сроку. Обязательным атрибутом данного вида тестирования является проверка обеспечения функциональных характеристик продукта в соответствии со спецификацией требований. Для этого проводится тщательный анализ спецификаций требований к системе в целом и к каждому компоненту. Важным является определение отличий текущего релиза от предыдущего и проведение по этим отличиям регрессионного тестирования. В процессе тестирования релиза тестировщик удостоверяется в успешном прохождение всех тест кейсов в системе.

**4.4 Отчетность**

Основными видами отчетов, которые должен составлять тестировщик, являются отчеты о дефектах и отчет о тестировании.

Отчет о дефекте (баг-репорт) составляется при обнаружении дефекта. Составление баг-репортов преследует следующие цели:

- предоставить информацию о проблеме - уведомить проектную команду и иных

заинтересованных лиц о наличии проблемы, описать суть проблемы;

- приоритизировать проблему - определить степень опасности проблемы для

проекта и желаемые сроки её устранения;

- содействовать устранению проблемы.

Отчет о тестировании на проекте [shop.bugred.ru](http://shop.bugred.ru) составляется еженедельно и предоставляется проектному менеджеру по пятницам. Отчет о тестировании обязательно должен содержать:

- информацию о тестовом окружении;

- сведения о написанных за неделю тест-кейсах, созданных тест-сьютах;

- количество обнаруженных за неделю дефектов;

- статистику по дефектам, включая их статусы и приоритет;

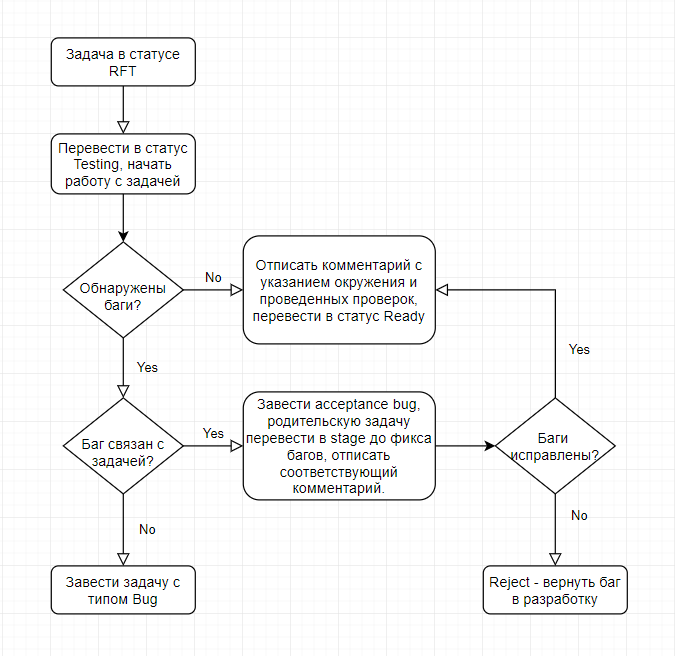
- актуальное состояние тестируемой системы.

Отчеты о тестировании загружаются в wiki проекта.

**4.5 Ответственные лица**

- Олег Наливайко

**Приложение А Рабочий процесс**

****

**Приложение Б Шаблон тест-кейса**

|  |  |
| --- | --- |
| Field | Description |
| Title | **Название тест-кейса**.   * Должно четко описывать, что проверяется в тесте. * Не должно быть слишком длинным. |
| Type | **Тип теста (F, R).**  **F** - функциональный тест, который будет прогнан один раз.  **R** - тест попадет в регрессию.  Перед включением теста в регрессию, ответьте на следующие вопросы:   * Тест проверяет основной и самый важный функционал фичи? * Если прогнать регрессионные тесты по фиче - можно сказать, что фича работает? |
| Priority | **Приоритет**.  **High** (Очень важный тест, обязательно должен быть прогнан. Если не будет работать - пользователь будет зол).  **Medium** (Важный тест, должен быть прогнан, если не будет работать - пользователь будет грустить).  **Low** (Неважный тест, хотелось бы прогнать, но если мало времени и не успеть - ничего страшного. Если не будет работать - пользователь возможно не заметит. Либо пользователь очень редко использует эту функциональность. Либо на работе пользователя скажется не сильно).  Обратить особое внимание на приоритеты тестов для регрессии: **Регрессионные тесты не могут быть приоритета Low!**  **High** - критически важная функциональность для пользователя.  **Medium** - не критически важная функциональность, но этот тест проверяет работоспособность фичи. |
| Precondition | Предусловия и шаги, предшествующие непосредственно тестируемому действию. |
| Steps | **Тестируемые шаги**.   * Т.к. один тест - одна проверка → в идеале должен быть один шаг. * Шаг должен непосредственно описывать, ЧТО мы проверяем. |
| Expected result | **Ожидаемый результат**.   * Может состоять из нескольких пунктов. * НЕ должен содержать дополнительных действий! |
| References | Номер задачи в Jira. |
| Attachments | В зависимости от сложности кейса |