

Test Plan.

Project Name: "Final project of the manual testing course"

WebSite: <https://qasvus.wixsite.com/ca-marketing/>

Prepared By: Artem Markov

Prepared By Date: 9.11.2021

TEST PLAN (Тест План):

Table of Contents (Оглавление):

1. Introduction (Введение)
2. Scope (Сфера применения)
3. Quality Objectives (Цели качества)
4. Test Approach (Подход к тестированию)
5. Entry And Exit Criteria (Критерии входа и выхода)
6. Suspension Criteria And Resumption Requirements (Критерии приостановки и требования к возобновлению)
7. Test Strategy(Стратегия тестирования)
8. Resource And Environment Needs (Потребности в ресурсах и окружающей среде)

1. INTRODUCTION (ВВЕДЕНИЕ)

The Test Plan has been created to facilitate communication within the team members. This document describes approaches and methodologies that will apply to the unit, integration and system testing of the <https://qasvus.wixsite.com/ca-marketing>. It includes the objectives, test responsibilities, entry and exit criteria, scope, schedule major milestones, entry and exit criteria and approach. This document has clearly identified what the test deliverables will be, and what is deemed in and out of scope.

(План тестирования был создан, чтобы облегчить общение между членами команды. В этом документе описываются подходы и методологии, которые будут применяться к модульному, интеграционному и системному тестированию <https://qasvus.wixsite.com/ca-marketing>. Он включает цели, обязанности по тестированию, критерии входа и выхода, объем, основные этапы. В этом документе четко определены результаты тестирования, а также то, что входит и выходит за рамки.)

2. SCOPE (ОБЪЕМ)

The document mainly targets:

- website manual test
- website automation test
- website API test

- website Performance Automation test with Lighthouse, GTMetrix and SpeedLab tools.

, and validating data in report output as per Requirements Specifications provided by Client.

(Документ в основном нацелен на ручное, автоматическое, API тестирования и тестирование производительности, а также проверку данных в выводе отчета в соответствии со спецификациями требований, предоставленными клиентом.)

3. QUALITY OBJECTIVES (ЦЕЛИ КАЧЕСТВА)

A primary objective of testing is to: assure that the system meets the full requirements, including quality requirements (functional and non-functional requirements) and fit metrics for each quality requirement and satisfies the use case scenarios and maintain the quality of the product. (Основная цель тестирования состоит в том, чтобы: гарантировать, что система соответствует всем требованиям, включая требования к качеству (функциональные и нефункциональные требования) и соответствующие метрики для каждого требования к качеству, а также удовлетворяет сценариям вариантов использования и поддерживает качество продукта.)

The secondary objectives of testing will be to identify and expose all issues and associated risks, communicate all known issues to the project team, and ensure that all issues are addressed in an appropriate matter before release.

(Вторичные цели тестирования будут заключаться в выявлении всех проблем и связанных с ними рисков, и сообщении обо всех известных проблемах команде проекта и обеспечении того, чтобы все проблемы были решены в соответствующем вопросе перед выпуском.)

4. TEST APPROACH (ТЕСТОВЫЙ ПОДХОД)

Analytical test approach was used, in accordance to requirements-based strategy, where an analysis of the requirements' specification form is the basis for planning, estimating and designing tests. Test cases will be created during exploratory testing. All project is using an Agile approach, with weekly iterations. At the end of each week the requirements identified for that iteration will be delivered to the team and will be tested.

(В соответствии со стратегией, основанной на требованиях, использовался аналитический подход к тестированию, при котором анализ формы спецификации требований является основой для планирования, оценки и разработки тестов. Тестовые случаи будут созданы во время исследовательского тестирования. Во всех проектах используется Agile-подход с еженедельными итерациями. В конце каждой недели требования, определенные для этой итерации, будут доставлены команде и проверены.)

5. ENTRY AND EXIT CRITERIA (Критерии входа и выхода)

Entry Criteria

- Proper test data should be available.
- All the necessary documentation, design, and requirements information should be available that will allow testers to operate the system and judge the correct behavior.
- All the standard software tools including the testing tools must have been successfully installed and functioning properly.
- All test hardware platforms must have been successfully installed, configured, and functioning properly.

(Критерии входа)

- Должны быть доступны надлежащие данные испытаний.
- Должна быть доступна вся необходимая документация, дизайн и информация о требованиях, что позволит QA работать с системой и оценивать ее правильное поведение.
- Все стандартные программные инструменты, включая инструменты тестирования, должны быть успешно установлены и функционируют нормально.
- Все тестовые аппаратные платформы должны быть успешно установлены, настроены и правильно работать.)

Exit Criteria

- No high priority or severe bugs are left outstanding.
- A certain level of requirements coverage has been achieved.
- All high-risk areas have been fully tested, with only minor residual risks left outstanding.
- The schedule has been achieved.

(Критерии выхода)

- Не остаются невыявленные высокоприоритетные или серьезные ошибки.
- Достигнут определенный уровень покрытия требований.
- Все зоны высокого риска были полностью протестированы, оставлены лишь незначительные остаточные риски.
- График был выполнен.)

6. SUSPENSION CRITERIA AND RESUMPTION REQUIREMENTS (КРИТЕРИИ ПРИОСТАНОВКИ И ТРЕБОВАНИЯ К ВОЗОБНОВЛЕНИЮ)

Suspension criteria (Критерии отстранения)

- Software/Hardware problems. (Программные / аппаратные проблемы.)
- Significant change in requirements suggested by client. (Существенное изменение требований, предложенных клиентом.)
- The build contains many serious defects which seriously or limit testing progress. (Сборка содержит много серьезных дефектов, которые серьезно ограничивают прогресс тестирования.)
- Assigned resources are not available when needed by test team. (Назначенные ресурсы недоступны для команды тестирования.)

Resumption criteria (Критерии возобновления)

- Resumption will only occur when the problem that caused the suspension has been resolved. (Возобновление произойдет только тогда, когда проблема, вызвавшая приостановку, будет решена.)

7. TEST STRATEGY (Стратегия тестирования)

<p>QA role in test process:</p> <ul style="list-style-type: none"> • Understanding Requirements. • Requirement specifications will be sent by client. <p>(Роль QA в процессе тестирования:</p> <ul style="list-style-type: none"> • Понимание требований. • Спецификации требований будут отправлены клиентом.) <p>Preparing Test Cases:</p> <ul style="list-style-type: none"> • QA will be preparing test cases based on the exploratory testing. This will cover all scenarios for requirements. <p>(Подготовка тестовых случаев:</p> <ul style="list-style-type: none"> • QA будет готовить тестовые примеры на основе исследовательского тестирования. Это покроет все сценарии требований.) <p>Preparing Test Matrix:</p> <ul style="list-style-type: none"> • QA will be preparing test matrix which maps test cases to respective requirement. This will ensure the coverage for requirements. <p>(Подготовка тестовой матрицы:</p> <ul style="list-style-type: none"> • QA подготовит тестовую матрицу, которая сопоставляет тестовые случаи с их соответствующими требованиями. Это обеспечит соответствие требованиям.) <p>Reviewing test cases and matrix:</p> <ul style="list-style-type: none"> • Review for test cases and test matrix will be conducted by QA Lead. • Any comments or suggestions on test cases and test coverage will be provided by reviewer • Suggestions or improvements will be updated by preparer and sent to QA Lead for approval. • Updates and improvements will be reviewed and approved by reviewer. <p>(Обзор тестовых случаев и матрицы:</p> <ul style="list-style-type: none"> • Проверка тестовых случаев и тестовой матрицы будет проводиться руководителем отдела обеспечения качества. • Любые комментарии или предложения по тестовым случаям и тестовым покрытиям будут предоставлены рецензентом. • Предложения или улучшения будут обновлены составителем и отправлены на утверждение руководителю отдела контроля качества. • Обновления и улучшения будут рассмотрены и утверждены рецензентом.) <p>Creating Test Data:</p> <ul style="list-style-type: none"> • Test data will be created by respective QA based on scenarios and Test cases. <p>(Создание тестовых данных:</p> <ul style="list-style-type: none"> • Тестовые данные будут созданы соответствующим QA на основе сценариев и тестовых случаев.) <p>Executing Test Cases:</p> <ul style="list-style-type: none"> • Test cases will be executed by respective QA based on designed scenarios, test cases and Test data. • Test result (Actual Result, Pass/Fail) will be updated in test case document Defect Logging and Reporting: QA will be logging the defect/bugs in Excel spreadsheet and JIRA, found during execution of test cases. <p>(Выполнение тестовых случаев:</p> <ul style="list-style-type: none"> • Тестовые примеры будут выполняться соответствующим QA на основе разработанных сценариев, тестовых примеров и тестовых данных. • Результат теста (фактический результат, пройден / не пройден) будет обновлен в документе тестового примера. Регистрация дефектов и создание отчетов: QA будет регистрировать дефекты / ошибки в электронной таблице Excel и JIRA, обнаруженные во время выполнения тестовых примеров.) <p>Retesting and Regression Testing:</p> <ul style="list-style-type: none"> • Retesting for fixed bugs will be done by respective QA once issue is resolved by respective developer and bug/defect status will be updated accordingly. In certain cases, regression testing will be done if required. <p>(Повторное и регрессионное тестирование:</p> <ul style="list-style-type: none"> • Повторное тестирование на исправленные ошибки будет выполнено соответствующим QA, как только проблема будет решена соответствующим разработчиком, и статус ошибки / дефекта будет соответствующим образом обновлен. В некоторых случаях при необходимости будет проведено регрессионное тестирование.) <p>Deployment/Delivery:</p> <ul style="list-style-type: none"> • Once all bugs/defect reported after complete testing is fixed and no other bugs are found,report will be deployed to the client, along with sample output by email to respective lead and Report group <p>(Развертывание / доставка:</p> <ul style="list-style-type: none"> • После того, как все ошибки / дефекты, сообщенные после полного тестирования, будут исправлены и никаких других ошибок не будет обнаружено, отчет будет отправлен клиенту вместе с образцом вывода по электронной почте соответствующему руководству и группе отчетов ») 			
Testing tasks (Задачи тестирования):			
1. Website manual test (Ручное тестирование вебсайта) https://amarkov.atlassian.net/jira/software/projects/AM/boards/1?selectedIssue=AM-5			
2. Website automation test (Автоматическое тестирование вебсайта) https://amarkov.atlassian.net/jira/software/projects/AM/boards/1?selectedIssue=AM-6			
3. Website API test https://amarkov.atlassian.net/jira/software/projects/AM/boards/1?selectedIssue=AM-7			
4. Website Performance Automation test with Lighthouse, GTMetrix and SpeedLab tools. https://amarkov.atlassian.net/jira/software/projects/AM/boards/1?selectedIssue=AM-8			
Testing subtasks (Подзадачи тестирования):			
Subtasks for task 1:			
1. Exploratory testing https://amarkov.atlassian.net/jira/software/projects/AM/boards/1?selectedIssue=AM-9			
2. Functional Testing https://amarkov.atlassian.net/jira/software/projects/AM/boards/1?selectedIssue=AM-10			
3. Positive testing https://amarkov.atlassian.net/jira/software/projects/AM/boards/1?selectedIssue=AM-11			
4. Negative testing https://amarkov.atlassian.net/jira/software/projects/AM/boards/1?selectedIssue=AM-13			
5. Boundary testing https://amarkov.atlassian.net/jira/software/projects/AM/boards/1?selectedIssue=AM-14			
6. Equivalence Partitioning testing https://amarkov.atlassian.net/jira/software/projects/AM/boards/1?selectedIssue=AM-15			
7. Ad-Hoc testing https://amarkov.atlassian.net/jira/software/projects/AM/boards/1?selectedIssue=AM-16			
Subtasks for task 4:			
1. Automation test with Lighthouse tool https://amarkov.atlassian.net/jira/software/projects/AM/boards/1?selectedIssue=AM-19			
2.Automation test with GTMetrix tool https://amarkov.atlassian.net/jira/software/projects/AM/boards/1?selectedIssue=AM-20			
3.Automation test with SpeedLab tool https://amarkov.atlassian.net/jira/software/projects/AM/boards/1?selectedIssue=AM-21			
Testing types (Виды тестирования):			
<p>Exploratory testing (Исследовательское тестирование)</p> <p>Exploratory testing will includes a type of software testing where Test cases are not created in advance but QA check system on the fly. QA may note down ideas about what to test before test execution. (Исследовательское тестирование будет включать в себя тип тестирования программного обеспечения, при котором тестовые случаи не создаются заранее, а система проверки качества на лету. QA может записывать идеи о том, что тестировать перед тестированием)</p>			
<p>Functional testing (Функциональное тестирование):</p> <p>Functional testing is carried out in order to find out unexpected behavior of the report. The characteristic of functional testing are to provide correctness, reliability, testability and accuracy of the report output/data. (Функциональное тестирование проводится с целью выявления неожиданного поведения для отчета. Характеристики функционального тестирования - обеспечить правильность, надежность, тестируемость и точность вывода / данных отчета.)</p>			
<p>Positive testing (Позитивное тестирование):</p> <p>Positive testing will includes the type of testing that can be performed on the system by providing the valid data as input. It checks whether an application behaves as expected with positive inputs. (Позитивное тестирование будет включать в себя тип тестирования, который можно выполнить в системе, предоставив действительные данные в качестве входных данных. Оно проверяет, работает ли приложение должным образом при положительных входных данных.)</p>			
<p>Negative testing (Негативное тестирование):</p>			

<p>Negative testing will includes is a method of testing an application or system that ensures that the plot of the application is according to the requirements and can handle the unwanted input and user behavior. Invalid data is inserted to compare the output against the given input. Negative testing is also known as failure testing or error path testing. When performing negative testing exceptions are expected. (Негативное тестирование будет включать в себя метод тестирования приложения или системы, который гарантирует, что график приложения соответствует требованиям и может обрабатывать нежелательный ввод и поведение пользователя. Вставлены недопустимые данные для сравнения вывода с заданным вводом. Отрицательное тестирование также известно как тестирование отказа или тестирование пути ошибки. При выполнении отрицательного тестирования ожидаются исключения.)</p>					
Boundary testing (Граничное тестирование):					
<p>Boundary Testing will give an opportunity to check the behavior of the product at the extreme (boundary) values of the input data. Boundary testing can also include tests that test system behavior on inputs that are out of range. (Граничное тестирование даст возможность проверить поведение продукта на крайних (граничных) значениях входных данных. Граничное тестирование также может включать тесты, проверяющие поведение системы на входных данных, выходящих за допустимый диапазон значений.)</p>					
Equivalence Partitioning testing (Эквивалентное разделение):					
<p>The idea behind testing with equivalence class splitting is to eliminate a set of inputs that make the system behave the same and give the same result when testing the program. This testing is done to reduce the number of tests tests in test fields. (Идея тестирования по методу разбиения классов эквивалентности состоит в том, чтобы исключить набор входных данных, которые заставляют систему вести себя одинаково и давать одинаковый результат при тестировании программы. Это тестирование проводится для уменьшения количества тестов в тестируемых полях)</p>					
Ad-Hoc testing (Специальное тестирование)					
<p>Ad-Hoc testing will includes an informal testing type with an aim to break the system. (Специальное тестирование будет включать неформальный тип тестирования с целью взлома системы.)</p>					
Performance Automation testing (Автоматическое тестирование производительности)					
<p>An automated website performance test will be performed using Lighthouse, GTMetrix and SpeedLab tools to test the speed, response time, reliability, resource utilization and scalability of the software under the expected workload through automation. The goal of performance testing is to eliminate software bottlenecks. (Автоматический тест производительности веб-сайта будет производиться с помощью инструментов Lighthouse, GTMetrix и SpeedLab для проверки скорости, времени отклика, надежности, использования ресурсов и масштабируемости программного обеспечения при ожидаемой рабочей нагрузке за счет автоматизации. Целью тестирования производительности является устранение узких мест в программном обеспечении.)</p>					
API testing (тестирование API (интерфейс прикладного программирования))					
<p>API testing will be conducted to eliminate errors between system modules or between systems. (Тестирование API будет проводиться для устранения ошибок взаимодействия между модулями системы или между системами.)</p>					
8. Resource And Environment Needs(Потребности в ресурсах и окружающей среде):					
Windows 10/64					
Google Chrome - 95.0.4638.69					
Microsoft Edg - 93.0.961.47					
Yandex - 21.8.2.381					
Opera - 78.0.4093.147					
Testing Tools (Инструменты тестирования)					
Test case creation (Создание тестового случая): Microsoft Word, Microsoft Excel, JIRA					
Test case tracking (Отслеживание тестовых случаев): JIRA, Confluence					
Test case execution (Выполнение тестового случая): Manual					
Test case management (Управление тестовыми случаями): Microsoft Excel, JIRA, Confluence					
Test reporting (Отчет об испытаниях): JIRA					
Check list creating (Создание контрольного списка): Microsoft Excel, JIRA					
Timeline					
Task Name (Название задачи)	Start	Finish			
Test Planning (Планирование тестирования)	09.11.2021	09.11.2021			
Review Requirements documents (Изучите документы с требованиями)	9.11.2021	9.11.2021			
Create test basis (Создать тестовую базу)	10.11.2021	10.11.2021			
Staff and train new test resources (Укомплектуйте личным составом и обучите новые ресурсы для тестирования)	10.11.2021	10.11.2021			
Exploratory testing (Исследовательское тестирование)	9.11.2021	9.10.2021			
Functional Testing (Функциональное тестирование)	11.11.2021	11.11.2021			
Positive testing (Позитивное тестирование)	11.11.2021	11.11.2021			
Negative testing (Негативное тестирование)	11.11.2021	11.11.2021			
Boundary Testing (Граничное тестирование)	11.11.2021	11.11.2021			
Equivalence Partitioning testing (Эквивалентное разделение)	11.11.2021	11.11.2021			
Ad-Hoc testing (Специальное тестирование)	11.11.2021	11.11.2021			
API Testing (тестирование API (интерфейс прикладного программирования))	12.11.2021	12.11.2021			
Automation testing (Автоматическое тестирование)	12.11.2021	12.11.2021			
Performance Automation testing (Автоматическое тестирование производительности)	12.11.2021	12.11.2021			
APPROVALS (УТВЕРЖДЕНИЯ):					
Project Manager: Sergey Efremov					
QA Lead: Sergey Efremov					