**План тестирования**

### <Разработка Telegram-бота для автоматизации

### процесса сбора заявок на обучение>.

**Product version v 1.0.0**

|  |  |
| --- | --- |
| **Предоставлено для:** | **<Компания-организатор>** |
| **Предоставил:** | **<Алхимики кода>** |
| **Дата:** | **26 января 2025 г.** |
| **Версия:** | **V 1.0.0** |
| **Исполнитель:** | **<Алхимики кода>** |

**2025**

Содержание

1. СПИСОК СОКРАЩЕНИЙ И АББРЕВИАТУР 3
2. ВВЕДЕНИЕ 4
   1. Содержание документа 4
   2. Цель документа 4
   3. Целевая аудитория 4
   4. Обзор программного продукта 4
   5. Цели тестирования 4
3. ОБЪЕМ ТЕСТИРОВАНИЯ 5
   1. Перечень тестируемых компонентов 5
   2. Перечень нетестируемых компонентов 5
4. ТРЕБУЕМЫЕ РЕСУРСЫ 6
   1. Проектная команда 6
   2. Тестовое обеспечение 6
5. СТРАТЕГИИ ТЕСТИРОВАНИЯ 7
   1. Функциональное тестирование 7
   2. Минимальный приемочные тест 7
   3. Регрессионное тестирование 7
   4. Функциональный тест 7
6. МЕТОДОЛОГИЯ ТЕСТИРОВАНИЯ 8
   1. Обзор процесса 8
   2. Тестовая документация 8
   3. Управление дефектами 8
   4. Определение степени важности дефектов 8
   5. Определение приоритета дефектов 9
   6. Регистрация дефектов 9
   7. Критерии качества продукта 10
7. ГРАФИК ТЕСТИРОВАНИЯ 11
   1. Общая оценка временных затрат 11
8. ОЦЕНКА РИСКОВ ПРОЕКТА ТЕСТИРОВАНИЯ 11
9. ОТЧЕТЫ О РЕЗУЛЬТАТАХ ТЕСТИРОВАНИЯ 13
   1. Журнал тестирования 13
   2. Отчет о зарегистрированных дефектах 13
10. **СПИСОК СОКРАЩЕНИЙ И АББРЕВИАТУР**

|  |  |
| --- | --- |
| **Сокращение / аббревиатура** | **Описание** |
| Заказчик | Организация, в интересах которой разрабатывается программный продукт, имеющая полномочия утверждать требования к программному продукту и принимать результат разработок. |
| Дефект | Несоответствие требованию |
| План тестирования | Документ, описывающий объем,  методы и задачи тестирования |
| Продукт | Результат, произведенный в ходе выполнения проекта |
| Проект | Ограниченная во времени деятельность, направленная на разработку уникального продукта |
| Тестовый сценарий | Документ, содержащий перечень проверок для группы функционала продукта без спецификации используемых тестовых данных и обеспечивающий функциональное покрытие тестами стремящееся к 100%; может включать проверки для отдельных полей, доступных опций, графического интерфейса, целостности вводимой информации, динамического поведения полей, дефолтных значений, системные проверки, корректность вводимых значений и сохранения данных, проверки защиты от несанкционированного пользования и попыток изменения или удаления информации |
| Тестовый случай | Документ, содержащий детальное описание теста для одной или более функций с обозначением конкретных данных необходимых для начала тестирования и во время его, а также ожидаемого результата теста. |
| Набор тестовых данных | Документ, содержащий набор пререквизитов, для последующего применения их в процессах функционального и автоматизированного тестирования |
|  |  |
|  |  |

1. **ВВЕДЕНИЕ**

**2.1 Содержание документа**

Описание процесса тестирования, стратегии и подходов к тестированию Telegram-бота для автоматизации процесса заявок на обучение.

Описание взаимодействия группы тестирования и команды разработчиков в процессе тестирования Telegram-бота.

* 1. **Цель документа**
* Спланировать управление тестированием и техническую поддержку тестирования в ходе жизненного цикла разработки  Telegram-бота для автоматизации процесса заявок на обучение;
* Обеспечение полноты тестирования в соответствии с требованиями к продукту, охват всех типов тестирования, описание необходимой проектной документации;
* Организация контроля процесса тестирования, определение методов регистрации дефектов, видов предоставления отчетов;
* Определить график работ, описать используемую методологию и стратегию тестирования;
* Определение критериев качества;
* Определение ресурсов, необходимых для реализации проекта;
* Описание документации, необходимой для обеспечения тестирования по проекту.
  1. **Целевая аудитория**

Документ предназначен для последующего использования сотрудниками группы тестирования проекта и служит для понимания целей и содержания работ по тестированию программного продукта, определяет и описывает перечень работ и этапы тестирования чат-бота.

Также документ предназначен для ознакомления команды разработчиков с процессами взаимодействия с группой тестирования, методами регистрации и отслеживания текущего состояния дефектов, с предоставляемой по проекту тестовой документацией.

* 1. **Обзор программного продукта**

Telegram-бота для автоматизации процесса заявок на обучение является программным решением для упрощения процесса для пользователей и организаторов. Бот позволяет быстро и удобно заполнять заявки, сокращает время на их обработку и уменьшает ошибки, связанные с ручным вводом данных.  
Для организаторов данный бот также дает возможность лучше управлять набором, прогнозировать потоки студентов и позволяет собирать аналитику по направлениям.

* 1. **Цели тестирования**
  2. Проверка корректности ввода и обработки личных данных пользователей;
  3. Проверка правильности работы логики заявления и предзаявки;
  4. Проверка получения пользователями соответствующих уведомлений;
  5. Проверка получения пользователями сообщений о некорректном вводе данных.
  6. Проверка функциональность админ-системы;
  7. Выявление проблем, связанных с несоответствием требованиям к разрабатываемому программному продукту;
  8. Снижение рисков проекта, связанных с качеством разрабатываемого продукта.

1. **ОБЪЕМ ТЕСТИРОВАНИЯ**
   1. **Перечень тестируемых компонентов**

Описание компонентов системы, подлежащих тестированию представлено в таблице:

|  |  |
| --- | --- |
| **Наименование, приложения/компонента** | **Функции / Ссылки / Комментарии** |
| Интерфейс пользователя | - Отображение сообщений  - Кнопки и интерактивные элементы  - Пользовательский опыт (UX) |
| Обработка входящих данных | - Корректная обработка текста и команд  - Игнорирование или обработка нераспознанных команд  - Валидация введенных данных |
| Логика чат-бота | Сценарии общения |
| Производительность | Время ответа на сообщения |

В данной таблице приводится перечень функций, для отображения общих рамок предстоящей работы, определяется набор функций высокого риска.

* 1. **Перечень нетестируемых компонентов**

Нагрузка на сервер во время пиковых нагрузок, стресс-тесты функциональности, защита от SQL-инъекций и XSS-атак, шифрование данных пользователя.

1. **ТРЕБУЕМЫЕ РЕСУРСЫ**
   1. **Проектная команда**

|  |  |  |  |
| --- | --- | --- | --- |
| **Роль** | **Имя и Фамилия** | **Место нахождения** | **Ответственность** |
| Тестировщик | Кристина Воробьева | г. Пенза | 1) Написание проектной тестовой документации;  2) Разработка наборов тестовых данных;  3) Разработка сценариев тестирования, тест-кейсов;  4) Функциональное тестирование. |
| Тестировщик | Андрей Литвинов | г. Пенза | 1) Разработка наборов тестовых данных;  2) Разработка сценариев тестирования, тест-кейсов;  3) Функциональное тестирование. |

* 1. **Тестовое обеспечение**

Разработанный программный продукт должен быть протестирован с помощью:

|  |  |  |
| --- | --- | --- |
| **№** | **Назначение** | **Тестовое обеспечение** |
| 1 | Графический клиент | pgAdmin4 v8.13 |
| 2 | Операционная система | Windows |
| 3 | Операционная система | iOS |
| 4 | Браузер | Google |
| 5 | Браузер | Safari |

1. **СТРАТЕГИЯ ТЕСТИРОВАНИЯ**

Основным методом проверки данного программного продукта будет ручное функциональное тестирование с использованием метода «черного ящика», который базируется на использовании требований и спецификаций, и не предполагает наличия каких-либо специальных знаний о конфигурации и внутренней структуре объекта испытаний.

В процессе тестирования Telegram-бота для автоматизации процесса заявок на обучение будут применяться следующие его типы:

* 1. **Функциональное тестирование**

Функциональное тестирование – это процесс тестирования программного продукта на предмет соответствия требованиям к программному продукту и правильности реализации всех характеристик, заложенных в продукт. Осуществляется вручную тестировщиком. Данный вид тестирования будет включать в себя два этапа:

* 1. **Минимальный приемочный тест (Smoke test)**

МПТ – это  короткий тест, проверяющий основную функциональность программного продукта и его работоспособность, длящийся не более 1 часа. По результатам приемочного теста  принимается решение о допуске версии программного продукта к дальнейшему тестированию.

Если очередная версия программного продукта не прошла приемочный тест, то по его результатам может быть принято решение о дальнейшем частичном тестировании программного продукта. Минимальный приемочный тест будет выполняться для каждой из поставленных версий продукта. Основная его задача – это оперативное отслеживание регрессии качества и получение высокоуровневой картины работоспособности продукта.

* 1. **Регрессионное тестирование**

Для обеспечения надлежащего качества продукта на каждом последующем этапе будет производиться регрессионное тестирование функциональности, разработанной на предыдущей стадии и ее совместимости с новой функциональностью. Регрессионное тестирование подразумевает под собой повторное прохождение всех тестовых случаев протестированных после завершения предыдущего этапа разработки системы, проверки исправления дефектов, допущенных на этом этапе разработки системы, непоявления новых, корректное взаимодействие и совместимость функций.

* 1. **Функциональный тест**

Функциональный тест – основной вид теста, во время которого проверяются основная функциональность программного продукта при стандартном его использовании, а так же проверяется нестандартное использование программного продукта, границы переполнения массивов данных, ввод специальных символов и т.п. Для проведения функционального теста используется весь объем созданной тестовой документации.

Основная цель функционального тестирования – обеспечить максимально возможное покрытие тестами программного продукта.

1. **МЕТОДОЛОГИЯ ТЕСТИРОВАНИЯ**
   1. **Обзор процесса**

Данный процесс включает в себя следующие этапы:

1. Получение требований к программному продукту (предоставляется заказчиком);
2. Анализ требований, определение целей и постановка задач по тестированию (определяется в данном документе);
3. Разработка проектной документации:

3.1 план тестирования:

* + 1. определение тестовой стратегии;
    2. выявление необходимых ресурсов для выполнения проекта;
    3. определение тестового обеспечения
    4. расчет трудоемкости выполнения задач;
    5. составление предварительного графика тестирования;

3.2 написание Test Cases для проведения функциональных тестов, разработка тестовых сценариев и наборов тестовых данных для осуществления технического и автоматизации функционального тестирования;

1. Проведение минимального приемочного теста (МПТ) и принятие решения о приемке версии продукта для тестирования или отказе.
2. Проведение полного цикла функциональных тестов; регистрация дефектов.
   1. **Тестовая документация**

Для обеспечения процесса тестирования требуется разработка следующей проектной тестовой документации:

* План тестирования (описание смотри в пункте 2 «Термины, сокращения, определения», данный документ), документ в формате Word;
* Сценарии тестирования (документ в формате Excel) <Тест-кейсы.xlsx>;
* Отчеты о дефектах (документ в формате Word) <Баг-репорты.docx>.
  1. **Управление дефектами**

Средством регистрации и отслеживанием текущего состояния дефектов для данного проекта выбран инструмент Word.

* 1. **Определение степени важности дефектов**

Важность дефекта (Severity) требуется для правильной оценки степени влияния дефекта на корректность выполнения операций конечным пользователем.

Различают следующуе степени важности дефектов:

* **S1 - Критический (Critical) - Дефект, который блокирует работу приложения и не позволяет продолжать тестирование. Это может быть, например, поломка функциональности, отсутствие доступа к системе и т.д.**
* **S2 - Высокий (High) - Дефект, который серьезно влияет на функциональность или работоспособность приложения, но система в целом продолжает работать. Например, ошибка в обработке платежей.**
* **S3 - Средний (Medium) - Дефект, который не влияет на основную функциональность, но может затруднить опыт пользователя или вызвать недоразумения. Например, некорректное отображение информации.**
* **S4 - Низкий (Low) - Дефект, который не имеет значительного влияния на функционирование системы и может быть легко исправлен. Например, опечатки в тексте или незначительные проблемы интерфейса.**
* **S5 - Тривиальный (Trivial) - Наименее критичный дефект, который не требует срочного исправления, как, например, косметические изменения или улучшения интерфейса.**
  1. **Определение приоритета дефектов**

Приоритет дефекта определяется, чтобы правильно оценить, насколько срочно требуется исправление дефекта.

Приоритет должен быть основан на:

* Серьезности дефекта.
* Критичности функции для пользователя.

Различают следующие приоритеты:

* **P1 – Высокий (High)** – требуется исправить в первую очередь;
* **P2 – Средний** **(Medium)** – требуется исправить во вторую очередь, когда нет дефектов с высоким приоритетом;
* **P3 – Низкий (Low)** – исправляется в последнюю очередь, когда все дефекты с более высоким приоритетом уже исправлены.

В качестве резолюции о причине закрытия дефекта из состояний «зарегистрирован», «открыт», «отложен» могут быть рассмотрены следующие варианты:

* Разработано согласно требованиям (дефект открыт ошибочно);
* Дефект не существенный и не требует исправления (не нарушены бизнес-требования);
* Коррекция дефекта нецелесообразна - исправление дефекта окажет критическое действие на большую часть функциональности  (предложение варианта по изменению требований к продукту);
  1. **Регистрация дефектов**

Шаблон регистрации дефектов:

**Номер**: Bug1

**Критичность:** Medium (возможные значения Blocker, Critical, Medium, Low)

**Приоритет:** Medium

**Тип работ:** DEV или ANA

**Название:** Расшифровочный отчет по коду 8769: некорректная точность для поля SUM\_8769: 4 знака после запятой вместо 2

**Описание:** согласно ФС пункт 1.2 поле SUM\_8769 имеется размерность «руб.коп.». В соответствии с этим, поле всегда должно иметь 2 знака после запятой.

**Шаги воспроизведения:**

1) Открыть страницу отчета и сформировать отчет за дату ХХ.ХХ.ХХХХ;

2) Посмотреть, как отображается сумма в графе SUM\_8769.

**Ожидаемый результат:**

Сумма отображается в виде RRRR.RR

**Фактический результат:**

Сумма отображается в виде RRRR.RRRR

**Вложения:**

При необходимости приложить скриншоты, запросы.

* 1. **Критерии качества продукта**

Программный продукт должен работать в соответствии со спецификациями и требованиями;

Функциональность должна полностью покрывать все требования, изложенные в спецификации;

Программный продукт не должен иметь известных дефектов со статусом «Критический» и «Высокий» к моменту поставки программного продукта заказчику.

1. **ГРАФИК ТЕСТИРОВАНИЯ**
   1. **Общая оценка временных затрат**

Общий объем работ по обновлению и разработке тестовой документации, проведению регрессионных тестов и проверки новой функциональности по Telegram-боту оценивается в 2 чел/3 дня. Из них 2 чел/1 день отводится на разработку и обновление тестовой документации по проекту, 2 чел/1 день на проведение функционального тестирования программного продукта, 2 чел/1 день отводится на подведение итогов тестирования, оформления отчетов о результатах тестирования, предоставление отчетов о качестве программного продукта заказчику. Общая длительность проекта составляет 3 рабочих дня в связи с распределением задач на нескольких человек и с учетом того, что функциональное тестирование стартует по окончании этапа проектирования тестов.

1. **ОЦЕНКА РИСКОВ ПРОЕКТА ТЕСТИРОВАНИЯ**

|  |  |  |  |
| --- | --- | --- | --- |
| **Риск** | **Вероятность наступления** | **Влияние на проект** | **Предварительные меры** |
| Изменение функциональных требований на завершающем этапе проекта | Низкая | Высокое | Ответственный за риск: Заказчик. Предварительные меры: тестовая документация должна иметь адаптивный характер, предусмотреть возможность быстрого внесения изменений (новых требований по проекту) и минимизирования затрат на проведение дополнительных тестов. |
| Программный продукт имеет известные дефекты со статусом «Критический» и «Высокий» к моменту поставки программного продукта заказчику | Низкая | Высокое | Ответственный за риск: Команда разработчиков. Предварительные меры: своевременная нотификация группой тестирования команды разработчиков обо всех новых дефектах, идентифицированных при проведении функциональных тестов. |
| Отставание от графика работ | Низкая | Среднее | Ответственный за риск: группа тестирования / команда разработчиков. Предварительные меры: чёткое планирование временных затрат на выполнение поставленных задач, своевременная нотификация заказчика обо всех проблемах, возникающих в процессе тестирования и имплементации исправлений по дефектам и внедрения новой функциональности. |
| Поставленный для проведения тестирования программный продукт имеет дефекты блокирующие основную функциональность | Низкая | Высокое | Ответственный за риск: Команда разработчиков. Требуется оперативная реакция со стороны команды разработки для предоставления исправлений. |
| Не налажен процесс взаимодействия команд разработчиков и тестирования | Низкая | Среднее | Ответственный за риск: Состав команд разработчиков и тестирования. Требуется: предварительное согласование постановки процесса взаимодействия между командами |
| Возможность обнаружения ошибок на стороне заказчика после проведения тестирования | Низкая | Высокое | Ответственный за риск: Команда тестирования. |

1. **ОТЧЕТЫ О РЕЗУЛЬТАТАХ ТЕСТИРОВАНИЯ**

Отчет о ходе тестирования будет предоставляться по окончании тестирования конечного программного продукта.

* 1. **Журнал тестирования**

В процессе тестирования, результаты выполнения будут заноситься в Документ формата Excel следующего содержания:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Номер** | **Название** | **Предусловие** | **Шаги** | **Данные** | **Ожидаемый результат** | **Фактический результат** | **Статус прохождения** |

Основной целью Журнала тестирования является высокоуровневое отображение качества продукта. В нем приведен общий перечень тестируемых функций без глубокой детализации с результатами работоспособности функций. Отчет предназначен для использования составом проекта для получения общей картины о ситуации с дефектами по программному продукту.

* 1. **Отчет о зарегистрированных дефектах**

Данный отчет содержит информацию о зарегистрированных дефектах и служит для удобства заказчика и разработчиков. Документ создается в формате Word.