**<Тестирование веб приложения Skype>**

**Тест План**

**Version <1.0>**

**Revision History**

| **Date** | **Version** | **Description** | **Author** |
| --- | --- | --- | --- |
| <09/12/22> | <1.0> | <Создание> | <Соболевская И.А.> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

[1. Введение](#_dmjho28hu1g5)

[1.1 Цель](#_30j0zll)

[1.2. Описание](#_d0s69xmcgyqx)

[1.3. Сфера применения](#_cv3wuv8eavs6)

[1.4. Идентификация проекта](#_md23p4dniwdu)

[2. Требования к тесту](#_1t3h5sf)

[3. Тестовая стратегия](#_xam77pjjqs3m)

[3.1. Виды тестирования](#_b16vzqmkhosd)

[3.1.1. Тестирование целостности данных и базы данных](#_17dp8vu)

[3.1.2 Функциональное тестирование](#_16zunmy14pew)

[3.1.3 Usability Testing](#_74vhkdcsghi4)

[3.1.4 UI Тестирование](#_4uolbglf7wmv)

[3.1.5 Конфигурационное тестирование](#_45zescfc4ho)

[3.1.6 Тестирование производительности](#_z337ya)

[3.1.7 Нагрузочное тестирование](#_90gdvmv7n146)

[3.1.8 Стрессовое тестирование](#_g1c6lwis926y)

[3.1.9 Объемное тестирование](#_2lvrmy9v2zp0)

[3.1.10 Тестирование безопасности и контроля доступа](#_ypjlnmuh9w73)

[3.1.11 Тестирование отказа и восстановления](#_i7xz7llypo8)

[3.2 Инструменты](#_ti6jvtg23dgo)

[4. Ресурсы](#_qrfmsd99rnod)

[4.1 Роли](#_q3k8m6i197me)

[4.2 Система](#_d33ak1h0e4e6)

[5. Основные этапы проекта](#_3o7alnk)

[6. Практические результаты](#_ihv636)

[6.2 Протоколы испытаний](#_1hmsyys)

[6.3 Отчеты о дефектах](#_odzkzemeitbu)

**Тест план**

# Введение

Заказчик хочет протестированное веб приложение, прошедшее полный цикл ручного тестирования. План тестирования создан с целью облегчения общения внутри членов команды. Он включает в себя цели, тест, обязанности, критерии входа и выхода, объем, график основных этапов, вход и выход.

## Цель

Цель Тест Плана <Тестирование веб приложения Skype>: описание процесса тестирования веб приложения Skype, включая позитивное и негативное тестирование для отдельных компонентов; функциональное, UI, тестирование производительности, юзабилити тестирование и др. для всего веб приложения.

Компоненты, которые следует протестировать:

* Профиль, включая Настройки, Учетную запись пользователя, Окно поиска, Панель набора номера, Чаты, Звонки, Контакты, Уведомления.
* Главную страницу, включая Изменение аватара, Поделиться профилем, Быстрое собрание, Панель набора ввода.
* Переход по ссылкам из Футера.

Рекомендации и описание используемых стратегий тестирования:

* **Функциональное тестирование** (следует убедиться, что наш продукт соответствует нужной функциональной спецификации. Здесь необходимо протестировать валидацию всех обязательных полей, функциональность доступных кнопок, протестировать, что при отказе функциональности пользователю будет выводиться сообщение об ошибке.
* **Тестирование производительности** (нагрузочное, стрессовое, объемное тестирование). Необходимо оценить какой объем данных будет обрабатываться, сколько пользователей будет находиться в системе, определить производительность, стабильность и масштабируемость приложения под разной нагрузкой.
* **Тестирование Usability** (проверить проходимость пользовательских сценариев — то есть целевых действий за максимально короткое время; оценить востребованность продукта и лояльность к нему пользователей — в частности, по сравнению конкурентами; проанализировать удобство навигации — как быстро покупатель может найти товар, сориентироваться в опциях; оценить пользу, полноту и доступность опубликованного контента — насколько информация понятна для посетителей.).
* **UI тестирование** (тестирование всех визуальных индикаторов и иконок, меню, переключателей, текстовых полей, флажков, панелей инструментов, цветов, шрифтов и других элементов управления).
* **Конфигурационное тестирование** (проверка работы программного обеспечения при различных конфигурациях системы (заявленных платформах, поддерживаемых драйверах, при различных конфигурациях компьютеров и т.д.).
* **Тестирование безопасности** нацелено на поиск недостатков и пробелов с точки зрения безопасности приложения. (Убедиться, что если пользователь вышел из системы или сессия завершена, он не может пользоваться сайтом, что учетная запись пользователя блокируется, если он несколько раз ввел пароль неверно, проверить сайт на SQL-инъекции, проверить права пользователей и их роли, убедиться, что значения сессий отображаются в адресной строке в зашифрованном виде, проверить приложение на устойчивость к XSS-атакам.
* **Тестирование на отказ/восстановление работоспособности приложения** (проверить тестируемый продукт с точки зрения способности противостоять и успешно восстанавливаться после возможных сбоев, возникших в связи с ошибками программного обеспечения, отказами оборудования или проблемами связи (например, отказ сети).

Возможные риски:

* Некорректная оценка трудозатрат.
* Изменяющиеся либо неверно расставленные приоритеты.
* Форс-мажор.

Среда для работы.

| Браузеры | Google Chrome (версия 108.0.5359.125), Mozilla Firefox (версия 108.0.1), Яндекс Браузер (версия 22.11.5.715). |
| --- | --- |
| Операционные системы | Windows 7, 8, 10, MacOS 10.15.7, 10.14.6, 10.13.6. |

Для данного проекта задействована команда из 6 тестировщиков.

Оценка усилий по тестированию:

* Запуск приложения (отображение Splash Screen) - 1 Story Point (SP).
* Тестирование авторизации - 3 SP.
* Работоспособность приложения - 5 SP.
* Валидация обязательных полей - 3 SP.
* Различный сигнал соединения - 1 SP.
* Загрузка файлов - 3 SP.
* Навигация между разделами приложения - 3 SP.
* Настройки и редактирование данных в профиле 5 SP.
* Корректное отображение ошибок - 3 SP.
* Протестировать чат - 5 SP.
* Тестирование видео-, аудиосвязи - 5 SP.
* Протестировать контакты и уведомления - 1 SP.
* Футер (ссылки на различные документы) - 1 SP.

Модули тестового проекта:

1. Авторизация.
2. Настройки (в них же профиль).
3. Поле Поиск.
4. Контакты.
5. Уведомления.
6. Чат.
7. Звонки.
8. Футер.

## **1.2.** Описание

Skype (МФА: [skaɪ̯p]) — бесплатное проприетарное программное обеспечение с закрытым кодом, обеспечивающее текстовую, голосовую и видеосвязь через Интернет между компьютерами (IP-телефония), опционально используя технологии пиринговых сетей, а также платные услуги для звонков на мобильные и стационарные телефоны.

Программа также позволяет совершать конференц-звонки, видеозвонки (до 50 абонентов, включая инициатора), а также обеспечивает передачу текстовых сообщений (чат) и передачу файлов. Есть возможность вместе с изображением с веб-камеры передавать изображение с экрана монитора, а также создавать и отправлять видеосообщения пользователям настольных версий программы.

Программные клиенты Skype выпущены для macOS, iOS, Windows, Linux, Windows Phone, Open webOS, Android, PSP, Maemo, Xbox 360, PlayStation Vita, BlackBerry. Также была выпущена версия для Java, для устройства Kindle Fire HD и Xbox One, ранее выпускался клиент для Symbian. Поддержка Skype была предусмотрена во многих современных телевизорах: LG, Panasonic, Philips, Samsung, Sharp, Sony Bravia, Toshiba, Vizio, но большинство производителей не поддерживает Skype на старых версиях своих телевизоров.

Компания Skype Technologies была основана в 2003 году шведом Никласом Зеннстремом и датчанином Янусом Фриисом. В создании программы Скайп участвовали эстонские программисты Ахти Хейнла ([эст.](https://ru.wikipedia.org/wiki/%D0%AD%D1%81%D1%82%D0%BE%D0%BD%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) Ahti Heinla), Прийт Касесалу ([эст.](https://ru.wikipedia.org/wiki/%D0%AD%D1%81%D1%82%D0%BE%D0%BD%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) Priit Kasesalu) и Яан Таллинн ([эст.](https://ru.wikipedia.org/wiki/%D0%AD%D1%81%D1%82%D0%BE%D0%BD%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) Jaan Tallinn), создавшие ранее программу для файлообмена KaZaA.

10 мая 2011 года корпорация Microsoft приобрела компанию-разработчика программы за 8,5 млрд долларов; начиная с этого момента все права на Skype перешли Microsoft. В версии 5.3 для Windows от 16 июня 2011 года появилась интеграция с Facebook через чат программы. В обновлении от 13 сентября (с 5.5.113 до 5.5.117) осуществлена поддержка Windows 8.

## **1.3***.* Сфера применения

Этот план тестирования описывает интеграционные системные и системные тесты, которые будут проводиться после интеграционного модульного тестирования. Важно протестировать интерфейс системы, а также производительность системы на этом раннем этапе.

Важными показателями производительности для тестирования являются: время отклика для входа в систему, время отклика между модулями приложения, время отклика, когда система загружена.

Предполагается, что модульное тестирование уже обеспечило тщательное тестирование черного ящика, обширный охват исходного кода и тестирование всех интерфейсов модуля.

Тестирование функциональности и возможностей системы будет проводится на системном уровне (проверка функциональных и нефункциональных требований, дефекты системы в целом: на базе требований, на базе use cases).

## 1.4. Идентификация проекта

В таблице указаны документация и доступность, использованные для разработки Тест плана:

| **Документ** | **Автор ресурса** | **Примечания** |
| --- | --- | --- |
| Технические требования | BA | - |
| Функциональная спецификация | BA | - |
| План проекта | Lead | - |

# **2.** **Требования к тесту**

* Система должна работать в Windows 7, 8, 10; MacOS 10.15.7, 10.14.6, 10.13.6.
* Система должна позволять добавлять пользователей.
* Нельзя добавить пользователя, существующего в системе.
* Проверить безопасность входа в систему при помощи имени пользователя и пароля.
* Навигация должна быть удобной.
* Стабильное высокое качество связи.
* Система должна быть простой в использовании и соответствовать целевой аудитории.
* Проверить время отклика на переход в веб приложение.
* Проверить реакцию системы при загрузке 150 человек, вошедших в систему.
* Проверить реакцию системы, когда 150 человек одновременно использует видео/аудио связь.

# 3. Тестовая стратегия

## 3.1. Виды тестирования

### 3.1.1. Тестирование целостности данных и базы данных

| Цель | Убедиться, что методы и процессы доступа к базе данных работают правильно и без повреждения данных. |
| --- | --- |
| Техника | - Вызвать каждый метод и процесс доступа к базе данных, заполняя каждый валидными и невалидными данными или запросами данных.  - Проверить базу данных, чтобы данные были заполнены корректно, все события базы данных произошли правильно, или просмотреть возвращенные данные, проверить, что были получены корректные данные. |
| Критерии завершения | Все методы и процессы доступа к базе данных функционируют в соответствии с проектом и без повреждения данных. |
| Особенности | - Для тестирования может потребоваться среда разработки СУБД или драйверы для ввода или изменения данных непосредственно в базах данных.  - Процессы следует запускать вручную.  - Следует использовать базы данных небольшого или минимального размера (ограниченное количество записей), чтобы повысить наглядность любых неприемлемых событий. |

### 3.1.2 Функциональное тестирование

Данный тип тестирования основан на методе черного ящика; это проверка приложения и его внутренних процессов путем взаимодействия с приложением через графический интерфейс пользователя (GUI) и анализа результатов.

| Цель | Обеспечить надлежащую функциональность цели тестирования. |
| --- | --- |
| Техника | Выполнение use cases или функций, используя валидные и невалидные данные, чтобы проверить следующее:  ∙ Ожидаемые результаты достигаются при использовании достоверных данных.  ∙ При использовании некорректных данных отображаются соответствующие сообщения об ошибках или предупреждения. |
| Критерии завершения | Выполнено более 85% запланированных тестов.  Устранены выявленные дефекты высокого и важного приоритета.. |
| Особенности | Из проблем, влияющих на тестирование можно выделить - недостаточность или неточность документации. |

### 3.1.3 Тестирование Usability

| Цель | * Выявление барьеров в дизайне продукта; * Обнаружение точек развития (все, что помогает сделать дизайн лучше); * Изучение поведения и предпочтений пользователей (из целевой аудитории). |
| --- | --- |
| Техника | Создать тесты для тестирования юзабилити. |
| Критерии завершения | Все запланированные тесты выполнены.  Все выявленные дефекты устранены. |
| Особенности | - |

### 3.1.4 UI Тестирование

| Цель | Убедиться, что пользовательский интерфейс предоставляет пользователю соответствующий доступ и навигацию по функциям объекта тестирования. |
| --- | --- |
| Техника | Создать тесты для каждого окна, чтобы проверить правильную навигацию и состояния объектов для каждого окна приложения и объектов. |
| Критерии завершения | Каждое окно успешно проверено, чтобы оставаться в пределах приемлемого стандарта, привычного для пользователя. |
| Особенности | - |

### 3.1.5 Конфигурационное тестирование

| Цель | Убедиться, что объект тестирования правильно работает на требуемых аппаратных и программных конфигурациях. |
| --- | --- |
| Техника | - Использовать сценарии функционального тестирования.  - Открывать и закрывать различное программное обеспечение, не связанное с тестом, либо как часть теста, либо до его начала.  - Выполнение выбранных транзакций для имитации взаимодействия actor с программным обеспечением, являющимся объектом тестирования, и программным обеспечением, не являющимся объектом тестирования.  - Повторить описанный выше процесс, сократив до минимума доступную обычную память на клиентской рабочей станции. |
| Критерии завершения | Для каждой комбинации целевого и нецелевого программного обеспечения все транзакции успешно завершены без сбоев. |
| Особенности | Все системы, сетевое ПО, сетевые серверы, базы данных и т. д. также должны быть задокументированы как часть этого теста. |

### 3.1.6 Тестирование производительности

| Цель | Проверить поведение производительности при следующих условиях:  - нормальная предполагаемая рабочая нагрузка,  - ожидаемая рабочая нагрузка в худшем случае. |
| --- | --- |
| Техника | Сценарии должны выполняться на одном компьютере (в лучшем случае для сравнительного анализа одного пользователя, одной транзакции) и повторяться с несколькими пользователями (виртуальными или реальными, см. раздел «Особые замечания» ниже). |
| Критерии завершения | - Одна транзакция или один пользователь: успешное завершение тестовых сценариев без каких-либо сбоев и в пределах ожидаемого или необходимого времени, выделенного на транзакцию.  - Несколько транзакций или несколько пользователей: успешное завершение тестовых сценариев без каких-либо сбоев и в рамках приемлемого распределения времени. |
| Особенности | Комплексное тестирование производительности включает фоновую нагрузку на сервер.  Для этого можно использовать несколько методов, в том числе:  ∙ «Направляйте транзакции» непосредственно на сервер, обычно в форме вызовов языка структурированных запросов (SQL).  ∙ Создайте «виртуальную» пользовательскую нагрузку для имитации множества клиентов, обычно нескольких сотен. Для выполнения этой задачи используются инструменты эмуляции удаленного терминала. Этот метод также можно использовать для загрузки сети «трафиком».  - Используйте несколько физических клиентов, каждый из которых запускает тестовые сценарии для загрузки системы.  Тестирование производительности должно выполняться на выделенной машине или в определенное время. Это обеспечивает полный контроль и точное измерение.  Базы данных, используемые для тестирования производительности, должны иметь либо фактический размер, либо одинаково масштабироваться. |

### 3.1.7 Нагрузочное тестирование

| Цель | Проверка времени поведения производительности для определенных транзакций или бизнес-кейсов в различных условиях рабочей нагрузки. |
| --- | --- |
| Техника | Изменить файлы данных, чтобы увеличить количество транзакций, или тесты, чтобы увеличить количество транзакций. |
| Критерии завершения | Несколько транзакций или несколько пользователей: успешное завершение тестов без каких-либо сбоев и в пределах приемлемого распределения времени. |
| Особенности | ∙ Нагрузочное тестирование должно выполняться на выделенной машине или в определенное время. Это обеспечивает полный контроль и точное измерение.  ∙ Базы данных, используемые для нагрузочного тестирования, должны иметь либо фактический размер, либо одинаково масштабироваться. |

### 3.1.8 Стрессовое тестирование

| Цель | Выявление и документирование стрессовых условий, при которых система НЕ работает должным образом. |
| --- | --- |
| Техника | - Используйте тесты, разработанные для тестирования производительности или нагрузочного тестирования.  - Для тестирования ограниченных ресурсов тесты должны выполняться на одной машине, а объем оперативной памяти должен быть уменьшен или ограничен.  - Для оставшихся стресс-тестов следует использовать несколько клиентов, выполняющих либо одни и те же тесты, либо дополнительные тесты для получения наихудшего объема или сочетания транзакций. |
| Критерии завершения | Выполняются все запланированные тесты, и указанные пределы системы достигаются или превышаются без отказа программного обеспечения или условий, при которых происходит отказ системы, выходящих за указанные условия. |
| Особенности | - Для нагрузки на сеть могут потребоваться сетевые инструменты для загрузки сети сообщениями или пакетами.  - Синхронизация одновременного доступа клиентов к одним и тем же записям или учетным записям данных. |

### 3.1.9 Объемное тестирование

| Цель | Убедиться, что объект тестирования успешно работает в следующих сценариях с большими объемами:  - Максимальное количество подключенных или смоделированных клиентов, выполняющих одну и ту же бизнес-функцию в наихудшем случае (производительность) в течение длительного периода времени.  - Достигнут максимальный размер базы данных (фактический или масштабированный), и одновременно выполняются несколько запросов или транзакций отчетов. |
| --- | --- |
| Техника | - Используйте тесты, разработанные для тестирования производительности или нагрузочного тестирования.  - Следует использовать несколько клиентов, выполняя либо одни и те же тесты, либо дополнительные тесты для получения наихудшего объема транзакций или сочетания (см. Стресс-тестирование выше) в течение длительного периода.  - Создается максимальный размер базы данных (фактический, масштабированный или заполненный репрезентативными данными), и несколько клиентов используются для одновременного выполнения запросов и отчетов о транзакциях в течение длительных периодов времени. |
| Критерии завершения | - Все запланированные тесты были выполнены, и заданные пределы системы достигнуты или превышены без сбоев программного обеспечения или падения программного обеспечения. |
| Особенности | - |

### 3.1.10 Тестирование безопасности и контроля доступа

Тестирование безопасности и контроля доступа сосредоточено на двух ключевых областях безопасности:

∙ Безопасность на уровне приложений, включая доступ к данным или бизнес-функциям.

∙ Безопасность на уровне системы, включая вход в систему или удаленный доступ к ней.

| Цель | * Безопасность на уровне приложений: Убедиться, что субъект может получить доступ только к тем функциям или данным, для которых его тип пользователя имеет разрешения. * Безопасность на уровне системы: Убедиться, что только те участники, которые имеют доступ к системе и приложениям, имеют к ним доступ. |
| --- | --- |
| Техника | * Безопасность на уровне приложений: Определить и перечислить каждый тип пользователя и функции или данные, для которых каждый тип имеет разрешения.   - Создать тесты для каждого типа пользователя и проверить каждое разрешение, создавая транзакции, характерные для каждого типа пользователя.  - Изменить тип пользователя и повторно запустить тесты для тех же пользователей. В каждом случае убедиться, что эти дополнительные функции или данные корректно доступны или запрещены.   * Безопасность на уровне системы: См. Особенности ниже |
| Критерии завершения | Для каждого известного типа субъектов доступны соответствующие функции или данные, и все транзакции функционируют, как ожидалось, и выполняются в предыдущих тестах функций приложения. |
| Особенности | Доступ к системе должен быть рассмотрен или обсужден с соответствующим сетевым или системным администратором. Это тестирование может не потребоваться, поскольку оно может быть функцией сетевого или системного администрирования. |

### 3.1.11 Тестирование отказа и восстановления

| Цель | Убедитесь, что процессы восстановления (вручную или автоматизированные) правильно восстанавливают базу данных, приложения и систему до желаемого известного состояния. В испытания должны быть включены следующие типы условий:  - отключение питания клиента  - сбой питания сервера  - прерывание связи через сетевые серверы  - неполные циклы (прерваны процессы фильтрации данных, прерваны процессы синхронизации данных).  - недопустимый указатель базы данных или ключи  - неверный или поврежденный элемент данных в базе данных |
| --- | --- |
| Техника | Тесты, созданные для тестирования функций и бизнес-циклов, следует использовать для создания серии транзакций. После достижения желаемой начальной контрольной точки необходимо выполнить или смоделировать следующие действия по отдельности:  - Сбой питания клиента: выключить компьютер.  - Прерывание питания сервера: смоделировать процедуры отключения питания сервера.  - Прерывание через сетевые серверы: имитировать потерю связи с сетью ( отключить связь или отключить сетевые серверы, маршрутизаторы).  Как только вышеуказанные условия или смоделированные условия достигнуты, должны быть выполнены дополнительные транзакции, и по достижении этого состояния второй контрольной точки должны быть вызваны процедуры восстановления.  Тестирование незавершенных циклов использует тот же метод, что и описанный выше, за исключением того, что сами процессы базы данных должны быть прерваны или преждевременно завершены. |
| Критерии завершения | Во всех вышеперечисленных случаях приложение, база данных и система после завершения процедур восстановления должны вернуться в известное желаемое состояние. Это состояние включает в себя повреждение данных, ограниченное известными поврежденными полями, указателями или ключами, а также отчеты, указывающие на процессы или транзакции, которые не были завершены из-за прерываний. |
| Особенности | - Тестирование восстановления. Процедуры отключения кабеля (имитация потери питания или связи) могут быть нежелательными или невыполнимыми. Могут потребоваться альтернативные методы, такие как диагностические программные средства.  - Эти тесты следует запускать в нерабочее время или на изолированной машине. |

## 3.2 Инструменты

Для этого проекта будут использованы следующие инструменты:

|  | Tool | Vendor/In-house (поставщик/внутренние) | Version |
| --- | --- | --- | --- |
| Управление тестированием | Microsoft Excel | - | - |
| Отслеживание дефектов | Microsoft Word | - | - |
| Управление проектом | Microsoft Word | - | - |

# 4. Ресурсы

Тестировщики проекта и их модули:

1. Баранов Сергей - тестирование Чат.
2. Кунцевич Владислав - Контакты, Уведомления.
3. Романовская Полина - Настройки (включая профиль).
4. Прокопович Анна - Футер, Чат.
5. Соболевская Ирина - Звонки.
6. Цупикова Ульяна - Авторизация, поле Поиск.

## 4.1 Роли

В этой таблице показаны кадровые предположения для проекта.

| Человеческие ресурсы | | |
| --- | --- | --- |
| Должность | Рекомендуемые минимальные ресурсы  (количество выделенных штатных ролей) | Конкретные обязанности или комментарии |
| Менеджер проекта | 1 | * Обеспечивает управленческий контроль. * Обязанности: * обеспечить техническое руководство * приобрести соответствующие ресурсы * предоставлять управленческую отчетность |
| Дизайнер | 1 | * Определяет, расставляет приоритеты и реализует тестовые примеры. * Обязанности: * создать Тест план * создать тестовую модель * оценить эффективность усилий по тестированию |
| Тестировщик | 4 | * Выполняет тесты. * Обязанности: * выполнять тесты * журнал результатов * восстановиться после ошибок * запросы на изменение документа |
| Администратор базы данных | 1 | * Обеспечивает управление и обслуживание среды тестовых данных (базы данных) и активов. * Обязанности: * администрирование тестовых данных (база данных) |

## 4.2 Система

В следующей таблице указаны системные ресурсы для проекта тестирования

| Система ресурсов | |
| --- | --- |
| Ресурс | Имя / Тип |
| Сервер базы данных |  |
| —Сеть или подсеть | TBD (подлежит уточнению) |
| —Имя сервера | TBD (подлежит уточнению) |
| —Имя базы данных | TBD |
| Клиентские тестовые ПК |  |
| Включить специальные требования к конфигурации | TBD |
| Тестовый репозиторий |  |
| —Сеть или подсеть | TBD |
| —Имя сервера | TBD |
| ПК для разработки тестов | TBD |

# **5. Основные этапы проекта**

| **Milestone Task (Контрольная задача)** | **Effort (Усилие)** | **Start Date** | **End Date** |
| --- | --- | --- | --- |
| Запланировать тест | 18 | 09.12.2022 | 11.12.2022 |
| Создать тест | 24 | 12.12.2022 | 15.12.2022 |
| Реализовать тест | 18 | 16.12.2022 | 18.12.2022 |
| Выполнить тест | 24 | 19.12.2022 | 22.12.2022 |
| Оценить тест | 8 | 23.12.2022 | 23.12.2022 |

# 6. Практические результаты

В процессе тестирования будут созданы чек лист, тест-кейсы, баг-репорт а также отчет о тестировании в Excel.

## 6.2 Протоколы испытаний

Отчет о тестировании будет оформлен в Excel.

## 6.3 Отчеты о дефектах

Дефекты будут оформлены в виде баг-репорта в Excel.