

Test plan

1.1 Введение

Основная информация

Документ описывает методы и подходы к тестированию, которые будут использоваться тестировщиками отдела тестирования, для тестирования сайта и приложения.

План тестирования может использоваться как тестировщиками, так и менеджерами, разработчиками.

Объект тестирования — это деятельность, направленная на проверку работоспособности функций сайта и приложения в браузерах Chrome, Firefox, и в операционных системах Windows, Linux, Android и iOS.

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

Область тестирования сайта

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

1. Регистрация
2. Восстановление логина и пароля
3. Идентификация пользователя
4. Загрузка медиафайлов.
5. Выбор страховки
6. Покупка полиса
7. Оплата
8. Работа электронного архива для хранения документов и печатных форм страхового полиса
9. Работа сервиса обмена данными со страховой компанией
10. Сохранение и редактирование профиля пользователя.

Область тестирования мобильного приложения

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

1. Создание учетной записи с помощью электронной почты.
2. Создание учетной записи с помощью социальных сетей.

3. Вход в систему под пользователем
4. Идентификация пользователя
5. Восстановление и изменение пароля.
6. Воспроизведение медиафайлов.
7. Выбор и покупка страховки
8. Оплата
9. Работа сервиса обмена данными со страховой компанией
10. Работа электронного архива
11. Поиск медиафайлов.
12. Профиль.

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. План и стратегия тестирования

1. 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.
- Интеграционное тестирование.
- Регрессионное тестирование.

Также будет проверено, как программный продукт запускается в браузерах и устройствах, которые должны его поддерживать, как он запускается и останавливается, сколько времени ему требуется для запуска.

В рамках тест-плана не будут выполняться виды тестирования:

- Тестирование безопасности.

1.1 Тестирование установки версии

Техника:	<ul style="list-style-type: none"> Запустить в браузерной версии или мобильном приложении Сбербанк Онлайн продукт «Удобная страховка» <p>Каналы продаж «Удобной страховки»:</p> <ol style="list-style-type: none"> Через строку поиска и в меню на сайте https://sber.ru/. Через строку поиска и в меню на сайте https://sber.insure/. В личном кабинете браузерной версии https://online.sberbank.ru/. В личном кабинете мобильного приложения СберБанк Онлайн.
Входные критерии	<ul style="list-style-type: none"> Подготовлено тестовое окружение, приложение готово к запуску
Критерий приемки:	<ul style="list-style-type: none"> Продукт запустился

1.2 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

Техника:	<p>Выполните каждый сценарий, используя допустимые и недопустимые данные:</p> <ul style="list-style-type: none"> Ожидаемые результаты возникают при использовании достоверных данных. Соответствующие сообщения об ошибках или предупреждения отображаются, когда используются неверные данные. Каждый пункт проверен.
Входные критерии	<ul style="list-style-type: none"> Подготовлено тестовое окружение, приложение готово к тестированию на тестовой площадке.
Критерий приемки:	<ul style="list-style-type: none"> Все дымовые тесты пройдены. Нет блокирующих багов. Все баги с высоким приоритетом поправлены. Тесты пройдены.

1.3 Интеграционное тестирование

Для определения необходимости интеграционного тестирования полезно перечислить все внешние системы, с которыми взаимодействует продукт, и указать, какие именно данные мы получаем и передаём.

Техника:	<p>Выполните каждый сценарий, используя допустимые и недопустимые данные:</p> <ul style="list-style-type: none"> Ожидаемые результаты возникают при использовании достоверных данных. Связь с внешними сервисами позволяет выполнить нужные действия Соответствующие сообщения об ошибках или предупреждения отображаются, когда используются неверные данные.
Входные критерии	<ul style="list-style-type: none"> Подготовлено тестовое окружение, приложение готово к тестированию на тестовой площадке.

Критерий приемки:	<ul style="list-style-type: none"> • Нет блокирующих багов. • Внешние сервисы работают • Все баги с высоким приоритетом поправлены. • Тесты пройдены.
--------------------------	---

1.4 Тестирование пользовательского интерфейса

Техника:	Проверить соответствие интерфейса визуальному образцу заказчика и насколько удобно расположены для пользователя элементы интерфейса: <ul style="list-style-type: none"> • Ни один элемент не выбивается из цветовой гаммы • Все надписи читабельны • Элементы интерфейса не съезжают
Входные критерии	<ul style="list-style-type: none"> • Подготовлено тестовое окружение, приложение готово к тестированию на тестовой площадке.
Критерий приемки:	<ul style="list-style-type: none"> • Дизайн соответствует требованиям клиента. • Все элементы интерфейса находятся на своих местах

1.5 Интеграционное тестирование

Техника:	Провести повторные тесты
Входные критерии	<ul style="list-style-type: none"> • Подготовлено тестовое окружение, приложение готово к тестированию на тестовой площадке.
Критерий приемки:	<ul style="list-style-type: none"> • Повторные тесты пройдены • Нет блокирующих багов. • Все баги с высоким приоритетом поправлены.

Сообщения 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 не критического 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 программного продукта.

- Номер версии программного продукта (при тестировании на мобильном устройстве).
- 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;
- Частота возникновения дефекта.
- Тяжесть дефекта.
- Дополнительная информация о дефекте в виде прикрепленных скриншотов или видеозаписей.

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

5. Ресурсы

Инструменты

Будут использованы следующие инструменты:

Наименование процесса	Инструмент
Баг трекинг-система	Jira
Тест кейсы	Testrail
Захват изображений	Screenshots / Video capture

Список браузеров

Наименование браузера	Версия браузера
Chrome	Последняя
Firefox	Последняя

Список устройств

Устройство	Операционная система
iPhone	Все поддерживаемые ОС
Android	Все поддерживаемые ОС
PC	Все поддерживаемые ОС

6. Критерии качества

Продукт должен работать в соответствии с требованиями и техническим заданием

(если таковые имеются).

Продукт не должен содержать критических и блокирующих дефектов в окончательной версии проекта.

7. Риски процесса тестирования

Следующие проблемы могут повлиять на результаты тестирования:

- Изменения и модификации программного продукта, которые не были запланированы и не обсуждались заранее с командой тестирования.
- Изменения в требованиях к программному обеспечению, которые не были предварительно обсуждены с командой тестирования.
- Задержки в исправлении/исправлении ошибок.
- Задержки в доставке новых сборок команде тестирования.

8. Ожидания команды тестирования

Команда тестирования должна быть обеспечена действительными, обновленными документами в течение всего процесса тестирования.

Все необходимое оборудование, приборы, устройства и программное обеспечение должны быть приобретены и подготовлены до начала процесса тестирования.

Все критические ошибки должны быть исправлены как можно скорее.

Примечание к выпуску следует добавлять к каждому выпуску программного обеспечения для команды тестирования. В примечании должно быть объяснено, какие элементы, функции и функции были добавлены в программу и как эти дополнения влияют на программное обеспечение.

Разработчики должны исправить все ошибки в программных модулях перед выпуском новой версии.

9. Обязанности участников тестовой группы

Менеджер проекта

- Управление процессом тестирования.
- Предоставление всех необходимых ресурсов для проведения испытаний.

Руководитель отдела тестирования

- Сбор и изучение требований.
- Проверка документации.
- Планирование работ по тестированию.
- Контроль за проведением испытаний, чтобы работы выполнялись в соответствии с планом.
- Отчеты о ходе выполнения, количестве и серьезности обнаруженных ошибок.

Тестирующий

- Процесс контроля качества, регистрация обнаруженных ошибок в утвержденной системе отслеживания ошибок.

10. Результаты

- План тестирования.
- Отчеты об ошибках и отчеты о ходе тестирования.