

Стратегия тестирования продукта "Удобная страховка" от Сбербанка

<b>Цель:</b> Сделать доступным продукт «Удобная страховка»: - в мобильном приложении «Сбербанк Онлайн», - в браузерной версии.  Интеграция нового сервиса в приложение и на сайт.	<b>Задачи:</b> Разработка и тестирование бэк- и фронт-систем банка. Обеспечение интеграционного взаимодействия с сервисами: 1. Идентификация клиента; 2. Платежная система; 3. Сервис обмена данными со страховой компанией; 4. Электронный архив хранения документов и печатных форм страхового полиса (БД Oracle).	<b>Анализ информации:</b> Продуктом будут пользоваться на мобильных ОС iOS, Android и в браузерах Chrome, Firefox, Opera, Safari, Яндекс. Каналы продаж «Удобной страховки»: 1. Через строку поиска и в меню: <a href="https://sber.ru/">https://sber.ru/</a> . 2. Через строку поиска и в меню: <a href="https://sber.insure/">https://sber.insure/</a> . 3. В личном кабинете: <a href="https://online.sberbank.ru/">https://online.sberbank.ru/</a> . 4. В личном кабинете моб. приложения СберБанк Онлайн. Аудитория: платежеспособная 18+.	<b>Цель тестирования:</b> Проверка соответствия реальных и ожидаемых результатов поведения программы. Основной функционал: Тестирование факта подключения страховки, оформления полиса и попадания данных в БД Oracle.
<b>Заинтересованные стороны:</b> Все участники проекта - владелец продукта, скрам-мастер, программисты, дизайнер, тестировщик.	<b>Условия начала тестирования:</b> Разработка требуемого функционала завершена. Необходимая документация подготовлена.	<b>Продукт считается реализованным:</b> При успешном выполнении тестирования в запланированном объеме и отсутствии ошибок с высоким уровнем критичности отсутствуют.	<b>Риски:</b> Не успеть выпустить задачу к дедлайну. Разработчики долго чинят баги после тестирования. Тестировщики часто переключаются между задачами. Разработка в параллель с тестированием.
<b>Время разработки сервиса:</b> 8 недель.	<b>Условия окончания тестирования:</b> Результаты проведенного тестирования соответствуют исходным требованиям к качеству продукта		

№	Этапы разработки	Объекты тестирования	Техники тестирования	Исполнители
1	<b>Бэкенд:</b> разработка системы - добавление новых Баз данных Oracle (хранение данных пользователей продукта "Удобная страховка")	Структура и хронология взаимодействия с Базами данных на сервере. Проверка сохранения целостности данных.	Структурное тестирование (проверка элементов для хранения данных). Функциональное тестирование (проверка соответствия трензаций требованиям). Нефункционально (нагрузочное, стресс-тестирование) Юнит-тесты. SQL-тесты. API-тесты.	Программист
2	<b>Фронтэнд:</b> Разработка нового пользовательского интерфейса для мобильного приложения и браузерной версии сайта.	Функции: вход/регистрация/авторизация; заполнение форм и оформление необходимой документации.	GUI - графическая часть UI - проверка элементов интерфейса Юзабилити - удобство и эффективности использования интерфейса Кросс-браузерное тестирование. Кросс-платформенное тестирование Функциональное тестирование. Нефункциональное (скорость, отзывчивость)	Тестировщик
3	Взаимодействие между фронтэнд и бэкенд	Работа протоколов передачи информации	Функциональное тестирование: <i>Black-box</i> (без доступа к коду системы). <i>White box</i> – с доступом к коду системы. Интеграционное тестирование (взаимодействие сервисов посредством интеграционной шины асинхронного типа взаимодействия).	Тестировщик, программист
4	Интеграция системы идентификации клиента	Модули хранения данных зарегистрированных и авторизованных пользователей на сервере, проверка функциональности регистрации и авторизации клиента.	Функциональное тестирование (регистрация). Юнит-тестирование: проверка правильности хранения и передачи информации в БД Oracle. Интеграционное тестирование. Регрессионное тестирование: проверка функциональностей "вход/ регистрация/авторизация".	Тестировщик, программист
5	Интеграция платёжной системы	Оформление и оплата страховки со стороны пользователя, хранение и передача информации о счете клиента со стороны сервера	Функциональное тестирование (проверка кнопок "Оформить страховку", "Купить", заполнение форм документов). Юнит-тестирование (передача в БД сведений о лицевом счете). Тестирование безопасности (насколько безопасна оплата). Интеграционное тестирование. Регрессионное тестирование	Тестировщик, программист
6	Интеграция сервиса обмена данными со страховой компанией	Сравнение информации от пользователя и сервера страховой компании. Корректность данных форм документов, хранящихся в БД конкретного пользователя	Функциональное тестирование (заполнить заявления на страхование, скачать pdf-формат страховки). Юнит-тестирование (взаимодействие с сервером страховой компании). Интеграционное тестирование. Регрессионное тестирование.	Тестировщик, программист
7	Интеграция электронного архива хранения документов и печатных форм страхового полиса	БД - сохраняются ли данные о страховке, в каком формате, возможно ли выгрузить данные со стороны пользователя. Порядок и сроки хранения данных со стороны пользователя и со стороны сервера.	Функциональное тестирование (кнопки "Просмотр", "Скачать", "Распечатать"). Регрессионное тестирование (оформить страховку). Юнит-тестирование (работа с БД). Нагрузочное тестирование (большое кол-во пользователей). Тестирование безопасности (корректность формата сохранности данных и их сохранность). Интеграционное тестирование. Регрессионное тестирование.	Тестировщик, программист
8	Проверки всех функций перед релизом проекта	Проверка и финальная отладка всех систем с позиции пользователя сервера. Отслеживание финальных багов.	Нефункциональное тестирование (скорость, безопасность). Нагрузочное тестирование. Тестирование UX (опыти тестировщика с акцентом на пользовательском опыте). Альфа-тестирование (силами штатных разработчиков или тестировщиков). Beta-тестирование (с привлечением добровольцев из числа будущих пользователей продукта).	Тестировщик