

Проект: ПРОГРАММА
План тестирования (Тест-план)
Версия 1.1

История изменений документа:

Дата	Версия	Описание	Автор
25.08.2015	1.0	Первая версия плана тестирования	Захаров В.В.
12.04.2016	1.1	Дополнение плана информацией	Захаров В.В.

Оглавление

1. Введение.....	3
1.1. Цель.....	3
1.2. Предпосылки.....	3
1.3. Объём работ.....	3
1.4. Идентификация проекта	4
2. Требования к тестированию	4
3. Стратегия тестирования	6
3.1. Типы тестирования	6
3.2. Инструментарий.....	7
4. Ресурсы	8
4.1. Исполнители и роли	8
4.2. Системные ресурсы	8
5. Основные этапы и трудозатраты	8
6. Критерии тестирования.....	9
7. Итоговые отчёты	9

1. Введение

1.1. Цель

Документ «План тестирования» для проекта «ПРОГРАММА» (далее «Продукт») преследует следующие цели:

- определить стратегии тестирования Продукта, которые планируется использовать;
- определить компоненты Продукта, которые должны быть протестированы;
- спланировать процесс тестирования и техническую поддержку тестирования;
- описать структуру и рамки тестирования, достаточные для достижения целей и решения задач тестирования в проекте;
- определить перечень инструментов, которые будут использоваться в процессе тестирования;
- определить график работ, этапы и основные вехи;
- определить обязанности, роли и ресурсы в процессе тестирования;
- определить критерии начала и окончания процесса тестирования.

Документ предназначен для руководства проекта, проектного офиса и руководства управления информационных технологий для согласования планов, и оценки затрат.

Документ предназначен группе тестирования для ознакомления с характером предстоящих работ, анализа и разбиения работ на подзадачи.

1.2. Предпосылки

Продукт предназначен для обмена юридически значимыми электронными документами. Особенности:

- при передаче документов используется электронная подпись (ЭП);
- продукт является SaaS-решением;
- при разработке Продукта использовались языки программирования Ruby (front-end), C++ (back-end);
- в качестве решения для баз данных используется PostgreSQL;
- для работы с ЭП используется стороннее решение – Крипто Про (на сервере и на клиентских местах).

1.3. Объём работ

Для обеспечения качества Продукта проводятся следующие виды тестирования: ручное тестирование, автоматизированное тестирование, дымовое тестирование, модульное тестирование, модульное интеграционное тестирование, системное интеграционное тестирование, функциональное тестирование, тестирование интерфейса, нагрузочное тестирование, тестирование базы данных, тестирование безопасности, тестирование конфигурации, регрессионное тестирование, тестирование установки, тестирование документации.

Текущий подход к контролю качества подразумевает следующие вехи проекта:

- система готова к демонстрации заказчику;
- система готова к промышленной эксплуатации.

Такое разбиение предполагает, как можно более раннюю поставку работающего прототипа заказчику с целью получения обратной связи.

Для проверки готовности прототипа служат приёма-сдаточные испытания. Критерий готовности - акт сдачи прототипа, подписанный приёма-сдаточной комиссией. Приёма-сдаточные испытания описываются в отдельном документе.

Для проверки готовности к промышленной эксплуатации используется полный набор запланированных тестов. Готовность определяется руководителем проекта, на основании представленных ему руководителем или менеджером тестирования отчётов о полноте тестового покрытия и списка значимых расхождений, оформленных в виде ошибок в системе отслеживания ошибок.

1.4. Идентификация проекта

Нижеследующая таблица содержит сведения о документации по проекту и ее доступности.

Документ	Создан или Доступен	Получен или Просмотрен	Автор или ресурс	Примечание
1	2	3	4	5
Технические требования				
Функциональные требования				
Use-Case (сценарии использования)				
План проекта				
Спецификация проекта (документ или комплект документов, которые описывают проект и используются в качестве основы для применения и интеграции изделия)				
Прототип				
Руководство пользователя				
Бизнес модель				
Модель потока данных				
Бизнес функции и правила				
Оценка проекта или бизнес-рисков				

В столбцах 2 и 3 таблицы указываем «+» (да) или «-» (нет).

2. Требования к тестированию

Контролю качества должны быть подвергнуты как программно-аппаратный комплекс в целом, так и его отдельные части. В ходе тестирования будут проверяться следующие комплексные показатели качества:

Функциональные возможности:

- *Пригодность.* Свойства и характеристики программного обеспечения, относящийся к наличию и соответствию набора функций конкретным задачам. Примерами соответствия является состав функций, ориентированных на задачу, из входящих в него подфункций и объемы таблиц.
- *Правильность.* Свойства и характеристики программного обеспечения, относящиеся к обеспечению правильности или соответствия результатов. Например, она включает необходимую степень точности вычисленных значений.
- *Способность к взаимодействию.* Свойства и характеристики программного обеспечения, относящиеся к способности его взаимодействовать с конкретными системами.
- *Согласованность.* Свойства и характеристики программного обеспечения, которые заставляют программу придерживаться соответствующих стандартов или соглашений, или положений законов, или подобных рекомендаций.
- *Защищенность.* Свойства и характеристики программного обеспечения, относящиеся к его способности предотвращать несанкционированный доступ, случайный или преднамеренный, к программам и данным.

Надёжность:

- *Стабильность.* Свойства и характеристики программного обеспечения, относящиеся к частоте отказов при ошибках в программном обеспечении.
- *Устойчивость к ошибкам.* Свойства и характеристики программного обеспечения, относящиеся к его способности поддерживать определенный уровень качества функционирования в случаях программных ошибок или нарушения определенного интерфейса.
- *Восстанавливаемость.* Свойства и характеристики программного обеспечения, относящиеся к его возможности восстанавливать уровень качества функционирования и восстанавливать данные, непосредственно поврежденные в случае отказа, а также к времени и усилиям, необходимым для этого.

Практичность:

- *Понятность.* Свойства и характеристики программного обеспечения, относящиеся к усилиям пользователя по пониманию общей логической концепции и ее применимости.
- *Обучаемость.* Свойства и характеристики программного обеспечения, относящиеся к усилиям пользователя по обучению его применению (например, оперативному управлению, вводу, выводу).
- *Простота использования.* Свойства и характеристики программного обеспечения, относящиеся к усилиям пользователя, но эксплуатации и оперативному управлению.

Эффективность:

- *Характер изменения во времени.* Свойства и характеристики программного обеспечения, относящиеся к временам отклика и обработки и к скоростям выполнения его функций.
- *Характер изменения ресурсов.* Свойства и характеристики программного обеспечения, относящиеся к объему используемых ресурсов и продолжительности такого использования при выполнении функции.

Сопровождение:

- *Анализируемость.* Свойства и характеристики программного обеспечения, относящиеся к усилиям, необходимым для диагностики недостатков или случаев отказов, или определения составных частей для модернизации.
- *Изменяемость.* Свойства и характеристики программного обеспечения, относящиеся к усилиям, необходимым для модификации, устранению отказа или для изменения условий эксплуатации.
- *Устойчивость.* Свойства и характеристики программного обеспечения, относящиеся к риску от непредвиденных эффектов модификации.
- *Тестируемость.* Свойства и характеристики программного обеспечения, относящиеся к усилиям, необходимым для проверки модифицированного программного обеспечения.

Мобильность:

- *Адаптируемость.* Свойства и характеристики программного обеспечения, относящиеся к удобству его адаптации к различным конкретным условиям эксплуатации, без применения других действий или способов, кроме тех, что предназначены для этого в рассматриваемом программном обеспечении.
- *Простота внедрения.* Свойства и характеристики программного обеспечения, относящиеся к усилиям, необходимым для внедрения программного обеспечения в конкретное окружение.
- *Соответствие.* Свойства и характеристики программного обеспечения, которые заставляют программу подчиняться стандартам или соглашениям, относящимся к мобильности.
- *Взаимозаменяемость.* Свойства и характеристики программного обеспечения, относящиеся к простоте и трудоемкости его применения вместо другого конкретного программного средства в среде этого средства. Взаимозаменяемость с конкретным программным средством не предполагает, что данное средство заменимо рассматриваемым программным средством. Взаимозаменяемость может включать атрибуты простоты внедрения и адаптируемости.

3. Стратегия тестирования

Основными задачами тестирования являются:

- проведение функционального тестирования каждого модуля и компонента системы для проверки его соответствия функциональным требованиям;
- проведение комплексного тестирования для обеспечения взаимодействия модулей и компонентов друг с другом согласно требованиям, к системе;
- определение и максимальное увеличение производительности системы и каждого отдельного модуля;
- проведение нагрузочного тестирования для обеспечения отказоустойчивости системы и каждого отдельного модуля;
- максимальная автоматизация процесса тестирования;
- разработка достаточного набора контрольных примеров для тестирования новых модулей и компонентов;
- своевременная разработка контрольных примеров для покрытия устраняемых ошибок;
- увеличение покрытия кода тестовыми примерами;
- тестирование удобства применения модулей, имеющих графический интерфейс.

3.1. Типы тестирования

Для решения указанных выше задач тестирования будут использоваться следующие виды тестирования:

- Ручное тестирование — выполнение тестировщиком прохода тестового цикла вручную, с последующей ручной фиксацией результатов по каждому тесту в отчете.
- Автоматизированное тестирование — автоматический проход тестового цикла, с последующим автоматическим уведомлением заинтересованных лиц о результатах.
- Дымовое тестирование — простейший вид тестирования, основанный на определении успешности сборки системы из ветви исходного кода, находящейся в разработке.
- Модульное тестирование — самый важный вид тестирования, основанный на проверке работоспособности функций, методов и свойств в условиях их нормального и ошибочного исполнения. Это тестирование проводится на уровне исходного кода каждого существующего класса. Что нужно тестировать на данном этапе:
 - класс правильно объявлен;
 - структура класса соответствует спецификации требований;
 - класс имеет достаточную функциональность;
 - класс совместим со средствами автоматической обработки кода (анализ покрытия, качества кода и т.п.);
 - некорректное функционирование и ошибочные ситуации корректно обрабатываются;
 - класс совместим со связанными классами в рамках используемого наследования, полиморфизма, процедур вызова и т.п.;
 - время выполнения, частота выполнения, нагрузка на ресурсы соответствуют требованиям;
 - класс не содержит утечек памяти и других ресурсов.
- Модульное интеграционное тестирование — после разработки тестов на отдельные классы необходимо проверить, как они будут работать вместе в рамках одного исполняемого процесса. Необходимо проверить, как соотносятся классы, разработанные по-разному разными разработчиками.
- Системное интеграционное тестирование — проверяет работоспособность компонентов системы на уровне взаимодействия нескольких отдельных исполняемых процессов. На данном этапе тестируется функционирование клиент-серверных систем, их взаимодействие внутри и с внешними компонентами.

- Функциональное тестирование — рассматривает продукт, состоящий из множества классов, процессов, компонентов, данных как единое целое. На этом этапе проверяется в целом его работоспособность, функциональные и технические характеристики, а также бизнес-логика.
- Тестирование интерфейса — проверка клиентских и административных интерфейсов пользователя на возможность выполнения с их помощью сценариев использования. Сценарий использования представляет собой последовательность действий пользователя, которые имитируют его активность при работе с интерфейсами системы. Сценарий использования должен покрывать спецификацию требований к пользовательскому интерфейсу. Такое тестирование производится в ручном и в автоматическом режиме с помощью специализированных утилит. Тестирование должно проверять корректность работы интерфейсной части приложения при любых возможных настройках экрана (различное разрешение, масштаб, шрифт), при изменениях фокуса, при работе с мышью и клавиатурой.
- Нагрузочное тестирование — определение и проверка характеристик производительности системы в заданной конфигурации оборудования и набора данных.
- Тестирование базы данных — проверка функционирования внешней базы данных и хранимых процедур в соответствии со спецификацией требований. Проверка политики безопасности доступа к базе в соответствии с ролями системы. Определение и проверка характеристик базы данных, таких как производительность, среднее время доступа, максимальное количество обслуживаемых клиентов, минимальная и максимальная длительность обработки запроса и т.п.
- Тестирование безопасности — определение ролей и проверка списка функций системы, доступных для каждой роли. Может осуществляться на уровне интерфейса, на уровне компонента, на уровне базы данных, на уровне модуля и на сетевом уровне. Включает проверку методов шифрования данных при хранении и передаче, отказа доступа к запрещенным функциям, перехвата данных, подделки удостоверения личности, отказа в обслуживании и других атак.
- Тестирование конфигурации — проверка работоспособности системы в заданном окружении конфигурации оборудования и набора данных.
- Регрессионное тестирование — повторное выборочное тестирование продукта с модифицированными частями после исправления ошибки, добавления новой функции, рефакторинга и изменения кода. Внесение изменений в исходный код может повлечь цепочку зависимостей и получение новых ошибок во взаимозависимых функциях. Данный вид тестирования минимизирует риск подобного события.
- Тестирование инсталляции — проверка корректной работы инсталляционного пакета, инсталляционных сценариев для копирования, обновления и последующей автоматической настройки работоспособности системы.
- Тестирование документации — проверка документации на полноту описания инструкций пользования в соответствии с «Требованиями по разработке пакета рабочей документации пользователя и администратора системы».

3.2. Инструментарий

Для тестирования Продукта будут использованы следующие средства:

	Инструмент	Поставщик	Версия
Инструмент хранения тест-кейсов	TestLink, MS Excel		
Инструмент для функционального тестирования	C#, Selenium WebDriver		
Инструмент для тестирования производительности и нагрузки	JMeter		
Система отслеживания ошибок	JIRA		
Инструменты тестирования СУБД	Database Benchmark	http://stssoft.com/products/database-benchmark	
Инструменты формирования отчётов	Word, Excel		

4. Ресурсы

Этот раздел представляет рекомендуемые для проекта ресурсы, их главные обязанности, знания и навыки.

4.1. Исполнители и роли

Исполнитель	Ресурсы (количество работников)	Обязанности, комментарии
Менеджер по тестированию	1	Управление отделом тестирования Формирование отчётов
Тест дизайнер	1	Определение, приоритизация и разработка тест-кейсов Оценка эффективности тест-кейсов
Инженер QA (автоматизатор)	1	Определение, приоритизация и разработка автоматизированных тестов
Тестирующий	1	Выполнение тестов Фиксирование результатов Воспроизведение ошибок Документирование ошибок
Администратор тестового окружения	1	Обеспечивает настройку, поддержку и администрирования тестовой среды
Администратор баз данных	1	Обеспечивает настройку, поддержку и администрирования тестовой среды (базы данных)
Разработчик	1	Разработка модульных тестов

4.2. Системные ресурсы

Нижеследующая таблица представляет системные ресурсы для тестирования проекта.

Ресурсы	Спецификация
Сервер базы данных	-
-- сеть	
-- имя сервера	
-- имя базы данных	
-- конфигурация	
Тестовый сервер (front-end)	-
-- сеть	
-- имя сервера	
-- конфигурация	
Тестовый сервер (back-end)	-
-- сеть	
-- имя сервера	
-- конфигурация	
Тестовый компьютер (Клиент)	-
-- конфигурация	

5. Основные этапы и трудозатраты

В таблице содержатся описание трудоёмкости для каждого объёма работ.

Задача	Дата начала	Дата окончания
Планирование тестов		
Разработка тест-кейсов		
Разработка автоматизированных тестов		
Выполнение тестов		
Оценка результатов тестов		

6. Критерии тестирования

Критерии успешности тестирования:

- все тест кейсы с высоким приоритетом закрыты с результатом «пройден».
- тестовое покрытие проверено и является достаточным, где критерий достаточности составляет не менее 99% покрытия требований тестами;
- итоговый отчёт составлен и утвержден руководителем тестирования и заказчиком.

Критерии прерывания и продолжения тестирования:

- критерием прерывания тестирования является появления и занесения в систему отслеживания ошибок блокирующих ошибок;
- критерием продолжения тестирования является закрытие блокирующей ошибки в системе отслеживания ошибок.

7. Итоговые отчёты

После окончания тестирования формируются отчёты, в которых отображаются следующие данные:

- количество всех тестов;
- количество: всех тестов; пройденных успешно; не пройденных успешно; тестов, которые не выполнялись;
- информация о критических ошибках, размещённых в системе отслеживания ошибок;
- информация о времени, затраченном на тестирование;
- информация о дате начала и завершения тестирования;
- информация о количестве ошибок: всего ошибок; закрыто ошибок; новые ошибки выявленные в процессе тестирования; новые ошибки выявленные в процессе эксплуатации.
- информация о критических ошибках, размещённых в системе отслеживания ошибок и влияющих на готовность Продукта;
- Оценка степени готовности Продукта.