

Тест план
проведения функционального тестирования
следующих частей
сайта Snov.io:
- регистрация
- авторизация
- создание списка элементов

Версия 1 . 00

Содержание

Введение.....	5
1.1 Цели.....	5
1.2 Объем	5
1 Функционал, подлежащий тестированию.....	5
1.1 Регистрация нового пользователя	5
1.1.1 Регистрация пользователя в системе	5
1.1.2 Просмотр страницы “Terms of Service”	5
1.1.3 Просмотр страницы “Privacy Policy”	6
1.1.4 Переход на страницу авторизации	6
1.2 Авторизация пользователя	6
1.2.1 Авторизация пользователя в системе	6
1.2.2 Возврат на страницу регистрации.....	6
1.2.3 Восстановление пароля	6
1.2.4 Режим авто-сохранения данных о пользователе	7
1.3 Создание списка персон по группам.....	7
1.3.1 Отображение раздела «Prospects»	7
1.3.2 Добавление новой группы в раздел «Lists».....	7
1.3.3 Поиск данных для группы	7
1.3.4 Добавление предварительного списка найденных персон в группу	7
1.4 Создание списка компаний и веб-сайтов.....	7
1.4.1 Отображение раздела «Companies & websites».....	7
1.4.2 Добавление новой группы в раздел «Companies list».....	8
1.4.3 Поиск данных для группы	8
1.4.4 Добавление предварительного списка найденных компаний в группу	8
2 Перечень основных типов устройств, операционных систем, браузеров, которые должны быть использованы при тестировании.....	8
2.1 Устройства, на которых будет проводиться тестирование	8
2.2 Операционные системы.....	9
2.3 Браузеры, обязательные для тестирования.....	9
2.4 Наиболее популярные сайты для выполнения поиска контактов	9
3 Виды и инструменты тестирования.....	9

3.1	Ручное тестирование.....	9
3.2	Автоматизированное тестирование.....	9
3.2.1	Автоматизированное тестирование api методов.....	9
3.2.2	Автоматизированное тестирование UI.....	10
3.2.3	Дымовое тестирование при помощи авто-тестов	10
3.2.4	Регрессионное тестирование	10
4	Тестовая документация.....	10
4.1	Чек-лист - описание, назначение, согласование	10
4.1.1	Описание документа Чек-лист	10
4.1.2	Назначение документа Чек-лист	11
4.1.3	Содержимое чек-листа и инструмент для ведения чек-листа	11
4.1.4	Согласование чек-листа	12
4.2	Тестовый сценарий (Тест-кейс)	12
4.2.1	Описание документа Тестовый сценарий	12
4.2.2	Назначение документа Тестовый сценарий	12
4.2.3	Содержимое тест-кейса и инструмент для ведения тест-кейса.....	12
4.3	Матрица покрытия тестами (МПТ)	13
4.3.1	Описание документа МПТ.....	13
4.3.2	Назначение документа МПТ.....	13
5	Порядок оформления дефектов в виде баг-репорта.....	14
5.1	Основные поля баг-репорта	14
5.2	Информация, которая будет указываться в описании баг-репорта.....	14
5.3	Жизненный цикл дефекта.....	14
6	Определение приоритета дефекта.....	15
6.1	Глобальный приоритет (Global Severity)	15
6.2	Принцип определения составляющих свойств	15
6.2.1	Серьезность (Severity)	15
6.2.2	Серьезность (Severity)	16
6.2.3	Серьезность (Severity)	16
6.3	Алгоритм определения глобального приоритета.....	16
7	Определение стратегии тестирования	17
7.1	Условия начала тестирования.....	17
7.2	График этапов тестирования.....	17
7.3	Критерии окончания тестирования	18

8	Определение рисков, которые могут возникнуть при тестировании	19
---	---	----

Введение

1.1 Цели

Этот документ определяет план тестирования части функционала сайта Snivio.io (регистрация, авторизация, создание списка), методы и техники, процессы и практику тестирования, которые будут использоваться для обеспечения качества и приемки разрабатываемого продукта.

1.2 Объем

Данный план тестирования будет охватывать следующее:

- Определение всех групп функциональности при регистрации, авторизации и создании списка, которые подлежат функциональному тестированию на данном этапе
- Определение методов и техник для качественного тестирования выделенных групп функциональности
- Указание основных инструментов для тестирования
- Описание основных тестовых документов, которые будут создаваться на протяжении всего жизненного цикла разрабатываемого продукта для выделенной области функциональности

1 Функционал, подлежащий тестированию

1.1 Регистрация нового пользователя

1.1.1 Регистрация пользователя в системе

При тестировании функционала регистрации будет выполняться тестирование страницы сайта с адресом : <https://app.snov.io/register>. Кроме того будет выполнено тестирование с целью проверки правильной валидации полей ввода при регистрации, проверка соответствия вида страницы регистрации требуемому дизайну, проверка отсылки письма для подтверждения емейла, проверка правильности письма для подтверждения емейла и приветственного письма, проверка того, что зарегистрированный пользователь есть в системе, проверка на невозможность повторной регистрации с одним и тем же емейлом

1.1.2 Просмотр страницы “Terms of Service”

При тестировании данного под-функционала будет выполняться проверка наличия ссылки и возможности перехода по ссылке «Terms of Service» со страницы регистрации. Тестирование соответствия заголовка и текста на странице тексту на ссылке по смыслу. Кроме того, будет

протестирована возможность возврата на страницу регистрации и авторизации с описываемой страницы.

Для данной страницы должны быть протестирована работа всех кнопок: верхние кнопки: Pricing, Chrome Extention, Email Verifier, Support, кнопка он-лайн помощи, кнопки на нижней панели: Pricing, Chrome Extention, Email Verifier, Support, Team, Blog, а также кнопки перехода на страницу Snov.io в соцсетях (FaceBook, Twitter, YouTube)

1.1.3 Просмотр страницы “Privacy Policy”

При тестировании данного под-функционала будет выполняться проверка наличия ссылки и возможности перехода по ссылке «Privacy Policy» со страницы регистрации. Тестирование соответствия заголовка и текста на странице тексту на ссылке по смыслу. Кроме того, будет протестирована возможность возврата на страницу регистрации и авторизации с описываемой страницы.

Для данной страницы должны быть протестирована работа всех кнопок: верхние кнопки: Pricing, Chrome Extention, Email Verifier, Support, кнопка он-лайн помощи, кнопки на нижней панели: Pricing, Chrome Extention, Email Verifier, Support, Team, Blog, а также кнопки перехода на страницу Snov.io в соцсетях (FaceBook, Twitter, YouTube)

1.1.4 Переход на страницу авторизации

Поскольку на странице регистрации расположена ссылка для перехода на страницу авторизации, то этот функционал так же должен быть протестирован.

1.2 Авторизация пользователя

1.2.1 Авторизация пользователя в системе

При тестировании функционала авторизации будет выполняться тестирование страницы сайта с адресом : <https://app.snov.io/login>. Кроме того будет выполнено тестирование, с целью проверки правильной валидации полей ввода при авторизации, проверка соответствия вида страницы авторизации требуемому дизайну, тестирование невозможности выполнить авторизацию без регистрации или не полностью проведенной регистрации. В ходе тестирования будет подвержено контролю качества поведение системы при успешной авторизации и проверка отображения правильных данных в системе, соответствующих авторизованному пользователю. Будет выполнены тестирование поведения системы при неуспешной авторизации.

1.2.2 Возврат на страницу регистрации

При тестировании страницы авторизации будет выполнена проверка на правильность работы ссылки «Sign Up» и переход на страницу регистрации

1.2.3 Восстановление пароля

При тестировании страницы авторизации будет тестироваться функционал восстановления пароля и проверка на возможность авторизации после восстановления пароля.

1.2.4 Режим авто-сохранения данных о пользователе

При тестировании страницы авторизации будет выполнена проверка авторизации после включенного и отключенного режима авто-сохранения данных.

1.3 Создание списка персон по группам

1.3.1 Отображение раздела «Prospects»

При тестировании данного функционала будет выполняться тестирование правильность отображения раздела «Prospects» после нажатия кнопки «Prospects», правильность заголовков блоков, отображения списка групп раздела, списка выбранных элементов, а так же их количества.

1.3.2 Добавление новой группы в раздел «Lists»

При тестировании функционала добавления новой группы для раздела «Lists» будет выполнена проверка, что новая группа добавлена, порядок сортировки групп в разделе соответствует требованию, проверка случая, когда выбрана новая группа на правильность отображения раздела при пустом списке элементов, правильность данных о группе, наличие обязательных кнопок и других элементов, которые предусмотрены дизайном.

1.3.3 Поиск данных для группы

При тестировании данной части функционала будет выполнена проверка, что найденные на сайте элементы при вызове дополнения Хрома Snovio включены в предварительный список для выбора. Тут будет выполняться проверка функций работы с предварительным списком – выбрать все, отменить все, выбор группы, в которую будет включен список. Кроме того, будет выполняться проверка кейса, когда элемент в предварительном списке уже есть в какой-либо группе и он должен быть помечен определенным образом.

1.3.4 Добавление предварительного списка найденных персон в группу

При тестировании данной части функционала будет выполнена проверка, что все отмеченные для добавления персоны из предварительного списка добавлены в соответствующую группу, в правильном порядке и при этом все данные перенесены правильно в список указанной группы.

1.4 Создание списка компаний и веб-сайтов

1.4.1 Отображение раздела «Companies & websites»

При тестировании данного функционала будет выполняться тестирование правильность отображения раздела «Companies & websites» после нажатия кнопки «Companies & websites», правильность заголовков блоков, отображения списка групп раздела, списка выбранных элементов, а так же их количества.

1.4.2 Добавление новой группы в раздел «Companies list»

При тестировании функционала добавления новой группы для раздела «Companies list» будет выполнена проверка, что новая группа добавлена, порядок сортировки групп в разделе соответствует требованию, проверка случая, когда выбрана новая группа на правильность отображения раздела при пустом списке элементов, правильность данных о группе, наличие обязательных кнопок и других элементов, которые предусмотрены дизайном.

1.4.3 Поиск данных для группы

При тестировании данной части функционала будет выполнена проверка, что найденные на сайте элементы при вызове дополнения Хрома Snovio включены в предварительный список для выбора. Тут будет выполняться проверка функций работы с предварительным списком – выбрать все, отменить все, выбор группы, в которую будет включен список. Кроме того, будет выполняться проверка кейса, когда элемент в предварительном списке уже есть в какой-либо группе и он должен быть помечен определенным образом. В окне предварительного списка список групп должен соответствовать списку, который создан на странице раздела.

1.4.4 Добавление предварительного списка найденных компаний в группу

При тестировании данной части функционала будет выполнена проверка, что все отмеченные для добавления персоны из предварительного списка добавлены в соответствующую группу и раздел в правильном порядке и при этом все данные перенесены правильно в список указанной группы.

2 Перечень основных типов устройств, операционных систем, браузеров, которые должны быть использованы при тестировании

2.1 Устройства, на которых будет проводиться тестирование

По требованию заказчика, самыми используемыми устройствами для тестируемого продукта являются персональные компьютеры и планшеты с размером экрана 2048x1536 и 2732x2048 пикселей

2.2 Операционные системы

Операционные системы, обязательные для тестирования: Windows 10, Windows 8, Windows 7, Windows 3, Windows XP, Android 5+, IOS 9+.

Для первоочередного уровня тестирования: Windows 10, Windows 7, Android 7, IOS 10.

2.3 Браузеры, обязательные для тестирования

Так как приложение имеет расширение для браузера Chrome, то этот браузер является обязательным при тестировании (версии начиная с 49+ и до последних)

2.4 Наиболее популярные сайты для выполнения поиска контактов

Обязательными для тестирования являются популярные поисковые сайты: LinkedIn, Facebook, ...

3 Виды и инструменты тестирования

3.1 Ручное тестирование

Ручное тестирование является обязательным на начальном этапе тестирования. А так же может выполняться и в дальнейшем для функционала, который не покрыт автотестами. Некоторый функционал целесообразно тестировать вручную, так как бывают случаи когда автотесты сделать довольно дорого из-за особенностей разработки кода.

На этапе выполнения ручного тестирования будут использоваться такие инструменты как: Postman, расширения Chrome, дающие возможность просмотреть вид страницы на экранах с разным расширением, на разных типах девайсов.

На этапе ручного тестирования будут созданы тест кейсы, сценарии для разработки автотестов.

Перед выполнением ручного тестирования достаточно сложных задач будет разрабатываться чек-лист, в котором заранее записываются кратко все кейсы, которые должны быть выполнены при ручном тестировании.

3.2 Автоматизированное тестирование

В ходе проведения функционального тестирования указанной части сайта для обеспечения высокого качества продукта, уменьшения времени на рутинное тестирование, сокращения расходов на ручное тестирование целесообразно автоматизировать некоторые этапы тестирования.

3.2.1 Автоматизированное тестирование API методов

Для быстрого выполнения тестов необходимо разработать тестирование API методов,

обеспечивающих работу тестируемого функционала. Для этого будет использоваться инструмент Postman или Py.Test с использованием библиотеки Requests.io (Python)

3.2.2 Автоматизированное тестирование UI

Для разработки автотестов UI будет использован самый популярный на данный момент инструмент Selenium Web Driver с использованием UnitTest, Py.test, Python 2.7 или 3.

С помощью автотестов UI можно будет выполнять быстрое смоук тестирование основного функционала после внесенных обновлений в систему.

Автоматизированные тесты ускорят прогонку основных end to end сценариев, что поможет сэкономить время тестировщиков для выполнения более интеллектуальных задач.

Входными артефактами для разработки автотестов являются:

- Тесткейсы, разработанные на этапе ручного тестирования
- Тестовые end to end сценарии для прохождения полного пути реализации какой либо операции в системе и имитации действий пользователя в реальной жизни при работе с приложением. Тестовые сценарии так же могут быть разработаны на этапе ручного тестирования.

3.2.3 Дымовое тестирование при помощи авто-тестов

После внесения изменений в код приложения необходимо выполнить дымовое тестирование основного функционала для определения целесообразности проведения более общего тестирования или отправки кода на доработку разработчику.

Для проведения дымового тестирования из всего набора тестов выбираются тесты, которые покрывают самый важный функционал, который должен работать обязательно. С помощью авто-тестов смоук тестирование не будет занимать много времени.

3.2.4 Регрессионное тестирование

Незначительные изменения в одном месте программы могут привести к возникновению ошибок в другом месте. Для выполнения тестирования с целью выявления таких дефектов проводится регрессионное тестирование. В ходе разработки авто-тестов будет реализован набор авто-тестов для регрессионного тестирования. Если изменения вносятся в код каждый день, то регрессионные тесты можно запускать ночью, чтобы не занимать время тестировщика или же на отдельных компьютерах.

4 Тестовая документация

4.1 Чек-лист - описание, назначение, согласование

4.1.1 Описание документа Чек-лист

Инженер по тестированию будет использовать Чек-лист для тестирования каждой задачи.

Чек-лист представляет собой документ, описывающий, что должно быть выполнено или проверено при тестировании конкретной задачи. Чек-лист может быть различного уровня детализации, в

зависимости от требований и сложности задачи.

Образец чек-листа для тестирования страницы регистрации пользователя на сайте Snov.io:

A	B	C	D	E	F	G
CLRI_1 Чеклист для тестирования страницы регистрации пользователя на сайте Snov.io						
	Описание проверки	Windows 10	Windows 7	Android	MAC IOS 10	Comment
		Chrome	Chrome	Crome	Chrome	
	Переходы на другие страницы по нажатию на кнопки и кнопки-ссылки					
CLRI_1_1	При нажатии на кнопку-ссылку "Log In" открылась страница вторизации	Passed	Passed	Passed	Passed	
CLRI_1_2	При нажатии на кнопку-ссылку "Terms of Service" открылась страница "Terms of Service" с правильным содержанием и соответствует дизайну требований	Failed	Failed	Failed	Failed	
CLRI_1_3	При нажатии на кнопку-ссылку "Privacy Policy" открылась страница "Privacy Policy" с правильным содержанием и соответствует дизайну требований	Passed	Passed	Passed	Passed	
CLRI_1_4	После правильного заполнения всех полей ввода для регистрации и нажатии кнопки "Sign me Up" открылась страница авторизации	Passed	Passed	Passed	Passed	
CLRI_1_5	После правильного заполнения всех полей ввода для регистрации и нажатии кнопки "Sign me Up" на открывшейся странице авторизации видим сообщение с текстом: "Thank you for signing up for Snovio. Please, check your email to confirm registration." о необходимости подтвердить регистрацию	Passed	Passed	Passed	Passed	
	Вид страницы регистрации					
CLRI_1_6	Форма регистрации содержит заголовок: "Sign up"	Passed	Passed	Passed	Passed	
CLRI_1_7	Форма регистрации содержит 4 поля ввода со следующими названиями: - Name - E-Mail Address - Password - Confirm Password	Passed	Passed	Passed	Passed	
CLRI_1_8	Форма регистрации содержит следующий текст под полями ввода: "By creating an account, you accept Snovio's Terms of Service and Privacy Policy"					

4.1.2 Назначение документа Чек-лист

Чек-лист используется для:

- отображения списка проверок, которые должны быть проверены при тестировании конкретной задачи, что защищает процесс тестирования от случаев, когда тестировщик забыл выполнить некоторые тесты для задачи
- сохранения отчетности и результатов тестирования
- возможности контроля за качеством тестирования
- создания в дальнейшем тест-кейсов на базе которых будут разрабатываться автоматизированные тесты
- если чек-лист подготовлен согласно требованиям к задаче еще до ее реализации, то разработчик может руководствоваться им, на случай, если какие-то проверки он не предусмотрел

4.1.3 Содержимое чек-листа и инструмент для ведения чек-листа

Чек-лист содержит такие данные:

- список проверок
- статус проверок
- название ОС, платформы, роли пользователя и др., для которого делается определенная проверка

Для ведения чек-листа будут использоваться таблицы GoogleDocs.

4.1.4 Согласование чек-листа

Если задача достаточно емкая и требует проверки множества случаев или же разработчик считает, что некоторые случаи проверки могут быть неявными или скрытыми от глаз тестировщика в коде, то тестировщик может согласовать чеклист для проверки задачи с разработчиком или Проектным менеджером.

4.2 Тестовый сценарий (Тест-кейс)

4.2.1 Описание документа Тестовый сценарий

Инженер по тестированию будет использовать Тест-кейс для тестирования какого либо функционала.

Тест-кейс представляет собой документ, описывающий какие действия необходимо совершить для проверки какого-либо функционала проекта.

4.2.2 Назначение документа Тестовый сценарий

Тест-кейс используется для:

- Обеспечения возможности быстрого ознакомления с функционалом проекта и проверки его работоспособности
- Создания автоматизированных тестов
- Выполнения смоук-тестирования
- Выполнения регрессионных тестов и их автоматизации
- Заполнения матрицы покрытия тестами (см. Описание документа в TS-04-03)

4.2.3 Содержимое тест-кейса и инструмент для ведения тест-кейса

Тест-кейс содержит такие данные:

- Идентификатор
- Название (Краткое описание)
- Предусловия, которые должны быть выполнены для успешного прохождения тест-кейса
- Шаги воспроизведения - последовательность действий, которая приведет к ожидаемому результату
- Ожидаемый результат

Для ведения Тест-кейсов будут использоваться таблицы Googl Docs.

Образец документа для ведения тест-кейсов, а также сохранения результатов тестирования по ним:

A	B	C	D	E	F	G
	Tets requerement:					
	Pass		Pending			
	Fail		Number of test cases:			
	Pass					
	Faild					
Id	Test case description	Test case procedure	Expected Output	Version (Bild)	Note	Result (Pass/Fail) TestDate
REG1	Открытие формы регистраци	1.Зайти на страницу регистрации на сайт Snov.io: https://app.snov.io/register	Открылась страница регистрации. На странице присутствуют такие элементы: 1) заголовок: "Sign up" 2) поля ввода: - Name, - E-Mail Address, - Password, - Confirm Password 3) текст под полями ввода: "By creating an account, you accept Snovio's Terms of Service and Privacy Policy.", где : - Terms of Service - Privacy Policy - кнопки-ссылки 4) кнопка для регистрации "Sign me Up" 5) кнопка-ссылка на страницу Авторизации с текстом перед ней "Already have an account? Login"			
REG2	Регистрация пользователя (Успешный кейс)	1.Попасть на форму регистрации (см. шаги в кейсе REG1) 3.Заполнить все поля (значения полей описаны в комментарии) 4.Нажать на кнопку "Sign me Up"	Открылась страница авторизации с текстом под заголовком: "Thank you for signing up for Snovio. Please, check your email to confirm registration."			

4.3 Матрица покрытия тестами (МПТ)

4.3.1 Описание документа МПТ

МПТ представляет собой таблицу, которая содержит соответствие функциональных требований и подготовленных тестовых сценариев.

4.3.2 Назначение документа МПТ

МПТ используется для:

- определения степени покрытия тестами требований по продукту

Общий пример документа «Матрица покрытия тестами» представлен ниже:

Матрица покрытия тестами для требований (Регистрация пользователя)				
Проект: Snov.io				
Функциональные требования			Документ с тесткейсами	
Номер требования	Номер подтребования	Описание требования	Приоритет	Номер тест-кейса
Регистрация пользователя				
16,1	1	Пользователь может зайти на страницу регистрации	1	REG1
	2	Пользователь может перейти на страницу авторизации по нажатию на кнопку-ссылку "Log In"	1	
	3	Пользователь может попасть на страницу авторизации после правильного заполнения полей регистрации и нажатия на кнопку "Sign In"	1	
	4	После перехода на страницу авторизации после успешной регистрации пользователь должен увидеть уведомление о том, что необходимо подтвердить регистрацию при помощи письма, которое отправлено на емейл.	1	REG2
	5	Пользователь, при незаполнении полей на странице регистрации или же при невалидном заполнении полей должен видеть уведомление о некорректно-заполненных полях, а так же выделение неверно заволенных полей и комментарии о требовании к заполнению поля.	2	

5 Порядок оформления дефектов в виде баг-репорта

5.1 Основные поля баг-репорта

При оформлении дефекта должны быть заполнены следующие поля:

- Приоритет (Срочный, высокий, нормальный, низкий)
- Тема (Краткое название дефекта)
- Описание
- Статус
- Номер спринта
- Фамилия и имя разработчика, который занимается исправлением дефекта или ранее исправил его

5.2 Информация, которая будет указываться в описании баг-репорта

Поле “Описание дефекта” должно содержать максимум полезных сведений о проблеме и включать:

- Окружение (устройство и его модель, ОС, окружение (тест/прод) и т.д. где воспроизводится дефект)
- Шаги с кратчайшим путем воспроизведения бага (может иметь вид текстового описания, в виде снимков экрана с нумерацией шагов, а также видео с воспроизведением бага (если это необходимо))
- Ожидаемый результат
- Логи системы, по которым возможно отследить причину ошибки, набор данных, которые подаются на вход и набор исходящих данных
- По необходимости список api - методов, которые вызываются во время возникновения проблемы
- Выводы о возможной причине бага, если проблема обсуждалась совместно тестировщиком и разработчиком

5.3 Жизненный цикл дефекта

Каждый оформленный в системе дефект имеет состояние, которое показывает, на каком этапе жизненного цикла он находится.

Возможные состояния дефекта:

- NEW - дефект заведен в баг-трекере в виде баг-репорта (в JIRA аналогичный статус SELECT FOR DEVELOPMENT)
- IN PROGRESS - дефект находится в работе - разработчик занимается исправлением дефекта
- READY FOR TEST - дефект исправлен, выложен на тестовую среду
- TESTING - дефект находится на стадии тестирования

- REOPENED - задача с дефектом переоткрыта если при тестировании исправляемого дефекта обнаружилось, что дефект не исправлен или исправлен не полностью или же обнаружены новые случаи появления такого дефекта (в JIRA задача опять переводится в статус SELECT FOR DEVELOPMENT)
- DONE - дефект исправлен и протестирован.

6 Определение приоритета дефекта

6.1 Глобальный приоритет (Global Severity)

Глобальный приоритет (Global Severity) бага определяется 3-мя составляющими:

- Серьезность (Severity) — свойство тестового артефакта, характеризующее влияние артефакта на работоспособность приложения. Является характеристикой, определяемой с точки зрения функциональности.
- Приоритет (Priority) - свойство тестового артефакта, влияющее на очередность выполнения задачи или устранения дефекта. Является характеристикой, определяемой с точки зрения бизнеса.
- Частота (Frequency) — свойство тестового артефакта, характеризующее частоту (%) возникновения при использовании приложения конечными пользователями. Является характеристикой, определяемой с точки зрения пользовательских сценариев использования продукта.

Таким образом, глобальный приоритет определяется на основании значений трёх свойств:
 $globalSeverity = F(Priority, Severity, Frequency)$

6.2 Принцип определения составляющих свойств

6.2.1 Серьезность (Severity)

Определяется QA специалистом, Lead QA или Team Lead после анализа требований, программного продукта и специфики конкретной проблемы:

- Blocker — дефект относится к критичной (с точки зрения работоспособности) функциональности или критичным данным. У пользователя нет возможности выполнить целевое действие другими способами.
- Critical — дефект относится к важной (с точки зрения работоспособности) функциональности или важным данным. Пользователь может выполнить целевое действие обходным путем, но он (путь) не очевиден или компания несет финансовые потери.
- Major — дефект относится к не приоритетной (с точки зрения работоспособности) функциональности или не приоритетным данным. Есть очевидный и простой обходной путь выполнения целевой функциональности.
- Minor — дефект не относится напрямую к функциональности и данным. Нет необходимости в обходных путях для выполнения целевого действия. Не влияет на продуктивность, скорость использования приложения.

6.2.2 Серьезность (Severity)

Определяется менеджером (PM/PO) проекта на основании влияния проблемы на бизнес задачи/цели программного продукта:

- Высокий (High): дефект должен быть исправлен как можно скорее, так как он влияет на наиболее критичный с точки зрения бизнеса функционал.
- Средний (Medium): дефект должен быть обязательно исправлен, так как он влияет на важный с точки зрения бизнеса функционал. Однако исправление может быть отложено до ближайших этапов/спринтов разработки.
- Низкий (Low): дефект не обязателен к исправлению, так как он не влияет на важный с точки зрения бизнеса, функционал и ее исправление может быть отложено до момента появления ресурсов.

6.2.3 Серьезность (Severity)

Определяется специалистом по маркетингу или заказчиком проекта на основании анализа поведенческих паттернов конечных пользователей:

- Высокая (High): более 80% конечных пользователей сталкиваются с проблемой.
- Средняя (Medium): менее 80%, но более 30% конечных пользователей сталкиваются с проблемой.
- Низкая (Low): менее 30%, но более 10% конечных пользователей сталкиваются с проблемой.
- Незначительная (Very low): менее 10% конечных пользователей сталкиваются с проблемой.

6.3 Алгоритм определения глобального приоритета

Частота (Frequency) проблемы прямым образом влияет на приоритет. Приоритет (Priority) и серьезность (Severity) прямым образом влияют на глобальный приоритет (Global Severity) дефекта: $globalSeverity = F(Priority, Severity)$, где $Priority = F(BasePriority, Frequency)$. Глобальный приоритет (globalSeverity) определяем по следующему алгоритму:

1. Первоначально изолировано от других факторов определяем серьезность дефекта (Blocker, Critical, Major, Minor).
2. Независимо от серьезности, определяем приоритет дефекта (High, Medium, Low).
3. Независимо от серьезности и приоритета, определяем частоту дефекта (High, Medium, Low, Very low).
4. Далее рассчитываем влияние частоты на первоначально определенный приоритет:

Частота	Изменение приоритета если он имеет высокую частоту
High	Medium → High Low → Medium
Medium	Low → Medium
Low	не изменяется

Very low	не изменяется
----------	---------------

5. Рассчитываем глобальный приоритет:

High	Critical	Critical	Major	Minor
Medium	Critical	Critical	Major	Minor
Low	-	-	Minor	Minor

7 Определение стратегии тестирования

7.1 Условия начала тестирования

Тестирование может быть начато на этапе окончания подготовки требований. Тестировщик изучает требования по задаче, выполняя их тестирование. На этом этапе задача тестировщика привести требования к наиболее понятному виду, чтобы у разработчика не возникали какие-то вопросы по их реализации, чтобы не было в требованиях неоднозначно описанных ситуаций. Т.е. тестировщик, обращаясь к бизнес-аналитику при возникновении вопросов или замечаний по требованиям, старается привести их к наиболее подробному, понятному и исчерпывающему виду для разработчика, с целью сократить его время на разработку и изучение задачи.

Если тестировщик приступает к тестированию уже разработанного функционала, то условием начала тестирования является выполнение всех ниже перечисленных пунктов:

- Статус задач по функционалу для тестирования – Ready for Test
- Функционал выложен на тестовый сервер
- Смоук тестирование показало, что система и ее сервисы работают, нет серьезных ошибок сервера и новый или обновленный функционал доступен для тестирования

7.2 График этапов тестирования

Данный график отображает перечень работ отдела тестирования, которые должны будут быть выполнены в ходе тестирования определенного функционала от начального этапа до завершения тестирования. Часть работ, которая на графике показана как следующий этап по тестированию, может быть выполнена на текущем этапе. Это зависит от количества тестировщиков, выделенных для

тестирования функционала а так же их профессиональной подготовки.

На графике не учтено время, которое тестировщик потратит на тестирование исправленных дефектов.

В зависимости от бюджета, выделенного на проект и времени, выделенного на тестирование, возможен вариант, когда будет принято решение об отмене этапов, связанных с разработкой автотестов или же будет оговариваться процент покрытия автотестами функционала.

	A	B	C	D	E	F	G
		График выполнения тестовых работ по контролю качества					
		функционала регистрации, авторизации, создания списка выборки по группам для персон и компаний					
		Вид работ по тестированию функционала	Этап 1	Этап 2	Этап 3	Этап 4	Этап 5
1		1 Установка программного обеспечения для создания автотестов	1 чел/день				
2		2 Изучение требований по задаче и подготовка чек-листа для ручного тестирования	2 чел/дня				
3		3 Ручное тестирование и создание баг-репортов		2 чел/дня			
4		4 Подготовка тесткейсов и сценариев для разработки автотестов		4 чел/дня			
5		5 Разработка автотестов					
6		6 Заполнение матрицы покрытия тестами			7 чел/дней		
7		7 Выбор автотестов для смоук-тестирования					
8		8 Подборка автотестов для регрессионного тестирования				1 чел/день	
9		9 Проведение регрессионного тестирования и анализ причин, по которым тесты не прошли					0.5 чел/дн. (Зависит от количества зафейленных тестов)
10		10 Создание баг-репортов по результатам регрессионных тестов					В зависимости от количества дефектов

7.3 Критерии окончания тестирования

Критерии, по которым можно сделать вывод, что тестирование закончено и можно делать выкладку на прод среду:

- Функционал удовлетворяет всем требованиям заказчика
- Части тестируемого функционала выполняют все надлежащие операции без ошибок и со скоростью, не меньшей чем оговорено в требованиях заказчика (или же, если в требованиях это не описано, то со скоростью, которая может удовлетворить конечного пользователя (Это изучается так же на этапе нагрузочного тестирования или тестирования производительности))
- Проведено общее тестирование сайта после выкладки нового функционала на тестовую среду и не найдено серьезных или средних по серьезности дефектов в системе. О том, могут ли быть не исправлены мелкие дефекты в данном релизе, вопрос решается с проектным менеджером и продукт-оунером.
- Выполнена выкладка на прод среду и проведенное общее тестирование сайта показало, что сайт работает стабильно и правильно. Новых дефектов не обнаружено.
- Так же могут быть условия, которые оговариваются с заказчиком, например, проведение демо и т.д.

При оформлении дефекта должны быть заполнены следующие поля:

- Приоритет (Срочный, высокий, нормальный, низкий)
- Тема (Краткое название дефекта)
- Описание
- Статус
- Номер спринта
- Фамилия и имя разработчика, который занимается исправлением дефекта или ранее исправил его

8 Определение рисков, которые могут возникнуть при тестировании

При выполнении работ по тестированию вышеописанного функционала могут возникнуть такие риски, что может повлечь за собой больших затрат времени на тестирование или больших трудозатрат:

- Могут возникнуть некоторые проблемы по срокам начала тестирования из-за невыполнения условий начала тестирования – проблема со стороны разработчиков с выкладкой функционала на тестовую среду или же появление дефектов, которые блокируют работу всей системы или тестируемого функционала
- Привлечение к тестированию человека, который не знаком с проектом. Это может привести к увеличению временных затрат а также к увеличению количества невыявленных дефектов. Данные проблемы легко устраняются, если в команде существует человек, не занятый разработкой, но хорошо знающий проект, который может в короткое время объяснить новому человеку в команде некоторые вопросы по проекту, что позволит уменьшить время на изучение задачи и требований тестировщиком и поможет ему в более быстрые сроки приступить к тестированию и не отвлекать команду вопросами и консультациями.
- На этапе усовновки программного обеспечения могут появиться некоторые трудности, связанные с несовместимостью версий инструментов тестирования, которые должны работать совместно. Такие проблемы так же могут привести к лишним растратам времени на тестирование
- Существует риск, что в ходе тестирования будет выявлен дефект, возникший по причине неправильно разработанной архитектуры проекта или структуры данных проекта. В таких случаях иногда разработчики выбирают путь исправления в структуре данных или проекта, что может повлечь за собой необходимость тестировать заново переделанные задачи после того как они уже были протестированы.
- При разработке автотестов есть риск того, что некоторые важные части функционала не смогут быть покрыты автотестами из-за специфики реализации. Например, если при

регистрации нового пользователя проверяется введенный емейл, на принадлежность к фейковым почтовым ящикам, то в таком случае будет затруднительно запускать автотесты по регистрации, так как система будет определять временные емейлы, а реальных емейлов невозможно создать много. В таком случае некоторые тесты придется выполнять вручную, постоянно проходя одни и те же сценарии. Еще возможен вариант, что для тестовой среды разработчики уберут проверки в коде для тестовой среды, которые затрудняют реализацию автотестов.

- Есть риск нехватки персонала, т.е. на тестировщика будут возложены задачи в таком объеме и в такие сроки, что просто физически он не сможет их выполнить
- Есть риск с трудностями воспроизведения некоторых дефектов (чаще прода) из-за недостатка в описании бага, который пришел от пользователя, или же баг может быть воспроизведен на устройстве, которого нет в доступе и нет возможности его найти.