|  |
| --- |
|  |
| План тестирования |
| Версия 1.0 |

|  |
| --- |
| Разработал Куликов В.Н.  25.07.2017 |

Оглавление

[Цель 3](#_Toc488923838)

[Идентификация объектов тестирования 3](#_Toc488923839)

[Стратегия тестирования 3](#_Toc488923840)

[Спецификация тест дизайна 4](#_Toc488923841)

[Используемые методики тест-дизайна 4](#_Toc488923842)

[Проверяемые атрибуты качества. (по ГОСТ Р ИСО/МЭК 9126-93) 4](#_Toc488923843)

[Спецификация требований 4](#_Toc488923844)

[Критерии завершения тестирования 4](#_Toc488923845)

[Критерии завершения тестирования 5](#_Toc488923846)

[Критерии прохождения тест-кейсов 5](#_Toc488923847)

[Модель жизненного цикла дефекта 5](#_Toc488923848)

[Модель серьёзности дефектов 5](#_Toc488923849)

[Модель приоритетов дефектов 6](#_Toc488923850)

[Требование к составу и квалификации участников процесса тестирования 6](#_Toc488923851)

[Необходимые ресурсы 7](#_Toc488923852)

[Календарный план процесса 7](#_Toc488923853)

[Тестовая документация 7](#_Toc488923854)

[Риски и ограничения 8](#_Toc488923855)

[Заключение 8](#_Toc488923856)

# Цель

Цель документа «План тестирования» - предоставление информации о качестве ПО конечному заказчику и повышение качества ПО.

Документ содержит описание общих для подсистем стратегии, подходов и видов тестов. Также определяет численные и квалификационные требования к персоналу, необходимые для успешного тестирования; необходимое программное и аппаратное обеспечение.

# Идентификация объектов тестирования

-программное обеспечение Calculator.java, представляющей собой калькулятор, ограниченного функционала (только с функцией деления). Язык разработки Java SE 8. Приложение разработано в среде IntelliJ IDEA.

-ТЗ к программному обеспечению.

# Стратегия тестирования

В процессе тестирования приложения Calculator будет применено функциональное тестирование, ad-hoc тестирование ввиду отсутствия строгой спецификации и ввиду ограниченности ресурсов на формализацию тестов, модульное тестирование отдельных компонентов приложения.

На первом этапе будет проведено системное тестирование, основанное на корректном сценарии работы пользователя с приложением, описывающих полный цикл работы с приложением. В ходе системного тестирования также планируется выявить usability-недоработки.

На втором этапе, в ходе проведения функционального тестирования, планируется выявить ошибки, основанные на взаимодействии функционалов путем реализации нетривиальных сценариев.

На третьем этапе будут протестированы модули приложения, и их взаимодействие с использованием Junit.

Тестирование нефункциональных требований не предусмотрено.

Цели проведения тестирования:

- убедиться, что пользователь может использовать основные функционалы программы в процессе реального использования;

- проверить степень удобства приложения для пользователя на основании общих критериев и субъективного восприятия;

- выявить функциональные ошибки, основанные на взаимодействии функционалов путем реализации нетривиальных сценариев.

# Спецификация тест дизайна

## Используемые методики тест-дизайна

При составлении тест-кейсов предполагается использование следующих техник:

1. Эквивалентное разделение.
2. Анализ граничных значений.
3. Техники ориентированные на опыт.

## Проверяемые атрибуты качества. (по ГОСТ Р ИСО/МЭК 9126-93)

**Пригодность (Suitability)**  
Атрибут программного обеспечения, относящийся к наличию и соответствию набора функций конкретным задачам.

**Правильность (Accuracy)**  
Атрибуты программного обеспечения, относящиеся к обеспечению правильности или соответствия результатов или эффектов.

**Устойчивость к ошибке (Faulttolerance)**  
Атрибуты программного обеспечения, относящиеся к его способности поддерживать определенный уровень качества функционирования в случаях программных ошибок или нарушения определенного интерфейса.

**Понятность (Understandability)**  
Атрибуты программного обеспечения, относящиеся к усилиям пользователя по пониманию общей логической концепции и ее применимости.

## Спецификация требований

В результате выявления и анализа сформулированы требования, на соответствие которым необходимо проверить предоставленное программное обеспечение. Представлены в документе SRS-1.0.

## Критерии завершения тестирования

Программное обеспечение готово для тестирования, требования к нему сформулированы, документация этапа разработки готова и утверждена.

## Критерии завершения тестирования

Окончание сроков процесса тестирования, задокументированность всех найденных дефектов, готов отчёт о тестировании.

## Критерии прохождения тест-кейсов

Тест-кейс считается пройденным, если он выполняется без дефектов.

## Модель жизненного цикла дефекта

В рамках данного проекта используемая модель жизненного цикла дефекта представлена на рисунке.

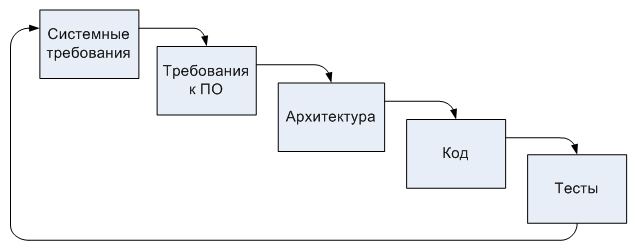


Рис. 1 «Модель жизненного цикла дефекта»

## Модель серьёзности дефектов

Классификация по уровням серьёзности построена на критерии нарушения работоспособности приложения и нарушения бизнес-требований. Предусмотрены следующие уровни серьёзности:

**S1 Блокирующая (Blocker)**  
Блокирующая ошибка, приводящая приложение в нерабочее состояние, в результате которого дальнейшая работа с тестируемой системой или ее ключевыми функциями становится невозможна. Решение проблемы необходимо для дальнейшего функционирования системы.

**S2 Критическая (Critical)**  
Критическая ошибка, неправильно работающая ключевая бизнес логика, дыра в системе безопасности, проблема, приведшая к временному падению сервера или приводящая в нерабочее состояние некоторую часть системы, без возможности решения проблемы, используя другие входные точки. Решение проблемы необходимо для дальнейшей работы с ключевыми функциями тестируемой системой.

**S3 Значительная (Major)**  
Значительная ошибка, часть основной бизнес логики работает некорректно. Ошибка не критична или есть возможность для работы с тестируемой функцией, используя другие входные точки.

**S4 Незначительная (Minor)**  
Незначительная ошибка, не нарушающая бизнес логику тестируемой части приложения, очевидная проблема пользовательского интерфейса.

**S5 Тривиальная (Trivial)**  
Тривиальная ошибка, не касающаяся бизнес логики приложения, плохо воспроизводимая проблема, малозаметная посредствам пользовательского интерфейса, проблема сторонних библиотек или сервисов, проблема, не оказывающая никакого влияния на общее качество продукта.

## Модель приоритетов дефектов

Классификация по уровням приоритетов основана на критерии срочности исправления, вероятности появления данного дефекта и уровня наносимого уровня. Предусмотрены следующие уровни приоритетов.

**P1 Высокий (High)**  
Ошибка должна быть исправлена как можно быстрее, т.к. ее наличие является критической для проекта.

**P2 Средний (Medium)**  
Ошибка должна быть исправлена, ее наличие не является критичной, но требует обязательного решения.

**P3 Низкий (Low)**  
Ошибка должна быть исправлена, ее наличие не является критичной, и не требует срочного решения.

# Требование к составу и квалификации участников процесса тестирования

Тест-менеджер – руководит организацией процесса тестирования, организует взаимодействием между участниками процесса, отвечает за составление стратегии и плана тестирования.

Тест-аналитик – отвечает за выявление и анализ требований, отвечает за выявление в ошибках в требованиях и документации, определяет спецификацию тест-дизайна, проводить анализ результатов тестирования.

Тест-дизайнер отвечает за составление тест-кейсов, их обновление, оптимизацию и актуализацию.

Тест-инженер отвечает за реализацию тест-кейсов, документирует дефекты.

Администратор тестового окружения- отвечает за настройку тестового стенда

Системный администратор- отвечает за настройку рабочей системы.

Примечание. В рамках тестирования рассматриваемого приложения возможно осуществление различных ролей одним человеком.

# Необходимые ресурсы

Ввиду простоты программного обеспечения достаточно будет одного ресурса, ответственного за весь процесс тестирования продукта. Конфигурации тестового стенда: Операционная система Windows 10, ноутбук на базе процессора Intel i3, JVM version 1.8.0\_031, JUnit 4.12.

# Календарный план процесса

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Этап** | **Дата начала** | **Дата окончание** | **Продолжительность** | **Ответственная роль** |
| Составление плана тестирования | 25.07.2017 | 25.07.2017 | 1 час | Тест- менеджер |
| Выявление и анализ требований | 25.07.2017 | 25.07.2017 | 0,5 часа | Тест-аналитик |
| Спецификация тест-дизайна | 25.07.2017 | 25.07.2017 | 0,5 часа | Тест-аналитик |
| Спецификация тестовых наборов данных | 25.07.2017 | 25.07.2017 | 0,5 часа | Тест-аналитик |
| Составление тест-кейсов | 25.07.2017 | 25.07.2017 | 1 час | Тест-дизайнер |
| Настройка рабочей системы и тестового стенда | 26.07.2017 | 26.07.2017 | 0,1 часа | Администратор тестового окружения |
| Проведение тест-кейсов и документирование дефектов | 26.07.2017 | 26.07.2017 | 1 час | Тест-инженер |
| Написание отчёта о тестирование и анализ процесса тестирования | 26.07.2017 | 26.07.2017 | 1 час | Тест аналитик. |

# Тестовая документация

1. План тестирования
2. Набор тест-кейсов (файл TC.docx)
3. Спецификация требований к ПО SRS-1.0 (файл SRS-1.0.docx)
4. Набор JUnit тест-кейсов (файл CalculatorTest.java)
5. Отчёт о тестировании (файл Test Results - CalculatorTest.html)

# Риски и ограничения

В силу недостатка времени и недостаточности информации по проекту риски и ограничения процесса тестирования не сформулированы.

# Заключение

Результатом проведение тестирования является отчёт о тестирование. Отчёт должен быть написан в общедоступной форме с привлечением стандартной и общепринятой терминологии. Адресаты: менеджер проекта, архитектор, руководитель группы разработчиков, системные аналитики.