  
МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
**«Дальневосточный федеральный университет»**

**ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ  
Департамент программной инженерии и искусственного интеллекта**

**РАЗРАБОТКА ИГРОВОГО ДВИЖКА С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИЙ КОЛЛЕКТИВНОЙ ПРОМЫШЛЕННОЙ РАЗРАБОТКИ**

КУРСОВОЙ ПРОЕКТ  
по дисциплине «Технологии коллективной промышленной разработки информационных систем» по образовательной программе подготовки бакалавров по направлению 09.03.04 «Программная инженерия»

Выполнили:  
студенты гр. Б9120-09.03.04прогин  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Рольщиков М.Е.  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Мамедов Ш.М.  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Фролов А.А.  
Руководитель:  
старший преподаватель департамента ПИиИИ  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Иваненко В.М.

**Оглавление**

[Введение 3](#_Toc124728865)

[1 План проекта 4](#_Toc124728866)

[2 Регламент проведения инспекции 5](#_Toc124728867)

[3 Модель состояний задач 8](#_Toc124728868)

[4 Презентация проекта 10](#_Toc124728869)

[5 Требований к проекту 13](#_Toc124728870)

[6 Разработка архитектуры проекта 15](#_Toc124728871)

[7 Измерения проекта 19](#_Toc124728872)

[8 Перечень задач проекта 21](#_Toc124728873)

[9 Правила по кодированию 24](#_Toc124728874)

[10 Разработка плана тестирования проекта 29](#_Toc124728875)

[11 Тестирование проекта 36](#_Toc124728876)

[Заключение 45](#_Toc124728877)

[Список литературы 46](#_Toc124728878)

# Введение

Промышленная разработка информационных систем включает в себя множество этапов, начиная от разработки плана проекта, заканчивая его тестированием, для чего необходимо множество специалистов различных профилей, а также унифицированные методы коммуникации между ними, с помощью которых можно разделить обязанности членов команды по их специализации.

Исходя из описанного выше, необходимо использовать определенные технологии коллективной разработки для повышения эффективности работы в группе и соответствия конечного продукта заявленным требованиям.

В данной курсовой работе рассматривается задача коллективной разработки программного средства «Игровой движок» и составление технической документации к нему.

Таким образом, целью курсовой работы является разработка программного средства с использованием подходов коллективной промышленной разработки.

Для достижения поставленной цели необходимо решить следующие задачи:

1. Разработать план проекта.
2. Разработать регламент проведения инспекции.
3. Разработать модель состояний задач.
4. Разработать презентацию проекта.
5. Разработать требования к проекту.
6. Разработать архитектуру проекта.
7. Разработать измерения проекта.
8. Разработать перечень задач проекта.
9. Разработать рекомендации по кодированию.
10. Разработать план тестирования проекта.
11. Протестировать проект.

# 1 План проекта

План проекта – это документ, содержащий подробную информацию о проекте: исполнителях, задачах и сроках. Документ является конечным результатом этапа планирования, утверждается до начала любых работ и становится самым главным и достоверным источником информации о грядущем проекте.

В нашем случае исполнителями являются следующие лица:

* Team Leader – Рольщиков Максим;
* Coder 1 – Рольщиков Максим;
* Coder 2 – Фролов Анатолий;
* Technical Writer – Мамедов Шамиль.

На рисунке 1 представлен перечень задач для выполнения и примерные сроки их реализации.

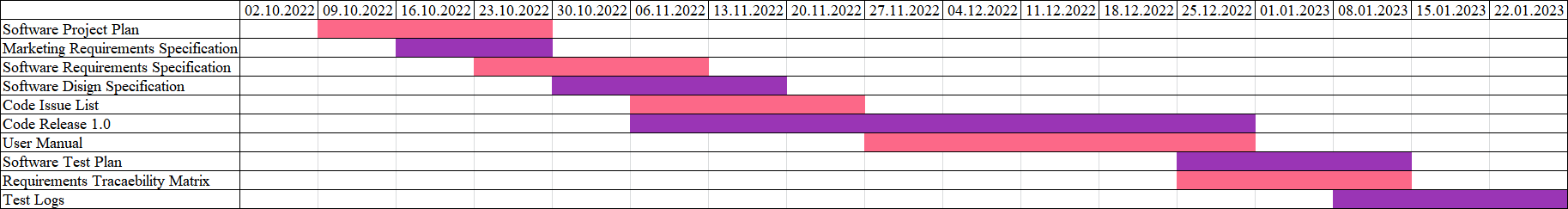


Рисунок 1 – План проекта

# 2 Регламент проведения инспекции

Верификация рабочих продуктов является неотъемлемой частью процесса по обеспечению их качества. Современной технологией программирования выработаны специальные стандарты, подходы и механизмы проведения верификаций рабочих продуктов, в формате так называемых инспекций.

Инспекция – это мероприятие по обеспечению качества рабочих продуктов проектов по разработке ПО и иной деятельности, которая проводится разработчиками, возможно – с участием представителей заказчика.

Концептуально инспекция имеет следующие цели:

* обнаружить ошибки в функциях, логике, содержании или реализации рабочих продуктов на ранних этапах их разработки и предотвратить их наследование;
* рационально донести замысел или реализацию продукта до всех заинтересованных лиц (через их участие);
* оптимизировать, оценить или улучшить рабочий продукт.

## Критерии отнесения к формальной/неформальной инспекции для различных типов рабочих продуктов

|  |  |  |
| --- | --- | --- |
|  | **Формальная инспекция в случае CRUD-операций над** | **Неформальная инспекция в случае CRUD-операций над** |
| **Требования** | >1 требованием | 1 требованием |
| **Документы дизайна** | >1 объектом дизайна внешнего вида готового продукта | 1 объектом дизайна внешнего вида готового продукта (изображение) |
| **Код** | >7 строчками | <8 строчками |
| **Тесты** | >3 тестами | < 4тестами |

## Перечень ролей участников инспекции и их обязанности

Автор (Author) – сотрудник, разработавший инспектируемый рабочий продукт, либо сделавший инспектируемые изменения в существующем рабочем продукте.

Инспектор (Inspector) – сотрудник, ответственный за эффективную проверку инспектируемого рабочего продукта.

## Этапы инспекции

## Планирование инспекции.

При планировании инспекции коллективно выбирается дата, время, формат (очный или заочный) и платформа (при заочной инспекции) проведения инспекции.

## Подготовка к инспекции.

Инспектор самостоятельно изучает предоставленный для инспекции рабочий продукт, используя накопленный опыт и стандарты.

## Собрание по инспекции.

На собрании происходит обсуждение замечаний и рекомендаций инспектора по рабочему продукту. На собрании по инспекции обязательно присутствует инспектор и автор рабочего продукта, требующего инспекции. Присутствие остальных участников команды разработки по желанию.

## Завершение инспекции.

Если рабочий продукт требует доработки, то автор фиксирует все замечания и рекомендации инспектора, разрабатывает план предстоящих работ и далее согласовывает его с инспектором. Если рабочий продукт не требует доработки, то инспектор подтверждает слияние рабочей ветки в «**main**» ветку.

## Перечень статусов и степени важности замечаний

1. Ошибка – проблема, которая найдена на той же фазе, на которой внесена. Допустимые значения степени серьёзности ошибки:

* Критическая (Critical)
* Средняя (Moderate)
* Мелкая, незначительная (Minor)
* Другие (Other)

1. Комментарий – это наблюдение, предложение, рекомендация или улучшение, предложенное для будущего выпуска рабочего продукта или вопрос, требующий разъяснения.

## Метрики, характеризующие эффективность инспекций

Стратегическая цель метрики – повысить качество разрабатываемого ПО.

В качестве метрики, характеризующей эффективность инспекции, была выбрана Inspection Preparation Rate (IPR):

IPR = (Количество инспекторов \* Размер продукта) / Общее время подготовки

Изучаемый объект метрики – подготовка к инспекции, измеряемый атрибут – производительность подготовки к инспекции.

Единица измерения – <страница, требование, LOC, тест> / час

# 3 Модель состояний задач

Каждая задача, являясь отражением делового процесса, проходит определенные состояния. Сначала идет создание задачи, потом идет выполнение работ по задаче, после выполнения задача завершается.

## Перечень состояний задач:

1. Ожидание – задача создана и находится в состоянии ожидания, над ней в данный момент никто не работает.
2. В работе – в данный момент задача находится в работе.
3. Тестирование – происходит тестирования внесенных изменений.
4. Закрыта – задача выполнена.

## Правила создания новой задачи

Созданием новых задач могут заниматься все участники команды. Происходит это на начальном этапе, когда необходимо организовать структурированную работу над проектом и в процессе работы над проектом в случае присутствия в плане слишком объемных задач, требующих упрощения. Также, когда необходимо организовать баг-фиксинг или разработать новый функционал.

## Правила перехода задачи из состояния в состояние

Состояния задач всегда идут последовательно друг за другом, в некоторых случаях пункты повторяться. Схема перехода из состояния в состояние показана на рисунке 2.

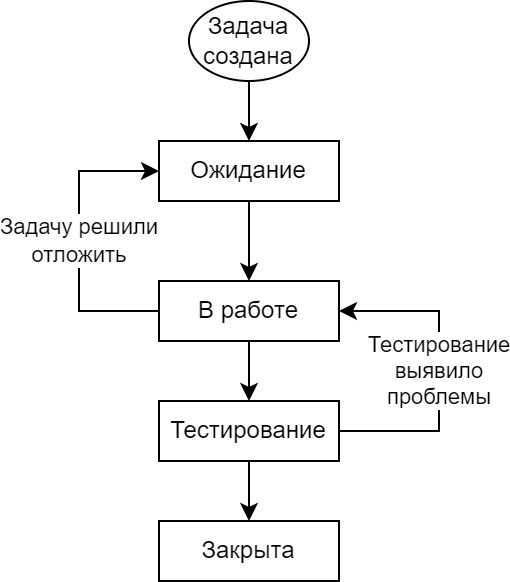


Рисунок 2 – Состояния задачи

# 4 Презентация проекта

На рисунке 3 представлена титульная страница презентации.

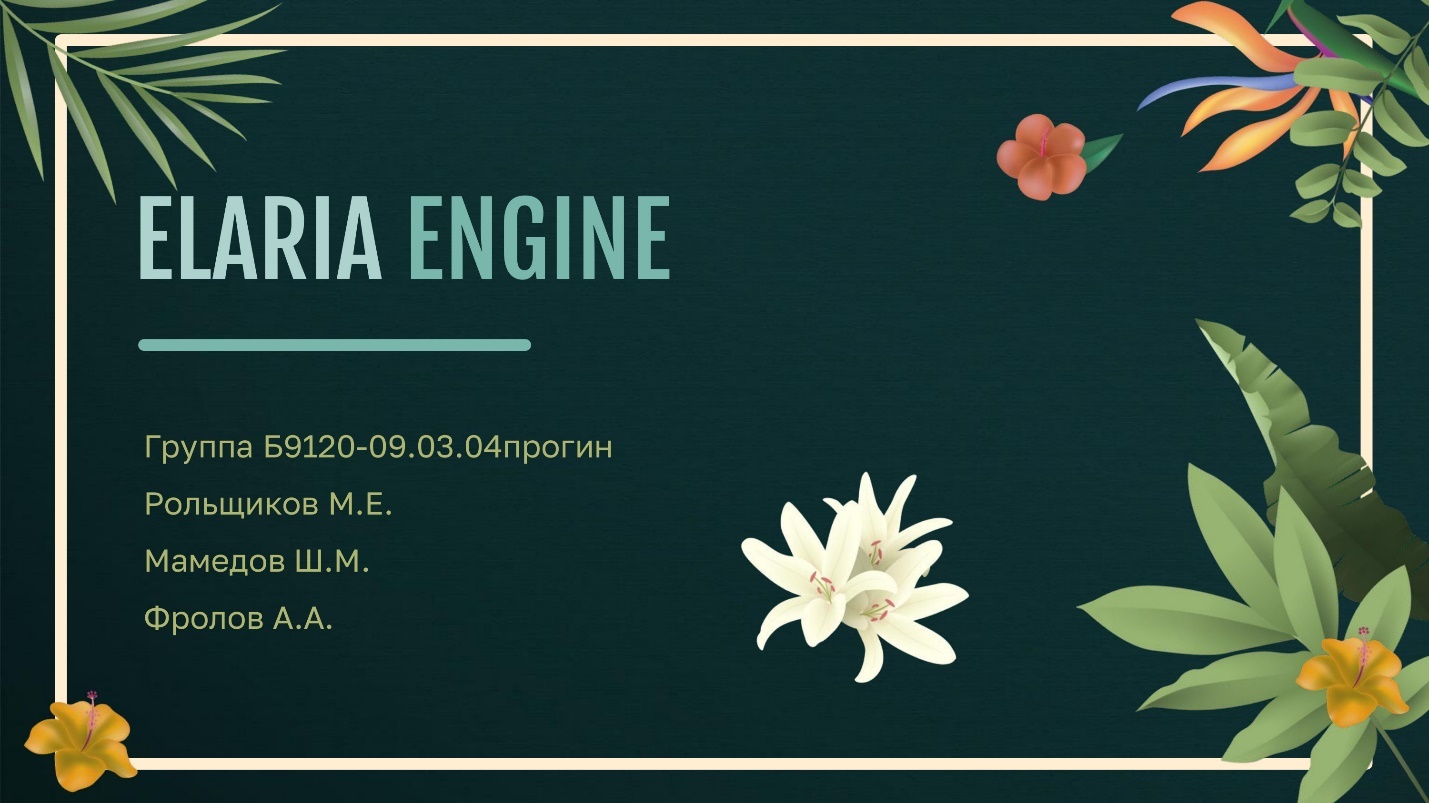


Рисунок 3 – Титульная страница

Проблемы, возникающие в предметной области разрабатываемого средства, представлены на рисунке 4.

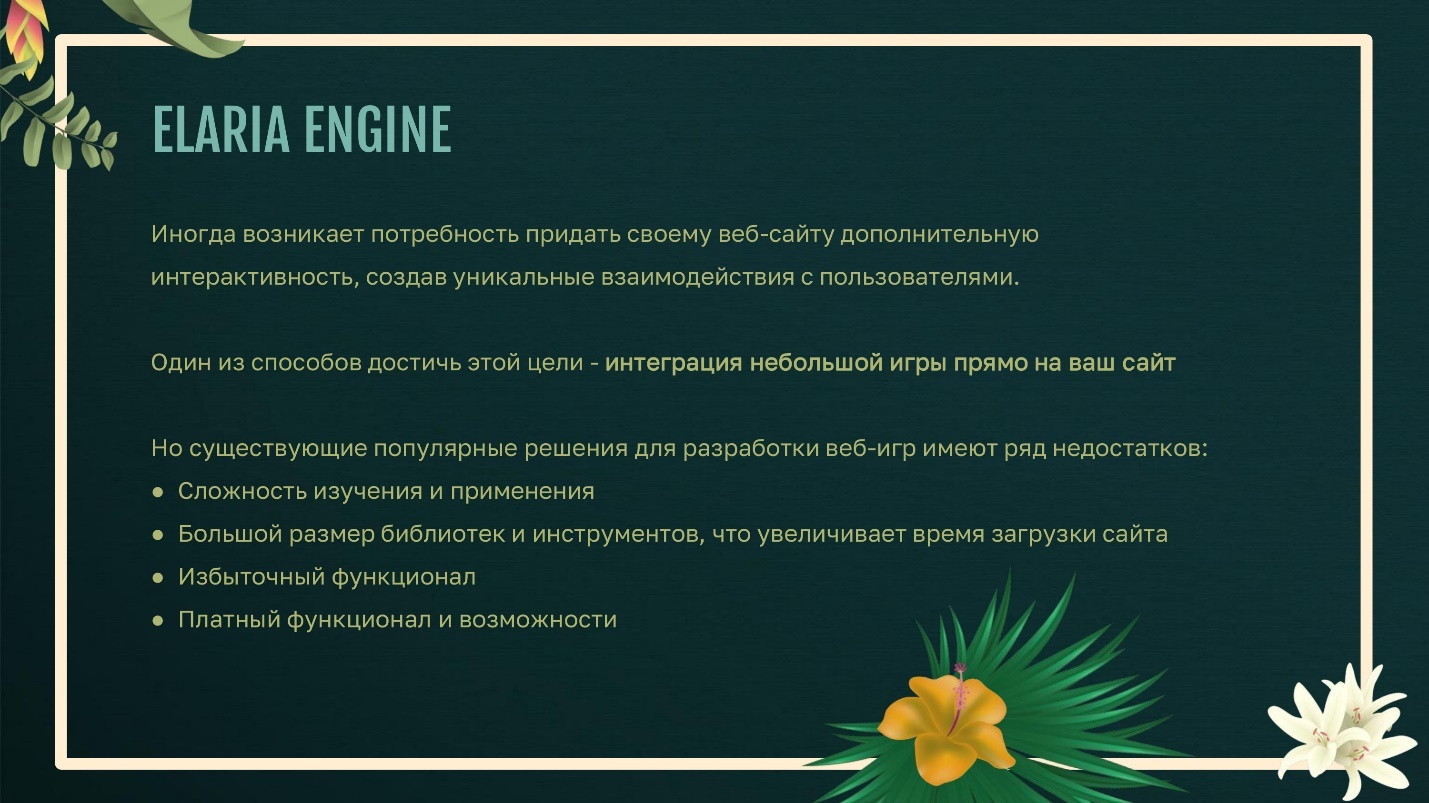


Рисунок 4 – Проблемы

На рисунке 5 демонстрируется, что можно получить и как, используя программное средство.

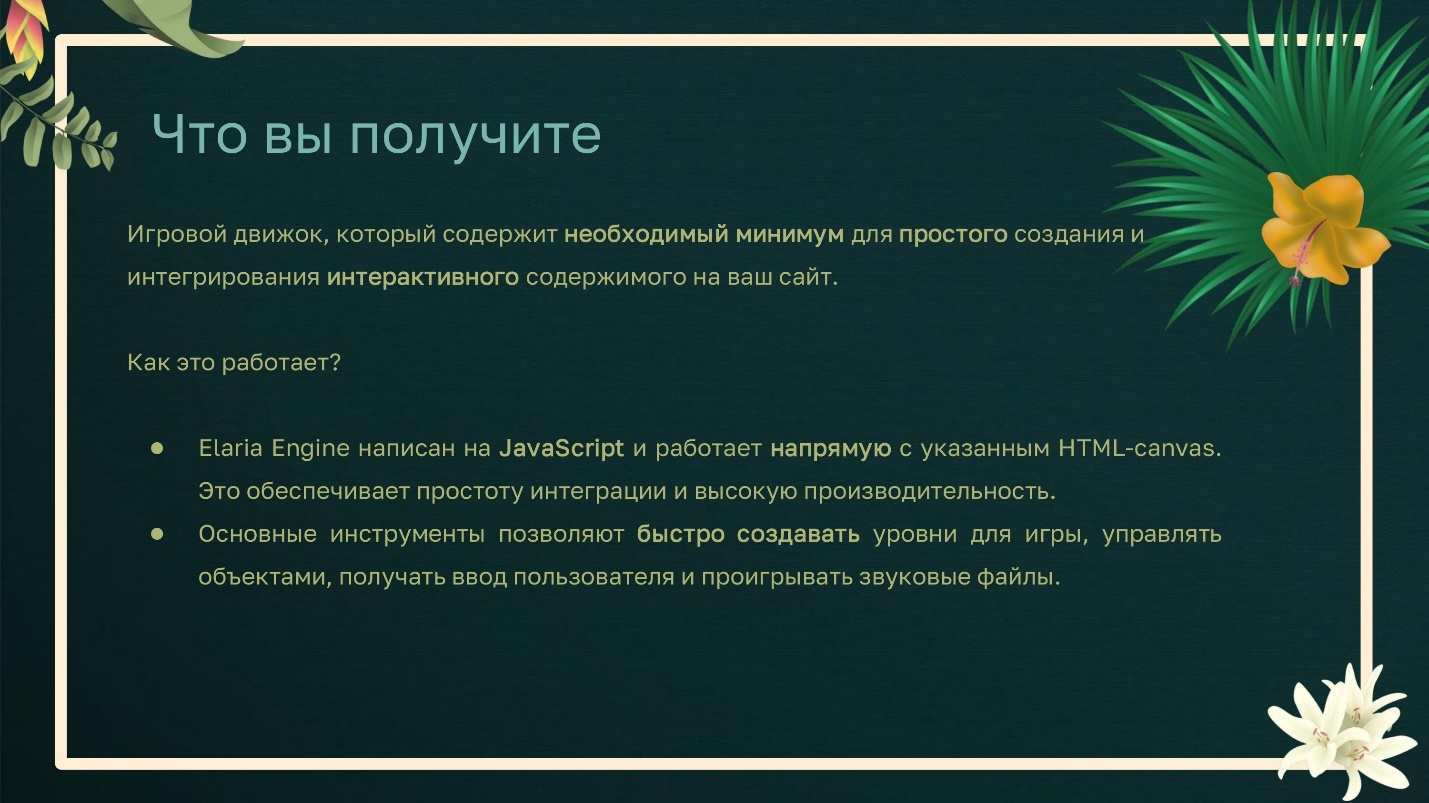


Рисунок 5 – Принцип работы

Внешний вид разрабатываемого средства представлен на рисунке 6.

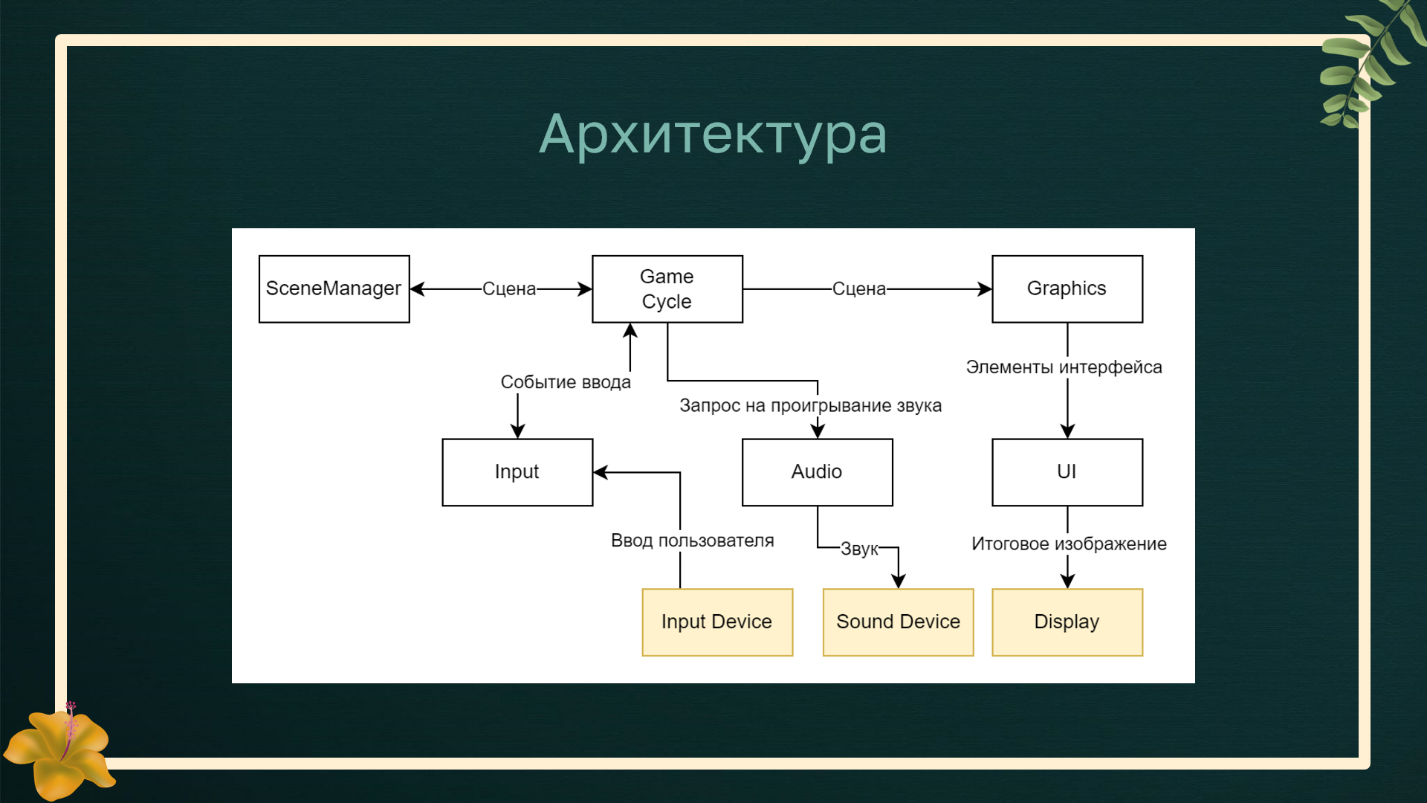


Рисунок 6 – Дизайн программного средства

Актуальность разрабатываемого средства представлена на рисунке 7.

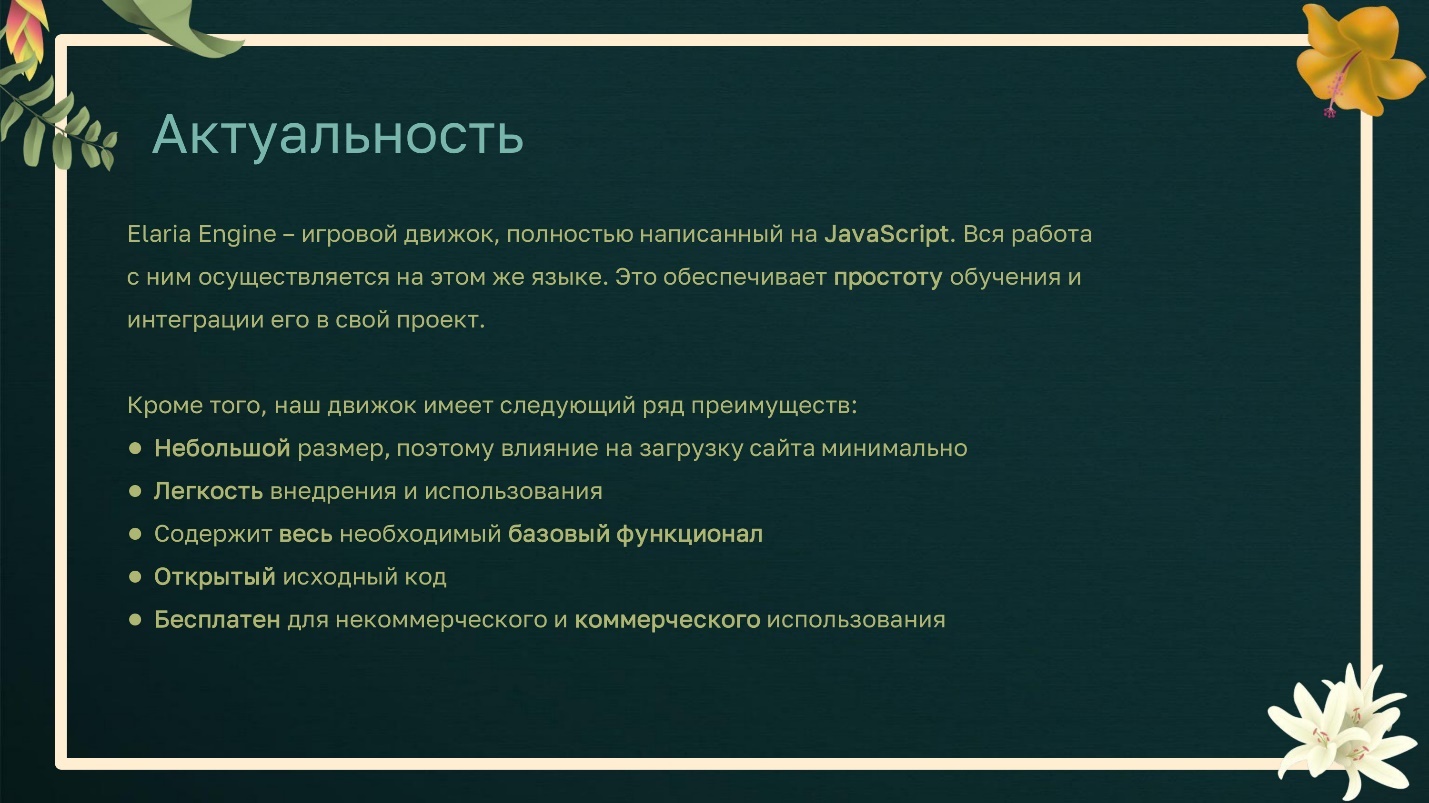


Рисунок 7 – Актуальность

# 5 Требования к проекту

Программный продукт «Игровой движок» предназначен для создания интерактивных элементов на сайте, с которыми пользователь может взаимодействовать.

Программный продукт «Игровой движок» состоит из следующих подсистем:

1. FE-1 – Ядро движка.
2. FE-2 – Графический модуль.
3. FE-3 – Модуль аудио.
4. FE-4 – Модуль получения пользовательского ввода.
5. FE-5 – Модуль пользовательского интерфейса.
6. FE-6 – Менеджер сцен.

## Матрица требований к подсистемам

|  |  |
| --- | --- |
| **Матрица требований** | **Идентификаторы** |
| Требования к ядру движка | FE-1 |
| Ядро движка должно загружать начальную сцену |  |
| Ядро движка должно запускать игровой цикл |  |
| Ядро движка должно позволять создавать и удалять игровые объекты на сцене |  |
| Ядро движка должно позволять добавлять и удалять компоненты игровым объектам |  |
| Требования к графическому модулю | FE-2 |
| Графический модуль должен определять компонент Sprite. Sprite должен отрисовывать определенное изображение. |  |
| Графический модуль должен определять компонент AnimatedSprite. AnimatedSprite должен отрисовывать последовательность изображений с указанной частотой |  |
| Требования к аудио модулю | FE-3 |
| Модуль аудио должен определять компонент AudioSource. Компонент AudioSource должен проигрывать переданный аудиофайл |  |
| Требования к модулю получения пользовательского ввода | FE-4 |
| Модуль пользовательского ввода должен уведомлять о событии нажатия клавиш клавиатуры |  |
| Модуль пользовательского ввода должен уведомлять о событии нажатия кнопок мыши |  |
| Модуль пользовательского ввода должен сохранять положение курсора |  |
| Требования к модулю пользовательского интерфейса | FE-5 |
| Модуль пользовательского интерфейса должен реализовывать компонент Button. Компонент Button обрабатывает событие нажатие левой кнопкой мыши на графический объект |  |
| Модуль пользовательского интерфейса должен реализовывать компонент Text. Компонент Text, который выводит текст с определенным шрифтом и размером |  |
| Требования к менеджеру сцен | FE-6 |
| Менеджер сцен должен загружать переданную сцену |  |
| Менеджер сцен должен автоматически выгружать текущую сцену после загрузки новой |  |

## Расшифровка идентификаторов

* UA – User Account (учетная запись пользователя)
* BP – Body Parameters (параметры тела пользователя)
* PD – Personal Data (личные данные пользователя)
* C – Catalog (каталог)
* CL – Comparison List (список сравнения)
* SP – Schema Pattern (схема выкройки)
* SM – Schema Measures (мерки схемы)
* UI – User Interface (пользовательский интерфейс)

# 6 Разработка архитектуры проекта

Архитектура программного обеспечения относится к фундаментальным структурам программной системы и дисциплине создания таких структур и систем. Каждая структура включает элементы программного обеспечения, отношения между ними, а также свойства как элементов, так и отношений.

Архитектура программной системы – это метафора, аналогичная архитектуре здания. Он функционирует как план для системы и проекта разработки, в котором излагаются задачи, которые должны быть выполнены командами разработчиков.

## Концептуальная модель

Концептуальная модель – это совокупность взаимосвязанных понятий, лежащих в основе исследовательского дизайна, системное описание исследуемой области. Концептуальная модель наглядно описывает структуру моделируемой предметной области и связи между ее элементами.

Концептуальная модель помогает решить, какие переменные наиболее важны, какие связи наиболее значимы и, следовательно, какую информацию необходимо собирать.

Схема концептуальной модели показана на рисунке 8.

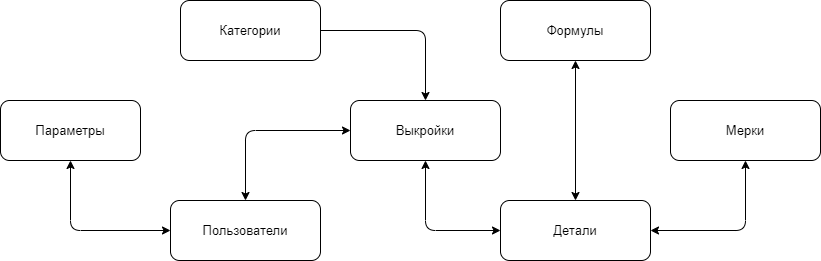


Рисунок 8 – Концептуальная модель

## Логическая модель

Целью построения логической модели является получение графического представления логической структуры исследуемой предметной области.

Логическая модель предметной области иллюстрирует сущности, а также их взаимоотношения между собой.

Схема данной модели представлена на рисунке 9.

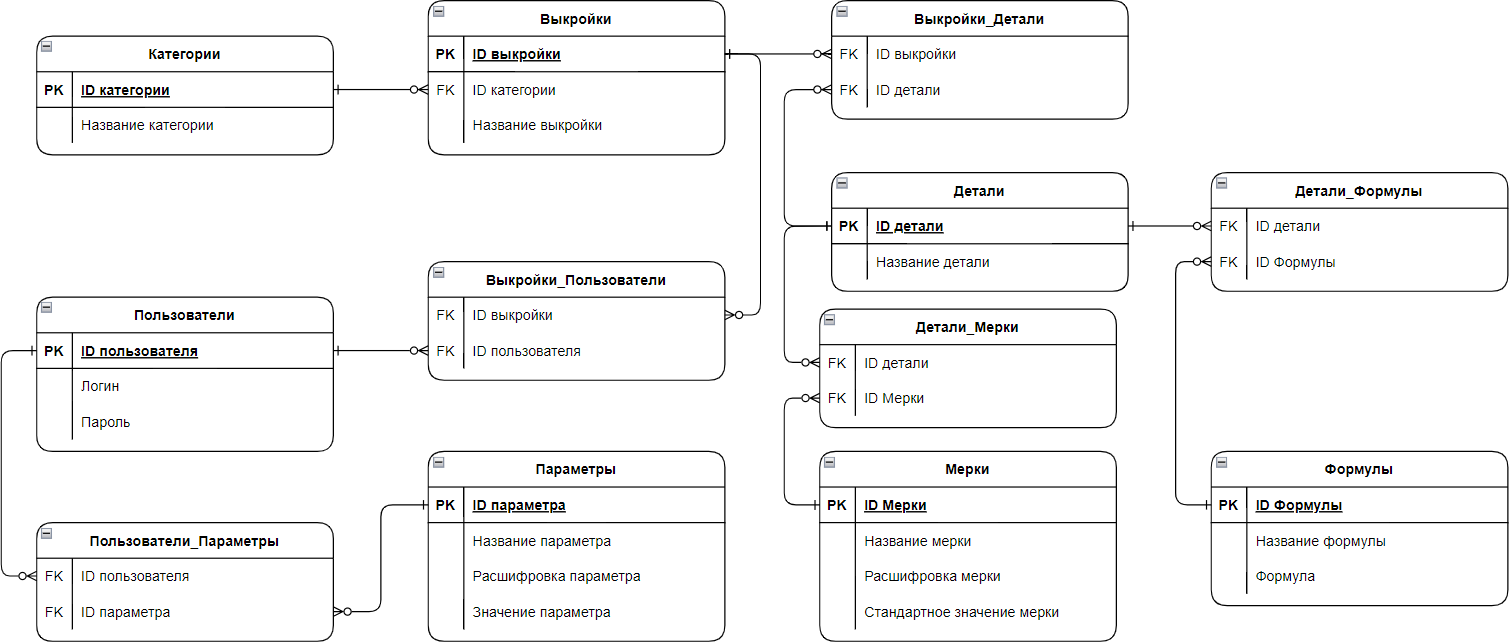


Рисунок 9 – Логическая модель

## Контекстная диаграмма потоков данных

Одним из способов представления архитектуры проекта является диаграмма потоков данных.

Диаграммы потоков данных (Data Flow Diagrams – DFD) представляют собой иерархию функциональных процессов, связанных потоками данных.

Цель такого представления – продемонстрировать, как каждый процесс преобразует свои входные данные в выходные, а также выявить отношения между этими процессами.

На рисунке 10 представлена диаграмма потоков данных нулевого уровня между подсистемами разрабатываемой системы.

****

Рисунок 10 – DFD0

На рисунке 11 представлена диаграмма потоков данных первого уровня между подсистемами разрабатываемой системы для модуля построения выкройки по параметрам.

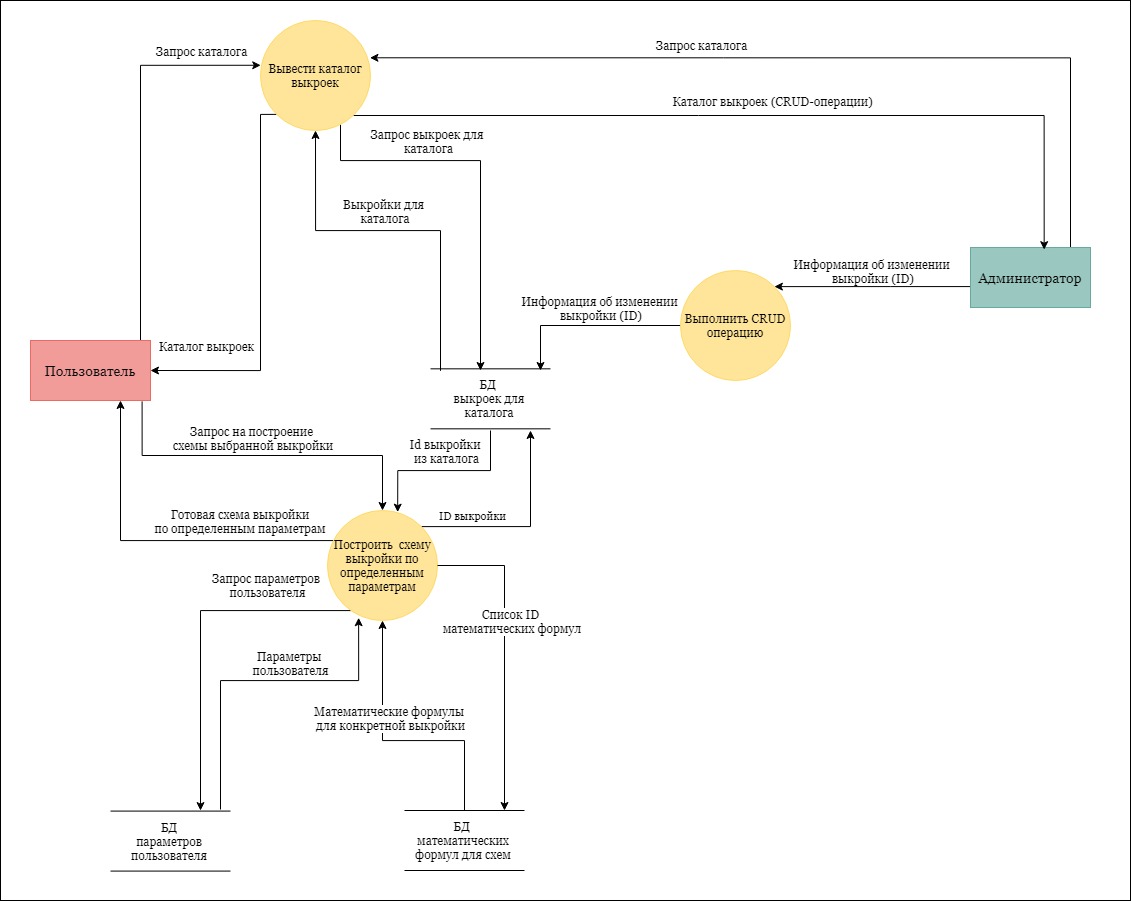
****

Рисунок 11 – DFD1

# 7 Измерения проекта

Контроль над производственным процессом и его результатами является ключевым видом деятельности на современном предприятии, производящем программное обеспечение на заказ. В силу специфики такого продукта, как программное обеспечение, для оценки эффективности процесса и качества конечного продукта применяются особые методы.

Комплекс мероприятий, направленных на количественную оценку эффективности работы, называется программой измерений.

## Метрика эффективности процесса производства

1. **Problem Resolution Rate (PRR)**

PRR = Количество дней на обработку задачи

Стратегическая цель метрики – сократить сроки выполнения проектов по разработке ПО.

Изучаемый объект метрики – задача, измеряемый атрибут – время обработки.

Единица измерения – день.

1. **Inspection Fault Density (IFD)**

IFD = (Количество найденных ошибок / Размер рабочего продукта)

Стратегическая цель метрики – повысить качество разрабатываемого ПО.

Изучаемый объект метрики – инспекция, измеряемый атрибут – плотность найденных в ходе инспекции ошибок.

Количество найденных ошибок – ошибки, обнаруженные на этапе инспектирования.

Единица измерения – ошибка / <страница, требование**,** LOC, тест>.

## Метрика качества продукта

1. **In Process Faults (IPF)**

IPF = (Число обнаруженных ошибок до выпуска его релиза) / LOC

Стратегическая цель метрики – повысить качество разрабатываемого ПО.

Изучаемый объект метрики – продукт, измеряемый атрибут – плотность неполадок.

Число обнаруженных ошибок – количество ошибок обнаруженных на этапе тестирования

Единица измерения – неполадка / LOC.

# 8 Перечень задач проекта

## ПОДСИСТЕМА FE-1

**TASK-1 UA**

**Учетная запись пользователя**

До 07.11.2022 реализовать предоставления доступа к личной информации пользователя, объединенной в совокупность данных в БД – REQ-UA-1 (см. Software Requirement Specification).

**TASK-2 BP**

**Параметры тела пользователя**

До 07.11.2022 реализовать предоставления доступа к параметрам тела в учетной записи, объединенных в совокупность данных в БД, редактирования и удаления совокупности/части совокупности данных в БД, представляющей параметры тела пользователя – REQ-BP-1 и REQ-BP-2 (см. Software Requirement Specification).

**TASK-3 PD**

**Личные данные пользователя**

До 07.11.2022 реализовать предоставления доступа к личным данным пользователя в учетной записи, объединенных в совокупность данных в БД и редактированию совокупности/части совокупности данных в БД – REQ-PD-1 (см. Software Requirement Specification).

## ПОДСИСТЕМА FE-2

**TASK-1 C**

***Каталог выкроек***

До 21.11.2022 реализовать предоставления доступа к совокупности данных в БД, представляющей выкройки системы и поиска части совокупности данных в БД, представляющей выкройки системы, по определенным критериям – REQ-C-1, REQ-C-2 и REQ-C-3 (см. Software Requirement Specification).

## ПОДСИСТЕМА FE-3

**TASK-1 CL**

***Список сравнения выкроек***

До 05.12.2022 реализовать предоставления доступа к списку сравнения, добавления в него части совокупности данных в БД, представляющей выкройки системы и удаления из него совокупности/части совокупности данных в БД, представляющей выкройки системы – REQ-CL-1, REQ-CL-2 и REQ-CL-3 (см. Software Requirement Specification).

## ПОДСИСТЕМА FE-4

**TASK-1 SM**

***Мерки схемы выкройки***

До 19.12.2022 реализовать добавление, редактирования и удаление совокупности/части совокупности данных в БД, представляющей значения мерок для построения схемы выкройки – REQ-SM-1, REQ-SM-2 и REQ-SM-3 (см. Software Requirement Specification).

**TASK-2 SP**

***Схема выкройки***

До 02.01.2023 реализовать предоставления доступа к построенной схеме выкройки, а также сохранение ее на локальное хранилище – REQ-SP-1 и REQ-SP-2 (см. Software Requirement Specification).

## ПОДСИСТЕМА FE-5

**TASK-1 UI**

***Интерфейс***

До 16.01.2023 реализовать веб-интерфейс для работы с системой. Интерфейс должен включать в себя элементы, указанных выше подсистем – REQ-UI -1 (см. Software Requirement Specification).

# 9 Правила по кодированию

Для создания качественного кода на любом языке программирования, обладающего таким свойствами, как удобочитаемость (readability) и понятность (understandability), необходимо следовать хорошо определённым стандартам и руководствам. Особенно это актуально при коллективной разработке программ.

Любой стандарт кодирования призван определить набор правил, которые способствуют разработке более единообразного кода и минимизации числа общераспространенных ошибок в нем, не ущемляя при этом права разработчика на творчество.

## Рекомендации

1. Комментарии должны быть законченными предложениями.

Если комментарий – фраза или предложение, первое слово должно быть написано с большой буквы, если только это не имя переменной, которая начинается с маленькой буквы (никогда не отступайте от этого правила для имен переменных).

Ставьте два пробела после точки в конце предложения. Следует использовать английский язык для написания комментариев.

1. Использование метода range.

Если есть необходимость пройти в цикле по ряду чисел, то метод range будет намного приемлемее, как минимум потому, что этот метод потребляет намного меньше памяти, чем вариант в блоке «Плохо».

Неправильно:

for i in [0, 1, 2, 3, 4, 5]:

print(i\*\*2)

Правильно:

for i in range(6):

print(i\*\*2)

1. Каждый импорт должен быть на отдельной строке.

Неправильно:

import sys, os

Правильно:

import os

import sys

1. Не используйте составные инструкции (несколько команд в одной строке).

Неправильно:

if foo == 'blah': do-blah-thing()

do-one(); do-two(); do-three()

Правильно:

if foo == 'blah':

do-blah-thing()

do-one()

do-two()

do-three()

## Запреты

1. Запрещено внутри блока для вложенных инструкций использовать разные отступы.

Неправильно:

def long-function-name(var-one, var-two):

print(var-one)

print(var-two)

Правильно:

def long-function-name(var-one, var-two):

print(var-one)

print(var-two)

1. При программировании на Python запрещается использовать пробелы.

* Непосредственно внутри круглых, квадратных или фигурных скобок.

Неправильно:

spam( ham[ 1 ], { eggs: 2 }

Правильно:

spam(ham[1], {eggs: 2})

* Непосредственно перед запятой, точкой с запятой или двоеточием:

Неправильно:

**if** x == 4 : print(x , y) ; x , y = y , x

Правильно:

**if** x == 4: print(x, y); x, y = y, x

* Сразу перед открывающей скобкой, после которой начинается список аргументов при вызове функции.

Неправильно:

spam (1)

Правильно:

spam(1)

* Сразу перед открывающей скобкой, после которой следует индекс или срез.

Неправильно:

dict ['key'] = list [index]

Правильно:

dict['key'] = list[index]

1. Запрещается использовать объявления импорта где-то кроме верхней части файла.

Они должны находится только после любых комментариев модуля и символов документации, а также перед объявлениями глобальных переменных и констант.

## Требования

1. Наследуйте свой класс исключения от Exception, а не от BaseException.

Прямое наследование от BaseException зарезервировано для исключений, которые не следует перехватывать.

1. Имена абстрактных классов (интерфейсов) дополняются префиксом I (ISocket), напротив – классы реализации могут снабжаться постфиксом Imp (SocketImp) или Impl (при использовании [идиомы Pimpl](https://pro-prof.com/forums/topic/cplusplus_pimpl_pattern)).

Класс в составе иерархии может содержать *префикс C* (CFigure), а классы исключений – *постфикс Exception* (BadArgumentsException). Есть множество других вариантов, которые часто противоречат друг другу.

1. Имена функций дополняются префиксами.

* *is-* – проверяет что-то и возвращает логический тип – is-digit;
* *has-* – выполняет поиск какого-либо значения в контейнере – has-primeNumber;
* *get-* и *set-* – метод возвращает или устанавливает значение какого-либо поля – set-volume, set-volume;

1. Используются схемы именования, позволяющие различать объекты и типы данных.

Имена пользовательских типов нужно начинать с заглавной буквы, а имена объектов – со строчной.

1. Для названий требуется использовать слова из предметной области решаемой задачи.

Имена классов должны выражаться существительными, функций и методов – глаголами. Имена классов должны быть хорошо различимы.

Неправильно:

x = 'Иван Петров'

y, z = x.split()

Правильно:

name = 'Иван Петров'

first-name, last-name = name.split()

# 10 Разработка плана тестирования проекта

## ПОДСИСТЕМА FE-1

* **TEST\_UA\_1**

Тестируемые требования: REQ-UA-1

Описание теста: Нажать на кнопку «Личный кабинет»

Ожидаемый результат: Переход на страницу личного кабинета с учетными данными пользователя

* **TEST\_BP\_1**

Тестируемые требования: REQ-UA-1, REQ-BP-1

Описание теста: Ввести новые параметры тела пользователя и сохранить изменения

Ожидаемый результат: Параметры были отредактированы

* **TEST\_BP\_2**

Тестируемые требования: REQ-UA-1, REQ-BP-1

Описание теста: Ввести слишком большие числа в параметры тела пользователя и сохранить изменения

Ожидаемый результат: Неверные параметры не были сохранены, поля ввода неправильных параметров стали красными

* **TEST\_BP\_3**

Тестируемые требования: REQ-UA-1, REQ-BP-1

Описание теста: Ввести не числовые значения в параметры тела пользователя и сохранить изменения

Ожидаемый результат: Неверные параметры не были сохранены, поля ввода неправильных параметров стали красными

* **TEST\_BP\_4**

Тестируемые требования: REQ-UA-1, REQ-BP-2

Описание теста: Удалить некоторые параметры тела пользователя и сохранить изменения

Ожидаемый результат: Параметры тела, выделенные ранее, были очищены. Поля ввода стали пустыми.

* **TEST\_PD\_1**

Тестируемые требования: REQ-UA-1, REQ-PD-1

Описание теста: Ввести новые личные данные пользователя и сохранить изменения

Ожидаемый результат: Личные данные были отредактированы

## ПОДСИСТЕМА FE-2

* **TEST\_C\_1**

Тестируемые требования: REQ-C-1

Описание теста: Нажать на кнопку «Поиск выкроек»

Ожидаемый результат: Переход на страницу каталога выкроек

* **TEST\_C\_2**

Тестируемые требования: REQ-C-1, REQ-C-2, REQ-C-3

Описание теста: Нажать на кнопку «Показать», не выбрав ни одного параметра

Ожидаемый результат: Вывод всех выкроек в каталоге

* **TEST\_C\_3**

Тестируемые требования: REQ-C-1, REQ-C-2

Описание теста: Нажать на кнопку «Показать», выбрав некоторые категории

Ожидаемый результат: Вывод всех выкроек в каталоге с выбранными категориями

* **TEST\_C\_4**

Тестируемые требования: REQ-C-1, REQ-C-3

Описание теста: Нажать на кнопку «Показать», введя интервал деталей

Ожидаемый результат: Вывод всех выкроек в каталоге с выбранным интервалом деталей

## ПОДСИСТЕМА FE-3

* **TEST\_CL\_1**

Тестируемые требования: REQ-CL-1, REQ-CL-2

Описание теста: Выбрать некоторые выкройки для сравнения и нажать на кнопку «Сравнить выкройки»

Ожидаемый результат: Переход на страницу сравнения ранее выбранных выкроек

* **TEST\_CL\_2**

Тестируемые требования: REQ-CL-1, REQ-CL-2

Описание теста: Выбрать больше 6 выкроек для сравнения и нажать на кнопку «Сравнить выкройки»

Ожидаемый результат: Переход на страницу сравнения ранее выбранных выкроек, вывод надписи «Выбрано больше 6 выкроек»

* **TEST\_CL\_2**

Тестируемые требования: REQ-CL-1, REQ-CL-2, REQ-CL-3

Описание теста: Выбрать некоторые выкройки для сравнения, а потом снова нажать на выбор выкройки, а затем на кнопку «Сравнить выкройки»

Ожидаемый результат: Переход на страницу сравнения ранее выбранных выкроек, выбранные дважды выкройки не показываются

## ПОДСИСТЕМА FE-4

* **TEST\_SM\_1**

Тестируемые требования: REQ-SM-1

Описание теста: Выбрать выкройку и нажать «Ввести параметры для построения», затем в пустые поля ввода ввести значения параметров

Ожидаемый результат: Переход на страницу построения выкройки со схемой выкройки по введенным ранее параметрам

* **TEST\_SM\_2**

Тестируемые требования: REQ-SM-2

Описание теста: Выбрать выкройку и нажать «Ввести параметры для построения», затем в непустые поля ввода ввести значения параметров

Ожидаемый результат: Переход на страницу построения выкройки со схемой выкройки по введенным ранее параметрам

* **TEST\_SM\_3**

Тестируемые требования: REQ-SM-3

Описание теста: Выбрать выкройку и нажать «Ввести параметры для построения», оставить некоторые поля ввода параметров пустыми

Ожидаемый результат: Переход на страницу построения выкройки, вывод надписи «Введены некорректные данные»

* **TEST\_SM\_4**

Тестируемые требования: REQ-SM-3

Описание теста: Выбрать выкройку и нажать «Ввести параметры для построения», ввести в некоторые поля ввода слишком большие числа

Ожидаемый результат: Переход на страницу построения выкройки, вывод надписи «Введены некорректные данные»

* **TEST\_SM\_5**

Тестируемые требования: REQ-SM-3

Описание теста: Выбрать выкройку и нажать «Ввести параметры для построения», ввести в некоторые поля ввода буквы

Ожидаемый результат: Переход на страницу построения выкройки, вывод надписи «Введены некорректные данные»

* **TEST\_SP\_1**

Тестируемые требования: REQ-SP-1

Описание теста: Выбрать выкройку и нажать «Ввести параметры для построения», ввести значения параметров тела

Ожидаемый результат: Переход на страницу построения выкройки со схемой выкройки по введенным ранее параметрам

* **TEST\_SP\_2**

Тестируемые требования: REQ-SP-1, REQ-SP-2

Описание теста: На странице построения выкройки нажать на кнопку «Скачать изображение»

Ожидаемый результат: Сохранение на локальное хранилище схемы выкройки

## ПОДСИСТЕМА FE-5

* **TEST\_SYS\_1**

Тестируемые требования: REQ-UI-1, REQ-UA-1, REQ-BP-1, REQ-PD-1, REQ-C-1, REQ-C-2, REQ-C-3, REQ-CL-1, REQ-CL-2, REQ-SM-1, REQ-SM-2, REQ-SP-1, REQ-SP-2

Описание теста: Нажать на кнопку перехода в личный кабинет, ввести новые личные данные и новые параметры и сохранить изменения, перейти на вкладку каталога выкроек, выбрать категории «Платья» и «Рубашки», ввести интервал деталей от 1 до 5, выбрать «Платье-футляр» и «Пляжная рубашка» для сравнения, нажать на кнопку сравнения выкроек, на пляжной рубашке нажать «Выбрать для построения» и «Ввести параметры для построения», ввести недостающие параметры, нажать на кнопку «Построить», нажать на кнопку «Скачать изображение»

Ожидаемый результат: Были представлены все страницы системы, отредактированы данные, был произведен поиск по каталогу, построение выкройки, редактирование данных выкройки, сравнение выкроек, сохранение на локальное хранилище схемы выкройки

* **TEST\_SYS\_2**

Тестируемые требования: REQ-UI-1, REQ-UA-1, REQ-C-1, REQ-C-2, REQ-C-3, REQ-CL-1, REQ-CL-2, REQ-SM-1, REQ-SM-2, REQ-SP-1, REQ-SP-2

Описание теста: Перейти на вкладку каталога выкроек, выбрать категории «Брюки» и «Юбки», ввести интервал деталей от 2 до 4, выбрать «Брюки карго» и «Брюки скинни» для сравнения, нажать на кнопку сравнения выкроек, на брюках скинни нажать «Выбрать для построения» и «Ввести параметры для построения», ввести недостающие параметры, нажать на кнопку «Построить», нажать на кнопку «Скачать изображение»

Ожидаемый результат: Были представлены все страницы системы, отредактированы данные, был произведен поиск по каталогу, построение выкройки, редактирование данных выкройки, сравнение выкроек, сохранение на локальное хранилище схемы выкройки

## Матрица покрытия тестами требований

Матрица соответствия требований используется QA-инженерами для валидации покрытия требований по продукту тестами.

Цель «Traceability Matrix» состоит в том, чтобы выяснить:

* какие требования «покрыты» тестами, а какие нет;
* избыточность тестов (одно функциональное требование покрыто большим количеством тестов).

В соответствии с написанными требованиями и тестами на рисунке 12 представлена матрица покрытия тестами требований.

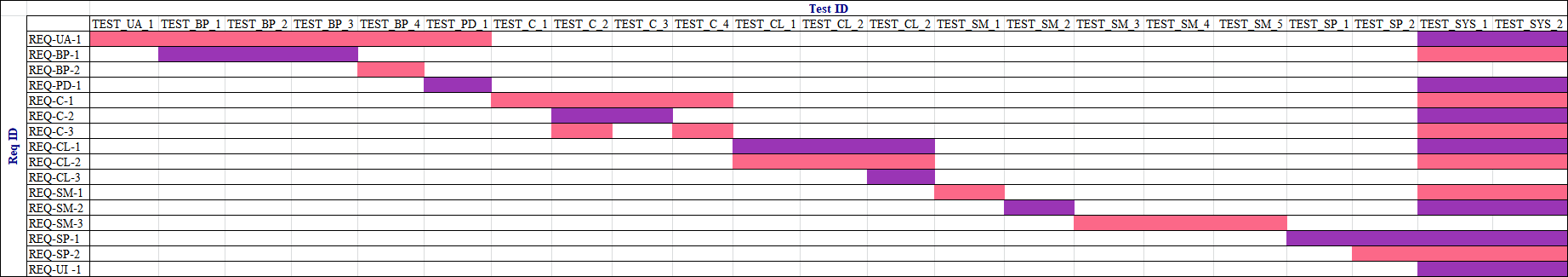


Рисунок 12 – Матрица покрытия тестами требований

# 11 Тестирование проекта

## ПОДСИСТЕМА FE-1

* **TEST\_UA\_1**

Тестируемые требования: REQ-UA-1

Описание теста: Нажать на кнопку «Личный кабинет»

Ожидаемый результат: Переход на страницу личного кабинета с учетными данными пользователя

Видимый результат: Полностью совпадает с ожидаемым

Резюме: Тест пройден

* **TEST\_BP\_1**

Тестируемые требования: REQ-UA-1, REQ-BP-1

Описание теста: Ввести новые параметры тела пользователя и сохранить изменения

Ожидаемый результат: Параметры были отредактированы

Видимый результат: Полностью совпадает с ожидаемым

Резюме: Тест пройден

* **TEST\_BP\_2**

Тестируемые требования: REQ-UA-1, REQ-BP-1

Описание теста: Ввести слишком большие числа в параметры тела пользователя и сохранить изменения

Ожидаемый результат: Неверные параметры не были сохранены, поля ввода неправильных параметров стали красными

Видимый результат: Полностью совпадает с ожидаемым

Резюме: Тест пройден

* **TEST\_BP\_3**

Тестируемые требования: REQ-UA-1, REQ-BP-1

Описание теста: Ввести не числовые значения в параметры тела пользователя и сохранить изменения

Ожидаемый результат: Неверные параметры не были сохранены, поля ввода неправильных параметров стали красными

Видимый результат: Полностью совпадает с ожидаемым

Резюме: Тест пройден

* **TEST\_BP\_4**

Тестируемые требования: REQ-UA-1, REQ-BP-2

Описание теста: Удалить некоторые параметры тела пользователя и сохранить изменения

Ожидаемый результат: Параметры тела, выделенные ранее, были очищены. Поля ввода стали пустыми.

Видимый результат: Полностью совпадает с ожидаемым

Резюме: Тест пройден

* **TEST\_PD\_1**

Тестируемые требования: REQ-UA-1, REQ-PD-1

Описание теста: Ввести новые личные данные пользователя и сохранить изменения

Ожидаемый результат: Личные данные были отредактированы

Видимый результат: Полностью совпадает с ожидаемым

Резюме: Тест пройден

## ПОДСИСТЕМА FE-2

* **TEST\_C\_1**

Тестируемые требования: REQ-C-1

Описание теста: Нажать на кнопку «Поиск выкроек»

Ожидаемый результат: Переход на страницу каталога выкроек

Видимый результат: Полностью совпадает с ожидаемым

Резюме: Тест пройден

* **TEST\_C\_2**

Тестируемые требования: REQ-C-1, REQ-C-2, REQ-C-3

Описание теста: Нажать на кнопку «Показать», не выбрав ни одного параметра

Ожидаемый результат: Вывод всех выкроек в каталоге

Видимый результат: Полностью совпадает с ожидаемым

Резюме: Тест пройден

* **TEST\_C\_3**

Тестируемые требования: REQ-C-1, REQ-C-2

Описание теста: Нажать на кнопку «Показать», выбрав некоторые категории

Ожидаемый результат: Вывод всех выкроек в каталоге с выбранными категориями

Видимый результат: Полностью совпадает с ожидаемым

Резюме: Тест пройден

* **TEST\_C\_4**

Тестируемые требования: REQ-C-1, REQ-C-3

Описание теста: Нажать на кнопку «Показать», введя интервал деталей

Ожидаемый результат: Вывод всех выкроек в каталоге с выбранным интервалом деталей

Видимый результат: Полностью совпадает с ожидаемым

Резюме: Тест пройден

## ПОДСИСТЕМА FE-3

* **TEST\_CL\_1**

Тестируемые требования: REQ-CL-1, REQ-CL-2

Описание теста: Выбрать некоторые выкройки для сравнения и нажать на кнопку «Сравнить выкройки»

Ожидаемый результат: Переход на страницу сравнения ранее выбранных выкроек

Видимый результат: Полностью совпадает с ожидаемым

Резюме: Тест пройден

* **TEST\_CL\_2**

Тестируемые требования: REQ-CL-1, REQ-CL-2

Описание теста: Выбрать больше 6 выкроек для сравнения и нажать на кнопку «Сравнить выкройки»

Ожидаемый результат: Переход на страницу сравнения ранее выбранных выкроек, вывод надписи «Выбрано больше 6 выкроек»

Видимый результат: Полностью совпадает с ожидаемым

Резюме: Тест пройден

* **TEST\_CL\_2**

Тестируемые требования: REQ-CL-1, REQ-CL-2, REQ-CL-3

Описание теста: Выбрать некоторые выкройки для сравнения, а потом снова нажать на выбор выкройки, а затем на кнопку «Сравнить выкройки»

Ожидаемый результат: Переход на страницу сравнения ранее выбранных выкроек, выбранные дважды выкройки не показываются

Видимый результат: Полностью совпадает с ожидаемым

Резюме: Тест пройден

## ПОДСИСТЕМА FE-4

* **TEST\_SM\_1**

Тестируемые требования: REQ-SM-1

Описание теста: Выбрать выкройку и нажать «Ввести параметры для построения», затем в пустые поля ввода ввести значения параметров

Ожидаемый результат: Переход на страницу построения выкройки со схемой выкройки по введенным ранее параметрам

Видимый результат: Полностью совпадает с ожидаемым

Резюме: Тест пройден

* **TEST\_SM\_2**

Тестируемые требования: REQ-SM-2

Описание теста: Выбрать выкройку и нажать «Ввести параметры для построения», затем в непустые поля ввода ввести значения параметров

Ожидаемый результат: Переход на страницу построения выкройки со схемой выкройки по введенным ранее параметрам

Видимый результат: Полностью совпадает с ожидаемым

Резюме: Тест пройден

* **TEST\_SM\_3**

Тестируемые требования: REQ-SM-3

Описание теста: Выбрать выкройку и нажать «Ввести параметры для построения», оставить некоторые поля ввода параметров пустыми

Ожидаемый результат: Переход на страницу построения выкройки, вывод надписи «Введены некорректные данные»

Видимый результат: Полностью совпадает с ожидаемым

Резюме: Тест пройден

* **TEST\_SM\_4**

Тестируемые требования: REQ-SM-3

Описание теста: Выбрать выкройку и нажать «Ввести параметры для построения», ввести в некоторые поля ввода слишком большие числа

Ожидаемый результат: Переход на страницу построения выкройки, вывод надписи «Введены некорректные данные»

Видимый результат: Полностью совпадает с ожидаемым

Резюме: Тест пройден

* **TEST\_SM\_5**

Тестируемые требования: REQ-SM-3

Описание теста: Выбрать выкройку и нажать «Ввести параметры для построения», ввести в некоторые поля ввода буквы

Ожидаемый результат: Переход на страницу построения выкройки, вывод надписи «Введены некорректные данные»

Видимый результат: Полностью совпадает с ожидаемым

Резюме: Тест пройден

* **TEST\_SP\_1**

Тестируемые требования: REQ-SP-1

Описание теста: Выбрать выкройку и нажать «Ввести параметры для построения», ввести значения параметров тела

Ожидаемый результат: Переход на страницу построения выкройки со схемой выкройки по введенным ранее параметрам

Видимый результат: Полностью совпадает с ожидаемым

Резюме: Тест пройден

* **TEST\_SP\_2**

Тестируемые требования: REQ-SP-1, REQ-SP-2

Описание теста: На странице построения выкройки нажать на кнопку «Скачать изображение»

Ожидаемый результат: Сохранение на локальное хранилище схемы выкройки

Видимый результат: Полностью совпадает с ожидаемым

Резюме: Тест пройден

## ПОДСИСТЕМА FE-5

* **TEST\_SYS\_1**

Тестируемые требования: REQ-UI-1, REQ-UA-1, REQ-BP-1, REQ-PD-1, REQ-C-1, REQ-C-2, REQ-C-3, REQ-CL-1, REQ-CL-2, REQ-SM-1, REQ-SM-2, REQ-SP-1, REQ-SP-2

Описание теста: Нажать на кнопку перехода в личный кабинет, ввести новые личные данные и новые параметры и сохранить изменения, перейти на вкладку каталога выкроек, выбрать категории «Платья» и «Рубашки», ввести интервал деталей от 1 до 5, выбрать «Платье-футляр» и «Пляжная рубашка» для сравнения, нажать на кнопку сравнения выкроек, на пляжной рубашке нажать «Выбрать для построения» и «Ввести параметры для построения», ввести недостающие параметры, нажать на кнопку «Построить», нажать на кнопку «Скачать изображение»

Ожидаемый результат: Были представлены все страницы системы, отредактированы данные, был произведен поиск по каталогу, построение выкройки, редактирование данных выкройки, сравнение выкроек, сохранение на локальное хранилище схемы выкройки

Видимый результат: Полностью совпадает с ожидаемым

Резюме: Тест пройден

* **TEST\_SYS\_2**

Тестируемые требования: REQ-UI-1, REQ-UA-1, REQ-C-1, REQ-C-2, REQ-C-3, REQ-CL-1, REQ-CL-2, REQ-SM-1, REQ-SM-2, REQ-SP-1, REQ-SP-2

Описание теста: Перейти на вкладку каталога выкроек, выбрать категории «Брюки» и «Юбки», ввести интервал деталей от 2 до 4, выбрать «Брюки карго» и «Брюки скинни» для сравнения, нажать на кнопку сравнения выкроек, на брюках скинни нажать «Выбрать для построения» и «Ввести параметры для построения», ввести недостающие параметры, нажать на кнопку «Построить», нажать на кнопку «Скачать изображение»

Ожидаемый результат: Были представлены все страницы системы, отредактированы данные, был произведен поиск по каталогу, построение выкройки, редактирование данных выкройки, сравнение выкроек, сохранение на локальное хранилище схемы выкройки

Видимый результат: Полностью совпадает с ожидаемым

Резюме: Тест пройден

# Заключение

В рамках курсовой работы было разработано программное средство «Игровой движок» с использованием подходов коллективной промышленной разработки, для чего были решены следующие поставленные задачи:

1. Разработан план проекта.
2. Разработан регламент проведения инспекции.
3. Разработана модель состояний задач.
4. Разработана презентация проекта.
5. Разработаны требования к проекту.
6. Разработана архитектура проекта.
7. Разработаны измерения проекта.
8. Разработан перечень задач проекта.
9. Разработаны рекомендации по кодированию.
10. Разработан план тестирования проекта.
11. Протестирован проект.

Таким образом, цель данной курсовой работы была достигнута.

По окончанию всех работ посчитаем метрики, описанные в 7 главе. Эффективность процесса разработки программного средства – 7 дней работы. Качество продукта выразилось в 28% плотности неполадок.

# Список литературы

1. Гриняк В.М. Лекции по дисциплине «Технологии коллективной промышленной разработки информационных систем». Электронный вариант.