  
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 ПРОГРАММНОЙ СИСТЕМЫ ПОДБОРА РАСТЕНИЙ ДЛЯ ДОМА**

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  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Кучапина С.С.  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Орлов Г.А. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Петров Ф.А. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Романова И.С.  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Чащин Т.А.  
Руководитель:  
ассистент департамента ПИиИИ  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Иваненко Ю.С.

г. Владивосток,  
2023

**Оглавление**

[**Введение** 4](#_Toc156642253)

[**1.** **План проекта** 5](#_Toc156642254)

[**2.** **Регламент проведения инспекции** 6](#_Toc156642255)

[**2.1** **Виды инспекций** 6](#_Toc156642256)

[**2.2 Участники инспекции** 7](#_Toc156642257)

[**2.1.** **Этапы инспекции** 7](#_Toc156642258)

[**2.1.1.** **Планирование инспекции.** 7](#_Toc156642259)

[**2.1.2.** **Назначение инспекции.** 8](#_Toc156642260)

[**2.1.3.** **Подготовка к инспекции.** 9](#_Toc156642261)

[**2.1.4.** **Собрание по инспекции.** 9](#_Toc156642262)

[**2.1.5.** **Завершение инспекции.** 9](#_Toc156642263)

[**2.2.** **Перечень статусов и степени важности замечаний** 10](#_Toc156642264)

[**2.3.** **Метрики, характеризующие эффективность инспекций** 10](#_Toc156642265)

[**3** **Модель состояний задач** 11](#_Toc156642266)

[**3.1** **Перечень состояний задач** 11](#_Toc156642267)

[**3.2** **Правила создания новой задачи** 12](#_Toc156642268)

[**3.3** **Правила перехода задач из состояния в состояние** 12](#_Toc156642269)

[**4** **Презентация проекта** 15](#_Toc156642270)

[**5** **Требования к проекту** 18](#_Toc156642271)

[**5.1** **Требования к пользовательскому интерфейсу** 18](#_Toc156642272)

[**5.2** **Требования к модулю подбора растений, по введённым характеристикам** 19](#_Toc156642273)

[**5.3** **Требования к базе данных растений** 19](#_Toc156642274)

[**5.4** **Модуль запросов к БД** 19](#_Toc156642275)

[**5.5** **Требования к модулю хостинга онлайн-сервиса** 20](#_Toc156642276)

[**5.6** **Нефункциональные требования** 20](#_Toc156642277)

[**6** **Архитектуры проекта** 21](#_Toc156642278)

[**7** **Измерения проекта** 23](#_Toc156642279)

[**Заключение** 25](#_Toc156642280)

# **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 «Программная система подбора растений для дома» и составление технической документации к данному средству.

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.

# **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.

Исполнителями данного проекта являются следующие лица:

* Team Leader – Орлов Георгий Александрович;
* Designer – Кучапина Софья Сергеевна;
* Coder 1 – Орлов Георгий Александрович;
* Coder 2 – Петров Филипп Александрович;
* Coder 3 – Чащин Трофим Александрович;
* Build Engineer – Орлов Георгий Александрович;
* Technical Writer – Романова Ирина Сергеевна.

На Рисунке 1 представлены перечень задач для выполнения и примерные сроки их реализации.

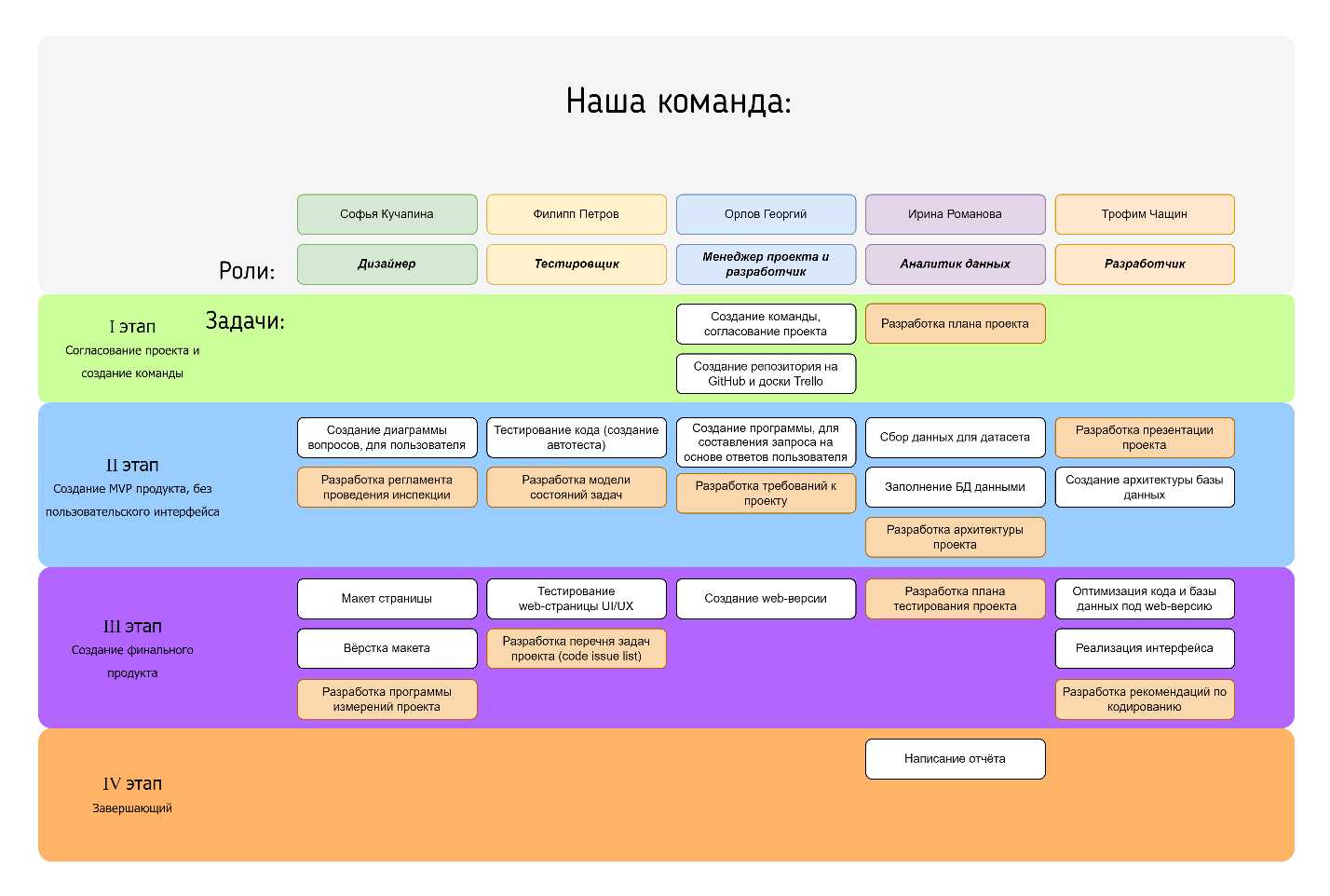


Рисунок 1 – План проекта

1. **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 (peer reviews).

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.
  1. **Виды инспекций**

Существует несколько разновидностей инспекций. В самом общем случае их подразделяют на формальные (проводимые по специальному регламенту) и неформальные (проводимые в рабочем порядке) инспекции. Критерием в пользу проведения того или иного типа инспекции служит размер инспектируемого рабочего продукта. Данный критерий по отношению к формальным инспекциям может быть, например, следующим:

Таблица 1 - Критерии инспекции

|  |  |
| --- | --- |
| **Рабочий продукт** | **Размер продукта или изменения в нем, не менее** |
| Требования | 5 страниц или 10% рабочего продукта |
| Документы по дизайну | 5 страниц или 10% рабочего продукта |
| Не комментированный код | 50 NCLOC |
| Тесты (Test Cases) | 5 страниц или 10% рабочего продукта |

**2.2 Участники инспекции**

Участники инспекции могут иметь следующие роли:

1. Автор (Author) - сотрудник, разработавший инспектируемый рабочий продукт, либо сделавший инспектируемые изменения в существующем рабочем продукте;
2. Председатель (Moderator) - ответственный сотрудник, выполняющий роль председателя инспекции;
3. Секретарь (Recorder) - сотрудник, ответственный за создание и распространение документации по инспекции;
4. Ведущий (Presenter) - сотрудник, представляющий рабочий продукт инспекторам;
5. Инспектор (Inspector) - сотрудник, ответственный за эффективную проверку инспектируемого рабочего продукта.

Один человек может играть сразу несколько ролей. В случае совмещения ролей рекомендуется, чтобы один человек играл не более двух ролей одновременно. В инспекции должны участвовать не менее 3 человек. При назначении одного человека на несколько ролей следует руководствоваться следующим правилом: не допустимо совмещение ролей Председатель и Автор, Автор и Ведущий, Автор и Секретарь, Автор и Инспектор. Все остальные сочетания ролей допускаются.

* 1. **Этапы инспекции**
     1. **Планирование инспекции.**

Все инспекции на конкретном проекте проводятся согласно принятому на предприятии стандартному процессу, планируются на фазе планирования проекта и фиксируются в документе SQAP (Software Quality Assurance Plan).

Кроме того, при изменении любого продукта должна быть проведена инспекция этих изменений в том случае, если объем этих изменений превышает определенные значения, служащие критериями проведения формальных инспекций.

При планировании инспекции коллективно выбирается дата, время, формат (очный или заочный) и платформа (при заочной инспекции) проведения инспекции. Эти параметры определяются с учетом удобства и доступности участников инспекции, обеспечивая эффективное проведение процесса и обмена необходимой информацией.

Секретарю, ведущему и инспектору необходимо принять или отклонить приглашение на формальную инспекцию с обязательным указанием причин.

* + 1. **Назначение инспекции.**

Автор должен оповестить руководителя проекта (или руководителя группы) о готовности рабочего продукта к формальной инспекции и зафиксировать промежуточную стабильную версию рабочего продукта. С момента предоставления рабочего продукта на формальную инспекцию до собрания включительно автор не имеет права вносить изменения. Также автор должен запросить руководителя проекта об имеющихся ресурсах на роль председателя инспекции. В случае необходимости, провести обзорное собрание, целью которого является ввести участников формальной инспекции в курс дела.

Председатель должен убедиться, что рабочий продукт удовлетворяет критерию готовности к формальной инспекции. Также председателю нужно определить необходимость проведения обзорного собрания и проверить, что все обязательные участники формальной инспекции приняли приглашение либо прислали отказ с обязательным указанием причин и/или предложением своих сроков проведения инспекции. В случае получения хотя бы одного отказа, председатель должен проанализировать сложившуюся ситуацию на предмет того, может ли быть проведена инспекция без отказавшегося участника.

Кроме того, председатель или автор должны обеспечить следующее: 1) определить материально – техническую сторону проведения инспекции, 2) подобрать команду участников и распределить роли, 3) оповестить всех участников формальной инспекции.

* + 1. **Подготовка к инспекции.**

Инспектор самостоятельно независимо от других участников изучает предоставленный для инспекции рабочий продукт, используя накопленный опыт и стандарты. Также ему требуется заполнить необходимые поля протокола подготовки к формальной инспекции и отправить его председателю формальной инспекции и автору.

На основе полученных от инспекторов протоколов подготовки к формальной инспекции, председателю необходимо принять одно из следующих решений: провести, перенести или отменить инспекцию. После принятия решения председатель должен оповестить всех участников формальной инспекции об изменениях.

До собрания автору необходимо ознакомиться с содержанием полученных от инспекторов протоколов подготовки к формальной инспекции, проанализировать изложенные в них замечания.

Секретарь должен проанализировать замечания, зафиксированные инспекторами в протоколах подготовки к формальной инспекции, выявить повторяющиеся.

* + 1. **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.

* + 1. **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 «master» ветку.

* 1. **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.
2. Дефект – проблема, которая найдена на фазе, отличной от той, на которой внесена.
3. Замечание для исследования – проблема, природа которой не может быть определена на собрании и требует дополнительного исследования. В результате дополнительного исследования такая проблема должна получить одно из вышеупомянутых значений статусов.
   1. **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.

# **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.

* 1. **Перечень состояний задач**

1. New – новая подзадача.

2. Analysis – в процессе анализа. В это состояние подзадачу переводит сотрудник после того, как начнёт её анализ.

3. Analysis completed – анализ завершён. Переводится сотрудником после завершения анализа задачи.

4. Terminated – прерванный. В это состояние задача может быть переведена CCB после рассмотрения и принятия решения о необходимости отмены, прекращения работы над задачей.

5. Forward – в данном случае имеет значение «переданный на дальнейшую разработку». В это состояние задача переводится CCB после анализа при назначении задачи на разработку конкретному сотруднику.

6. Coding – кодирование. В это состояние задача переводится сотрудником разработчиком, при начале работы по кодированию, связанному с задачей.

7. Inspected – проинспектировано. В это состояние задача переводится сотрудником-разработчиком после завершения кодирования и инспектирования изменений рабочего продукта.

8. Resolved – проверено. Переводится после проверки изменений рабочего продукта по результатам инспектирования специалистом-экспертом (code expert).

9. Integrated – заинтегрировано. Переводится сотрудником, осуществляющим интеграцию изменений в основную ветку рабочего продукта после успешной интеграции этих изменений.

10. Tested - протестировано. Переводится сотрудником, осуществляющим тестирование изменений в рабочий продукт (tester).

11. Closed – закрыто. В это состояние задача переводится CCB по результатам отчёта о тестировании сделанных изменений.

* 1. **Правила создания новой задачи**
     + 1. *Описание задачи*. Задача должна быть ясно и чётко описана, чтобы каждый член команды мог вникнуть в суть. В описание необходимо включить цели и ожидаемые результаты;
       2. *Требования и критерии завершённости*. Необходимо указать требования к задаче и определить критерии завершённости для чёткой оценки выполнения;
       3. *Приоритет и сроки*. Для эффективной разработки проекта необходимо указать приоритет новой задачи относительно других задач и установить реалистичные сроки выполнения;
       4. *Ресурсы*. Требуется определить необходимые ресурсы и обозначить зависимости от других задач или компонентов проекта;
       5. *Тестирование*. Нужно определить требования к тестированию и проверке новой задачи и указать возможные тестовые сценарии;
       6. *Документация*. Завершить задачу с необходимой документацией, чтобы облегчить понимание и поддержание кода.

Так как информирование о создании новой задачи осуществляется для всей команды, то приступить к разработке может приступить любой член команды. Решением новой задачи будут заниматься участник/участники, у которых меньше всего работы на тот момент времени, когда задача появилась. Это сделано для того, чтобы работа распределялась равномерно.

* 1. **Правила перехода задач из состояния в состояние**

В основном, состояния задач идут последовательно друг за другом (по стрелкам). В некоторых случаях состояния могут возвращаться на несколько назад или пропускать некоторые состояния. Схема перехода из состояния в состояние показана на рисунке 2.

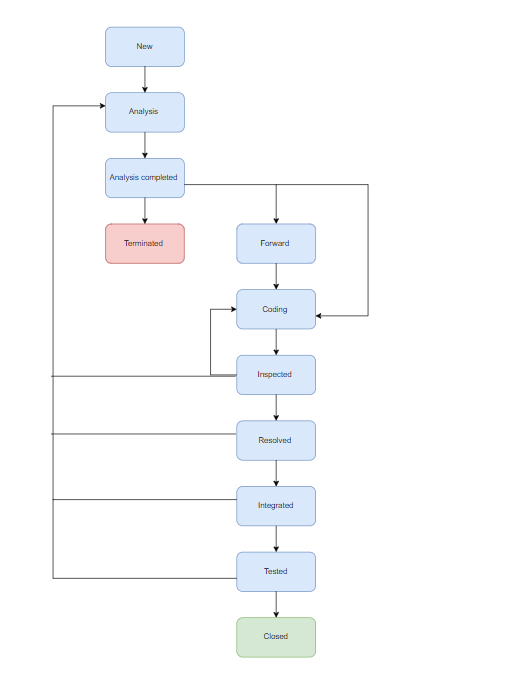
****

Рисунок 2 – Схема перехода из состояния в состояние

Примечания:

* Задача переходит в состояние *Terminated* только в самых критических ситуациях: когда не хватает ресурсов или при анализе были сделаны выводы о нецелесообразности работы над данной задачей;
* После состояния Analysis completed задача переходит в состояние

Forward. На данном этапе Team Leader может назначать на задачу исполнителя и инспектора. Также из состояния Analysis completed можно перейти в Coding, если на выполнение задачи не переназначали исполнителя/исполнителей;

* Из состояния Inspected возможно вернуться в Coding и Analysis, если в ходе инспектирования нашлись ошибки или возникли вопросы к решению задач;
* Из состояний Resolved, Integrated и Tested тоже есть возможность вернуться в состояние Analysis, так как появляются ошибки и новые идеи по преобразованию решения задачи;
* В состояние Closed можно перейти только в случае полной уверенности в выполнении задачи.

# **Презентация проекта**

На рисунке 3 представлена титульная страница презентации.

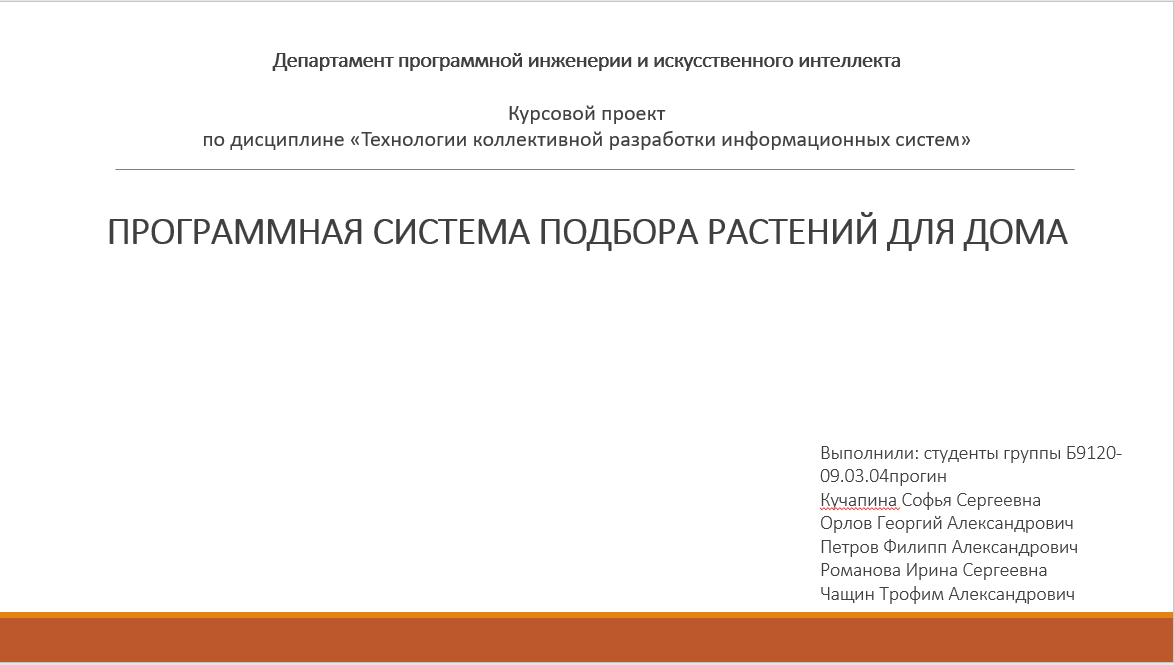


Рисунок 3 – Титульная страница

Актуальность данного проекта представлена на рисунке 4.

Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Автоматически созданное описание

Рисунок 4 – Актуальность проекта

На рисунке 5 продемонстрировано, какие есть преимущества у данного проекта.

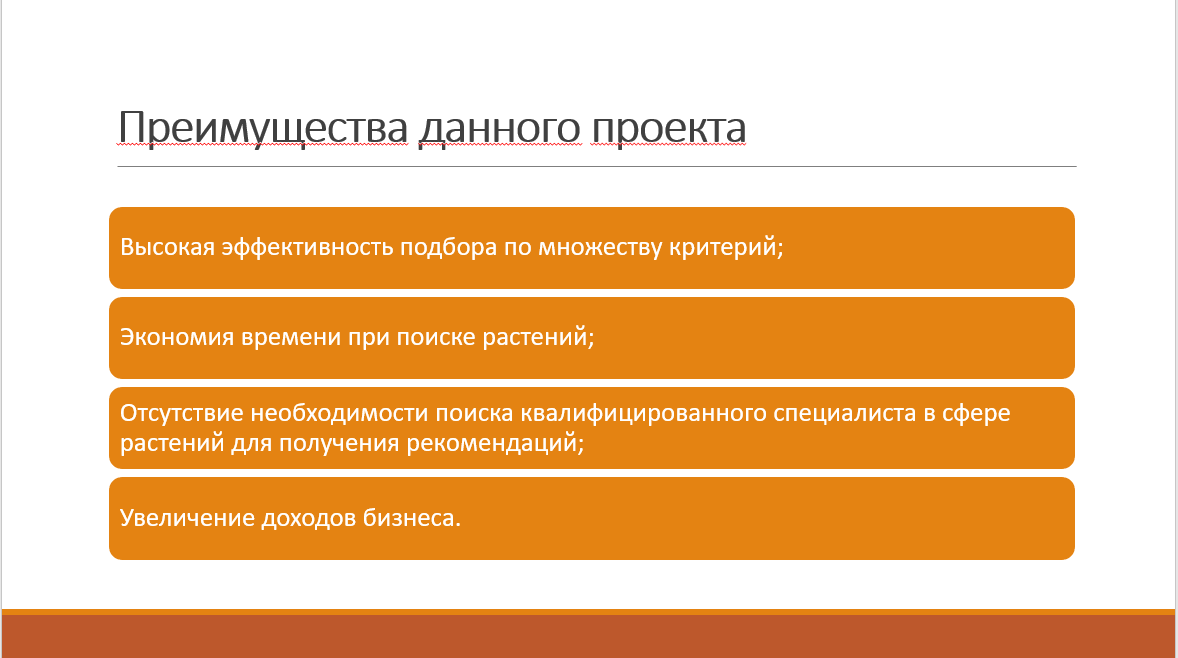


Рисунок 5 – Преимущества проекта

Внешний вид разрабатываемого средства представлен на рисунках 6 и 7.

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Автоматически созданное описание

Рисунок 6 – Пример окна интерфейса выбора параметров



Рисунок 7 – Пример окна интерфейса вывода результата

# **Требования к проекту**

Программный продукт «Программная система подбора растений для дома» предназначен для подбора растений по выбранным пользователем параметрам.

Программный продукт «Программная система подбора растений для дома» состоит из следующих подсистем:

1. FS-1 – Пользовательский интерфейс
2. FS-2 – Модуль подбора растений
3. FS-3 – База данных растений
4. FS-4 – Модуль запросов к БД
5. FS-5 – Модуль хостинга онлайн-сервиса

## **Требования к пользовательскому интерфейсу**

Требование REC FS-1 001 Пользовательский интерфейс должен запускаться и корректно отображаться на любом устройстве, поддерживающим выход в интернет, и просмотр web-страниц. В том числе:

1. Настольные компьютеры.
2. Планшеты.
3. Телефоны.

Требование REC FS-1 002 Пользовательский интерфейс должен содержать каталог всех растений. В него можно перейти, нажав на главной странице кнопку “Все растения”. На данной странице должны отображаться все растения, имеющиеся в БД, с их фотографиями. Изначально, на странице открывается небольшое число записей о растениях (10–15), и записи должны автоматически загружаться, когда пользователь пролистывает страницу вниз. Внизу и вверху страницы должна быть кнопки “На главную страницу”.

Требование REC FS-1 003 Пользовательский интерфейс должен содержать информацию о проекте. Ссылка для перехода на данную страницу должна располагаться в футере, на всех других страницах. На странице должна быть кнопка “На главную страницу”.

Требование REC FS-1 004 Пользовательский интерфейс должен содержать кнопку “Приступить к подбору”. После её нажатия открывается страница с параметрами для выбора растений. Для выхода на главную страницу должна существовать кнопка “На главную страницу”. После выбора всех параметров появляется кнопка “Подобрать растения”, нажав на которую, пользователь переходит на с результатами подбора.

Требование REC FS-1 005 Если при подборе растений в вопросе есть поле ввода, оно должно проверять введённые данные на корректность, и выводить соответствующее сообщение о некорректно введённых данных.

Требование REC FS-1 006 При изменении одного из предыдущих ответов, все последующие ответы должны сбрасываться и возвращаться в первоначальное состояние.

* 1. **Требования к модулю подбора растений, по введённым характеристикам**

Требование REC FS-2 001 Модуль должен содержать алгоритм подбора растений.

Требование REC FS-2 002 Модуль должен подобрать одно или несколько растений (у пользователя существует выбор при подборке параметров: вывести одно растение или вывести несколько растений). В случае, когда для перечня введённых характеристик нет ни одного растения в Базе данных, продукт должен уведомить пользователя о том, что ему не удалось подобрать ни одного растения.

* 1. **Требования к базе данных растений**

Требование REC FS-3 001 В Базе данных (далее - БД) должны содержаться записи о растениях, покрывающие большее количество возможных запросов, с указанием характеристик (> 90% всех запросов должны возвращать результат), однако, не учитывая характеристики, связанные с цветом растения.

* 1. **Модуль запросов к БД**

Требование REC FS-4 001 Модуль запросов к БД должен быть защищён от SQL-инъекций.

* 1. **Требования к модулю хостинга онлайн-сервиса**

Требование REC FS-5 001 Модуль хостинга онлайн-сервиса должен быть всегда доступен.

## **Нефункциональные требования**

Требование UF\_001 Контроль версий, на всех стадиях разработки продукта, должен вестись на платформе GitHub.

Требование UF\_002 Алгоритм подбора растений, должен получать набор характеристик введённых пользователем, преобразовывать их в SQL-запрос, и передавать модулю запросов к БД.

# **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.

Была разработана Архитектурно-контекстная диаграмма разрабатываемой программной системы по подбору растений.

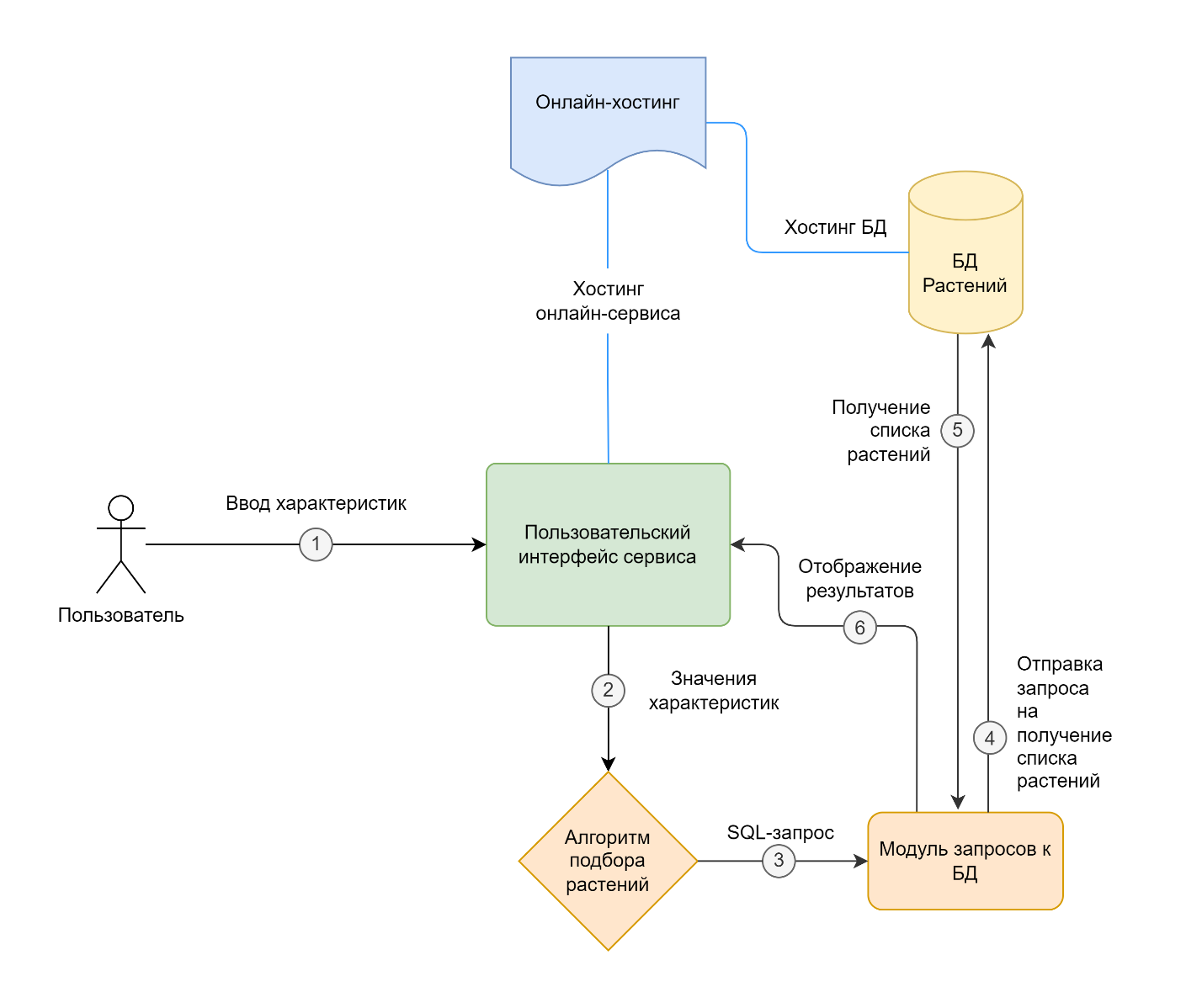
**

Рисунок 8 – Архитектурно-контекстная диаграмма

Также была разработана графический сценарий алгоритма подбора растений.

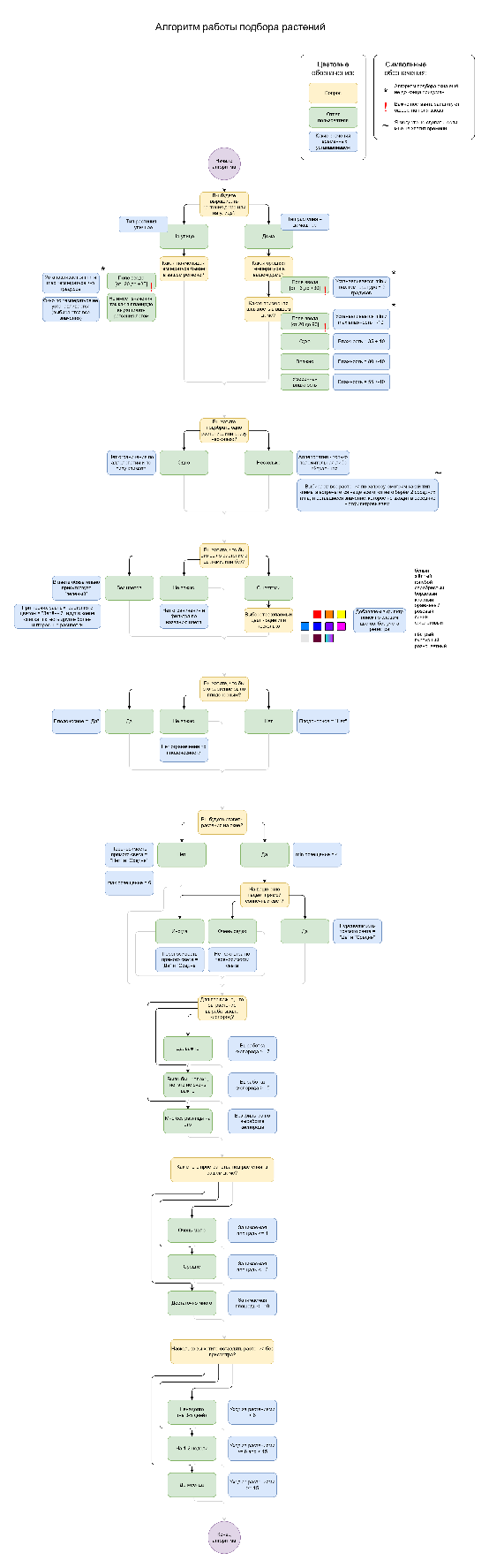


Рисунок 9 – Графический сценарий алгоритма подбора растений.

# **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.

**Метрика плотности дефектов у продукта**

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.

# **Заключение**

В рамках работы над проектом «Программная система для подбора растений для дома» 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.

**Список литературы**

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