**Содержание**

[Введение 2](#_Toc186123679)

[Глава 1. Теоретические основы разработки требований к программной системе 4](#_Toc186123680)

[1.1. Требования к программной системе и обоснование их разработки 4](#_Toc186123681)

[1.2. Пользовательские требования 4](#_Toc186123682)

[1.3. Функциональные и нефункциональные требования 5](#_Toc186123683)

[Глава 2. Предпроектное исследование. Обоснование необходимости разработки ПС 7](#_Toc186123684)

[2.1. Анализ предметной области 7](#_Toc186123685)

[2.2. Сравнительный анализ существующих решений 7](#_Toc186123686)

[2.3. Моделирование бизнес-процессов организации 8](#_Toc186123687)

[2.4. Концепции продукта и границы проекта 18](#_Toc186123688)

[Глава 3. Разработка требований к программной системе Альфа 23](#_Toc186123689)

[3.1. Определение и документирование пользовательских требований 23](#_Toc186123690)

[3.2. Определение и документирование функциональных требований 24](#_Toc186123691)

[3.3. Определение и документирование нефункциональных требований 32](#_Toc186123692)

[Заключение 35](#_Toc186123693)

[Список использованной литературы 36](#_Toc186123694)

# Введение

Несмотря на более чем пятидесятилетнее существование компьютерной отрасли, многие компании-разработчики программного обеспечения по-прежнему прикладывают значительные усилия для сбора, документирования и управления требованиями к ПО. Недостаточный объем информации, поступающей от пользователей, требования, сформулированные не полностью, их кардинальное изменение и неправильно понятые бизнес-цели — вот основные причины, из-за которых зачастую командам, работающим в области информационных технологий, не удается успешно завершить проект.

**Объект исследования**

Объектом исследования является компания для разработки программных продуктов «Цифровые решения».

**Предмет исследования**

Предметом исследования является процесс разработки программной системы управления разработками цифровой компании, включая планирование, распределение задач, контроль выполнения и управление требованиями.

**Цель работы**

Целью данной курсовой работы является разработка и документирование требований к программной системе для управления разработками цифровой компании, обеспечивающей автоматизацию и оптимизацию процессов разработки и взаимодействия внутри команды.

**Задачи**

Для достижения поставленной цели необходимо решить следующие задачи:

1. Провести исследование предметной области и изучить существующие решения в сфере управления разработками цифровых компаний.
2. Сформулировать и задокументировать пользовательские, функциональные и нефункциональные требования к системе.
3. Определить границы применения системы и разработать её концепцию.
4. Смоделировать бизнес-процессы, связанные с управлением разработками, и построить соответствующие диаграммы вариантов использования.
5. Провести анализ рисков и ограничений, связанных с разработкой и внедрением системы.

# Глава 1. Теоретические основы разработки требований к программной системе

## Требования к программной системе и обоснование их разработки

Требования к программной системе определяют функции, характеристики и качества, которые должна обеспечивать система. Они подразделяются на несколько уровней: бизнес-требования, требования пользователей и функциональные требования.

* Бизнес-требования отражают цели и задачи, которые система должна решать с точки зрения бизнеса. Эти требования формируются на основе анализа существующих бизнес-процессов и долгосрочных целей компании.
* Требования пользователей описывают, какие действия должна выполнять система с точки зрения конечных пользователей. Они обычно оформляются как сценарии использования или пользовательские истории.
* Функциональные требования включают конкретные функции и возможности, которые система обязана предоставить, и описываются через требования к её функционалу.

Процесс разработки требований включает следующие этапы:

1. Сбор требований: получение информации от заинтересованных сторон.
2. Анализ требований: изучение собранной информации и формулировка требований.
3. Документирование требований: подготовка документации, описывающей требования.
4. Утверждение требований: согласование и утверждение требований с заинтересованными сторонами.

## Пользовательские требования

Пользовательские требования описывают, какие функции система должна выполнять с точки зрения конечного пользователя. Эти требования обычно представлены в виде сценариев использования и пользовательских историй.

Сценарии использования объясняют, как пользователь будет взаимодействовать с системой для выполнения определённых задач. Они включают описание действий пользователя, входных и выходных данных, а также результатов выполнения задачи.

Пользовательские истории — это краткие описания нужных функций системы, сформулированные с точки зрения пользователя. Они обычно представляют собой фразы в формате: "Как [роль], я хочу [действие], чтобы [результат]".

Для фиксации пользовательских требований часто применяются диаграммы вариантов использования и таблицы пользовательских историй.

## Функциональные и нефункциональные требования

Функциональные требования определяют конкретные функции и возможности, которые должна предоставлять система. Эти требования описываются через функциональные возможности системы. Примеры функциональных требований:

* Система должна позволять отслеживать статус выполнения задач в реальном времени.
* Система должна поддерживать автоматическую отправку уведомлений пользователям о важных событиях или изменениях.
* Система должна обеспечивать интеграцию с внешними сервисами для обмена данными (например, с CRM или бухгалтерским ПО).
* Система должна поддерживать создание отчетов в различных форматах (PDF, Excel, CSV).
* Система должна предоставлять возможность настройки пользовательских панелей и отображения информации в удобном формате.
* Система должна поддерживать возможность импорта и экспорта данных через API.

Нефункциональные требования описывают характеристики системы, такие как производительность, надежность, безопасность и удобство использования. Эти требования касаются качества работы системы. Примеры нефункциональных требований:

* Система должна быть защищена от основных видов атак, таких как SQL-инъекции и XSS.
* Система должна иметь минимальную нагрузку на сервер, обеспечивая работу при высоких объемах данных.
* Время загрузки страницы не должно превышать 3 секунд.
* Система должна поддерживать локализацию на нескольких языках.
* Все данные, сохраняемые в системе, должны быть защищены с использованием современных методов шифрования.
* Система должна работать без сбоев в условиях пиковых нагрузок.
* Система должна обеспечивать высокий уровень отказоустойчивости и автоматически восстанавливаться после сбоя.

# Глава 2. Предпроектное исследование. Обоснование необходимости разработки ПС

## 2.1. Анализ предметной области

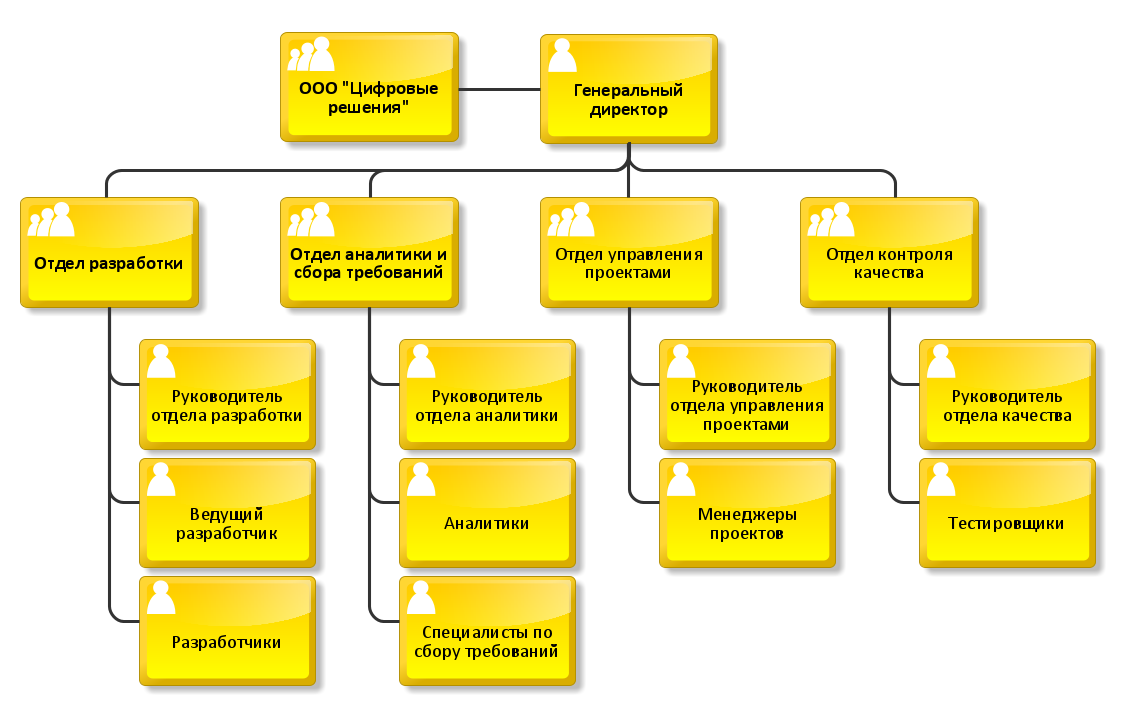
Организационная модель компании представлена на рисунке 1. 

Рисунок 1 – Диаграмма организационный структуры компании “Цифровые решения”

Диаграмма включает в себя следующие отделы:

* Отдел разработки
* Отдел аналитики и сбора требований
* Отдел управления проектами
* Отдел контроля качества

Каждый отдел включает в себя определенные отделы и должности, ответственные за выполнение конкретных задач.

## 2.2. Сравнительный анализ существующих решений

Сравнительный анализ существующих решений охватывает обзор программных систем, применяемых для управления разработками в цифровых компаниях. В данном разделе проводится анализ аналогичных систем, их сильных и слабых сторон, а также сравнительный разбор с разрабатываемой системой.

Примеры систем-аналогов:

System X:

* Преимущества — удобные инструменты для планирования проектов, интеграция с популярными сервисами для управления задачами.
* Недостатки — высокая стоимость подписки, ограниченные возможности настройки под специфические нужды компании.

System Y:

* Преимущества — модульная архитектура, что позволяет добавлять или отключать функционал, адаптируя систему под задачи.
* Недостатки — ограниченная поддержка русского языка, трудности с обучением пользователей.

System Z:

* Преимущества — высокая скорость работы, встроенная аналитика для оценки эффективности команд.
* Недостатки — отсутствие функций для управления рисками, сложность в интеграции с устаревшими системами.

## 2.3. Моделирование бизнес-процессов организации

На данном этапе проведен анализ текущей ситуации в компании, в частности ключевых процессов, связанных с управлением разработками цифровой компании. Были выделены три основных направления бизнес-деятельности:

* Планирование и управление проектами разработки программного обеспечения.
* Контроль статуса задач, распределение ресурсов и управление приоритетами.
* Сбор, анализ и документирование требований к программным продуктам.

Концептуальное описание бизнес-процессов в состоянии «as is»

В состоянии «as is» бизнес-процессы компании функционируют по устоявшейся модели, без оптимизации и значительных улучшений. Эти процессы включают в себя три основные функции: планирование и управление требованиями, управление реализацией проектов и контроль качества и управление версиями. Рассмотрим концептуально, как они функционируют в текущем состоянии.

1. **Планирование и управление требованиями**

Описание:

* На этапе инициации проекта собираются исходные данные от заказчика, включая основные требования, ограничения и ожидания.
* Проводится предварительный анализ полученной информации, но его глубина зачастую недостаточна из-за отсутствия стандартизированных подходов. Это может приводить к неполноте требований.
* На основании собранных данных формируется план требований и проектный план. Документы проходят согласование, но их качество сильно зависит от опыта аналитика.
* Возникают проблемы с недочётами в требованиях, что приводит к переработкам на более поздних этапах реализации.

**2. Управление реализацией проектов**

Описание:

* После утверждения проектного плана начинается процесс распределения задач между командами разработки и контроля над выполнением.
* Управление ресурсами и сроками осуществляется вручную или с использованием базовых инструментов, что снижает оперативность в случае изменений.
* Коммуникация между командами и менеджерами проектов осуществляется по различным каналам, часто разрозненным, что приводит к увеличению времени на передачу информации.
* Статус выполнения задач фиксируется нерегулярно, что затрудняет мониторинг и контроль.

**3. Контроль качества и управление версиями**

Описание:

* Контроль качества включает в себя проверку всех выполненных задач на соответствие установленным требованиям и стандартам качества. Проверки выполняются на каждом этапе реализации проекта, чтобы исключить ошибки и несоответствия.
* Управление версиями предусматривает фиксирование всех изменений в продукте, создание стабильных версий и управление ветками разработки. Версионный контроль обеспечивает прозрачность и позволяет отслеживать внесённые изменения на каждом этапе разработки.
* По завершении проекта выполняется финальная проверка качества и подготовка итоговой документации. Формируются отчёты, содержащие информацию о выполненных задачах, используемых ресурсах и итоговом качестве продукта. Эти документы передаются заинтересованным сторонам для анализа и принятия решений.

Границы разрабатываемой системы и её связи со всем остальным миром

Для определения границ разрабатываемой системы важно выделить, какие функции и процессы относятся непосредственно к системе, а какие — к внешним элементам.

Разрабатываемая система охватывает следующие процессы:

* Управление сбором и анализом требований, включая их верификацию.
* Планирование и управление ресурсами, задачами и сроками.
* Мониторинг статуса выполнения задач и управление коммуникациями.
* Контроль качества выполненных задач и управление версиями продукта.

Детализация бизнес-процессов до 3 уровня глубины декомпозиции

Декомпозиция представлена в виде 3-х уровней на рисунках 2-6.

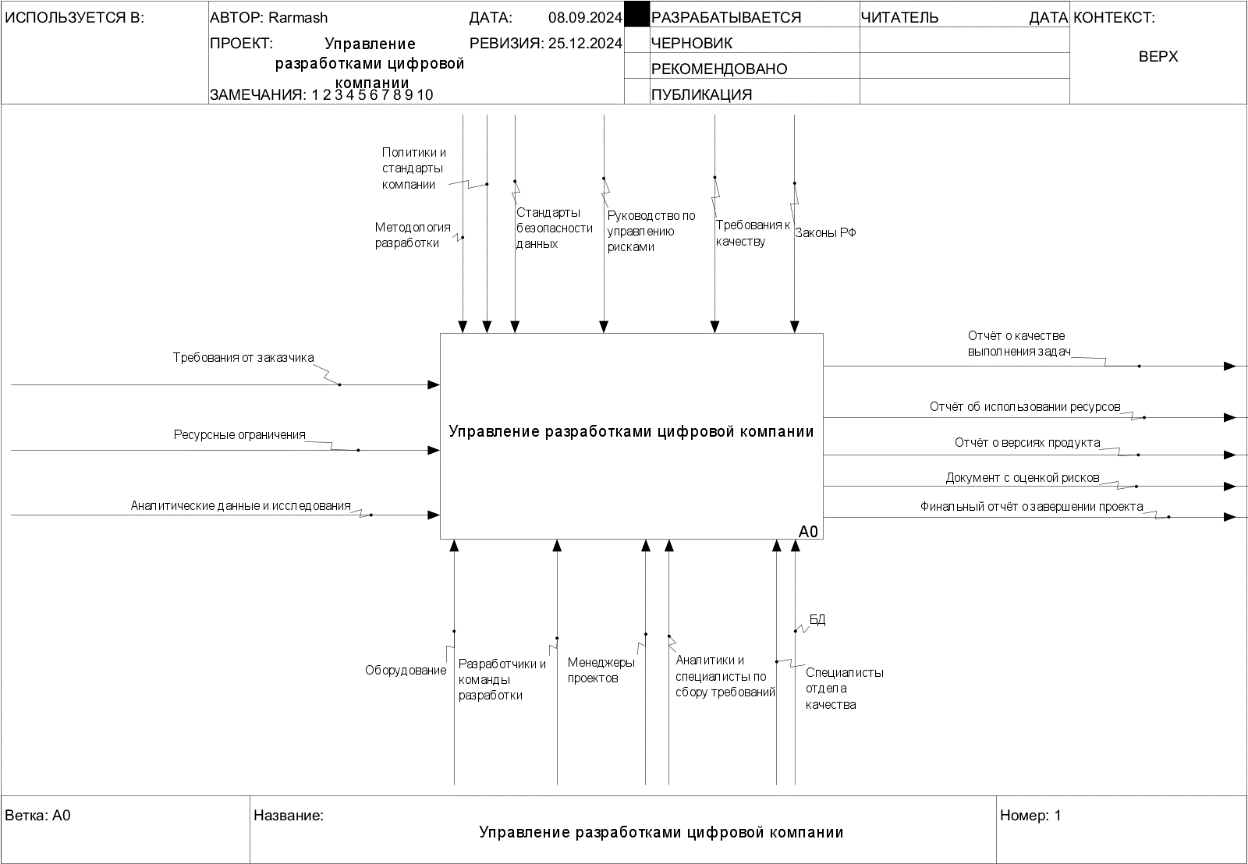


Рисунок 2 – Первый уровень декомпозиции

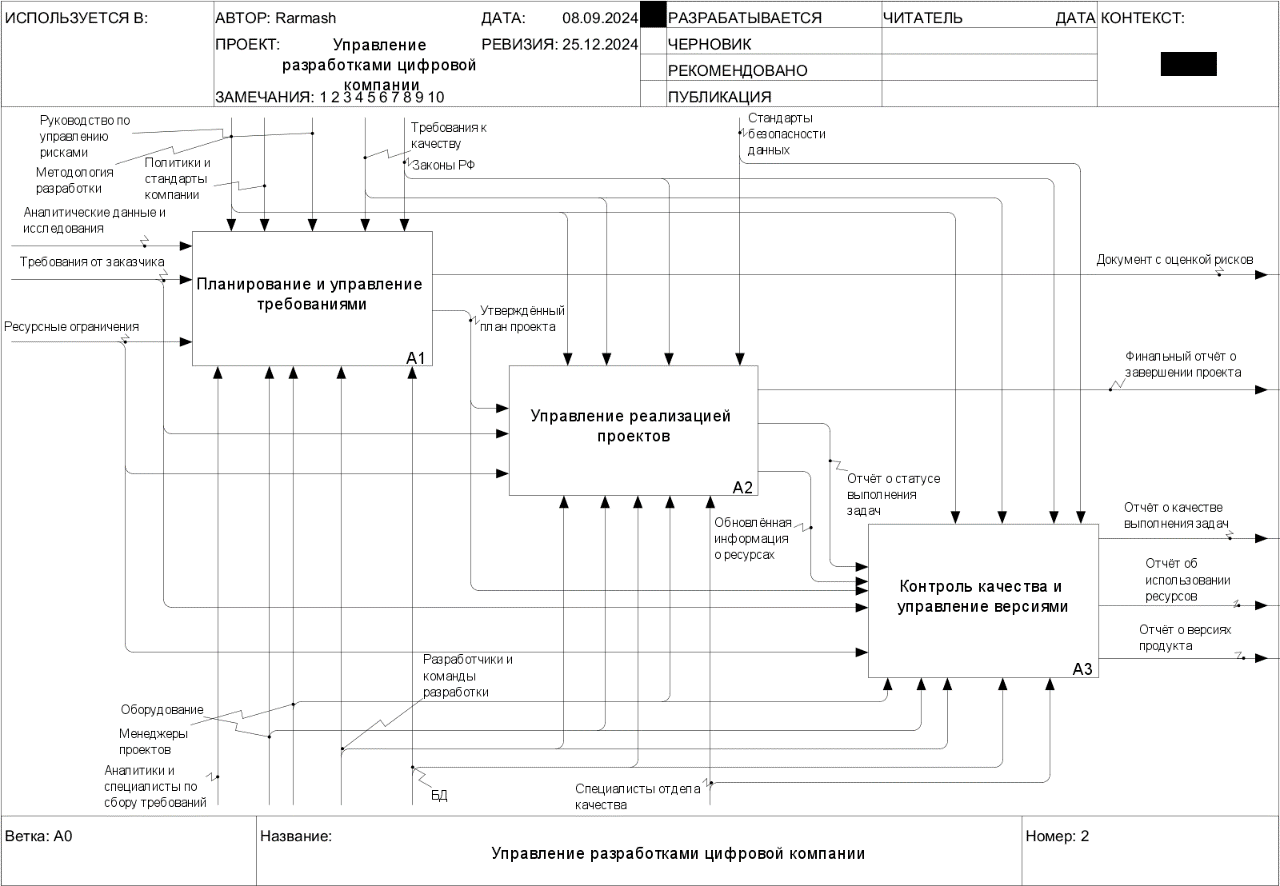


Рисунок 3 – Второй уровень декомпозиции

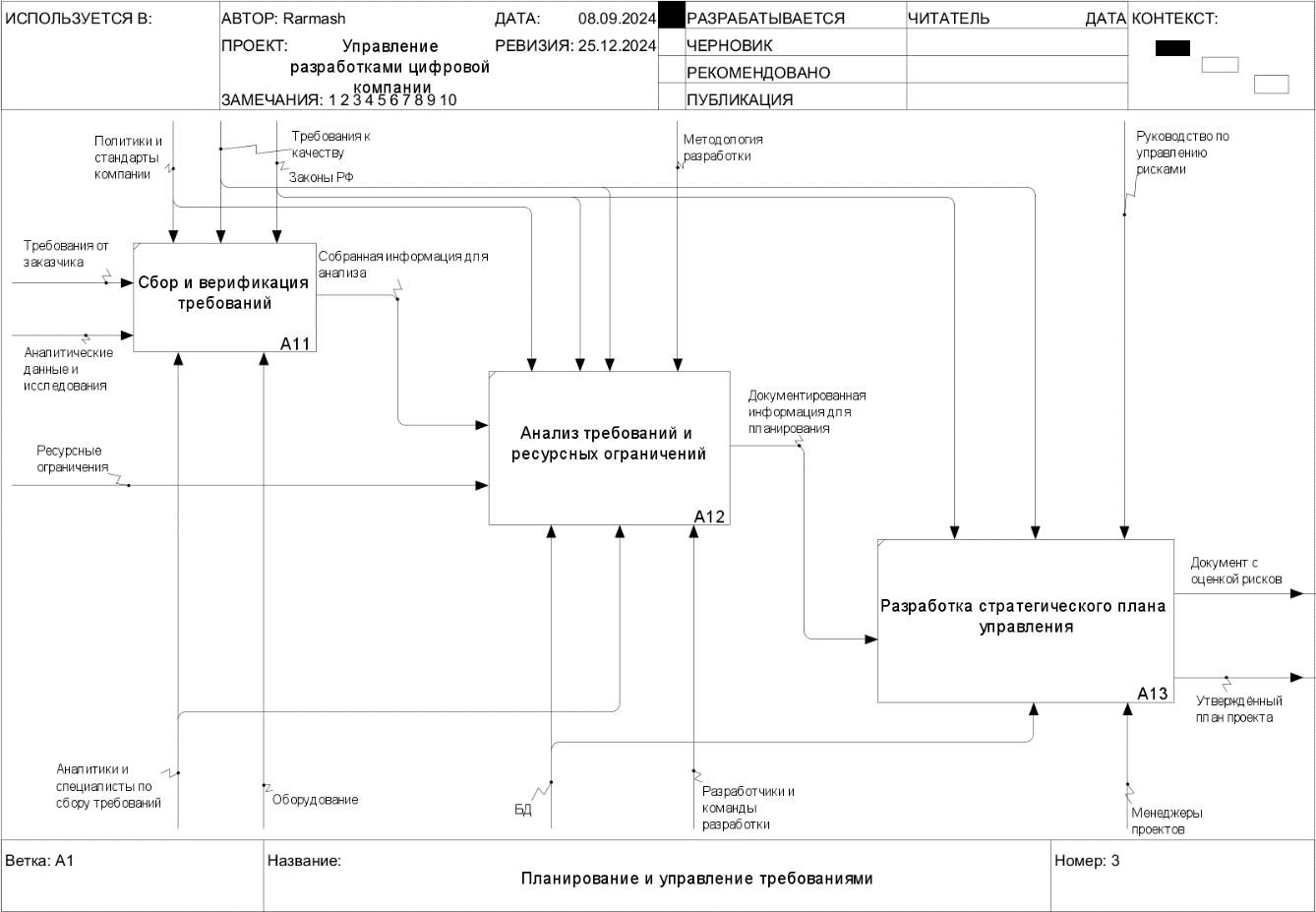


Рисунок 4 – Третий уровень декомпозиции “Планирование и управление требованиями”

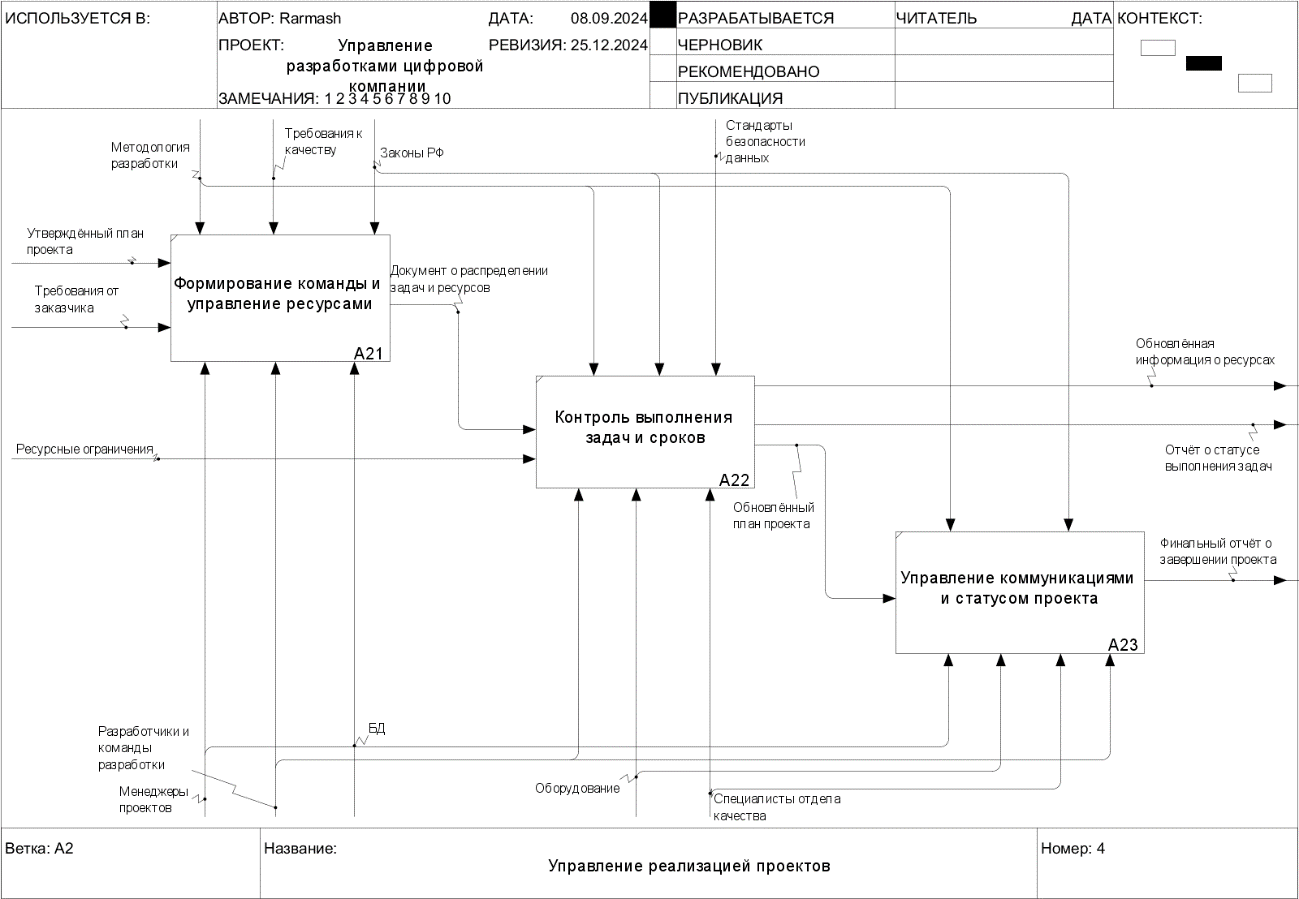


Рисунок 5 – Третий уровень декомпозиции “Управление реализацией проектов”

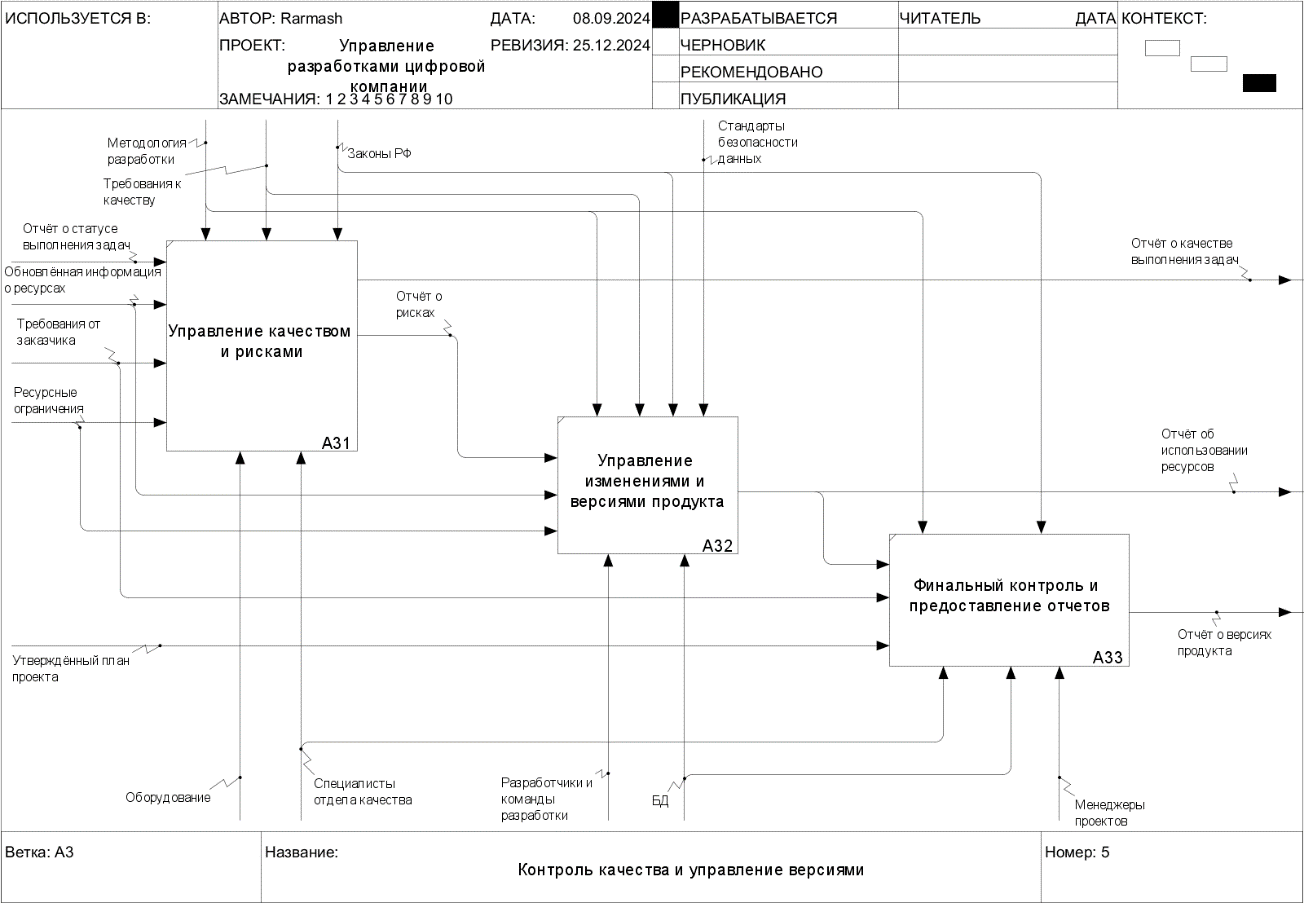


Рисунок 6 – Третий уровень декомпозиции “Ко”

**Разработка или модернизация программной системы направлена на устранение выявленных проблем.** Для начала строится модель текущего состояния («as is»), где выявляются существующие «узкие места» — проблемы, снижающие эффективность процессов. После этого определяется цель разработки или модернизации системы. Затем осуществляется поиск решений выявленных проблем с использованием методов анализа, включая морфологический анализ и синтез. На основе найденных решений формируется модель желаемого состояния системы («to be»).

Цель разработки новой системы, направленная на решение выявленных проблем, обычно слишком широкая и сложная для прямой реализации. Поэтому её разбивают на более мелкие подцели с помощью метода декомпозиции. Этот подход позволяет сосредоточиться на достижении каждой конкретной задачи, необходимой для реализации общей цели.

Для создания логичной структуры целей применяется метод построения «дерева целей». Это структурированная модель, где:

* Генеральная цель находится на вершине дерева.
* Подцели первого, второго и последующих уровней формируют ветви дерева.

Такой подход упрощает управление проектом, помогая определить приоритеты и последовательность действий.

Нахождение “узких мест”

На основании анализа текущих процессов были выделены следующие основные проблемы:

1. Управление требованиями**.** Сложности в сборе и актуализации требований приводят к потере времени и несогласованности между участниками проекта.
2. Контроль выполнения задач**.** Недостаточная прозрачность процесса выполнения задач и отсутствие актуальной информации о статусах проектов.
3. Управление версиями и качеством**.** Трудности с координацией команд разработки, недостаточное тестирование промежуточных версий и проблемы с выпуском стабильных релизов.

Эти проблемы ограничивают эффективность работы системы и требуют решения для улучшения управления разработкой.

Дерево целей

Дерево целей для компании по разработке цифровой компании представлено на рисунке 7.



Рисунок 7 – Дерево целей

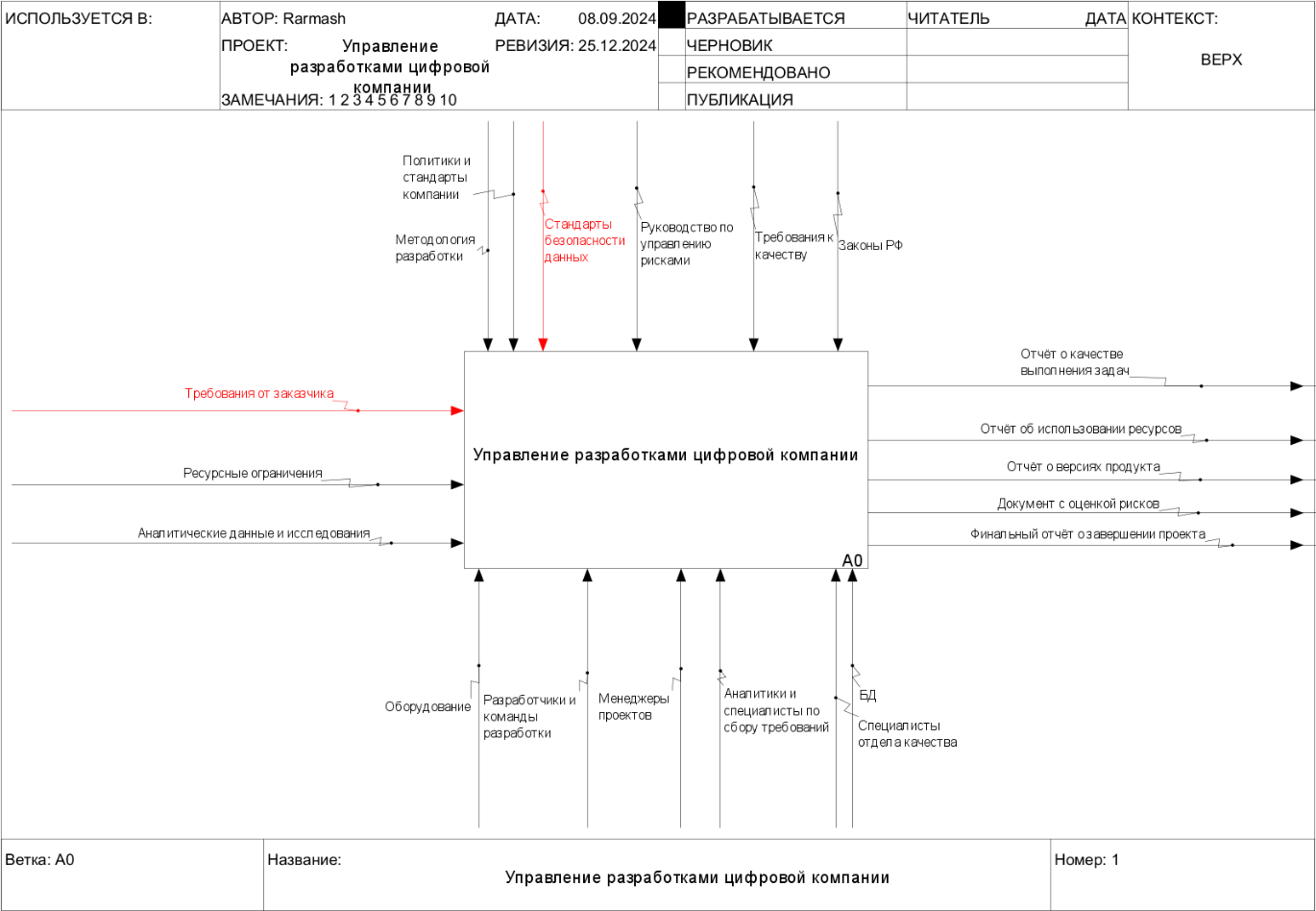
Диаграмма нотации IDEF0 “to be”  
****

Рисунок 8 – Концептуальный уровень

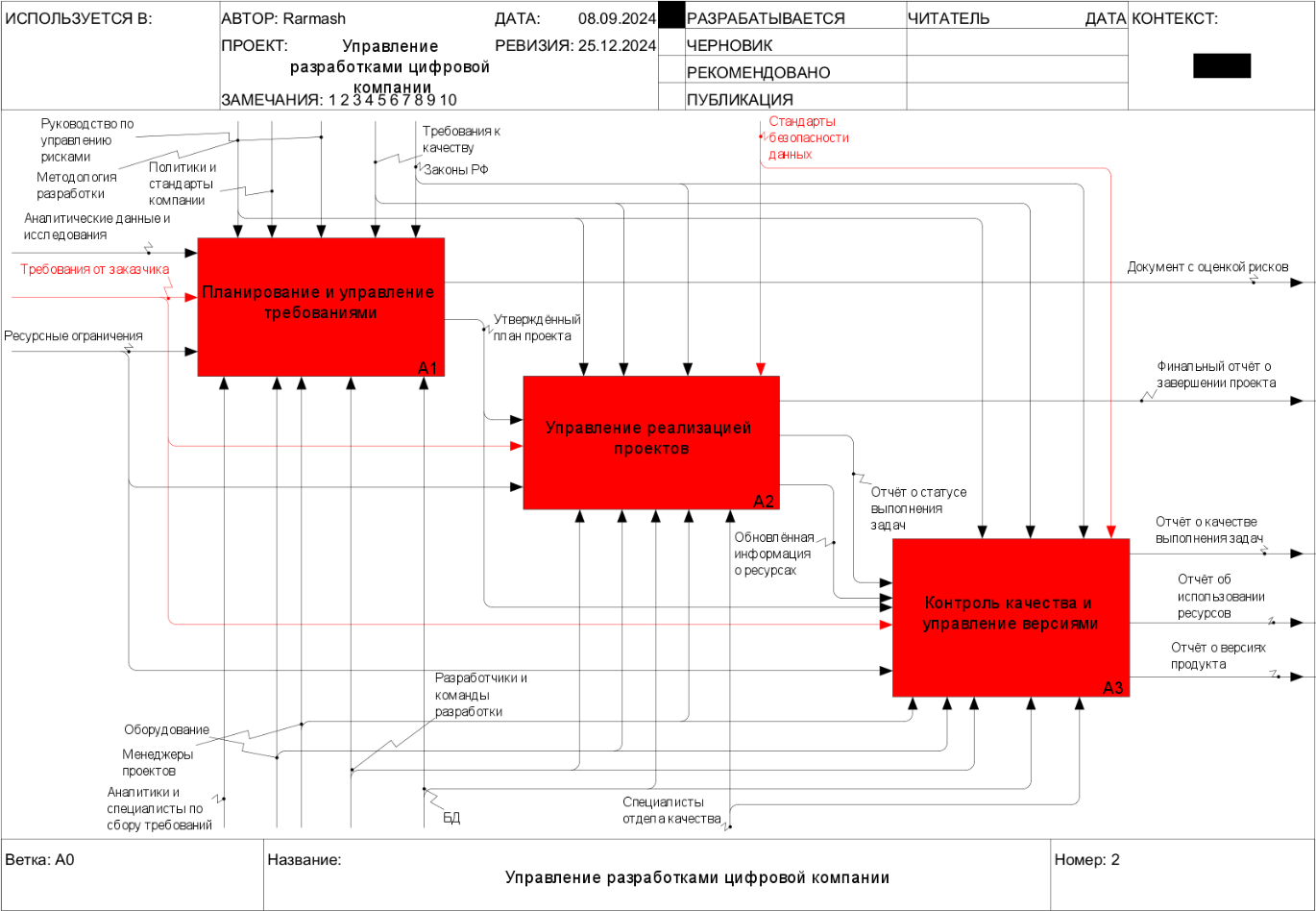


Рисунок 9 – Второй уровень декомпозиции

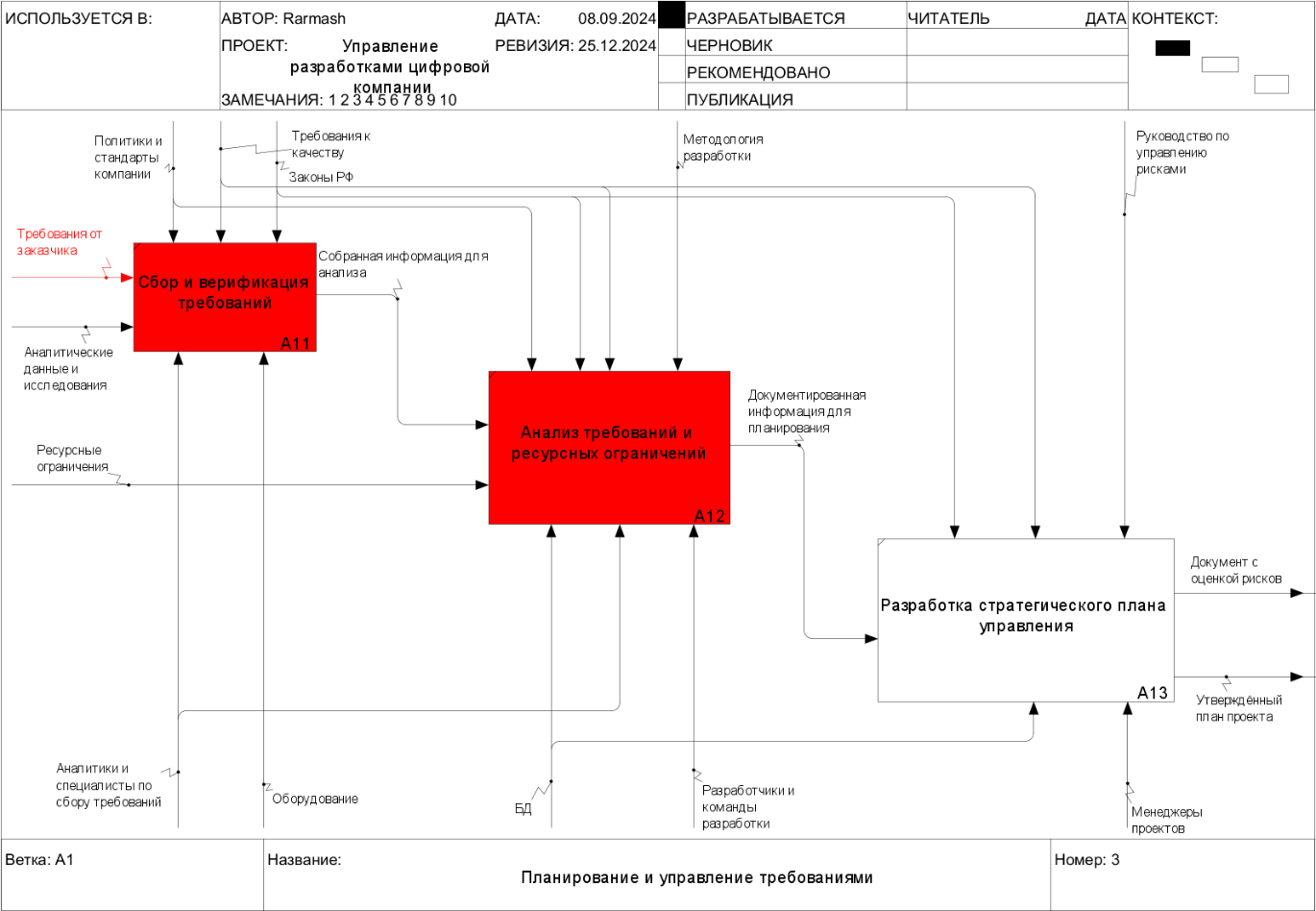


Рисунок 10 – Третий уровень декомпозиции “Планирование и управление требованиями”

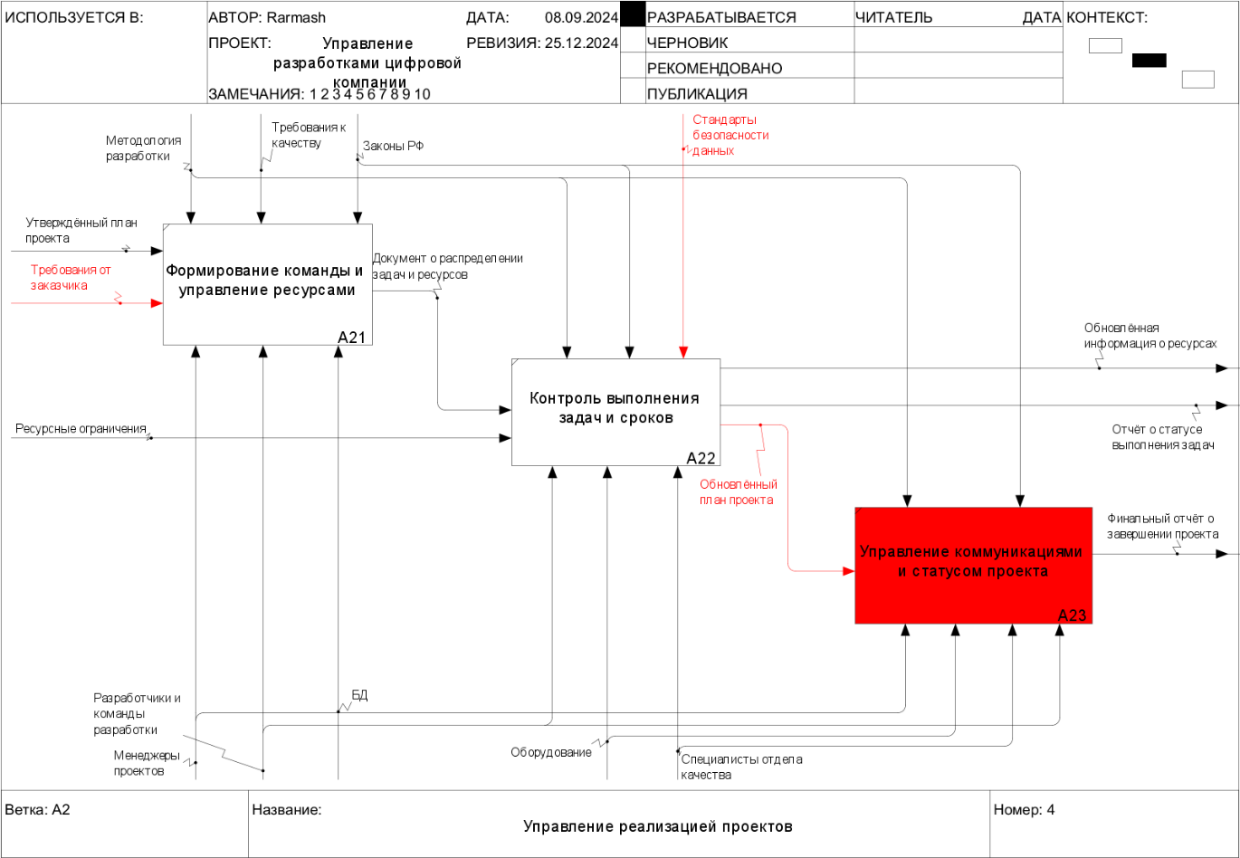


Рисунок 11 – Третий уровень декомпозиции “Управление реализацией проектов”

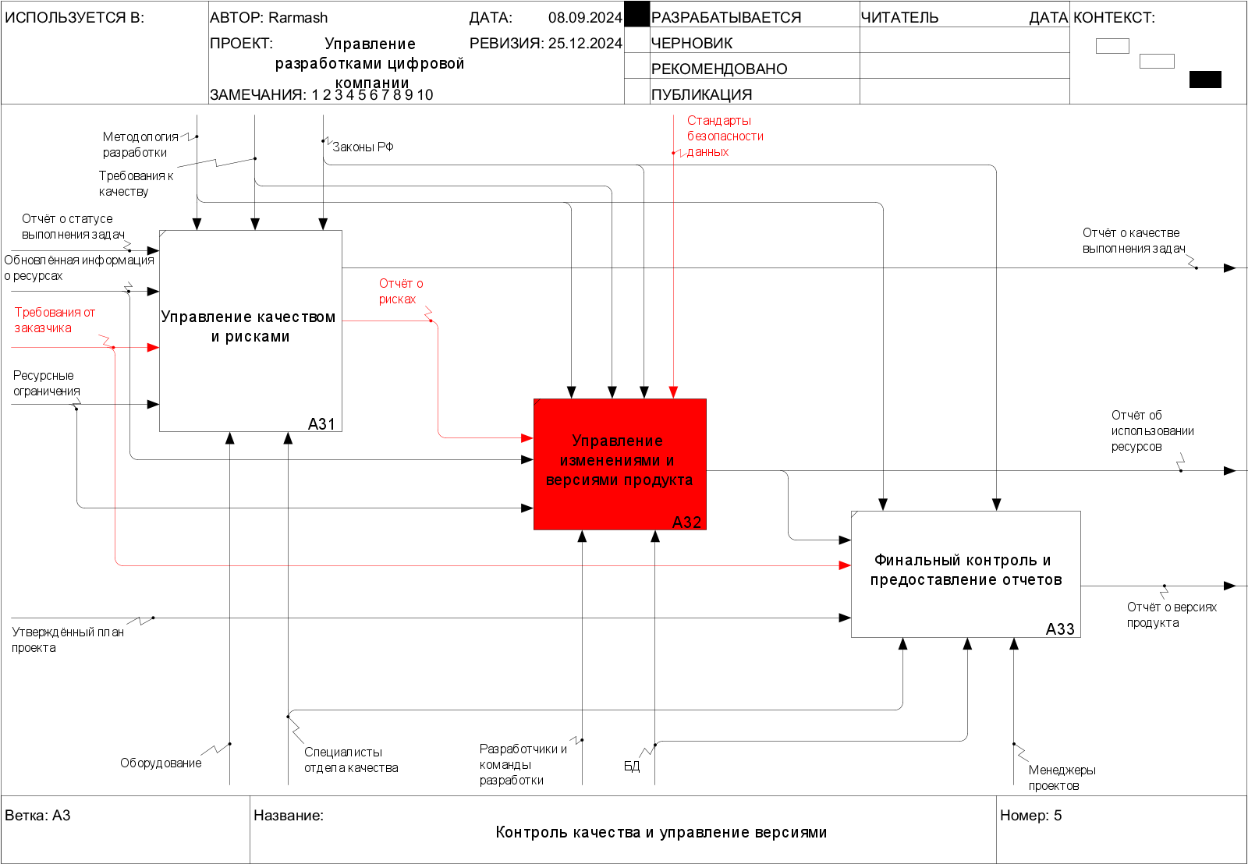


Рисунок 12 – Третий уровень декомпозиции “Контроль качества и управление версиями”

После решения проблем на диаграмме “to be” система получила улучшенную стабильность и скорость работы.

## 2.4. Концепции продукта и границы проекта

**Представление о концепции продукта**

Для руководителей разработки, которым необходимо контролировать и управлять процессом создания цифровых продуктов, система **DigitalSolutions Manage** является информационной системой, которая обеспечит единую точку доступа ко всем подразделениям, задействованным в разработке. Система будет отслеживать статус выполнения каждой задачи и предоставлять всю необходимую информацию вместе с документами. Эта система позволит сократить время на выполнение задач, исключив затраты на ручную координацию между командами. В отличие от существующих ручных методов управления разработкой, наш продукт будет автоматически формировать отчеты, предоставлять доступ к ключевым данным и отслеживать статус выполнения проектов.

Пробелы в требованиях:

* + - 1. Управление изменениями задач:
  + Неопределенность в механизмах отслеживания изменений задач и историй их выполнения может затруднить анализ и контроль над ходом проекта.
    - 1. Интеграция с внешними системами:
  + Недостаточная детализация требований к интеграции с существующими корпоративными системами и сервисами может ограничить возможности автоматизации и обмена данными.
    - 1. Размытость пользовательских сценариев:
  + В требованиях отсутствует четкое описание ключевых сценариев использования, что может привести к созданию функционала, не соответствующего реальным потребностям.

Избыточные требования:

* + - * 1. Чрезмерная детализация в отчетности:
  + Избыточные требования к настройке множества мелких параметров отчетов могут увеличить сложность системы и потребовать дополнительных ресурсов на их реализацию, не принося значимой пользы.
    - * 1. Необоснованное расширение функций управления рисками:
  + Предложение функционала, охватывающего редкие и маловероятные сценарии рисков, может усложнить систему и привести к удорожанию разработки, при этом не будучи востребованным на практике.

**Документ о концепции и границах**

1. **Бизнес-требования**
   1. **Исходные данные**

Компания ООО "Цифровые решения" сталкивается с рядом проблем в управлении разработками цифровых продуктов. В настоящее время процесс разработки затягивается из-за неэффективного планирования и координации между различными командами и отделами. Среднее время разработки одного продукта составляет 6 месяцев, из которых 2 месяца уходит на согласование требований и планирование задач. Это приводит к задержкам в реализации проектов и снижению удовлетворенности клиентов. Кроме того, из-за недостаточной информации о статусе проекта часто возникают ошибки в планировании ресурсов и версий продукта, что увеличивает количество исправлений и откладывает финальные сроки.

* 1. **Возможности бизнеса**

Многие сотрудники и менеджеры компании ООО "Цифровые решения" выразили необходимость в создании системы управления разработками, которая позволила бы автоматизировать процессы планирования и координации между различными командами. Такая система сможет значительно сократить время разработки, уменьшить количество ошибок, связанных с неверным распределением ресурсов и контролем версий, а также повысить удовлетворенность клиентов за счет своевременной и качественной реализации проектов.

* 1. **Бизнес-цели**
* Сократить среднее время разработки одного проекта на 30% в течение 6 месяцев после первого внедрения системы.
  + Способ измерения: анализ данных системы управления разработками.
* Снизить количество ошибок и несоответствий в проектах на 20% в течение 12 месяцев после первого внедрения системы.
  + Способ измерения: анализ данных системы контроля качества, своевременное предоставление отчетов о выполнении задач и версий.
* Увеличить удовлетворенность клиентов на 15% в течение 6 месяцев после первого внедрения системы.
  1. **Критерии успеха**

Достичь увеличения среднего рейтинга по ежеквартальному опросу об удовлетворенности работой системы на 0,5 балла по сравнению с третьим кварталом 2024 года по шкале от 1 до 10 в течение 3 месяцев после первого внедрения системы и на 1,0 балла в течение 12 месяцев.

Увеличить количество завершённых проектов и улучшить эффективность разработки на 25% в течение первого года после ввода системы в эксплуатацию.

* 1. **Видение решения**

DigitalSolutions Manage предназначен для IT-компаний, стартапов и проектных команд, стремящихся упростить и автоматизировать управление разработками. Это программное обеспечение обеспечивает централизованный подход к планированию, распределению задач, контролю версий и отслеживанию прогресса. В отличие от разрозненных инструментов, система интегрирует все ключевые процессы в единой платформе, что позволяет снизить риски ошибок, сократить сроки реализации проектов и улучшить взаимодействие между командами.

* 1. **Бизнес-риски**
* Недостаточное количество сотрудников могут сразу адаптироваться к новой системе **DigitalSolutions Manage**, что снизит ожидаемую отдачу от инвестиций в разработку системы и изменения в процессе управления разработками. (Вероятность = 0,3; ущерб = 9.)
* Имеющихся функциональных возможностей системы может быть недостаточно, что приведет к сбоям в управлении проектами и задержкам в выполнении задач, что может негативно сказаться на сроках разработки цифровых продуктов. (Вероятность = 0,5; ущерб = 6.)
  1. **Предположения и зависимости**
* У работников компании будут доступны системы с соответствующими интерфейсами для обработки и управления проектами в рамках системы **DigitalSolutions Manage**.
* У клиентов компании будут доступны системы с соответствующими интерфейсами для отслеживания статуса своих проектов, что обеспечит прозрачность и улучшит взаимодействие.

1. **Бизнес-требования**
   1. **Основные функции**

* Планирование и координация процессов управления проектами.
* Создание, просмотр, изменение и удаление производственных задач и проектов.
* Просмотр списка ресурсов и сведений об их использовании в рамках проектов.
* Интеграция с отделом разработки для управления процессами разработки и выполнения проектов.
* Просмотр статуса проектов со стороны клиентов для обеспечения прозрачности выполнения.
* Обеспечение доступа к системе через корпоративную интрасеть, смартфон, планшет или через внешнее подключение к Интернету для авторизованных сотрудников.

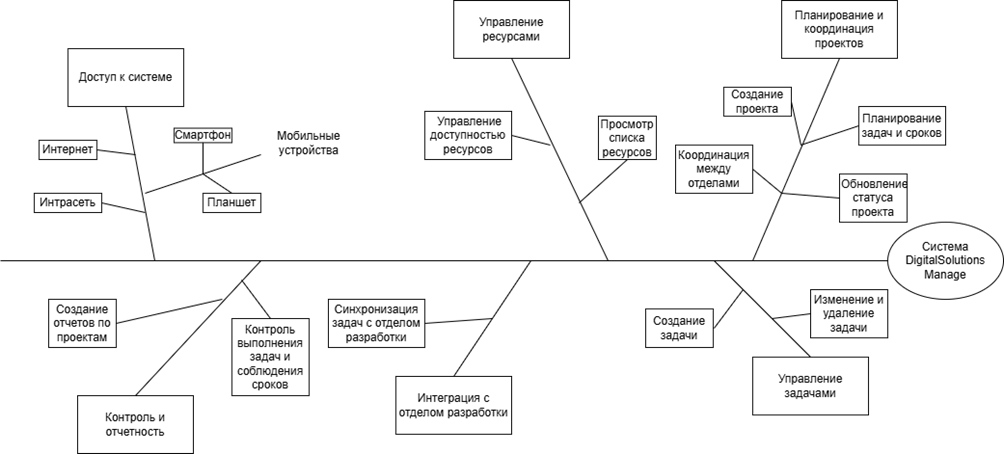


Рисунок 13 – Частичное дерево функций системы

# Глава 3. Разработка требований к программной системе Альфа

## 3.1. Определение и документирование пользовательских требований

(построение диаграммы вариантов использования в нотации UML, составление документа пользовательских историй).

Для определения и документирования пользовательских требований построим диаграмму варианта использования (рис. 14).

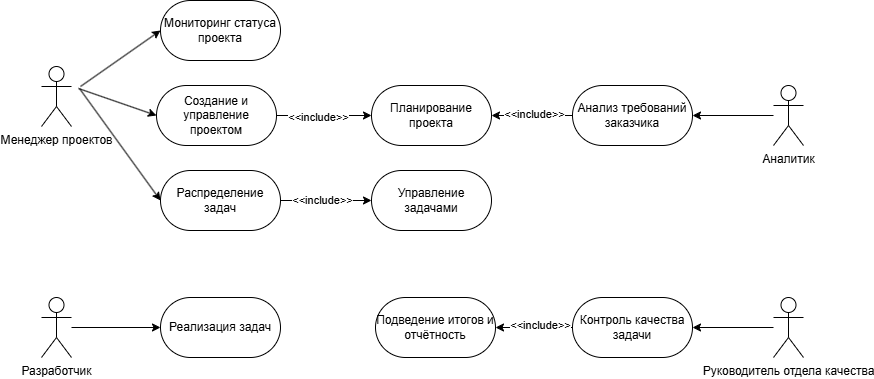


Рисунок 14 – Диаграмма варианта использования

Пользовательские истории:

* **Как аналитик,** я хочу собирать и анализировать требования от заказчика и ресурсы компании, чтобы создать техническое задание для команды разработки и обеспечить соответствие конечного продукта потребностям клиента.
* **Как руководитель отдела качества,** я хочу проверять промежуточные этапы разработки и составлять отчеты о контроле качества кода, чтобы убедиться, что продукт соответствует стандартам качества и безопасности.
* **Как менеджер проектов,** я хочу отслеживать выполнение задач и контроль сроков на каждом этапе разработки, чтобы обеспечить своевременное завершение проекта и оперативно вносить коррективы в план при необходимости.
* **Как разработчик,** я хочу выполнять задачи для каждого этапа разработки, чтобы обеспечить выполнение проекта.

## 3.2. Определение и документирование функциональных требований

Диаграммы классов анализа, последовательности и деятельности представлены на рисунках 15, 16 и 17 соответственно. Они были построены для определения и документирования функциональных требований.

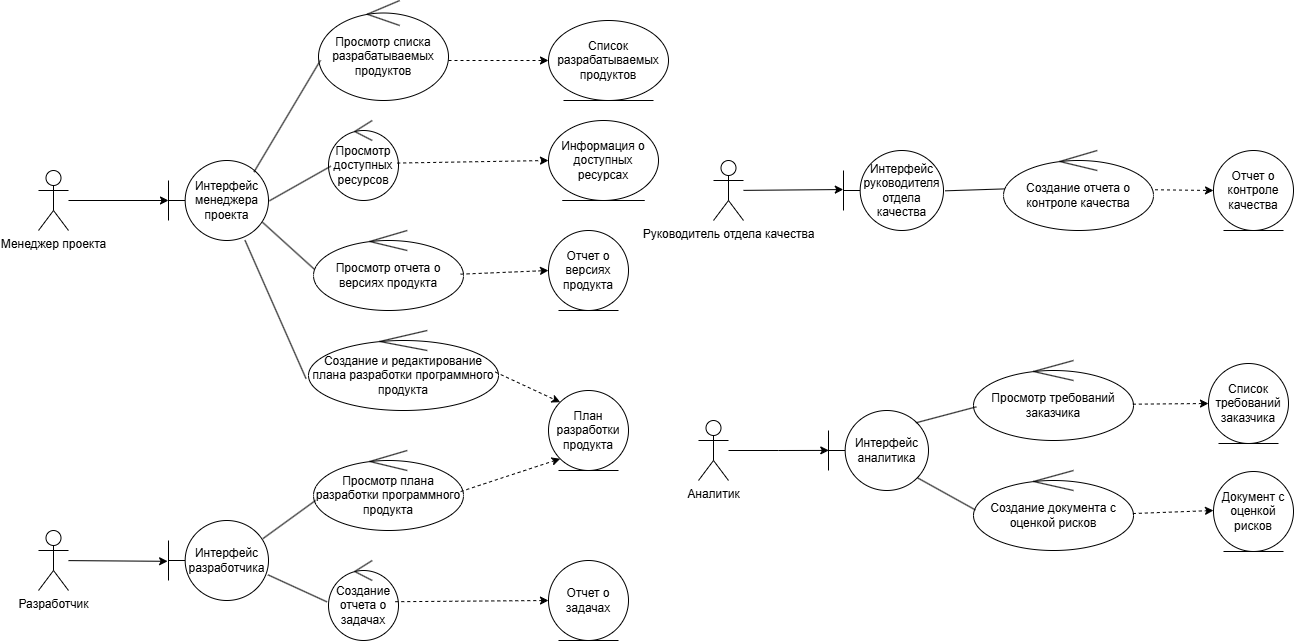
****

Рисунок 15 – Диаграмма классов анализа

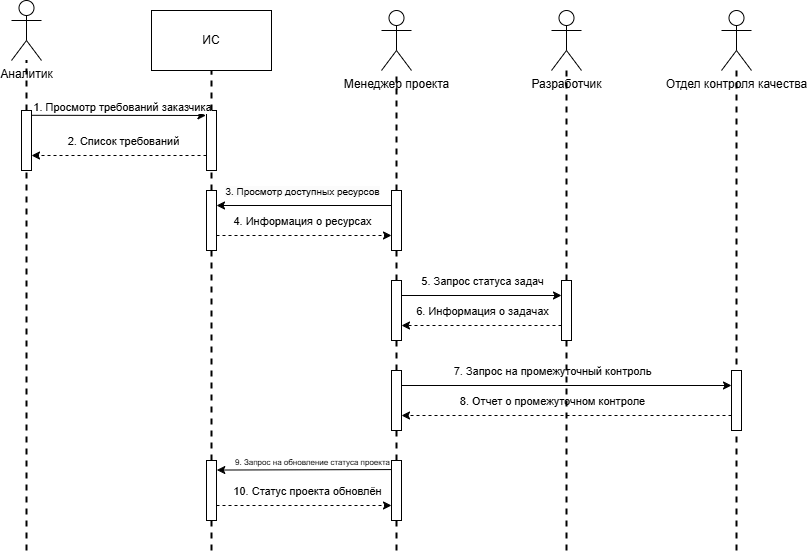
****

Рисунок 16 – Диаграмма последовательности

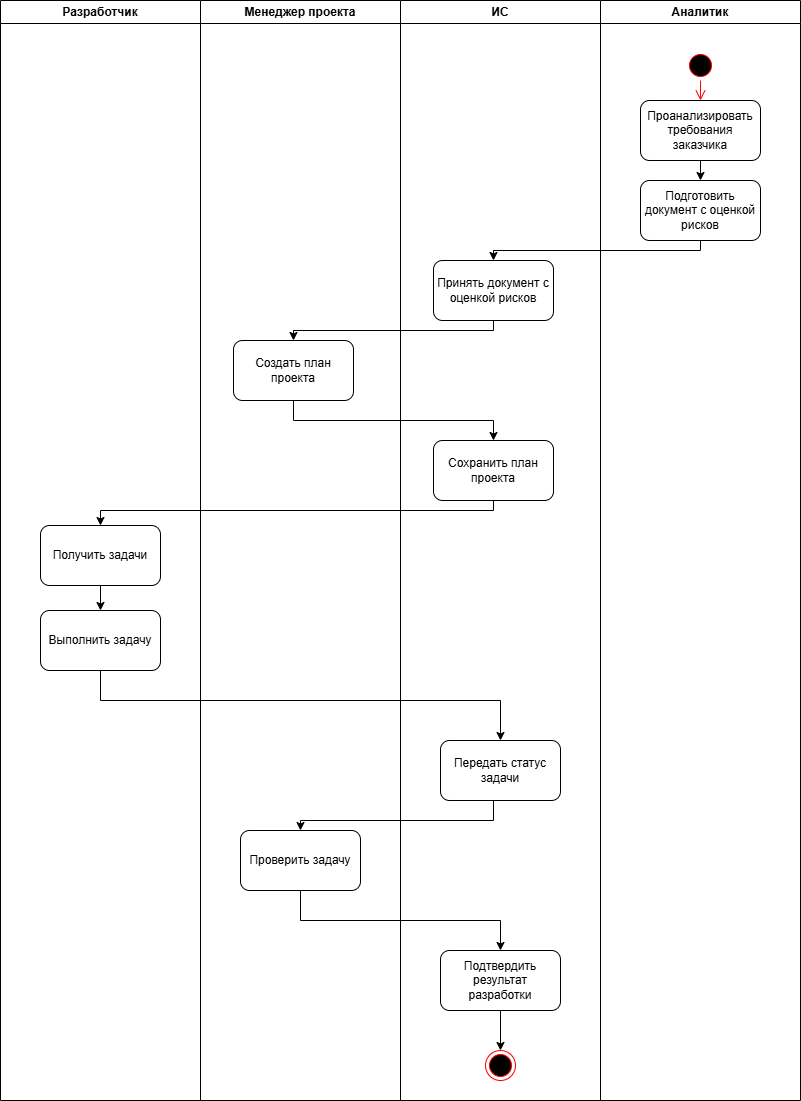
****

Рисунок 17 – Диаграмма деятельности

Также, для определения и документирования функциональных требований построим диаграммы классов и «сущность связь» (рис. 18 и 19).

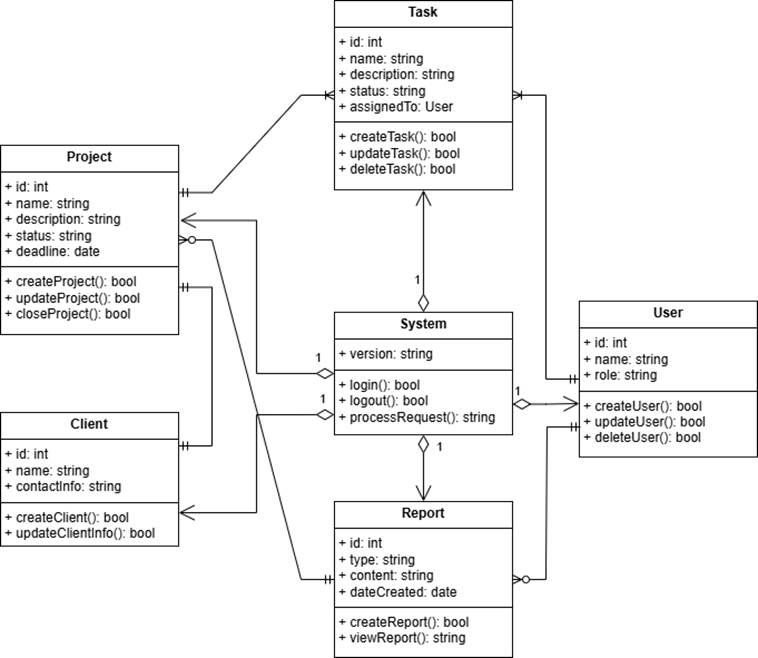
****

Рисунок 18 – Диаграмма классов

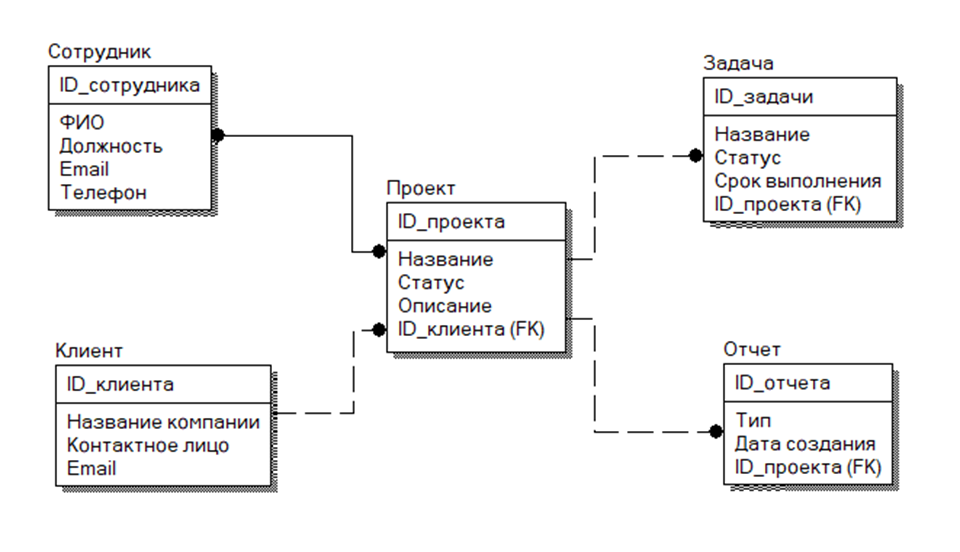


Рисунок 19 – Диаграмма «сущность - связь»

Таблица 1 – Словарь данных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование сущности | Описание | Значения | Длина | Структура или тип данных |
| Сотрудник | Описывает сотрудника, который участвует в проектах и задачах. | ID\_сотрудника |  | int |
| ФИО | 100 | varchar |
| Должность | 100 | varchar |
| Email | 100 | varchar |
| Телефон | 50 | varchar |
| Клиент | Описывает клиента, который заказывает проект. | ID\_клиента |  | int |
| Название компании | 50 | varchar |
| Контактное лицо | 50 | varchar |
| Email | 100 | varchar |
| Проект | Описывает проект, который разрабатывается в системе. Связан с клиентом. | ID\_проекта |  | int |
| Название | 100 | varchar |
| Статус | 50 | varchar |
| Описание | 400 | varchar |
| ID\_клиента |  | int |
| Задача | Описывает задачи, входящие в проект. | ID\_задачи |  | int |
| Название | 100 | varchar |
| Статус | 50 | varchar |
| Срок выполнения |  | datetime |
| ID\_проекта |  | int |
| Отчёт | Описывает отчёт, связанный с проектом. | ID\_отчёта |  | int |
| Тип | 50 | varchar |
| Дата создания |  | datetime |
| ID\_проекта |  | int |

Таблица 2 – Матрица CRUD

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Вариант использования \ Сущность | Сотрудник | Клиент | Проект | Задача | Отчёт |
| Добавить нового сотрудника | C |  |  |  |  |
| Назначить проект клиенту |  | R, U | U |  |  |
| Создать новый проект |  |  | C |  |  |
| Добавить задачу в проект |  |  | R | C |  |
| Прочитать данные о задаче |  |  | R | R |  |
| Обновить описание задачи |  |  |  | U |  |
| Удалить задачу |  |  |  | D |  |
| Создать отчёт по проекту |  |  | R |  | C |
| Прочитать отчёт |  |  |  |  | R |
| Удалить отчёт |  |  |  |  | D |

1. Требования к качеству программной системы:

Внешние атрибуты качества:

* 1. Доступность
* AVL-1: Система должна быть доступна не менее 99,8% времени в течение месяца.
* AVL-2: Плановые работы допускаются только в ночное время (с 02:00 до 05:00) и не чаще двух раз в месяц.
* AVL-3: Среднее время восстановления после сбоя не должно превышать 15 минут.
  1. Производительность
* PER-1: Время отклика интерфейса на запрос пользователя не более 1,5 секунд.
* PER-2: Поддержка одновременной работы до 200 пользователей без снижения производительности.
* PER-3: Обработка до 10 000 объектов данных в день.
  1. Безопасность
* SEC-1: Двухфакторная аутентификация для всех пользователей.
* SEC-2: Шифрование данных при передаче (протокол TLS 1.3).
* SEC-3: Логирование всех действий, связанных с изменением критически важных данных.
* SEC-4: Разграничение прав доступа в зависимости от ролей пользователей.
  1. Удобство использования
* USE-1: Интуитивно понятный интерфейс, доступный для понимания пользователем без предварительного обучения.
* USE-2: Наличие интерактивных подсказок и обучающих материалов прямо в системе.
* USE-3: Поддержка корректной работы на современных версиях популярных браузеров (Google Chrome, Mozilla Firefox, Safari).
  1. Надёжность
* REL-1: Среднее время между отказами (MTBF) должно составлять не менее 180 дней.
* REL-2: Система должна автоматически сохранять критически важные данные каждые 5 минут.
  1. Совместимость
* IOP-1: Программа должна корректно работать на операционных системах Windows 10, macOS 11.0 и Linux.
* IOP-2: Поддержка стандартных форматов данных для обмена: JSON, XML.

Внутренние атрибуты качества:

* + - 1. Эффективность
* EFF-1: На пике нагрузки приложение не должно потреблять более 75% процессорного времени и 50% оперативной памяти.
  + - 1. Масштабируемость
* SCA-1: Возможность увеличения нагрузки до 300% в течение одного года без изменения архитектуры системы.
  + - 1. Модифицируемость
* MOD-1: Обновление функциональности должно быть возможно без изменения основного кода системы более чем на 15%.
  + - 1. Переносимость
* POR-1: Возможность миграции системы на другой сервер без потери данных или необходимости значительных изменений.
  + - 1. Проверяемость
* VER-1: Каждый модуль системы должен иметь отдельные тесты, которые можно выполнить независимо от других модулей.
  + - 1. Повторное использование
* REU-1: Функции, обрабатывающие данные, должны быть реализованы в виде библиотек, которые можно повторно использовать в других проектах.

Ограничения программной системы:

Технические ограничения:

* CON-1: Поддержка только современных веб-браузеров с актуальными версиями (не старше двух лет).
* CON-2: Использование реляционной СУБД PostgreSQL версии не ниже 13.
* CON-3: Необходима поддержка HTTPS протокола
* CON-4: Необходимость развертывания системы на серверах с минимальными характеристиками: 4 ядра CPU, 16 ГБ RAM, SSD-хранилище.

Организационные ограничения:

* CON-5: Соответствие требованиям GDPR по защите данных
* CON-6: Соблюдение корпоративных стандартов кодирования
* CON-7: Все сторонние библиотеки и фреймворки должны иметь лицензии, совместимые с корпоративными стандартами.
* CON-8: Регулярный аудит безопасности не реже одного раза в год.

Определим взаимодействие прототипов форм для работы программной системы (рис. 20).

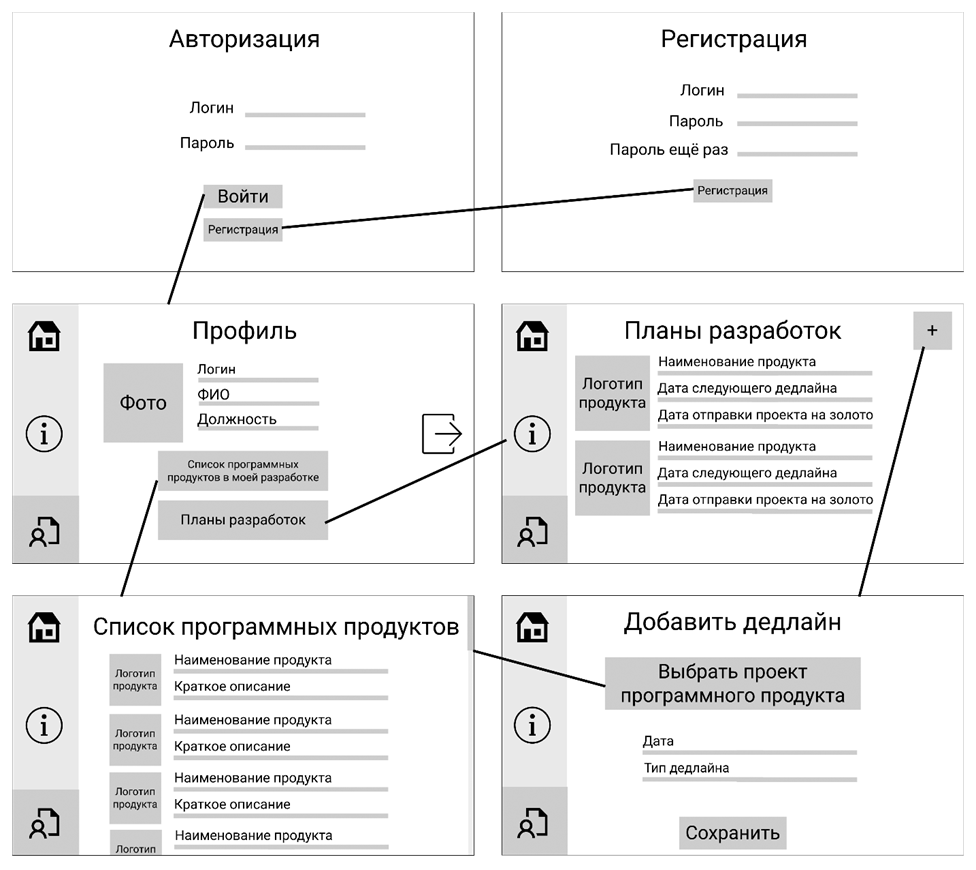


Рисунок 20 – Взаимодействие прототипов форм

## 3.3. Определение и документирование нефункциональных требований

Для определения и документирования нефункциональных требований построим диаграммы компонентов и размещения (рис. 21 и 22).



Рисунок 21 – Диаграмма компонентов

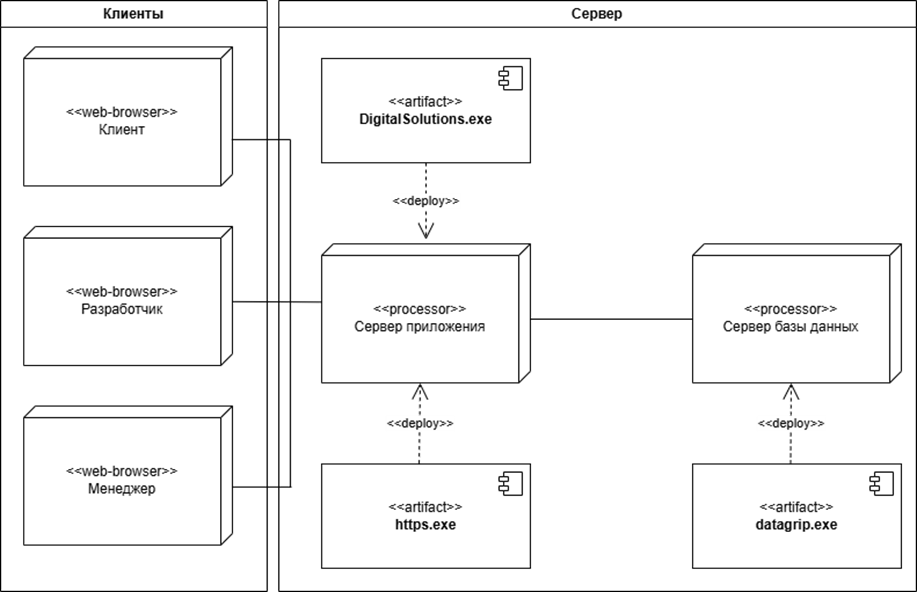


Рисунок 22 – Диаграмма размещения

Коммуникационные интерфейсы для системы управления разработками цифровой компании:

* + - 1. Сетевые протоколы:
* HTTPS (TLS 1.3): Защищенная передача данных между клиентом и сервером.
* WebSocket: Для двусторонней связи в режиме реального времени.
* HTTP/3: Оптимизация скорости взаимодействия между клиентом и сервером.
* SMTP/SMTPS: Для отправки email-уведомлений о статусах задач и проектах.
* OAuth 2.0: Для обеспечения безопасной аутентификации пользователей.

1. Форматы обмена данными:

* XML: Основной формат для передачи данных между внутренними и внешними системами.
* JSON: Для легковесных API-запросов, используемых клиентами.
* CSV: Для экспорта данных.
* PDF: Для генерации документов

1. Безопасность коммуникаций:

* SSL/TLS шифрование: Защита всех передаваемых данных (длина ключа 256 бит)
* JWT (JSON Web Token): Для авторизации и передачи пользовательских прав.
* Двухфакторная аутентификация: Для повышения уровня безопасности.
* Механизмы защиты от DoS-атак: Ограничение количества запросов от одного пользователя.
* Подпись и проверка целостности XML-документов: Для защиты от несанкционированных изменений данных.

1. Интеграционные интерфейсы:

* REST API: Для интеграции с внешними системами управления проектами.
* SOAP API: Для взаимодействия с корпоративными системами, где требуется повышенная надежность.
* Webhook: Для отправки событийных уведомлений
* OpenAPI спецификация: Для документирования всех доступных REST API.
* GraphQL: Для выборочного получения данных по сложным запросам.

1. Характеристики передачи:

* Максимальный размер payload: 15 МБ.
* Время отклика сервера: Не более 300 мс.
* Пропускная способность: До 150 Мбит/с.
* Сжатие данных: Поддержка gzip для уменьшения объема передаваемых данных.

1. Ограничения интерфейсов:

* Поддерживаемые форматы вложений: XML, JSON, CSV, PDF, PNG, JPG, DOC, DOCX.
* Максимальный размер файлов для загрузки: 10 МБ.
* Максимальное время установки соединения: 4 секунды.
* Автоматический разрыв соединений: Через 20 минут неактивности.

1. Механизмы синхронизации:

* NTP: Для синхронизации времени между сервером и клиентами.
* Очереди задач: Использование брокеров сообщений (например, RabbitMQ или Kafka) для управления обработкой данных.
* Кэширование: Снижение нагрузки на сервер через использование Redis.
* Репликация базы данных: Для обеспечения высокой доступности и отказоустойчивости.
* Синхронизация XML-документов: Механизм контроля версий и слияния изменений.

# Заключение

В заключение стоит подчеркнуть, что разработка и документирование требований к системе управления разработками для цифровой компании играет важную роль в создании эффективного и надежного инструмента. Анализ области применения, моделирование процессов и формулировка требований помогают создать систему, которая будет соответствовать нуждам компании, обеспечивать автоматизацию процессов разработки и повышать эффективность работы команды.

# Список использованной литературы

1. Шевчук, А. В. (2020). *Системный анализ и моделирование бизнес-процессов: теория и практика*. Москва: Инфра-М.  
   [Ссылка на ресурс: <https://www.infra-m.ru/>]
2. Обоснование и разработка требований к программным системам : учебное пособие / А. А. Бирюкова, А. М. Володина, К. В. Гусев, А. Н. Миронов. — Москва : РТУ МИРЭА, 2022. — 157 с. — Текст : электронный // Лань : электронно-библиотечная система. [Ссылка на ресурс: <https://e.lanbook.com/book/240089>]
3. Парахин, С. А. (2018). *Методы анализа и проектирования информационных систем*. Санкт-Петербург: БХВ-Петербург.  
   [Ссылка на ресурс: <https://bhv.spb.ru/>]
4. Кузнецов, И. Н. (2019). *Системный подход к разработке и внедрению информационных систем*. Москва: Эксмо.  
   [Ссылка на ресурс: <https://eksmo.ru/>]
5. Остроух, А.В. Интеллектуальные информационные системы и технологии : монография / А.В. Остроух, А.Б. Николаев. — Санкт-Петербург : Лань, 2019. — 308 с. — ISBN 978-5-8114-3409-1. [Ссылка на ресурс: <https://e.lanbook.com/book/115518>]
6. ISO/IEC 25010:2011. *Системы и программное обеспечение — Оценка качества и системное качество*. Доступно через: [https://www.iso.org](https://www.iso.org/)
7. Денисов, А. А. (2017). *Управление бизнес-процессами: основы и методы*. Москва: Юрайт.  
   [Ссылка на ресурс: <https://urait.ru/>]
8. Современные тенденции информационных технологий: автоматизация бизнес-процессов и управление логистикой. (2021). Журнал информационных технологий, 15(4), 45-60.  
   [Ссылка на журнал: [https://www.journalit.ru](https://www.journalit.ru/)]
9. Дьячков, Е. П. (2017). *Проектирование и внедрение ERP-систем на предприятиях*. Москва: Логистика и технологии.  
   [Ссылка на ресурс: <https://logistics-tech.ru/>]
10. Литвинов, А. Ю. (2018). *Анализ и моделирование производственных процессов*. Санкт-Петербург: СПбГУ.  
    [Ссылка на СПбГУ: <https://english.spbu.ru>]
11. Ахмедова, Х. Г. Написание и оформление курсовой работы по дисциплине «Обоснование и разработка требований к программной системе» : методические указания / Х. Г. Ахмедова, А. А. Бирюкова, А. В. Овсянникова. — Москва : РТУ МИРЭА, 2023. — 23 с. — Текст : электронный // Лань : электронно-библиотечная система. [Ссылка на ресурс: <https://e.lanbook.com/book/368720>]