



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Институт информационных технологий (ИИТ)
 Кафедра математического обеспечения и стандартизации информационных технологий
(МОСИТ)

ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ
по дисциплине «Обоснование и разработка требований к программным
системам»

Практическое занятие № 3
Вариант № 13. Обоснование и разработка требований к программной
системе управления разработками цифровой компании

Студент группы *ИКБО-65-23, Олефиров ГГ*

(подпись)

Преподаватель *Бирюкова А.А.*

(подпись)

Отчет представлен «____ » 202__ г.

Москва 2025 г.

1. Цель занятия

Научиться выявлять пользователей системы и преобразовывать потребности пользователей в письменные требования и диаграммы, пригодные для понимания, анализа и использования целевой аудиторией.

2. Постановка задачи

Задачами работы являются:

- разработать организационную диаграмму в методологии ARIS для выявления пользователей;
- построить диаграмму вариантов использования (Use Case diagram) в нотации UML;
- описать пользовательские истории;
- построить таблицу «события-реакция»;
- установить взаимосвязь между используемыми методами описания потребностей пользователей.

В первой практической работе были определены следующие классы пользователей:

Клиент, команда разработчиков, аналитики и специалисты по сбору требований.

3. Результат работы

3.1. Организационная модель предприятия

Организационная диаграмма в методологии ARIS, построенная для выявления пользователей программной системы представлена на рисунке 1.

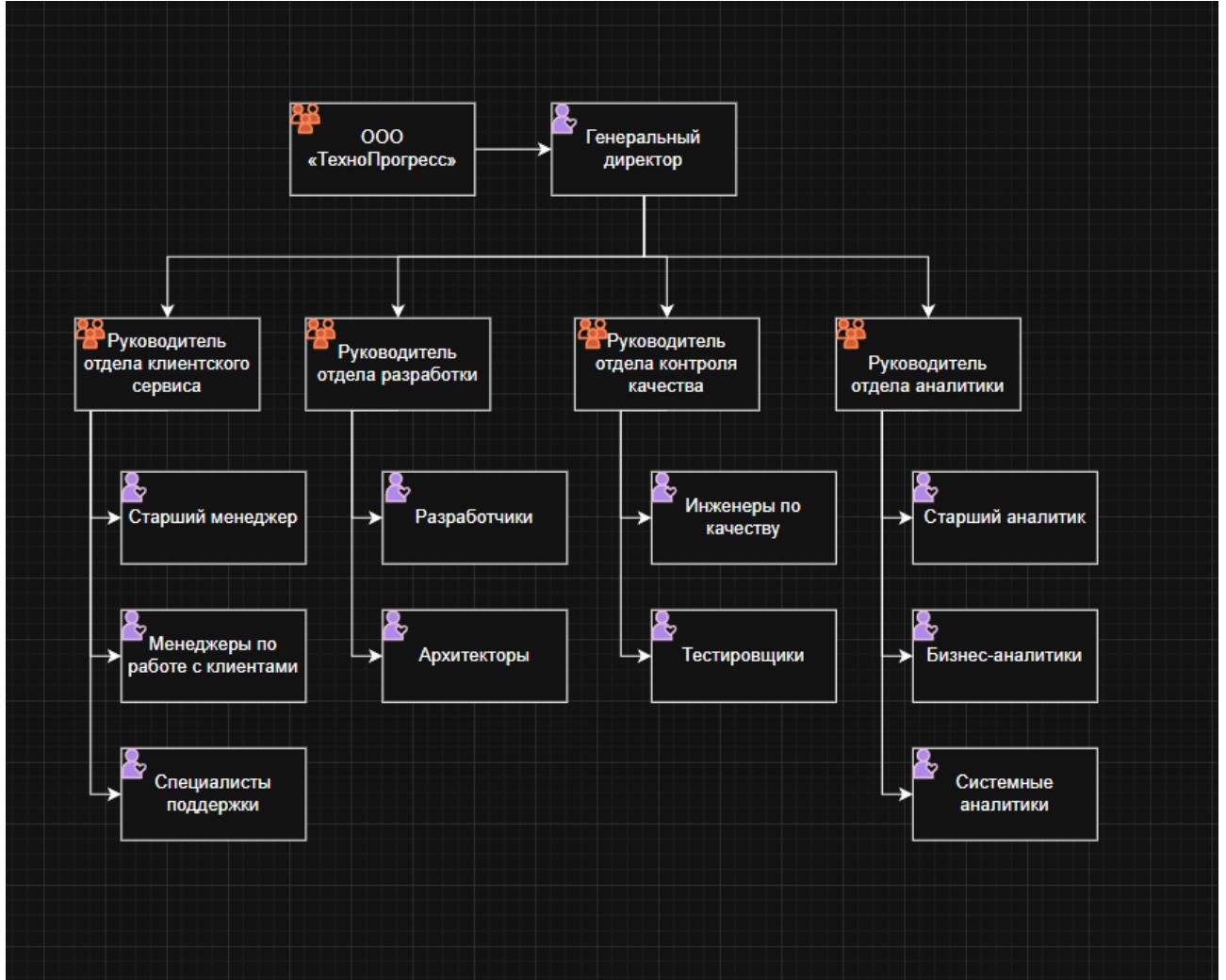


Рисунок 1 - Диаграмма организационной структуры

3.2. Определение пользовательских требований

3.2.1. Диаграмма вариантов использования (use case)

Диаграмма вариантов использования описывает функциональное назначение системы, т.е. то, что система будет делать в процессе своего функционирования.

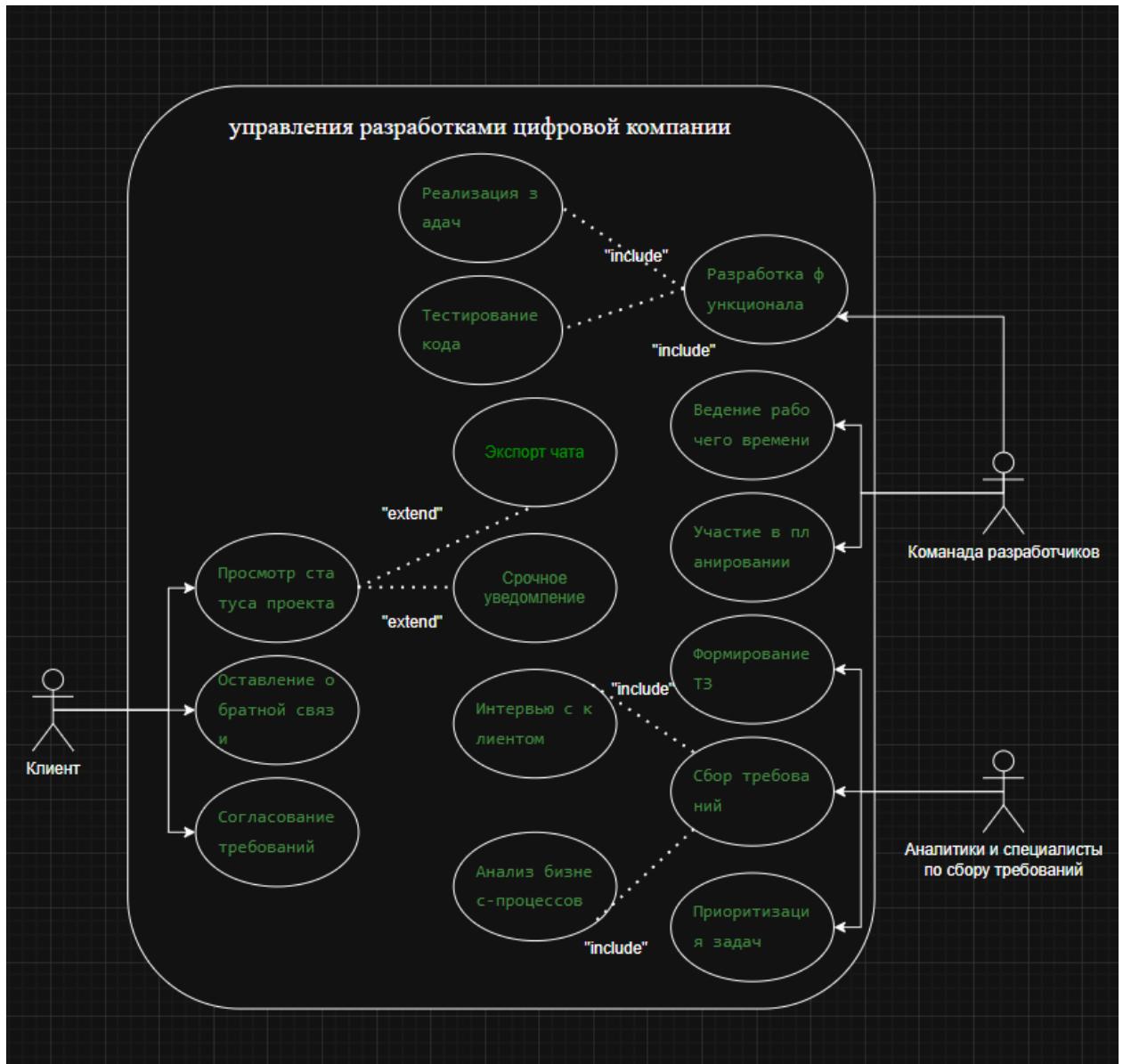


Рисунок 2 – Диаграмма вариантов использования для управления разработками цифровой компании

3.2.2. Пользовательские истории

Опишем потребности пользователей, используя пользовательские истории.

Как менеджер проектов, я хочу иметь возможность создавать задачи и назначать их конкретным разработчикам с установкой сроков выполнения, чтобы эффективно распределять workload команды [2].

Как разработчик, я хочу иметь возможность просматривать назначенные мне задачи и изменять их статус (в работе, выполнено, на проверке), чтобы команда видела актуальный прогресс по проекту [1].

Как тестировщик, я хочу иметь возможность создавать баг-репорты с прикреплением скриншотов и подробным описанием шагов воспроизведения, чтобы разработчики могли быстро понять и исправить проблему [2].

Как бизнес-аналитик, я хочу иметь возможность формировать технические задания на основе собранных требований и приоритизировать задачи по бизнес-ценности, чтобы разработка велась в соответствии с потребностями заказчика [1].

Как клиент, я хочу иметь возможность просматривать текущий статус проекта и этапы выполнения в реальном времени, чтобы быть в курсе прогресса и своевременно вносить корректировки [2].

Как руководитель, я хочу иметь возможность анализировать эффективность работы команд через автоматические отчеты и метрики, чтобы принимать обоснованные управленческие решения

Как клиент, я хочу иметь возможность экспортить отчет о статусе проекта в PDF формате, чтобы делиться информацией с другими stakeholders компании

3.3. Таблица событий и реакций

Опишем пользовательские требования с помощью определения внешних событий, на которые система должна реагировать.

№	Событие	Состояние системы	Реакция системы
1	Пользователь создает новую задачу	Система готова к работе, проект активен	Задача добавляется в бэклог, назначается ID, уведомление назначается ответственному [1]
2	Разработчик изменяет статус задачи на "Выполнено"	Задача в статусе "В работе"	Статус обновляется, уведомление отправляется тестировщику, задача перемещается в колонку "На тестировании"

3	Тестировщик создает баг-репорт	Задача в статусе "На тестировании"	Создается связанная задача-баг, уведомление отправляется разработчику, первоначальная задача возвращается в "В работе" [1]
4	Клиент запрашивает отчет по проекту	Проект активен, есть данные о задачах	Система формирует PDF отчет с текущими метриками, статусами задач и прогрессом
5	Наступает дедлайн задачи	Задача не завершена, срок истек	Система отправляет уведомление менеджеру и разработчику, задача помечается красным цветом
6	Менеджер назначает задачу исполнителю	Задача создана, но не назначена	Задача появляется в списке исполнителя, отправляется уведомление о новой задаче
7	Аналитик завершает сбор требований	Требования собраны, но не структурированы	Система предлагает шаблон ТЗ, создает карточку проекта с базовой информацией
8	Руководитель фильтрует отчет по датам	Система содержит данные за выбранный период	Формируется кастомный отчет с метриками за указанный период времени [2,]
9	Пользователь выполняет поиск по задачам	В системе есть активные и архивные задачи	Отображается список задач, соответствующих поисковому запросу, с возможностью фильтрации

Список использованных источников и литературы:

1. Ахмедова, Х. Г. Обоснование и разработка требований к программным системам : учебное пособие / Х. Г. Ахмедова. — Москва : РТУ МИРЭА, 2023. — 104 с. — ISBN 978-5-7339-1934-8. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/382694> (дата обращения: 22.09.2025). — Режим доступа: для авториз. пользователей.
2. Завьялов, А. В. Диаграммы UML для анализа и проектирования информационных систем : учебно-методическое пособие / А. В. Завьялов. — Москва : РТУ МИРЭА, 2021. — 65 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/218630> (дата обращения: 22.09.2025). — Режим доступа: для авториз. пользователей.