Федеральное государственное автономное

образовательное учреждение

Высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

институт

Программная инженерия

кафедра

**ОТЧЕТ О ПРАКТИЧЕСКОЙ РАБОТЕ №1**

Сбор и тестирование требований

тема

|  |
| --- |
| Преподаватель |
| подпись, дата                    инициалы, фамилия |
| Студент     КИ23-17/1Б, 32321533                                         Р. А. Карандаев |
| номер группы, зачетной книжки           подпись, дата                    инициалы, фамилия |
| Студент     КИ23-17/1Б, 32324850                                      М. В. Музалевский |
| номер группы, зачетной книжки           подпись, дата                    инициалы, фамилия |
|  |

Красноярск 2025

# Цель

Данная работа направлена на развитие комплексных навыков анализа, формализации и верификации требований, а также на отработку эффективного взаимодействия в команде при решении реальных задач проектирования. Студенты учатся выявлять существенные аспекты бизнес-потребностей, корректно структурировать и оформлять требования с применением стандартных моделей, что способствует более глубокому пониманию процессов коммуникации с заказчиком и повышения качества проектной документации.

Задачи

– Выбрать один вариант из списка предполагаемых бизнес-требований;

– Составить перечень вопросов для уточнения бизнес-требований и на их основе сформировать пользовательские требования (преподаватель выступает в роли представителя заказчика и отвечает на вопросы из списка);

– Формализовать пользовательские требования с применением UML-диаграммы прецедентов;

– Предоставить набор пользовательских требований своему коллеге для анализа и формирования замечаний;

– Скорректировать пользовательские требования с учетом замечаний, уточнив при необходимости у преподавателя недостающую информацию;

– Сформировать проектные требования и повторно предоставить своему коллеге для анализа и формирования замечаний;

– Устранить замечания и оформить отчет, содержащий информацию со всех этапов формирования требований.

# Ход выполнения

Был выбран второй вариант работы. Ниже скопирован его текст.

Приложение для планирования выполнения домашних заданий (в школе/университете и тд). Учет заданного, дата наиболее позднего возможного момента для начала выполнения, учет времени просрочки “дедлайна”. Каждому заданию выдавать приоритет.

Ниже представлено интервью с заказчиком.

1. Вопрос: "Какую основную проблему должно решать приложение?" Ответ: "Распределение времени на выполнение заданий".

2. Вопрос: "Какими основными функциями должно обладать приложение (минимально необходимый функционал)?" Ответ: "Создать задачу, назначить ей длительность, показ, когда человек должен начать выполнять задачу, чтобы успеть к указанному пользователем времени".

3. Вопрос: "Требуется ли регистрация и авторизация пользователей?" Ответ: "Да".

4. Вопрос: "Какого уровня защиты данных ожидается? (Нужны ли шифрование, соответствие GDPR и т.п.)" Ответ: "Нет".

5. Вопрос: "Будет ли приложение бесплатным, платным или условно-бесплатным (Freemium)?" Ответ: "Бесплатным".

6. Вопрос: "Рассматривается ли размещение рекламы?" Ответ: "Нет".

7. Вопрос: "На какую возрастную категорию (школьники, студенты вуза, аспиранты и т.д.) ориентировано приложение?" Ответ: "Пользоваться может любой".

8. Вопрос: "Насколько технически подкованы предполагаемые пользователи?" Ответ: "В состоянии заполнить форму, пользоваться Goggle календарем".

9. Вопрос: "На каких платформах (iOS, Android, Web) планируется выпуск приложения?" Ответ: "Web".

10. Вопрос: "Как должен определяться приоритет: вручную пользователем или автоматически по каким-то критериям (сложность, срочность и т.д.)?" Ответ: "Вручную".

11. Вопрос: "Сколько уровней приоритетов планируется: низкий/средний/высокий или более гибкая градация?" Ответ: "Пять уровней: очень высокий, высокий, нормальный, низкий, очень низкий".

12. Вопрос: "Какие статусы заданий необходимы? (Например, «Назначено», «В процессе», «Выполнено», «Просрочено» и т.д.)" Ответ: "К выполнению, выполняется, на паузе, отменено, выполнено, удалено".

13. Вопрос: "Хотите ли вы, чтобы приложение само рассчитывало «наиболее поздний момент начала выполнения» задачи на основе заданного дедлайна и оценочного времени выполнения?" Ответ: "Оно и должно этим заниматься".

14. Вопрос: "Нужно ли учитывать регулярные (повторяющиеся) задания?" Ответ: "Да, мы можем создавать регулярные, повторяющиеся задания, указав их длительность, интервалы между заданиями и с какого по какое число их повторять".

15. Вопрос: "Будет ли функциональность календаря (с визуализацией дат и дедлайнов) встроенной в приложение?" Ответ: "Да".

16. Вопрос: "Нужно ли автоматизировать смену статуса при наступлении дедлайна или окончании времени выполнения?" Ответ: "Нет, статус меняется вручную пользователем. Если дедлайн наступил, то система должна уведомить пользователя и предложить выбрать выполнено или невыполнено задание (присвоится статус отменено)".

17. Вопрос: "Как вы планируете использовать информацию о времени просрочки: для уведомлений, рейтингов, статистики?" Ответ: "Для уведомлений".

18. Вопрос: "Требуется ли какая-то специальная логика обработки просроченных заданий? (Например, отдельный список или выделение цветом)" Ответ: "Просроченные задания могут быть выделены цветом".

19. Вопрос: "Нужно ли учитывать время, выделяемое на внезапные обстоятельства (например, дополнительный буфер перед дедлайном)? Может ли пользователь самостоятельно регулировать этот буфер?" Ответ: "В таком случае мы ставим задачу на паузу, однако время пока задача на паузе не добавляется к дедлайну".

20. Вопрос: "Требуется ли учитывать тип задания (реферат, лабораторная работа, проект, тест и т.д.)? Планируется ли хранить разную дополнительную информацию в зависимости от типа?" Ответ: "Нет".

21. Вопрос: "Есть ли необходимость указывать, что выполнение одного задания зависит от завершения другого? (Например, «Часть 2» задачи может быть начата только после «Части 1».)" Ответ: "Да".

22. Вопрос: "Нужно ли позволять нескольким пользователям совместно работать над одним заданием (делить прогресс, видеть общие дедлайны)? Требуются ли уведомления для каждого участника группы об изменениях статуса?" Ответ: "Нет. У нас нет параллельности задач, нет параллельности пользователей".

23. Вопрос: "Нужно ли собирать статистику о том, сколько времени уходит на задания разных типов, сложностей, дисциплин и т.д.?" Ответ: "Можно сделать".

24. Вопрос: "Планируется ли система поощрений (очков, бейджей, рейтингов) за своевременное выполнение заданий?" Ответ: "Нет".

25. Вопрос: "Нужно ли импортировать задания из внешних источников (PDF-документов, LMS, электронных журналов)? Требуется ли возможность экспорта расписания или списка заданий в Excel, PDF или другие форматы?" Ответ: "Импорт задач из google календаря".

26. Вопрос: "Нужно ли настраивать язык интерфейса, поддержку нескольких языков?" Ответ: "Поддержка мультиязычности".

27. Вопрос: "Нужно ли интегрировать приложение с другими календарями (Google Calendar, Outlook и т.д.)?" Ответ: "Google, Yandex календарь".

28. Вопрос: "Требуются ли интеграции с системами обучения (например, Moodle, Google Classroom)?" Ответ: "Нет".

29. Вопрос: "Какие роли пользователей планируются? (Студент, Преподаватель, Администратор)" Ответ: "Ролей нет. Но есть отдельно админстратор".

30. Вопрос: "Нужно ли хранить вложения (файлы, ссылки, изображения), связанные с заданием?" Ответ: "Нужно".

31. Вопрос: "Нужно ли учитывать каникулы, праздничные и выходные дни, чтобы не ставить дедлайны на такие даты?" Ответ: "Нет. Это можно подсвечивать, но дедлайн можно поставить в такие дни".

32. Вопрос: "Где должны отображаться созданные задания?" Ответ: "Задания должны отображаться, как списком, так и календарем".

33. Вопрос: "Как будут располагаться добавленные задания?" Ответ: "Задания будут располагаться перед или на «наиболее позднем моменте начала выполнения», т.е. если мы добавим новое задание, то система будет пытаться расположить его среди единой линейной последовательности заданий как можно ближе к последнему моменту, когда его можно успеть начать и сделать (если смотреть на рассчетное время)".

34. Вопрос: "Если у нас будет такая ситуация, когда есть 2 задания с рассчетным временем в 6 месяцев и их дедлайн примерно в одно и то же время через 7 месяцев, то их же не получится поместить в этот временной промежуток, учитывая отсутствие параллельных задач?" Ответ: "В таком случае система должна уведомить пользователя, но при этом не запрещать такое делать (тогда получится, что начало выполнения более раннего задания уйдет в прошлое относительно текущей даты".

35. Вопрос: "Как будет влиять приоритет на порядок заданий?" Ответ: "Задание с более высоким приоритетом будет стоять раньше среди ряда заданий, однако все равно ряд заданий будет выровнен вправо к «наиболее поздним моментам начала выполнения»".

Цели проекта обозначены ниже.

Суть проекта

Приложение для планирования выполнения домашних заданий – это веб-инструмент, предназначенный для оптимизации распределения времени на выполнение учебных и иных заданий. Основные функции приложения включают.

1. Создание и управление заданиями: пользователь может создать задание, указав его название, описание, длительность, дедлайн, приоритет (по 5-уровневой шкале: очень высокий, высокий, нормальный, низкий, очень низкий) (можно глобально настроить как сильно приоритет влияет на эффективный дедлайн) и текущий статус (к выполнению, выполняется, на паузе, отменено, выполнено, удалено).

2. Автоматический расчет времени начала выполнения: система на основе заданного дедлайна, оценочного времени выполнения, приоритета и компоновки соседних заданий определяет «наиболее поздний момент начала выполнения», позволяющий успеть завершить задачу в срок.

3. Интеграция с календарями: возможность импорта заданий из Google и Yandex календарей, а также визуальное отображение задач как в календарном виде, так и в виде списка.

4. Поддержка повторяющихся заданий и зависимостей: пользователь может задать повторяемость задания, указать интервалы повторений, а также установить зависимость выполнения одного задания от завершения другого (отдельным экземплярам повторяющихся заданий можно также ставить зависимости).

5. Уведомления и визуальное выделение: система уведомляет пользователя о приближении дедлайнов и просроченных заданиях (задачи выделяются цветом), что помогает вовремя реагировать на изменения.

Метрики достижения целей

1. Точность расчета времени начала выполнения: не менее 90% случаев автоматического расчета должны соответствовать реальным требованиям для выполнения задания до дедлайна.

2. Сокращение времени планирования: время, затрачиваемое на распределение и планирование заданий, должно уменьшиться минимум на 50% по сравнению с традиционными методами ручного планирования.

3. Успешная интеграция с календарями: корректная синхронизация с Google и Yandex календарями, подтвержденная тестами импорта и экспорта данных, без потери информации.

4. Положительный пользовательский опыт: удовлетворенность пользователей (например, по результатам опросов) должна составлять не менее 80% положительных отзывов.

5. Низкий процент ошибок в логике планирования: количество ситуаций, когда система рекомендует старт выполнения, уже прошедший или не соответствующий дедлайну, должно быть менее 5% от общего числа расчетов.

Риски:

1. Некорректный расчет стартового времени: сложные зависимости между заданиями или неточная оценка длительности выполнения могут привести к неверному расчету «наиболее позднего момента начала выполнения».

2. Проблемы с интеграцией календарей: возможны сбои или ошибки при импорте/экспорте данных из внешних календарей, что может вызвать дублирование или потерю информации.

3. Перегруженность алгоритма планирования: большое количество заданий при ограниченном временном интервале может привести к конфликтам в расписании и вызвать путаницу у пользователя.

4. Проблемы пользовательского интерфейса: недостаточная интуитивность или избыточная функциональность могут снизить эффективность работы с приложением.

5. Ошибки при ручном изменении статусов: неверное или запоздалое изменение статусов заданий пользователем может нарушить логику работы системы и вызвать конфликтные ситуации в расписании.

6. Ситуации невозможности автоматического размещения заданий: при наличии конфликтующих дедлайнов система может не справиться с автоматическим распределением задач, что потребует вмешательства пользователя и может снизить доверие к функционалу автоматизации.

Список составленных требований написан ниже.

Бизнес-требования

БТ-1: Видение и цели проекта

БТ-2: Цели бизнеса

БТ-3: Ожидаемая ценность (бизнес-ценность)

Пользовательские требования

ПТ-1: Сценарии использования (Use Cases)

ПТ-2: Диаграмма сценариев использования (Use Case Diagram)

ПТ-3: Типы пользователей (User Types)

ПТ-4: Архетипы пользователей (User Personas)

ПТ-5: Истории пользователей (User Stories)

Функциональные требования

ФТ-1: Основные функции системы

ФТ-2: Входные и выходные данные

ФТ-3: Логика обработки данных

ФТ-4: Ограничения и правила работы системы

ФТ-5: Интеграционные требования

Нефункциональные требования

НФТ-1: Производительность (Performance)

НФТ-2: Безопасность (Security)

НФТ-3: Надежность и отказоустойчивость (Reliability & Fault Tolerance)

НФТ-4: Удобство использования (Usability)

НФТ-5: Масштабируемость (Scalability)

НФТ-6: Поддерживаемость (Maintainability)

НФТ-7: Совместимость (Compatibility)

НФТ-8: Соответствие стандартам (Compliance)

Системные требования

СТ-1: Аппаратные требования

СТ-2: Программные зависимости

СТ-3: Сетевые требования

СТ-4: Ограничения платформы

СТ-5: Требования к развертыванию

Правовые и нормативные требования

ПиНТ-1: Соответствие законодательству

ПиНТ-2: Стандарты индустрии

ПиНТ-3: Политики конфиденциальности и защиты данных

# Вывод

В результате выполнения практической работы студенты получили практический опыт системного подхода к сбору и обработке требований, научились задавать целенаправленные вопросы для уточнения потребностей и формировать корректные пользовательские и проектные спецификации. Отчет отражает важность итеративного взаимодействия и обмена обратной связью в процессе работы, демонстрируя, что качественная формализация требований является ключевым элементом успешной реализации проекта. Полученные навыки и умения являются важным шагом в профессиональном развитии и могут быть успешно применены в реальных условиях разработки программного обеспечения.