

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего образования
«Северо-Кавказский федеральный университет»

Колледж СКФУ в г. Ставрополе

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ
для практических занятий**

ОП. 08 Управление ИТ-проектами

Специальность	09.02.11 Разработка и управление программным обеспечением
Форма обучения	очная

Ставрополь, 2026 г.

Методические указания по выполнению практических работ составлены в соответствии с ФГОС и рабочей программой учебной дисциплины ОП.08 Управление ИТ-проектами 09.02.11 Разработка и управление программным обеспечением.

Разработчик:

Дечева А.В., преподаватель колледжа СКФУ в г. Ставрополе

1. Пояснительная записка

Методические указания по организации и проведению практических занятий составлены в соответствии с ФГОС СПО по специальности 09.02.11 Разработка и управление программным обеспечением и рабочей программой учебной дисциплины ОП.08 Управление ИТ-проектами.

Целью выполнения практических занятий является систематизация и закрепление теоретических знаний и формирование практических умений.

Особое значение для усвоения содержания дисциплины и привития практических навыков имеет правильная и четкая организация проведения и выполнения студентами практических работ под контролем преподавателя.

Перед началом выполнения каждой работы студенты должны ознакомиться с ее основными положениями, порядком выполнения работы.

По каждому практическому занятию предусматривается индивидуальный отчет перед преподавателем.

В результате освоения дисциплины обучающийся **должен знать:**

- основные этапы разработки и реализации проекта;
- психологические основы деятельности коллектива;
- создавать и управлять ролями и правами доступа к данным;
- анализировать требования к программному обеспечению и составлять планы тестирования;
- принципы работы информационных систем;
- современные технологии и методы модернизации информационных систем.

В результате освоения дисциплины обучающийся **должен уметь:**

- находить интересные проектные идеи, грамотно их формулировать и документировать;
- оценивать жизнеспособность проектной идеи, составлять план проекта;
- организовывать работу коллектива и команды;
- участвовать в проекте по модернизации информационной системы компании;
- анализировать текущее состояние информационной системы и выявить ее слабые места;
- участвовать в проекте по внедрению новых технологий в информационную систему компании;
- формировать и представлять отчетности о подготовке к выполнению задания на тестирование ПО в соответствии с установленными регламентами.

2. Перечень практических занятий

Наименование практического занятия	Количество часов
Раздел 1. Основы управления проектами и методологии	6
Практическое занятие №1. Анализ примеров ИТ-проектов.	2
Практическое занятие №2. Выбор методологии под проект. Сравнительный анализ методологий управления проектами.	2
Практическое занятие № 3. Разработка проектной документации. Знакомство с программным обеспечением для управления проектами.	2
Раздел 2. Планирование и выполнение ИТ-проектов	18
Практическое занятие №4. Подготовка деловой беседы.	2
Практическое занятие №5. Формирование бэклога продукта и написание User Stories.	2
Практическое занятие №6. Разработка технического задания.	2
Практическое занятие №7. Создание диаграммы Ганта.	2
Практическое занятие №8. Составление бюджета проекта.	2
Практическое занятие № 9. Выполнение SWOT-анализа.	2
Практическое занятие №10. Разработка плана управления рисками.	2
Практическое занятие №11. Планирование задач в Scrum.	2
Практическое занятие №12. Ведение отчетности.	2
Раздел 3. Мониторинг, контроль и завершение проекта	16
Практическое занятие №13. Анализ отклонений.	2
Практическое занятие №14. Корректировка проекта.	2
Практическое занятие №15. Разработка плана передачи проекта в эксплуатацию и миграции данных.	2
Практическое занятие №16. Контроль качества продукта.	2
Практическое занятие №17. Подготовка итоговой документации.	2
Практическое занятие №18. Презентация проекта.	2
Практическое занятие №19. Анализ работы команды.	2
Практическое занятие №20. Защита итогового проекта.	2
Итого:	40

Раздел 1. Основы управления проектами и методологии

Тема 1.1. Введение в управление проектами

Практическое занятие №1. Анализ примеров ИТ-проектов.

Цель работы: Научиться определять признаки проекта, анализировать факторы успеха и причины провала ИТ-проектов, работать в команде.

Оснащение: раздаточный материал с кейсами, шаблоны таблиц, ручка, бумага для заметок.

Краткие теоретические сведения

Проект — это временное предприятие, направленное на создание уникального продукта, услуги или результата.

Признаки проекта: ограниченность во времени, уникальность результата, ограниченность ресурсов (люди, бюджет).

Успех проекта оценивается по трём критериям (треугольник качества):

- соблюдение сроков,
- соблюдение бюджета,
- соответствие результата требованиям.

Типичные причины неудач ИТ-проектов:

- нечёткие или постоянно меняющиеся требования;
 - недостаточная коммуникация в команде и с заказчиком;
 - технические ошибки и недостаточное тестирование;
 - сопротивление пользователей;
 - нереалистичные сроки и бюджет.
-

Ход работы

Этап 1. Анализ общего кейса (работа в группах)

Кейс №1. «Разработка сайта для стоматологии»

Небольшая веб-студия взялась разработать сайт для стоматологической клиники с функцией онлайн-записи. Срок – 2 недели, бюджет – 50 000 руб. Через неделю работы заказчик попросил добавить интернет-магазин средств гигиены. Исполнитель согласился, не меняя сроки и бюджет. В итоге сайт был сдан с опозданием на месяц, с ошибками в модуле оплаты. Заказчик отказался платить, студия понесла убытки.

Задание:

Прочитайте кейс, обсудите в группе и заполните таблицу 1.

Таблица 1 – Анализ кейса №1

Вопрос для анализа	Ответ
Можно ли назвать эту деятельность проектом? Почему?	
Кто выступает заказчиком, а кто исполнителем?	
Перечислите основные ошибки исполнителя (не менее 3)	
Что нужно было сделать, чтобы проект стал успешным?	

Этап 2. Анализ индивидуального кейса

Каждая группа получает от преподавателя карточку с описанием реального или учебного ИТ-проекта. Ваша задача – проанализировать ситуацию и заполнить таблицу 2. Определите, успешен проект или провален, и почему.

Таблица 2 – Анализ кейса №_____
(укажите номер или название вашего кейса)

Параметр анализа	Описание
Название / тип проекта	
Заказчик / пользователи	
Цель проекта (что хотели получить?)	
Полученный результат	
Проблемы, возникшие в ходе реализации	
Главная причина успеха / провала (одна ключевая)	
Ваш вердикт: успех / провал / спорный	

Карточки для групп

Карточка А. Внедрение ERP-системы на заводе

Крупный завод решил автоматизировать управление ресурсами и заказал внедрение ERP-системы у известного интегратора. Проект длился 2 года, бюджет был превышен в 1,5 раза. Система была установлена, но сотрудники бухгалтерии и склада отказались в ней работать, так как она была неудобной и требовала двойного ввода данных. В итоге завод продолжает работать в Excel, а система используется лишь для отчетности перед руководством.

Карточка Б. Разработка мобильной игры

Инди-команда из 3 человек решила создать простую игру-головоломку. Они четко определили концепцию, сделали прототип за месяц, протестировали на знакомых, учли пожелания и выпустили игру в сторас за 3 месяца. Бюджет составил 50 тыс. руб. (свои сбережения). Игру скачали 100 тыс. раз за первый месяц, разработчики окупили затраты и получили прибыль.

Карточка В. Создание корпоративного портала

IT-отдел крупной компании начал разработку внутреннего портала для сотрудников. Заказчиком выступило руководство, которое постоянно меняло требования: сначала нужен был только документооборот, потом добавили соцсеть, потом календарь мероприятий. Сроки сдвигались 4 раза. Через год портал запустили, но сотрудники им не пользовались, потому что интерфейс был сложным, а многие функции не работали стабильно.

Карточка Г. Система видеонаблюдения в кампусе

Руководство колледжа решило установить систему видеонаблюдения для безопасности. Был проведен тендер, выбрана компания. Перед началом работ компания предложила сделать пилотную зону в одном корпусе, чтобы проверить работу камер и учесть пожелания охраны. После успешного пилота систему масштабировали на все корпуса. Проект сдан в срок, бюджет соблюден, охрана и студенты довольны.

Этап 3. Презентация результатов

Каждая группа выступает с кратким (2–3 минуты) сообщением о своём кейсе. Необходимо озвучить:

- какой проект анализировали;
- успешен он или нет;
- какая главная причина привела к такому итогу;
- один важный урок, который можно извлечь из этого проекта.

Остальные группы могут задавать вопросы.

Этап 4. Оформление отчета

Оформите результаты работы в виде письменного отчета, который должен содержать:

- титульный лист (название работы, ФИО участников группы);
- заполненную таблицу 1;
- заполненную таблицу 2 (по вашему индивидуальному кейсу);
- вывод по работе (2–3 предложения о том, что нового узнали и какие навыки приобрели).

Контрольные вопросы

1. Назовите три основных признака проекта.
2. Перечислите возможные причины провала ИТ-проектов.
3. Почему важно фиксировать требования заказчика?
4. Что такое «треугольник качества» проекта?

Этап 5. Найти информацию

Найдите информацию об известном провальном ИТ-проекте (например, Apple Maps, Google+, запуск [Healthcare.gov](https://www.healthcare.gov/) и др.). Письменно ответьте на вопрос: «Какая из причин, обсуждавшихся на занятии, стала главной в этом провале?» (3–5 предложений).

Тема 1.2. Методологии и подходы к управлению проектами

Практическое занятие №2. Выбор методологии под проект. Сравнительный анализ методологий управления проектами.

Цель работы

Научиться различать основные методологии управления ИТ-проектами, понимать их сильные и слабые стороны, уметь выбирать подходящую методологию в зависимости от характеристик проекта.

Краткие теоретические сведения

Методология управления проектом – это совокупность принципов, практик и процедур, определяющих, как проект будет реализован от начала до конца.

Наиболее распространённые методологии в ИТ-сфере:

1. Waterfall (Водопад, каскадная модель)

- Проект разбивается на последовательные этапы (требования → проектирование → разработка → тестирование → внедрение).
- Переход на следующий этап только после полного завершения предыдущего.
- Требования чётко фиксируются в начале.
- *Когда применять:* проект с ясными, стабильными требованиями; небольшие проекты; когда качество документации критично.

2. Agile (Гибкие методологии)

- Итеративный подход: работа ведётся короткими циклами (спринтами/итерациями), после каждого цикла получается работающий продукт.
- Требования могут меняться в процессе.
- Тесное взаимодействие с заказчиком.
- *Когда применять:* проекты с неопределёнными или изменяющимися требованиями; инновационные разработки; стартапы.

○ **Scrum** – подвид Agile: роли (Product Owner, Scrum Master, команда), события (спринт, daily scrum, обзоры, ретроспективы), артефакты (бэклог продукта, бэклог спринта).

○ **Kanban** – подвид Agile: визуализация потока работ на доске, ограничение количества задач на этапах (WIP), непрерывная поставка.

3. Гибридные модели

– Сочетают элементы Waterfall (на уровне планирования) и Agile (на уровне разработки). Часто применяются в крупных компаниях с жёсткими требованиями к документации.

Критерии выбора методологии:

- Степень определённости требований.
- Размер и распределённость команды.
- Частота поставки продукта пользователям.
- Уровень вовлечённости заказчика.
- Критичность документации и формальной отчётности.

Ход работы

Работа выполняется в группах по 3–4 человека (допускается индивидуальное выполнение по решению преподавателя). Все ответы фиксируются в отчёте.

Этап 1. Сравнительный анализ методологий

Задание 1. Изучите описание методологий (Waterfall, Scrum, Kanban) и заполните сравнительную таблицу 1.

Таблица 1 – Сравнение методологий управления проектами

Критерий сравнения	Waterfall	Scrum	Kanban
Степень гибкости (возможность менять требования)			
Частота поставки результата заказчику			
Требования к размеру команды			
Роли в команде			
Необходимость детального планирования в начале			
Документирование (насколько обязательно)			
Основные артефакты (что используется)			
Примеры проектов, для которых подходит			

Примечание: для заполнения можно использовать информацию выше или дополнительные источники

Этап 2. Анализ ситуаций и выбор методологии

Задание 2. Для каждого из приведённых ниже проектов определите наиболее подходящую методологию управления (Waterfall, Scrum, Kanban или их комбинацию). Обоснуйте свой выбор в 2–3 предложениях, указав ключевые факторы.

Кейс 2.1. Разработка системы управления складом для государственного предприятия

Требования жёстко регламентированы государственным стандартом, все детали описаны

в техническом задании, подписанном заказчиком. Изменения в процессе разработки маловероятны. Бюджет фиксирован, сроки сдачи – 31 декабря текущего года. Команда – 10 человек, ранее работавших по каскадной модели.

Ваш выбор:

Обоснование:

Кейс 2.2. Мобильное приложение для стартапа (доставка еды)

Идея приложения только сформирована, но требования постоянно уточняются на основе обратной связи от первых пользователей. Запустить минимально жизнеспособный продукт (MVP) нужно через 2 месяца. Команда – 5 разработчиков, дизайнер, владелец продукта активно участвует.

Ваш выбор:

Обоснование:

Кейс 2.3. Поддержка и развитие интернет-магазина

Интернет-магазин уже работает. Задачи поступают хаотично: то срочно исправить баг, то добавить небольшую функциональность, то обновить дизайн. Важно быстро реагировать на запросы, но строгих спринтов нет, приоритеты могут меняться каждый день.

Ваш выбор:

Обоснование:

Кейс 2.4. Создание бортового программного обеспечения для спутника

Требования строго фиксированы, ошибки недопустимы. Каждая строка кода должна быть документирована и проверена. Заказчик – космическое агентство, которое требует полного соответствия стандартам и отчётам на каждом этапе. Изменения требований практически исключены.

Ваш выбор:

Обоснование:

Этап 3. Проектирование методологии для учебного проекта

Задание 3. Представьте, что ваш колледж решил разработать мобильное приложение для студентов (расписание, новости, заявки в техподдержку). Требования на старте не до конца ясны, но есть общее видение. Команда – 4 студента старших курсов (вы), куратор от преподавателей. Бюджет – 0 руб., срок – 4 месяца.

Ответьте на вопросы:

1. Какую методологию вы выберете? Почему?
2. Какие роли нужно определить в команде?
3. Как часто вы будете показывать промежуточные результаты заказчику (куратору)?
4. Как будете отслеживать прогресс?

Этап 4. Оформление отчёта

Отчёт должен содержать:

- Титульный лист с указанием темы, ФИО участников группы.
 - Заполненную таблицу 1.
 - Ответы на кейсы 2.1–2.4 (с обоснованием).
 - Ответы на вопросы задания 3.
 - Вывод по работе (что нового узнали, какой опыт приобрели).
-

Контрольные вопросы

1. Чем отличается Scrum от Kanban?
2. В каких случаях каскадная модель предпочтительнее Agile?
3. Что такое MVP и почему его важно выпускать как можно раньше?
4. Можно ли комбинировать Waterfall и Agile? Приведите пример.
5. Назовите основные роли в Scrum и их зоны ответственности.

Дополнительное задание

Нарисуйте схему жизненного цикла проекта (от идеи до релиза) по выбранной вами методологии для проекта из задания 3. Схема может быть выполнена от руки и сфотографирована или создана в любом графическом редакторе. Поясните основные этапы.

Тема 1.3. Документация и инструменты для управления проектной деятельностью

Практическое занятие №3. Разработка проектной документации. Знакомство с программным обеспечением для управления проектами.

Цель работы

Научиться разрабатывать основные документы ИТ-проекта (техническое задание, протоколы собраний, отчёты) и познакомиться с программными инструментами для управления проектами (российские аналоги Jira, Trello, MS Project).

Ход работы

Этап 1. Разработка фрагментов проектной документации

Задание 1.1. Составление Технического задания (ТЗ)

Изучите структуру технического задания. Для приведённого ниже описания проекта разработайте разделы ТЗ:

Описание проекта: Мобильное приложение «CollegeHelper» для студентов колледжа. Приложение должно отображать расписание занятий (с возможностью выбора группы), новости колледжа, позволять подавать заявки в техподдержку (сломанный принтер, проблема в аудитории) и просматривать свою успеваемость (оценки). Заказчик — администрация колледжа. Пользователи — студенты и преподаватели.

Заполните таблицу:

Раздел ТЗ	Содержание (ваш вариант)
Цель проекта	
Функциональные требования (перечислить 5–7)	
Нефункциональные требования (платформа, безопасность, производительность)	
Ограничения проекта (сроки, бюджет, технологии)	
Критерии приемки (как поймём, что проект готов)	

Задание 1.2. Оформление протокола собрания

Представьте, что в вашей команде прошло собрание по проекту «CollegeHelper». Используя шаблон, составьте протокол встречи .

Исходные данные:

- Дата собрания: сегодняшняя
- Присутствуют: Руководитель проекта (вы), Аналитик (Иванов), Разработчик (Петров), Тестировщик (Сидорова)
- Повестка: 1. Утверждение функциональных требований; 2. Распределение задач на спринт; 3. Сроки первой версии (MVP)
- Решения: Добавить функцию авторизации через госуслуги; MVP выпустить через 3 недели; Иванову доработать ТЗ; Петрову настроить репозиторий; Сидоровой подготовить план тестирования

Шаблон протокола:

text

Дата: _____

Присутствуют: _____

Отсутствуют: _____

1. Начало собрания в _____

Повестка:

1.1 _____

1.2 _____

1.3 _____

2. Ход собрания:

По пункту 1.1: _____

По пункту 1.2: _____

По пункту 1.3: _____

3. Принятые решения и задачи:

- Кому: _____ Задача: _____ Срок: _____

- Кому: _____ Задача: _____ Срок: _____

- Кому: _____ Задача: _____ Срок: _____

4. Окончание собрания в _____

Задание 1.3. Индивидуальный отчёт разработчика

Заполните отчёт о своей работе за один день (предположите, что вы разработчик в проекте). Используйте формат и аббревиатуры из примера .

Аббревиатуры видов деятельности:

- ME — Встречи
- LC — Другие виды общения
- PP — Составление плана проекта
- PR — Проектирование
- CO — Кодирование
- TE — Тестирование
- DO — Документирование
- RE — Разработка требований

Формат отчёта:

text

<дата> <вид деятельности> <время> <описание>

Пример:

text

15.03.2026 МЕ 1.5 Планирование спринта с командой

15.03.2026 СО 4.0 Разработка модуля авторизации

15.03.2026 ДО 1.0 Написание комментариев к коду

Ваш отчёт (минимум 3 записи):

text

Этап 2. Знакомство с программным обеспечением

Задание 2.1. Анализ инструментов

Изучите таблицу с описанием программных продуктов. Для каждого инструмента определите, для каких типов проектов он подходит лучше всего, и заполните последний столбец.

Инструмент	Тип	Основные возможности	Для каких проектов подходит
Projecto	Универсальная платформа	Задачи, база знаний, Kanban, кастомные статусы	
EvaProject	Agile-платформа	Scrum/Kanban доски, бэклоги, диаграммы Ганта, учёт времени	
Naumen Project Ruler	Комплексная система	Управление рисками, ресурсами, изменениями, Agile и Waterfall	
WEEEEK	Лёгкое решение	Доски, календари, списки, бесплатный тариф	
MS Project	Планирование	Диаграммы Ганта, календарное планирование, ресурсы	
Strive	Для разработки	Спринты, Agile, оценки задач, метрики	

Задание 2.2. Выбор инструмента для своего проекта

Представьте, что вы руководитель проекта «CollegeHelper». Команда — 4 человека (вы, аналитик, разработчик, тестировщик). Проект будет вестись по Scrum: спринты по 2 недели, ежедневные стендапы, планирование в начале спринта. Бюджет на инструменты — 0 рублей (нужны бесплатные или условно-бесплатные решения).

Ответьте на вопросы:

1. Какой инструмент (или комбинацию инструментов) вы выберете для управления задачами? Почему?
2. Какой инструмент вы выберете для хранения документации (ТЗ, протоколы, отчёты)?

3. Будете ли вы использовать инструмент для планирования сроков (диаграмма Ганта)? Если да, то какой?
 4. Как вы организуете доступ команды к инструментам?
-

Задание 2.3. Работа в инструменте (по возможности)

Если у вас есть доступ к интернету и преподаватель предоставил доступ к одной из систем (например, WEEEEK, Kaiten, YouGile или демо-версии Projecto), выполните следующие действия:

1. Создайте проект «CollegeHelper».
 2. Создайте доску (Kanban или Scrum).
 3. Добавьте колонки: «Бэклог», «В работе», «На проверке», «Готово».
 4. Создайте 5 задач (например: «Разработать модуль расписания», «Сверстать экран новостей», «Настроить базу данных», «Написать тесты для авторизации», «Создать макет главного экрана»).
 5. Назначьте исполнителей (можно вымышленных).
 6. Установите сроки для задач.
 7. Сделайте скриншот получившейся доски и вставьте в отчёт (если работаете в аудитории) или опишите, какие действия выполняли.
-

Этап 3. Оформление отчёта

Отчёт должен содержать:

- Титульный лист с указанием темы, ФИО студента (или участников группы).
 - Результаты выполнения Задания 1.1 (таблица с ТЗ).
 - Результаты выполнения Задания 1.2 (заполненный протокол собрания).
 - Результаты выполнения Задания 1.3 (индивидуальный отчёт).
 - Заполненную таблицу из Задания 2.1.
 - Ответы на вопросы Задания 2.2.
 - Скриншот или описание работы в инструменте (Задание 2.3).
 - Вывод по работе: что нового узнали, какие навыки приобрели, какой инструмент показался наиболее интересным и почему.
-

Контрольные вопросы

1. Какие документы обязательно должны быть в любом ИТ-проекте?
 2. Чем техническое задание отличается от устава проекта?
 3. Для чего нужны протоколы собраний? Кто их должен вести?
 4. Какие российские аналоги Jira вы знаете? В чём их преимущества?
 5. Что такое диаграмма Ганта и в каких инструментах она реализована?
 6. Можно ли управлять проектом без специального ПО? Какие риски?
-

Дополнительное задание

1. Зарегистрируйтесь в одном из бесплатных инструментов (WEEEEK, Kaiten, YouGile или Trello). Создайте личный проект (например, «Подготовка к экзаменам» или «Планирование путешествия») с задачами, сроками и исполнителями. Пришлите скриншот доски.

2. Найдите в интернете пример реального технического задания (можно фрагмент). Проанализируйте его структуру и сравните с тем, что разработали на занятии. Кратко опишите различия.

Раздел 2. Планирование и выполнение ИТ-проектов

Тема 2.1. Планирование проекта

Практическое занятие №4. Подготовка деловой беседы.

Цель работы

Научиться планировать и готовить деловую беседу с заказчиком или партнёром по ИТ-проекту, освоить структуру подготовки, формулировки целей, технику постановки вопросов и документирование результатов встречи.

Краткие теоретические сведения

Деловая беседа – это форма межличностного общения, основанная на обмене мнениями и информацией между двумя или несколькими собеседниками. В отличие от переговоров, деловая беседа не обязательно завершается заключением договора или подписанием обязательных документов, но может предварять переговоры или быть их частью.

Основные функции деловой беседы :

- взаимное общение работников из одной деловой среды;
- совместный поиск, выдвижение и оперативная разработка идей и замыслов;
- контроль и координирование уже начатых мероприятий;
- поддержание деловых контактов;
- стимулирование деловой активности.

Три этапа подготовки и проведения деловой беседы :

1. **Докоммуникативный этап** – планирование, оценка собеседников и обстановки, сбор материала, анализ и редактирование, подготовка тезисов.
2. **Коммуникативный этап** – собственно беседа (установление контакта, изложение позиции, выяснение позиции собеседника, совместный анализ проблемы).
3. **Посткоммуникативный этап** – анализ результатов, фиксация договорённостей.

Структура деловой беседы (пять фаз) :

1. Начало беседы (установление контакта, создание рабочей атмосферы).
2. Передача информации.
3. Аргументирование.
4. Опровержение доводов собеседника.
5. Принятие решений.

Модель GROW для структурирования беседы :

- **G (Goal)** – определить цель встречи;
 - **R (Reality)** – оценить текущую ситуацию;
 - **O (Options)** – обсудить возможные варианты;
 - **W (Wrap-up)** – зафиксировать договорённости и следующие шаги.
-

Ход работы

Этап 1. Анализ понятия и структуры деловой беседы

Задание 1.1. Изучите приведённую ниже таблицу. На основе теоретического материала заполните пустые ячейки.

Характеристика	Деловая беседа	Переговоры
Основная цель	Обмен мнениями, информирование, поиск идей	
Результат	Договорённости (не всегда юридически оформленные)	
Обязательность документирования	По необходимости, в произвольной форме	
Пример в ИТ-проекте	Встреча с заказчиком для уточнения требований к интерфейсу	

Задание 1.2. Прочитайте описание этапов подготовки деловой беседы.

Расположите действия в правильной последовательности (пронумеруйте от 1 до 5):

Действие	Порядковый номер
Сбор материала (поиск информации о собеседнике, компании, предмете обсуждения)	
Редактирование текста, окончательная шлифовка тезисов	
Планирование (определение темы, целей, участников, времени и места)	
Анализ собранного материала, выделение главного	
Систематизация материала, раскладка по "ячейкам" плана	

Этап 2. Планирование деловой беседы (работа с кейсом)

Задание 2.1. Прочитайте описание ситуации.

Ситуация: Вы — руководитель небольшой ИТ-студии (3–5 человек). К вам обратился потенциальный заказчик — владелец сети кофеен «Кофе&Книги». Он хочет обсудить разработку мобильного приложения для программы лояльности (накопление бонусов, push-уведомления о скидках, возможность оплаты через приложение). Заказчик назначает встречу на следующей неделе. О компании заказчика известно: сеть насчитывает 7 точек в городе, активно развивается, владелец — молодой предприниматель, не разбирается в технических деталях, но хочет "современное и удобное приложение, как у крупных сетей".

Используя чек-лист подготовки к деловой беседе, выполните задания:

1. Сформулируйте цель предстоящей беседы (заполните пропуски):

Цель моей встречи:

○ Узнать

○ Выяснить

- Договориться о

-
- Понять
-

Проверьте себя: цель должна быть конкретной и измеримой .

2. Определите, какую информацию о собеседнике и его компании нужно собрать ДО встречи (составьте список из 5–7 пунктов, которые помогут вам лучше подготовиться).

3. Подготовьте вопросы к заказчику. Используйте технику открытых и закрытых вопросов. Сформулируйте по 2–3 вопроса каждого типа:

Тип вопроса	Примеры вопросов
Открытые вопросы (начинаются с "что", "как", "почему", "расскажите")	1. 2. 3.
Закрытые вопросы (требуют ответа "да"/"нет" или выбора варианта)	1. 2. 3.

4. Спланируйте структуру беседы по фазам. Заполните таблицу:

Фаза беседы	Что вы будете делать/говорить?
1. Начало (установление контакта, первые 2–3 минуты)	
2. Передача информации (что расскажете о себе и студии)	
3. Аргументирование (почему стоит работать с вами)	
4. Опровержение доводов (как ответите на возможные возражения)	
5. Завершение (что предложите в качестве следующих шагов)	

5. Подготовьте аргументы. Заказчик может сомневаться: "У вас маленькая студия, справитесь ли вы?" или "Почему вы, а не известная компания?". Напишите 2–3 аргумента в свою пользу.

- Аргумент 1:

-
- Аргумент 2:

-
- Аргумент 3:
-

6. Определите место и время встречи. Опишите, где и когда вы предложите встретиться и почему выбрали именно эти параметры.

Этап 3. Приёмы начала беседы

Задание 3.1. Изучите четыре приёма начала беседы:

Приём	Суть приёма
Снятие напряжённости	Комплименты, шутка, тёплые слова, чтобы растопить лёд
"Зацепка"	Краткое изложение ситуации или проблемы, связанной с темой беседы, необычный факт, личное впечатление
Стимулирование воображения	Постановка в начале беседы множества вопросов по проблеме
Прямой подход	Непосредственный переход к делу без вступления (для кратких контактов)

Для ситуации из Задания 2.1 придумайте по одному варианту начала беседы для каждого приёма:

- **Снятие напряжённости:**

" _____ "

- **"Зацепка":**

" _____ "

- **Стимулирование воображения:**

" _____ "

- **Прямой подход:**

" _____ "

Какой приём вы выберете и почему?

Этап 4. Документирование результатов беседы

Деловая беседа должна завершаться фиксацией договорённостей. Это помогает избежать недопонимания и сохранить контекст для дальнейшей работы .

Задание 4.1. Изучите структуру документа "Запись деловой беседы". Реквизиты документа:

- наименование структурного подразделения;
- вид документа (ЗАПИСЬ ДЕЛОВОЙ БЕСЕДЫ);
- дата;
- заголовок к тексту;
- текст;
- подпись.

Задание 4.2. Представьте, что встреча из Задания 2.1 состоялась. По её итогам вы договорились:

- Заказчик готов рассмотреть сотрудничество;
- Нужно подготовить коммерческое предложение с тремя вариантами (минимальный, оптимальный, расширенный);
- Через неделю вы пришлёте набросок технического задания;
- Заказчик пришлёт вам логотип и брендбук до пятницы;
- Следующая встреча (онлайн) планируется через две недели для обсуждения ТЗ.

Составьте документ "Запись деловой беседы" по итогам этой встречи.

Используйте приведённый ниже шаблон:

[НАИМЕНОВАНИЕ ВАШЕЙ КОМПАНИИ]

ЗАПИСЬ ДЕЛОВОЙ БЕСЕДЫ

Дата: «» _____ 20 г.

Место проведения:

Присутствовали:

- со стороны заказчика:

-
- со стороны исполнителя:

Тема беседы:

Обсуждённые вопросы:

1. _____
2. _____
3. _____

Достигнутые договорённости:

Что нужно сделать Кто делает Срок

Следующие шаги:

Подпись: _____ / _____ /

Этап 5. Итоговое задание: разработка полной подготовки к деловой беседе

Задание 5.1. Выберите один из предложенных сценариев (или предложите свой, согласовав с преподавателем):

1. **Сценарий А.** Вы встречаетесь с заказчиком для обсуждения разработки сайта для благотворительного фонда. Бюджет ограничен, но проект интересный и может стать кейсом для портфолио.

2. **Сценарий Б.** Вы пришли на встречу с потенциальным партнёром (компания-интегратор), чтобы предложить сотрудничество по субподряду (вы берёте небольшие задачи на аутсорс разработки).

3. **Сценарий В.** Вы руководитель проекта и проводите встречу с заказчиком, который недоволен сроками разработки (проект задерживается на 2 недели). Нужно обсудить причины и новые сроки.

Разработайте полный комплект подготовки к этой беседе:

1. Сформулируйте цель беседы.
2. Составьте список информации для предварительного изучения.
3. Подготовьте 5–7 вопросов к собеседнику.
4. Напишите тезисы для рассказа о себе/компании.
5. Подготовьте аргументы и ответы на возможные возражения.
6. Выберите приём начала беседы и запишите первую фразу.

7. Составьте шаблон для записи итогов (заполните договорённости гипотетически, как если бы встреча уже прошла успешно).

Результаты оформите в виде отчёта (можно в свободной форме, но структурированно).

Этап 6. Оформление отчёта

Отчёт по практической работе должен содержать:

- Титульный лист (тема, ФИО студента/участников группы).
- Результаты выполнения Заданий 1.1 и 1.2.
- Результаты выполнения Заданий 2.1 (все пункты 1–6).
- Результаты выполнения Задания 3.1.
- Заполненный документ "Запись деловой беседы" из Задания 4.2.
- Результаты выполнения Итогового задания 5.1.
- Вывод по работе (что нового узнали, какие навыки приобрели, что показалось самым сложным/интересным).

Контрольные вопросы (для самопроверки)

1. Чем деловая беседа отличается от переговоров?
2. Какие пять фаз включает структура деловой беседы?
3. Почему важно заранее собирать информацию о собеседнике и его компании?
4. Какие существуют приёмы начала беседы? Какой из них вы бы выбрали для встречи с новым клиентом?
5. Зачем нужно документировать итоги деловой беседы?
6. Какие реквизиты включает документ "Запись деловой беседы"?

Дополнительное задание

1. Проведите мини-опрос среди знакомых, работающих в ИТ или бизнесе: "Какие ошибки в деловых беседах и переговорах вы замечали чаще всего?" Составьте список топ-5 ошибок (с примерами). Подготовьте краткое сообщение (3–5 минут).

2. Найдите в интернете видеоклип деловой встречи или переговоров (можно из художественного фильма, например, "Социальная сеть" или любой другой).

Проанализируйте его по плану:

- Какая цель встречи?
- Кто участники?
- Какие приёмы общения использовались?
- Была ли достигнута цель?
- Что можно было сделать лучше?

Приложение. Памятка "Как подготовиться к деловой беседе"

1. **Определите цель** – конкретную, измеримую, реалистичную.
2. **Изучите собеседника** – компания, позиция, потребности, "боли", стиль общения.
3. **Подготовьте вопросы** – открытые (для сбора информации) и закрытые (для уточнения).

4. **Продумайте структуру** – как начнёте, что скажете в основной части, чем завершите.
5. **Подготовьте аргументы** – почему вам стоит доверять, почему ваше решение подходит.
6. **Предусмотрите возражения** – подумайте, что может смущать собеседника, и подготовьте ответы.
7. **Организуйте встречу** – выберите удобное время и место (на своей территории или нейтральной).
8. **Зафиксируйте договорённости** – отправьте итоговое письмо или запишите в документ сразу после встречи.

Практическое занятие №5. Формирование бэклога продукта и написание User Stories.

Цель работы

Научиться формировать бэклог продукта, правильно формулировать пользовательские истории (User Stories) в соответствии с шаблоном и критериями INVEST, а также применять методы приоритизации задач для выделения MVP (Minimum Viable Product).

Краткие теоретические сведения

1. Бэклог продукта (Product Backlog)

Бэклог продукта – это упорядоченный по приоритету список всего, что может потребоваться в продукте. Он является единственным источником требований для любых изменений в продукте. Бэклог никогда не бывает полным – он постоянно развивается и меняется вместе с продуктом и рыночными условиями.

В бэклог могут входить:

- пользовательские истории (User Stories);
- исправления ошибок;
- технические задачи;
- исследовательские задачи (спайки).

Ответственный за бэклог – владелец продукта (Product Owner), который управляет приоритетами и следит за тем, чтобы бэклог был прозрачным и понятным для команды.

2. Пользовательские истории (User Stories)

User Story – это короткое описание функциональности, написанное с точки зрения пользователя или заказчика. Она представляет небольшую ценность для бизнеса, которую команда может реализовать в течение одной итерации.

Стандартный шаблон User Story :

text

As a <тип пользователя/роль>

I want to <действие/функция>

so that <ценность/выгода>

Пример:

As a customer, I want to be able to order products online so that I can avoid going to the store .

Три составляющих User Story :

1. Краткое описание потребности (сама история);
2. Обсуждения (conversations) – детали, которые выясняются в ходе уточнения;
3. Критерии приёмки (acceptance criteria) – условия, при которых история считается выполненной.

3. Критерии INVEST для качественной User Story

Хорошая пользовательская история должна соответствовать критериям INVEST :

Критерий	Значение
Independent	Независимая – истории можно разрабатывать в любом порядке
Negotiable	Обсуждаемая – детали могут уточняться в ходе обсуждения
Valuable	Ценная – история приносит ценность пользователю или заказчику
Estimable	Оцениваемая – команда может оценить трудоёмкость
Small	Маленькая – история реализуется в рамках одной итерации
Testable	Тестируемая – есть чёткие критерии проверки

4. Критерии приёмки (Acceptance Criteria)

Acceptance Criteria – это набор условий, которые должны быть выполнены, чтобы история считалась завершённой . Они определяют "готовность" (Definition of Done) для конкретной истории.

Рекомендуемый формат :

text
Given that <контекст/ситуация>
When <действие/событие>
Then <результат>

Пример:

Given that пользователь находится на странице оплаты, When он выбирает оплату картой и вводит корректные данные, Then заказ оформляется, и пользователь видит подтверждение .

5. Методы приоритизации бэклога

Метод MoSCoW

Категория	Значение
Must have	Обязательные требования без которых продукт не имеет смысла
Should have	Важные требования, но можно отложить, если нет времени
Could have	Желательные требования, которые реализуются, если останутся ресурсы
Won't have	Требования, которые не будут реализованы сейчас (но могут быть в будущем)

При планировании итерации рекомендуется придерживаться соотношения: **Must have: 60%, Should have: 20%, Could have: 20%.**

User Story Mapping

User Story Mapping – это техника визуализации бэклога, которая помогает:

- увидеть всю картину продукта;
- понять пользовательский путь (user journey);
- выделить MVP (минимально жизнеспособный продукт).

Карта строится следующим образом:

1. Сверху располагаются **активности/этапы** пользовательского пути (по горизонтали слева направо);
2. Под каждым этапом – **пользовательские задачи** (действия);
3. Задачи располагаются по вертикали: **приоритетные – выше**, менее важные – ниже;
4. Горизонтальная линия отсекает MVP – всё, что выше линии, входит в первый релиз.

6. Эпики (Epics)

Еpic – это крупная пользовательская история, которую невозможно реализовать в одной итерации. Эпики декомпозируются на несколько более мелких историй по мере продвижения к релизу .

Ход работы

Работа выполняется в группах по 3–4 человека (допускается индивидуальное выполнение). Все ответы фиксируются в отчёте.

Этап 1. Анализ и написание User Stories

Задание 1.1. Разбор структуры User Story

Изучите приведённые ниже формулировки задач. Определите, какие из них являются правильно сформулированными пользовательскими историями, а какие – нет. Для некорректных формулировок укажите, что именно не так (опираясь на шаблон и INVEST), и предложите исправленный вариант.

№	Формулировка	Корректная? (Да/Нет)	Если нет – что не так?	Исправленный вариант
1	Реализовать модуль авторизации через соцсети			
2	As a user, I want to login with my social media account so that I can access the app faster			
3	Сделать кнопку "Заказать" зелёной			
4	As a customer, I want to receive email notification when my order status changes so that I can track my delivery			

№	Формулировка	Корректная? (Да/Нет)	Если нет – что не так?	Исправленный вариант
5	As a student, I want to view my grades so that I can track my academic progress			
6	Разработать базу данных для хранения информации о пользователях			
7	Система должна быть быстрой и удобной			
8	As a site administrator, I want to batch process customer records so that I can update their subscription status quickly			

Задание 1.2. Написание User Stories по описанию

Для каждого из приведённых ниже описаний функциональности сформулируйте пользовательскую историю по шаблону **As a... I want to... so that...**

Описание 1: Пользователь интернет-магазина хочет добавить товары в список желаний (wish list), чтобы потом можно было легко их найти и купить .

User Story:

text

As a _____
I want to _____
so that _____

Описание 2: Менеджер по продажам хочет просматривать отчёты по продажам в разрезе регионов, чтобы принимать решения о стратегии продаж.

User Story:

text

As a _____
I want to _____
so that _____

Описание 3: Студент колледжа хочет получать push-уведомления об изменениях в расписании, чтобы не пропустить замену занятий.

User Story:

text

As a _____
I want to _____
so that _____

Этап 2. Критерии приёмки (Acceptance Criteria)

Задание 2.1. Разработка Acceptance Criteria

Для каждой из трёх User Stories, составленных в задании 1.2, разработайте критерии приёмки в формате **Given/When/Then** (минимум 2 критерия на историю).

User Story 1 (список желаний):

Критерий приёмки

1. Given that _____
When _____
Then _____

2. Given that _____
When _____
Then _____

User Story 2 (отчёты по продажам):

Критерий приёмки

1. Given that _____
When _____
Then _____

2. Given that _____
When _____
Then _____

User Story 3 (уведомления о расписании):

Критерий приёмки

1. Given that _____
When _____
Then _____

2. Given that _____
When _____
Then _____

Этап 3. Формирование бэклога продукта

Проект для работы: Мобильное приложение "CollegeHelper" (продолжение работы с предыдущего занятия). Напоминание: приложение должно помогать студентам колледжа отслеживать расписание, новости, оценки и подавать заявки в техподдержку.

Задание 3.1. Создание бэклога

На основе описания проекта и вашего понимания функциональности составьте начальный бэклог продукта. Включите в него **не менее 10 элементов** (пользовательских историй), которые могут потребоваться в проекте.

Используйте таблицу (можно продолжить на отдельном листе):

ID	User Story	Приоритет (пока не заполняем)	Еpic (если применимо)
1	As a student, I want to view my daily schedule so that I know which classes to attend		
2			
3			
4			
5			

ID	User Story	Приоритет (пока не заполняем)	Еpic (если применимо)
6			
7			
8			
9			
10			

Этап 4. Приоритизация бэклога методом MoSCoW

Задание 4.1. Приоритизация

Используя метод MoSCoW, распределите все истории из вашего бэклога по категориям приоритета. Заполните таблицу:

Категория	ID историй	Пояснение (почему именно эта категория)
Must have (обязательно в первой версии)		
Should have (важно, но можно позже)		
Could have (хорошо бы, но не критично)		
Won't have (не в этом релизе)		

Рекомендация: при распределении старайтесь придерживаться примерного соотношения 60/20/20 для Must/Should/Could.

Задание 4.2. Обоснование MVP

На основе категории **Must have** сформируйте состав MVP (Minimum Viable Product) – минимально жизнеспособного продукта, который можно показать заказчику и получить обратную связь.

Перечислите функциональность, которая войдёт в MVP:

- _____
- _____
- _____
- _____
- _____

Почему именно эти функции являются обязательными?

Этап 5. User Story Mapping (практикум)

Задание 5.1. Построение карты для утреннего сценария (по методике)

Представьте, что вы проводите воркшоп по User Story Mapping для проекта "CollegeHelper". Для разминки выполните упражнение на примере повседневной активности.

Шаг 1. Индивидуально напишите на стикерах (или в таблице ниже) все действия, которые вы выполняете с момента пробуждения до выхода из дома. На каждое действие –

отдельная строка. Постарайтесь вспомнить как можно больше деталей (минимум 10 действий).

№	Действие
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	

Шаг 2. Объедините похожие действия в категории (например, "Водные процедуры", "Завтрак", "Одевание", "Сборы"). Заполните таблицу :

Категория	Действия (перечислите ID из шага 1)

Шаг 3. Расположите категории в хронологическом порядке (слева направо) и запишите получившуюся последовательность:

Хронология категорий: _____ → _____ → _____ → _____ → _____

Шаг 4. Представьте, что вы проспали и у вас осталось только **30 минут** на все сборы (вместо обычных 1.5 часов). Выделите **самые важные действия**, которые обязательно нужно успеть – это ваш MVP для утреннего сценария .

Обязательные действия (MVP):

- Из категории _____ : _____
- Из категории _____ : _____
- Из категории _____ : _____
- Из категории _____ : _____

Задание 5.2. Применение Story Mapping к проекту "CollegeHelper"

Используя полученный опыт, спроектируйте карту пользовательских историй для ключевого сценария в приложении "CollegeHelper" – "Просмотр расписания на сегодня".

1. Опишите пользовательский путь (последовательность действий, которые совершает студент, чтобы узнать расписание и увидеть детали занятия):

Шаг	Действие пользователя
1	Открывает приложение
2	
3	
4	
5	

2. Какие функции обязательно должны быть в первой версии (MVP), чтобы пользователь мог успешно узнать расписание?

MVP для сценария "Просмотр расписания":

- _____
- _____
- _____
- _____

Этап 6. Оформление отчёта

Отчёт по практической работе должен содержать:

1. Титульный лист (тема, ФИО студента/участников группы).
2. Результаты выполнения Задания 1.1 (таблица с анализом User Stories).
3. Результаты выполнения Задания 1.2 (три User Story по шаблону).
4. Результаты выполнения Задания 2.1 (Acceptance Criteria для трёх историй).
5. Бэклог продукта из Задания 3.1 (таблица с 10+ историями).
6. Результаты приоритизации MoSCoW из Задания 4.1.
7. Описание MVP из Задания 4.2.
8. Результаты упражнения по Story Mapping (Задания 5.1 и 5.2).
9. Вывод по работе (что нового узнали, какие навыки приобрели, какие сложности возникли при написании User Stories).

Контрольные вопросы (для самопроверки)

1. Что такое бэклог продукта и кто отвечает за его ведение?
2. Какова структура правильно написанной User Story? Приведите пример.
3. Что означает аббревиатура INVEST? Перечислите все критерии.
4. Чем отличаются категории Must have и Should have в методе MoSCoW?
5. Что такое критерии приёмки (Acceptance Criteria) и в каком формате их рекомендуется записывать?
6. Как User Story Mapping помогает выделить MVP?
7. Почему важно включать в User Story часть "so that" (чтобы получить)?

Дополнительное задание

1. **Декомпозиция эпика.** Возьмите крупную функциональность (эпик) для проекта "CollegeHelper" – например, "Система уведомлений" или "Личный кабинет студента". Разбейте её на 5–7 отдельных пользовательских историй, каждая из которых соответствует критериям INVEST. Оформите в виде таблицы.

2. **Критерии приёмки.** Для трёх историй, полученных в результате декомпозиции, напишите подробные критерии приёмки в формате Given/When/Then.

3. **Работа с инструментами (по желанию/возможности).** Зарегистрируйтесь в одном из инструментов для ведения бэклога (YouGile, WEEEEK, Kaiten или Trello). Создайте проект "CollegeHelper", добавьте туда 5–7 user stories из вашего бэклога, назначьте приоритеты (можно использовать метки Must/Should/Could). Пришлите скриншот доски.

Приложение. Шпаргалка "Как писать хорошие User Stories"

1. **Всегда используйте шаблон:** *As a [роль], I want to [действие], so that [ценность]*.

2. **Часть "so that" обязательна!** Она связывает историю с бизнес-целью и помогает при приоритизации.

3. **Избегайте технических задач** в формулировках историй – истории должны быть о ценности для пользователя, а не о том, как это реализовано технически.

4. **Не пишите слишком подробно** на ранних этапах – детали появятся в ходе обсуждения.

5. **Проверяйте по INVEST:**

- Независима ли история?
- Можно ли её обсудить?
- Есть ли ценность?
- Можно ли оценить?
- Достаточно ли она маленькая?
- Можно ли протестировать?

6. **Критерии приёмки пишите до начала разработки** – это синхронизирует понимание команды и заказчика.

7. **Группируйте истории в эпики и темы** – это помогает видеть общую картину.

8. **Избегайте расплывчатых формулировок:** "удобный", "быстрый", "интуитивный", "простой" – такие слова нельзя проверить.

Практическое занятие №6. Разработка технического задания.

Цель работы

Научиться разрабатывать техническое задание (ТЗ) на создание ИТ-продукта, освоить структуру ТЗ, правила формулирования функциональных и нефункциональных требований, а также применять полученные знания для составления фрагмента ТЗ на примере проекта мобильного приложения.

Краткие теоретические сведения

1. Техническое задание (ТЗ): определение и назначение

Техническое задание – это официальный документ, который определяет цели, требования, ограничения и порядок разработки программного продукта или информационной системы. ТЗ является основным документом, регламентирующим взаимоотношения между заказчиком и исполнителем: в нём фиксируется, что именно должно быть создано, в какие сроки и с какими характеристиками.

Основные функции ТЗ:

- устанавливает единое понимание целей и задач проекта между заказчиком и разработчиком;
- служит основой для планирования работ и оценки сроков;
- используется для проверки готового продукта (приёмочные испытания);
- защищает интересы обеих сторон в случае споров.

2. Структура технического задания

Существуют различные стандарты оформления ТЗ (ГОСТ 34.602-89, ГОСТ 19.201-78, международные стандарты IEEE). Для большинства современных ИТ-проектов применяется следующая типовая структура:

Раздел	Содержание
1. Введение	Краткая информация о проекте, заказчике, исполнителе, основания для разработки
2. Основание для разработки	Документы, на основании которых ведётся разработка (договор, приказ, устав проекта)
3. Назначение разработки	Цели создания системы, пользователи, область применения
4. Требования к программе или ПО	Функциональные требования, нефункциональные требования (надёжность, производительность, безопасность, эргономика, совместимость)
5. Требования к программной документации	Состав документации, которая должна быть разработана (руководство пользователя, руководство администратора и т.д.)
6. Техничко-экономические показатели	Ориентировочная эффективность, экономические преимущества
7. Стадии и этапы разработки	Календарный план, этапы, сроки, результаты этапов
8. Порядок контроля и приёмки	Виды испытаний, критерии приёмки, состав приёмочной комиссии
9. Требования к составу и содержанию работ	Что должно быть сделано на каждом этапе
10. Требования к эргономике и технической эстетике	Дизайн, удобство использования
11. Дополнительные требования	Особые условия (например, необходимость обучения пользователей)

На практике часто используют сокращённую структуру, адаптированную под масштаб проекта.

3. Функциональные и нефункциональные требования

Функциональные требования описывают, что именно система должна делать: какие функции, операции, сервисы предоставлять. Они обычно формулируются в виде пользовательских историй (User Stories) или списка функций.

Пример: "Система должна позволять пользователю просматривать расписание занятий на текущую неделю".

Нефункциональные требования описывают, как система должна это делать, то есть её характеристики и ограничения:

- **Производительность:** время отклика, количество одновременных пользователей;
- **Надёжность:** время безотказной работы, восстановление после сбоев;
- **Безопасность:** разграничение доступа, шифрование данных;
- **Совместимость:** поддерживаемые платформы, браузеры, интеграция с другими системами;
- **Эргономика:** удобство интерфейса, соответствие стандартам юзабилити;
- **Технологические ограничения:** язык программирования, СУБД, фреймворки.

4. Этапы разработки ТЗ

1. **Сбор исходных данных:** интервью с заказчиком, изучение бизнес-процессов, анализ аналогичных продуктов.
2. **Определение целей и границ проекта:** что входит в проект, а что – нет.
3. **Формулирование функциональных требований:** составление списка функций, пользовательских историй.
4. **Определение нефункциональных требований:** уточнение технических параметров.
5. **Согласование и утверждение:** обсуждение с заказчиком, внесение правок, подписание.

5. Типичные ошибки при разработке ТЗ

- Требования сформулированы неоднозначно (размытые формулировки типа "удобный интерфейс", "быстрая работа").
- Отсутствуют критерии приёмки – непонятно, как проверять выполнение требований.
- Требования противоречат друг другу.
- Не учтены нефункциональные требования (безопасность, нагрузка).
- Изменения требований в процессе не фиксируются (ТЗ не обновляется).

Ход работы

Работа выполняется индивидуально или в группах по 2–3 человека (по указанию преподавателя). Для выполнения заданий используется проект мобильного приложения "**CollegeHelper**" (рассмотренный на предыдущих занятиях).

Напоминание по проекту:

Мобильное приложение для студентов колледжа должно предоставлять следующие основные возможности:

- просмотр расписания занятий (с выбором группы);
 - просмотр новостей колледжа;
 - подача заявок в техподдержку (сломанное оборудование, проблемы в аудитории);
 - просмотр успеваемости (оценок).
-

Этап 1. Анализ исходных данных и определение целей проекта

Задание 1.1. Прочитайте описание проекта. Сформулируйте цель разработки системы (что должен делать продукт с точки зрения бизнеса) и назначение (для кого создаётся).

- **Цель разработки:**

-
- **Назначение (целевая аудитория):**
-

Задание 1.2. Определите границы проекта: что НЕ входит в первую версию (MVP), но может быть добавлено позже. Заполните таблицу.

Входит в первую версию (MVP)	Не входит в первую версию
(возьмите из прошлого занятия)	

Этап 2. Разработка функциональных требований

Задание 2.1. Используя бэклог, составленный на предыдущем занятии (или создав новый), сгруппируйте пользовательские истории по функциональным областям. Для каждой истории укажите её идентификатор (ID) и краткое описание.

ID	Функциональная область	Пользовательская история (кратко)
	Расписание	
	Новости	
	Заявки в техподдержку	
	Успеваемость	
	...	

Задание 2.2. Для трёх наиболее важных пользовательских историй (из разных областей) напишите детальное описание, включая:

- **Название** (краткое);
- **User Story** (полная формулировка As a... I want to... so that...);
- **Критерии приёмки** (2–3 условия в формате Given/When/Then);
- **Приоритет** (Must / Should / Could).

Используйте шаблон:

Элемент	Описание
ID:	F_01
Название:	Просмотр расписания на сегодня
User Story:	As a student, I want to view my today's schedule so that I know which classes to attend
Критерии приёмки:	1. Given that user is logged in and on the home screen, when user taps "Schedule", then the system displays list of today's classes with time and location.

Элемент	Описание
	2. Given that there are no classes today, when user views schedule, then system shows message "No classes today".
Приоритет:	Must

(Выполните для трёх историй аналогично)

Этап 3. Разработка нефункциональных требований

Задание 3.1. Для проекта "CollegeHelper" определите нефункциональные требования по следующим категориям. Заполните таблицу, указав конкретные измеримые показатели, где это возможно.

Категория	Требование (с указанием конкретных значений)
Платформа	(под какие ОС разрабатывается: iOS, Android, веб-версия?)
Производительность	(время запуска приложения, время отклика на действия пользователя, количество одновременных пользователей)
Надёжность	(доступность системы, время восстановления после сбоя)
Безопасность	(авторизация, защита данных, шифрование)
Удобство использования (эргономика)	(требования к интерфейсу, соответствие гайдлайнам платформы)
Технологические ограничения	(языки программирования, фреймворки, СУБД)
Интеграция	(с какими внешними системами должна взаимодействовать)

Этап 4. Описание ограничений и допущений

Задание 4.1. Укажите ограничения проекта, которые влияют на разработку:

- **Бюджетные ограничения:**

- **Временные ограничения (сроки):**

- **Организационные ограничения:** (например, доступ к данным, согласования)

- **Допущения:** (что считается верным для успеха проекта, но не гарантировано)

Этап 5. Составление структуры ТЗ

Задание 5.1. На основе типовой структуры, приведённой в теории, составьте оглавление (план) технического задания для проекта "CollegeHelper". Номера разделов и подразделов можно изменять, но основные разделы должны присутствовать.

Пример начала:

1. Введение

1.1. Наименование программы

1.2. Краткая характеристика области применения

2. Основание для разработки
3. Назначение разработки

...

Заполните план (не менее 8 разделов):

1. _____
 2. _____
 3. _____
 4. _____
 5. _____
 6. _____
 7. _____
 8. _____
- ...

Этап 6. Написание раздела ТЗ "Требования к программе"

Задание 6.1. Используя материалы, полученные на этапах 2 и 3, напишите полный текст раздела **"Требования к программе"** для ТЗ проекта "CollegeHelper". Оформите в соответствии со структурой, приведённой ниже.

Требования к программе:

1. Требования к функциональным характеристикам

Перечислите все функциональные требования (можно ссылаться на ID пользовательских историй). Группируйте по подсистемам или модулям.

Пример:

1.1. Модуль расписания:

- Приложение должно отображать расписание на текущий день.
- Должна быть возможность выбора группы (для студентов разных специальностей).

...

2. Требования к надёжности

- Время восстановления после сбоя не более 5 минут.
- Приложение должно сохранять данные при отсутствии интернета и синхронизировать при появлении соединения.

...

3. Условия эксплуатации

• Тип пользователей: студенты и преподаватели, владеющие базовыми навыками работы со смартфоном.

- Количество одновременных пользователей: до 500.

4. Требования к составу и параметрам технических средств

• Мобильное устройство на базе iOS (версия не ниже 15) или Android (версия не ниже 10) с доступом в интернет.

5. Требования к информационной и программной совместимости

- Доступ к данным через REST API.
- Формат обмена данными: JSON.

6. Требования к программной документации

- В составе документации должны быть: руководство пользователя, руководство администратора, описание API (если применимо).

7. Требования к эргономике и технической эстетике

- Интерфейс должен соответствовать гайдлайнам Material Design (Android) и Human Interface Guidelines (iOS).
- Размер шрифтов должен регулироваться системными настройками.

Этап 7. Оформление отчёта

Отчёт по практической работе должен содержать:

1. Титульный лист (тема, ФИО студента/участников группы).
2. Результаты выполнения Задания 1.1 (цель и назначение).
3. Результаты выполнения Задания 1.2 (границы проекта).
4. Результаты выполнения Задания 2.1 (таблица функциональных областей).
5. Результаты выполнения Задания 2.2 (детальное описание трёх историй с критериями приёмки).
6. Результаты выполнения Задания 3.1 (таблица нефункциональных требований).
7. Результаты выполнения Задания 4.1 (ограничения и допущения).
8. План ТЗ из Задания 5.1.
9. Текст раздела "Требования к программе" из Задания 6.1.
10. Вывод по работе (что нового узнали, какие навыки приобрели, с какими трудностями столкнулись).

Контрольные вопросы (для самопроверки)

1. Что такое техническое задание и для чего оно необходимо?
2. Перечислите основные разделы типового технического задания.
3. Чем функциональные требования отличаются от нефункциональных? Приведите примеры.
4. Почему важно указывать в ТЗ критерии приёмки?
5. Какие ошибки чаще всего допускают при написании ТЗ?
6. Как ТЗ связано с пользовательскими историями (User Stories)?
7. В каком виде обычно оформляются нефункциональные требования?

Дополнительное задание

1. **Анализ реального ТЗ.** Найдите в интернете пример технического задания на разработку программного продукта (можно фрагмент). Проанализируйте его структуру и содержание. Опишите, какие разделы присутствуют, какие требования сформулированы удачно, а какие – размыто. Сформулируйте предложения по улучшению.

2. **Разработка полного ТЗ.** Для проекта "CollegeHelper" допишите остальные разделы ТЗ (кроме "Требований к программе") согласно плану, составленному в задании 5.1. Объём – 2–3 страницы.

3. **Подготовка презентации (для групп).** Подготовьте краткую презентацию (5–7 слайдов) основных требований к вашему проекту, как если бы вы защищали ТЗ перед заказчиком.

Приложение. Шаблон титульного листа отчёта

Специальность 09.02.11 Информационные системы и программирование

ОТЧЁТ

по практическому занятию №5

«Разработка технического задания»

Выполнил (и) студент (ы) группы _____

(Ф.И.О.)

Проверил преподаватель: _____

[Город]

2026

Практическое занятие №7. Создание диаграммы Ганта.

Цель работы

Научиться планировать выполнение работ по проекту, строить диаграмму Ганта, определять последовательность и длительность задач, выявлять критический путь и использовать диаграмму для управления сроками проекта.

Краткие теоретические сведения

1. Что такое диаграмма Ганта?

Диаграмма Ганта – это инструмент планирования и управления проектами, который визуально отображает календарный график работ. Она представляет собой набор горизонтальных полос (столбцов), расположенных вдоль оси времени. Каждая полоса соответствует отдельной задаче (работе), её длина показывает длительность задачи, а положение на временной шкале – даты начала и окончания.

Диаграмма названа в честь американского инженера Генри Л. Ганта (1861–1919), который разработал этот метод в начале XX века.

2. Основные элементы диаграммы Ганта

- **Задачи (работы, этапы)** – список всех действий, которые необходимо выполнить.
- **Длительность** – время, отведённое на выполнение задачи (в днях, часах, неделях).
- **Даты начала и окончания** – календарные или относительные (например, день 1, день 2).
- **Связи между задачами** – показывают зависимость: задача В не может начаться, пока не завершится задача А.
 - *Окончание-начало (FS)*: самая распространённая – следующая задача начинается после окончания предыдущей.
 - *Начало-начало (SS)*: задачи начинаются одновременно.
 - *Окончание-окончание (FF)*: задачи заканчиваются одновременно.

- *Начало-окончание (SF)*: редко используется.
- **Вехи (milestones)** – ключевые события (нулевая длительность), например, утверждение ТЗ, сдача прототипа.
- **Ресурсы** – люди, оборудование, материалы, назначенные на задачу.
- **Критический путь** – самая длинная последовательность взаимосвязанных задач, определяющая минимальное время выполнения проекта. Любая задержка на критическом пути приведёт к задержке всего проекта.

3. Правила построения диаграммы Ганта

1. Составить полный список работ (иерархическая структура работ – WBS).
2. Определить логические связи и последовательность выполнения.
3. Оценить длительность каждой задачи.
4. Назначить ответственных (ресурсы).
5. Построить диаграмму, расположив задачи на временной оси.
6. Выделить критический путь.
7. Регулярно обновлять диаграмму по мере выполнения проекта.

4. Инструменты для создания диаграммы Ганта

Инструмент	Тип	Особенности
Microsoft Project	Десктопный	Профессиональное управление проектами, автоматический расчёт критического пути, ресурсное планирование. Платный.
GanttProject	Десктопный	Бесплатный, open-source, простой интерфейс, экспорт в PDF/PNG.
LibreOffice Calc / Excel	Табличный процессор	Можно создать диаграмму с помощью условного форматирования или встроенных шаблонов. Бесплатно.
Онлайн-инструменты (GanttPRO, WEEEEK, Kaiten, Smartsheet)	Веб-приложения	Удобны для совместной работы, многие имеют бесплатные тарифы для небольших проектов.
Бумага и карандаш	Ручной	Для самых маленьких проектов или эскизов.

На этом занятии мы будем использовать **Microsoft Excel** (или Google Sheets) как наиболее доступный инструмент. При наличии специализированного ПО можно работать в нём.

Ход работы

Работа выполняется индивидуально или в группах по 2–3 человека. Проект для планирования – разработка мобильного приложения «CollegeHelper» (рассмотрен на предыдущих занятиях).

Этап 1. Определение состава работ (WBS)

Задание 1.1. На основе бэклога и функциональных требований составьте иерархический список работ (структуру разбиения работ – WBS) для проекта «CollegeHelper». Разбейте проект на основные фазы (этапы) и задачи. Минимальное количество задач – 15–20.

Рекомендуемые фазы:

- Управление проектом (собрания, планирование, отчётность);

- Анализ и проектирование (сбор требований, прототипирование, ТЗ);
- Дизайн (UI/UX, макеты);
- Разработка (бэкенд, фронтенд, модули);
- Тестирование (модульное, интеграционное, приёмочное);
- Внедрение и сопровождение.

Заполните таблицу:

ID	Фаза	Задача	Длительность (дни)	Предшественники	Ресурсы (исполнители)
1	Управление	Инициация проекта (устав, собрание)	1	—	PM
2	Анализ	Интервью с заказчиком	2	1	Аналитик, PM
3	Анализ	Разработка ТЗ	3	2	Аналитик
...	...	...	...	...	...

Примечание: столбец «Предшественники» указывает ID задач, которые должны завершиться до начала данной.

Этап 2. Оценка длительности и определение зависимостей

Задание 2.1. Для каждой задачи из таблицы задания 1.1 укажите:

- Предполагаемую длительность в днях (используйте экспертные оценки: 1–5 дней для мелких задач, до 10 дней для крупных).
- Связи с другими задачами (укажите ID задач-предшественников).
- Назначьте условные ресурсы (исполнителей): PM (руководитель проекта), Аналитик, Дизайнер, Разработчик (Dev), Тестировщик (QA).

Подсказка: при определении последовательности учитывайте логику разработки: сначала анализ, потом дизайн, потом разработка, потом тестирование. Некоторые задачи могут выполняться параллельно (например, разработка разных модулей).

Этап 3. Построение диаграммы Ганта в Excel

Задание 3.1. Создайте диаграмму Ганта на основе вашего списка работ. Можно использовать один из двух способов:

Способ А. Условное форматирование (быстрый способ)

1. В Excel создайте таблицу: столбцы – ID, Задача, Длительность, Начало, Конец.
2. Рассчитайте даты начала и окончания, учитывая предшественников. Для первой задачи можно задать начальную дату (например, 01.03.2026).
3. Для остальных задач используйте формулу: $\text{Начало} = \text{МАКС}(\text{Конец всех предшественников}) + 1$ (если рабочие дни подряд).
4. Справа от таблицы создайте календарную шкалу (например, столбцы с датами).
5. Используйте условное форматирование, чтобы закрашивать ячейки в строках задач для дат, попадающих в интервал [Начало; Конец].

Способ Б. Использование диаграммы «Линейчатая с накоплением»

1. Подготовьте данные: задачи, даты начала, длительность, даты окончания.
2. Вставьте диаграмму «Линейчатая с накоплением».

3. Настройте оси так, чтобы отображались даты (скройте синюю полосу, оставьте оранжевую как длительность).

4. Отформатируйте.

Для ускорения можно воспользоваться готовым шаблоном Gantt Chart из интернета.

Результат: диаграмма, на которой видно все задачи, их длительность и последовательность.

Задание 3.2. Добавьте на диаграмму вехи (ключевые события):

- Утверждение ТЗ;
- Готовность прототипа;
- Завершение разработки;
- Начало тестирования;
- Релиз MVP.

Вехи изображаются ромбиками или точками с нулевой длительностью.

Этап 4. Определение критического пути

Задание 4.1. Проанализируйте полученную диаграмму и определите критический путь проекта. Критический путь – это самая длинная последовательность взаимосвязанных задач, определяющая общую длительность проекта.

Методика определения:

1. Выпишите все пути от начала до конца проекта (последовательности задач с учётом зависимостей).
2. Сложите длительности задач в каждом пути.
3. Самый длинный путь – критический. Задачи на этом пути не имеют резервов времени.

Пример: если проект длится 45 дней, а сумма длительностей по некоторому пути даёт 45 дней – это критический путь.

Заполните таблицу:

№ пути	Последовательность задач (ID)	Суммарная длительность (дней)
1	...	...
2	...	...
...	...	...

Критический путь: задачи с ID: _____ (перечислите)

Задание 4.2. Выделите задачи критического пути на диаграмме (например, красным цветом или жирной рамкой).

Этап 5. Анализ и оптимизация

Задание 5.1. Представьте, что заказчик попросил сократить сроки проекта на 20%. Предложите не менее двух способов сокращения длительности (например, параллельное выполнение некоторых задач, увеличение ресурсов, сокращение функциональности). Опишите, как изменится диаграмма Ганта и критический путь.

Задание 5.2. Ответьте на вопросы:

- Какие задачи имеют резервы времени (не лежат на критическом пути)?

- Можно ли начать тестирование до завершения всей разработки? Если да, перерисуйте фрагмент диаграммы для модуля «Расписание», где тестирование начинается сразу после завершения разработки этого модуля (перекрытие фаз).

Этап 6. Оформление отчёта

Отчёт должен содержать:

1. Титульный лист (тема, ФИО студента/группы).
2. Таблицу со списком работ (задание 1.1 и 2.1 – объединённая таблица).
3. Скриншот или рисунок диаграммы Ганта (задание 3.1), на которой отмечены вехи (задание 3.2) и выделен критический путь (задание 4.2).
4. Результаты определения критического пути (задание 4.1).
5. Анализ оптимизации (задание 5.1).
6. Ответы на вопросы задания 5.2.
7. Вывод по работе: что нового узнали, какие навыки приобрели, с какими трудностями столкнулись.

Контрольные вопросы

1. Что показывает диаграмма Ганта и для чего она используется в управлении проектами?
2. Какие элементы обязательны для построения диаграммы Ганта?
3. Что такое критический путь и почему важно его знать?
4. Какие типы связей между задачами существуют? Приведите примеры.
5. Как изменение длительности задачи на критическом пути влияет на сроки всего проекта?
6. Какие инструменты можно использовать для построения диаграммы Ганта? Назовите их достоинства и недостатки.
7. Для чего на диаграмме отмечают вехи?

Дополнительное задание

1. **Работа с профессиональным инструментом.** Скачайте и установите бесплатную программу GanttProject (или используйте онлайн-версию GanttPRO, WEEEEK). Перенесите ваш список работ в эту программу, постройте диаграмму, настройте связи и ресурсы. Сравните с диаграммой из Excel. Приложите скриншот.
2. **Расчёт расписания вручную.** Для небольшого проекта из 5–7 задач (например, организация мероприятия) рассчитайте даты начала и окончания вручную, постройте диаграмму на миллиметровке или в тетради. Сфотографируйте и вставьте в отчёт.
3. **Подготовка к защите.** Будьте готовы объяснить логику вашего плана, обосновать длительности задач и выбор критического пути.

Приложение. Пример оформления таблицы работ

ID	Название задачи	Длит. (дн.)	Начало	Окончание	Предшественник и	Ресурсы
1	Инициация	1	01.03.26	01.03.26	–	PM

ID	Название задачи	Длит. (дн.)	Начало	Окончание	Предшественник и	Ресурсы
2	Сбор требований	3	02.03.26	04.03.26	1	Аналитик
3	Разработка ТЗ	5	05.03.26	09.03.26	2	Аналитик
4	Дизайн-макеты	7	05.03.26	11.03.26	2	Дизайнер
5	Разработка бэкенда	15	10.03.26	24.03.26	3	Dev
...	...	...	...	...	...	...

Примечание: начало задачи 5 – после завершения задачи 3 (ТЗ), даже если задача 4 ещё не закончена (параллельная работа).

Практическое занятие №8. Составление бюджета проекта.

Цель работы

Научиться определять статьи затрат ИТ-проекта, оценивать стоимость ресурсов, рассчитывать бюджет с учётом прямых и косвенных расходов, а также анализировать риски, связанные с бюджетом.

Краткие теоретические сведения

1. Бюджет проекта: определение и назначение

Бюджет проекта – это документ, который содержит сметную стоимость всех работ, ресурсов и затрат, необходимых для реализации проекта в утверждённые сроки. Бюджет является основой для финансового планирования, контроля расходов и отчётности перед заказчиком и инвесторами.

Основные функции бюджета:

- определение общей стоимости проекта;
- распределение средств по этапам и статьям затрат;
- контроль расходов в ходе выполнения;
- прогнозирование денежных потоков;
- обоснование экономической эффективности.

2. Классификация затрат в ИТ-проектах

По способу отнесения на проект:

- **Прямые затраты** – непосредственно связаны с выполнением проекта (зарплата команды, оборудование, лицензии ПО, командировки).
- **Косвенные затраты** – не могут быть прямо отнесены на проект и распределяются пропорционально (аренда офиса, административные расходы, коммунальные услуги).

По зависимости от объёма работ:

- **Постоянные затраты** – не меняются при изменении объёма работ (например, аренда сервера на весь срок).

- **Переменные затраты** – зависят от объёма (например, оплата облачных сервисов по факту использования).

По этапам проекта:

- Затраты на инициацию и планирование.
- Затраты на разработку и тестирование.
- Затраты на внедрение и сопровождение.

3. Методы оценки стоимости

Метод	Описание	Применимость
Аналоговый (по аналогии)	Используются данные о стоимости аналогичных проектов в прошлом	На ранних стадиях, для быстрой оценки
Параметрический	Оценка на основе параметров (например, стоимость строки кода, стоимость часа работы)	При наличии статистических данных
Снизу вверх	Детальная оценка каждой работы/задачи, затем суммирование	Точный метод, требует детального планирования
Сверху вниз	Общая сумма определяется, затем распределяется по этапам	При ограниченном бюджете

4. Типовая структура бюджета ИТ-проекта

Статья затрат	Содержание
1. Трудовые затраты	Зарплата команды (РМ, аналитики, разработчики, тестировщики, дизайнеры) с учётом налогов
2. Оборудование и ПО	Серверы, компьютеры, лицензии на ПО, облачные сервисы, инструменты разработки
3. Командировки и обучение	Поездки к заказчику, курсы повышения квалификации
4. Услуги сторонних организаций	Аутсорсинг, консультации, хостинг, поддержка
5. Прочие прямые затраты	Канцтовары, связь, транспорт
6. Резерв на непредвиденные расходы	Обычно 5–15% от общей суммы
7. Накладные расходы (косвенные)	Аренда, административные расходы (если не включены в ставки)

5. Управление резервами

- **Резерв на известные риски** – включается в бюджет для покрытия предвидимых событий (например, рост курса валют).
- **Управленческий резерв** – для непредвиденных обстоятельств, утверждается руководством.

6. Инструменты для составления бюджета

- Электронные таблицы (Excel, Google Sheets) – наиболее доступны.
- Специализированное ПО для управления проектами (Projecto, Naumen, MS Project) – позволяют привязывать бюджет к задачам и ресурсам.
- Бухгалтерские системы.

Ход работы

Работа выполняется индивидуально или в группах по 2–3 человека. В качестве основы используется проект «CollegeHelper» – мобильное приложение для студентов колледжа.

Этап 1. Определение состава команды и трудовых затрат

Задание 1.1. Исходя из объёма работ (используйте результаты практических занятий №4–6), определите роли, необходимые для реализации проекта, и время их участия. Заполните таблицу 1.

Таблица 1 – Состав команды и трудозатраты

Роль	Количество человек	Ставка (руб./час)	Загрузка (часов всего)	Стоимость (руб.)
Руководитель проекта	1			
Аналитик	1			
Дизайнер UI/UX	1			
Разработчик мобильный (iOS/Android)	2			
Разработчик бэкенда	1			
Тестировщик	1			
ИТОГО трудозатраты				

Примечание: Ставки можно взять усреднённые по рынку для вашего региона (например, 500–1500 руб./час в зависимости от роли). Длительность проекта – из диаграммы Ганта (предыдущее занятие). Переведите длительность в часы (1 рабочий день = 8 часов).

Этап 2. Определение прямых материальных затрат

Задание 2.1. Определите, какое оборудование, программное обеспечение и внешние сервисы потребуются для проекта. Заполните таблицу 2.

Таблица 2 – Прямые затраты (оборудование, ПО, услуги)

Наименование	Тип затрат (покупка/аренда/подписка)	Количество	Цена за ед. (руб.)	Период (мес.)	Сумма (руб.)
Компьютеры для разработки (если нет своих)	Покупка			—	
Лицензии IDE (например, JetBrains)	Подписка				
Серверное ПО / облачный хостинг	Аренда				
Инструменты дизайна (Figma)	Подписка				

Наименование	Тип затрат (покупка/аренда/подписка)	Количество	Цена за ед. (руб.)	Период (мес.)	Сумма (руб.)
СУБД лицензия	Покупка/подписка				
Прочее (домен, SSL)	Покупка			—	
ИТОГО прямые материальные затраты					

Этап 3. Учёт косвенных затрат и резервов

Задание 3.1. Оцените косвенные затраты, связанные с выполнением проекта (например, аренда офиса, интернет, электроэнергия, административный персонал). Если точные данные отсутствуют, можно принять **20–30% от прямых трудовых затрат** как типичный коэффициент накладных расходов для ИТ-компаний.

- Процент накладных расходов: _____ %
- Сумма накладных расходов: _____ руб. (рассчитать)

Задание 3.2. Определите резерв на непредвиденные расходы. Обычно он составляет **5–15%** от общей суммы прямых затрат (трудовые + материальные + накладные). Обоснуйте выбранный процент.

- Процент резерва: _____ % (укажите причину: например, проект инновационный – 15%, типовой – 5%)
- Сумма резерва: _____ руб.

Этап 4. Сводный бюджет проекта

Задание 4.1. Сведите все рассчитанные показатели в итоговую таблицу 3. Рассчитайте общую стоимость проекта.

Таблица 3 – Сводный бюджет проекта «CollegeHelper»

№	Статья затрат	Сумма (руб.)	Примечание
1	Трудовые затраты (ФОТ с налогами?)		Указать, включая налоги или нет
2	Прямые материальные затраты		Из таблицы 2
3	Накладные расходы		Из задания 3.1
4	Итого прямые затраты (1+2+3)		
5	Резерв на непредвиденные		Из задания 3.2
6	ИТОГО бюджет проекта		(4+5)

Важно: Уточните, входят ли налоги на зарплату (страховые взносы) в трудовые затраты. Обычно ставка указывается уже с налогами (гросс) или отдельно добавляется ~30% к сумме зарплат «на руки». Договоритесь с преподавателем о варианте.

Этап 5. Анализ и оптимизация бюджета

Задание 5.1. Проанализируйте структуру бюджета. Какие статьи затрат самые весомые? Предложите два способа сокращения бюджета без потери качества (например,

замена части команды на менее дорогих специалистов, использование opensource решений, сокращение длительности за счёт параллельной работы и т.д.). Оцените, как изменится итоговая сумма.

- Самые затратные статьи:

-
- Способ 1:

Новая сумма по этой статье: _____ руб.

- Способ 2:

Новая сумма по этой статье: _____ руб.

- Общая экономия: _____ руб.
- Новый бюджет: _____ руб.

Задание 5.2. Рассмотрите два сценария: оптимистичный (все идёт по плану, резерв не используется) и пессимистичный (резерв израсходован полностью). Какова будет итоговая стоимость в каждом случае? Заполните:

- Оптимистичный бюджет: _____ руб.
- Пессимистичный бюджет: _____ руб.

Этап 6. Оформление отчёта

Отчёт по практической работе должен содержать:

1. Титульный лист (тема, ФИО студента/группы).
2. Заполненные таблицы 1, 2, 3.
3. Расчёты накладных расходов и резерва (задание 3.1, 3.2).
4. Анализ и предложения по оптимизации (задание 5.1, 5.2).
5. Вывод по работе: что нового узнали, какие навыки приобрели, с какими трудностями столкнулись.

Контрольные вопросы

1. Что такое бюджет проекта и для чего он нужен?
2. Какие существуют методы оценки стоимости проекта? В каких случаях каждый из них применяется?
3. Чем отличаются прямые и косвенные затраты? Приведите примеры для ИТ-проекта.
4. Зачем в бюджете предусматривают резерв на непредвиденные расходы?
5. Какие факторы могут повлиять на увеличение бюджета в процессе реализации?
6. Как можно контролировать соблюдение бюджета во время выполнения проекта?
7. Что такое накладные расходы и как они распределяются на проекты?

Дополнительное задание

1. **Анализ реального бюджета.** Найдите в открытых источниках информацию о стоимости разработки какого-либо известного приложения (или воспользуйтесь примером из практики знакомых). Сравните со своим расчётом. Напишите краткий анализ причин расхождений.

2. **Расчёт точки безубыточности.** Для проекта «CollegeHelper» предположите, что после разработки приложение будет приносить доход (например, от рекламы или подписки). Оцените ежемесячные доходы и расходы на поддержку. Рассчитайте, за сколько месяцев окупится разработка.

3. **Подготовка презентации.** Подготовьте слайд с итоговым бюджетом для защиты проекта перед заказчиком.

Приложение. Пример расчёта трудозатрат (условный)

Роль	Кол-во	Ставка (руб/час)	Часов	Сумма
PM	1	800	160	128 000
Аналитик	1	600	120	72 000
Разработчик	2	700	320	448 000
Тестировщик	1	500	100	50 000
Итого				698 000

Налоги (30%) = 209 400 руб.

Всего трудовые с налогами = 907 400 руб.

Тема 2.2. Оценка управления рисками

Практическое занятие №9. Выполнение SWOT-анализа.

Цель работы

Научиться проводить стратегический анализ ИТ-проекта с использованием метода SWOT, выявлять внутренние и внешние факторы, влияющие на успех проекта, и разрабатывать на их основе стратегические решения.

Краткие теоретические сведения

1. Что такое SWOT-анализ?

SWOT-анализ – это метод стратегического планирования, используемый для оценки факторов и явлений, влияющих на проект или бизнес. Название метода образовано от первых букв английских слов :

- **Strengths** (сильные стороны) – внутренние характеристики, которые дают преимущества;
- **Weaknesses** (слабые стороны) – внутренние характеристики, которые ослабляют позицию;
- **Opportunities** (возможности) – внешние факторы, которые можно использовать в своих интересах;
- **Threats** (угрозы) – внешние факторы, которые могут создать проблемы.

Метод был разработан в 1960-х годах в Гарвардской школе бизнеса и с тех пор стал классикой делового анализа.

2. Классификация факторов

Внутренние факторы (сильные и слабые стороны) – находятся под контролем команды проекта. Это может быть компетенция сотрудников, технологии, финансы, репутация.

Внешние факторы (возможности и угрозы) – не зависят от команды, но влияют на проект: рыночные тенденции, конкуренты, законодательство, экономическая ситуация.

3. Матрица SWOT-анализа

	Позитивное влияние	Негативное влияние
Внутренняя среда	S – Сильные стороны	W – Слабые стороны
Внешняя среда	O – Возможности	T – Угрозы

4. TOWS-анализ: от анализа к стратегии

TOWS – это расширение SWOT-анализа, которое помогает разработать стратегические действия на основе пересечения факторов:

	Возможности (O)	Угрозы (T)
Сильные стороны (S)	SO-стратегии – как использовать сильные стороны для реализации возможностей	ST-стратегии – как использовать сильные стороны для нейтрализации угроз
Слабые стороны (W)	WO-стратегии – как преодолеть слабости, используя возможности	WT-стратегии – как минимизировать слабости и избегать угроз

5. Типы SWOT-анализа

Тип	Описание	Когда применять
Базовый SWOT	Простое перечисление факторов без количественной оценки	Для быстрого анализа, в учебных целях
TOWS-анализ	Формирование стратегий на основе пересечения факторов	При разработке стратегии развития
Количественный SWOT	Оценка весов и значимости факторов	Для объективного сравнения и приоритизации
SWOT с визуализацией	Графическое представление факторов	Для наглядной презентации

6. Пошаговый алгоритм проведения SWOT-анализа

1. **Определить цель анализа** – что именно оцениваем (проект в целом, этап, продукт).
 2. **Собрать информацию** – данные о проекте, рынке, конкурентах.
 3. **Провести мозговой штурм** – выявить факторы по каждой категории.
 4. **Заполнить матрицу** – распределить факторы по квадрантам.
 5. **Уточнить и приоритизировать** – оставить наиболее значимые факторы.
 6. **Разработать стратегии** (для TOWS) – сформировать план действий.
 7. **Сделать выводы** – определить дальнейшие шаги.
-

Ход работы

Работа выполняется индивидуально или в группах по 2–3 человека. В качестве объекта анализа используется проект «CollegeHelper» – мобильное приложение для студентов колледжа (разработка продолжается с предыдущих занятий).

Этап 1. Подготовка к анализу

Задание 1.1. Сформулируйте цель SWOT-анализа для проекта «CollegeHelper». Что именно вы хотите оценить? Например: «Оценка перспектив вывода приложения на рынок образовательных приложений» или «Выявление факторов, влияющих на успешную разработку MVP».

Цель анализа:

Задание 1.2. Заполните таблицу, распределив приведённые факторы по категориям SWOT. Добавьте по одному своему фактору в каждую категорию.

Фактор	Категория (S/W/O/T)
Команда разработчиков имеет опыт создания мобильных приложений	
Отсутствие опыта продвижения мобильных приложений	
Рост популярности мобильных образовательных сервисов	
Появление нового конкурента с аналогичным функционалом	
Бесплатные инструменты разработки (open source)	
Ограниченный бюджет проекта	
Поддержка со стороны администрации колледжа	
Риск утечки персональных данных студентов	
<i>(добавьте свой фактор)</i>	
<i>(добавьте свой фактор)</i>	

Этап 2. Проведение SWOT-анализа проекта «CollegeHelper»

Задание 2.1. Проведите мозговой штурм в группе (или индивидуально) и заполните матрицу SWOT для проекта «CollegeHelper». Постарайтесь указать не менее 5 факторов в каждом квадранте.

Сильные стороны (S)	Слабые стороны (W)
1.	1.
2.	2.
3.	3.
4.	4.
5.	5.

Сильные стороны (S)	Слабые стороны (W)
Возможности (O)	Угрозы (T)
1.	1.
2.	2.
3.	3.
4.	4.
5.	5.

Рекомендации по заполнению:

- **Сильные стороны:** преимущества проекта, уникальные компетенции, ресурсы.
- **Слабые стороны:** недостаток ресурсов, нехватка опыта, технологические ограничения.
- **Возможности:** рыночные тренды, партнёрства, гранты, новые технологии.
- **Угрозы:** конкуренты, изменения законодательства, экономические риски.

Задание 2.2. Выберите 3 наиболее значимых фактора в каждом квадранте (обведите или отметьте звёздочкой). Обоснуйте свой выбор.

Почему эти факторы самые важные?

Этап 3. TOWS-анализ: разработка стратегий

Задание 3.1. Используя метод TOWS , заполните матрицу стратегий. Для этого сопоставьте факторы из разных квадрантов и сформулируйте конкретные действия.

	Возможности (O)	Угрозы (T)
Сильные стороны (S)	SO-стратегии (как использовать сильные стороны для реализации возможностей)	ST-стратегии (как использовать сильные стороны для нейтрализации угроз)
	1.	1.
	2.	2.
	3.	3.
	Возможности (O)	Угрозы (T)
Слабые стороны (W)	WO-стратегии (как преодолеть слабости, используя возможности)	WT-стратегии (как минимизировать слабости и избегать угроз)
	1.	1.
	2.	2.
	3.	3.

Пример формулировки стратегии :

- **SO:** Используя опыт команды в мобильной разработке (S), создать приложение быстрее конкурентов и занять нишу до выхода новых игроков (O).
- **WT:** Минимизировать риск утечки данных (T) путём привлечения внешнего эксперта по безопасности, так как у команды нет такого опыта (W).

Задание 3.2. Из всех сформулированных стратегий выберите **три приоритетные** для реализации в ближайшее время. Заполните таблицу:

№	Стратегия	Почему это приоритетно?	Срок реализации
1			
2			
3			

Этап 4. Визуализация результатов

Задание 4.1. Создайте визуальную карту SWOT-анализа. Используйте один из способов:

- Нарисуйте матрицу на листе А4 с цветовым кодированием квадрантов.
- Используйте онлайн-доску (Miro, Whiteboard) для создания стикеров.
- Создайте диаграмму в любом графическом редакторе.

Требования к визуализации:

- Четыре квадранта чётко обозначены.
- Факторы представлены кратко (ключевые слова).
- Использованы разные цвета для внутренних и внешних факторов.
- Три наиболее важных фактора выделены (размером, цветом или иконкой).

В отчёт вставьте фото/скриншот вашей визуализации.

Этап 5. Анализ и выводы

Задание 5.1. Ответьте на вопросы:

1. Какие сильные стороны проекта являются наиболее устойчивыми и долгосрочными?
2. Какие слабые стороны требуют немедленного устранения? Почему?
3. Какая возможность, на ваш взгляд, самая перспективная для проекта?
4. Какая угроза самая опасная? Как вы планируете её минимизировать?
5. Как изменится SWOT-анализ через год после запуска приложения? Какие факторы могут появиться/исчезнуть?

Задание 5.2. Сформулируйте общий вывод по результатам SWOT-анализа: насколько проект «CollegeHelper» перспективен, какие риски требуют особого внимания, какие стратегии будут реализованы в первую очередь.

Вывод:

Этап 6. Оформление отчёта

Отчёт по практической работе должен содержать:

1. Титульный лист (тема, ФИО студента/участников группы).
2. Цель анализа (задание 1.1).

3. Заполненную таблицу классификации факторов (задание 1.2).
4. Матрицу SWOT с 5+ факторами в каждом квадранте (задание 2.1).
5. Обоснование выбора трёх наиболее значимых факторов (задание 2.2).
6. Матрицу TOWS со стратегиями (задание 3.1).
7. Таблицу с тремя приоритетными стратегиями (задание 3.2).
8. Визуализацию SWOT-анализа (задание 4.1).
9. Ответы на вопросы анализа (задание 5.1).
10. Итоговый вывод (задание 5.2).

Контрольные вопросы (для самопроверки)

1. Что означает аббревиатура SWOT? Перечислите все элементы .
 2. Чем внутренние факторы отличаются от внешних? Приведите примеры .
 3. В чём отличие базового SWOT-анализа от TOWS-анализа?
 4. Почему важно не только выявить факторы, но и определить их приоритетность?
 5. Как часто следует проводить SWOT-анализ в ходе реализации проекта?
 6. Какие ошибки чаще всего допускают при проведении SWOT-анализа?
 7. Как результаты SWOT-анализа могут повлиять на планирование проекта?
-

Практическое занятие №10. Разработка плана управления рисками.

Цель работы

Научиться идентифицировать риски ИТ-проекта, проводить их качественный и количественный анализ, разрабатывать стратегии реагирования и составлять реестр рисков (план управления рисками).

Краткие теоретические сведения

1. Понятие риска в управлении проектами

Риск – это неопределённое событие или условие, которое в случае возникновения может оказать положительное или отрицательное влияние на цели проекта (сроки, стоимость, содержание, качество). Риски бывают:

- **Угрозы** – негативные события, которые могут навредить проекту.
- **Благоприятные возможности** – позитивные события, которые можно использовать для улучшения результатов.

2. Процесс управления рисками

Управление рисками включает следующие этапы (согласно PMBOK):

1. **Планирование управления рисками** – определение подходов, методов и ролей.
2. **Идентификация рисков** – выявление и описание возможных рисков.
3. **Качественный анализ** – оценка вероятности и влияния, приоритизация.
4. **Количественный анализ** – численная оценка последствий (не всегда обязателен).
5. **Планирование реагирования** – разработка мер по снижению угроз и использованию возможностей.
6. **Мониторинг и контроль** – отслеживание рисков, реализация мер.

3. Методы идентификации рисков

- Мозговой штурм с командой.
- Анализ документации (аналогичные проекты, ТЗ, бюджет).
- Опрос экспертов.
- SWOT-анализ (используйте результаты предыдущего занятия).
- Составление контрольных списков (checklists).

4. Качественный анализ рисков

Риски оцениваются по двум параметрам:

- **Вероятность** – от 0 до 1 (или по шкале: очень низкая, низкая, средняя, высокая, очень высокая).
- **Влияние** – степень воздействия на цели проекта (шкала от очень низкого до очень высокого).

На основе этих оценок строится **матрица вероятности и влияния**, которая позволяет ранжировать риски по степени опасности (красная зона – критические, жёлтая – средние, зелёная – незначительные).

5. Стратегии реагирования на риски

Для угроз (негативных рисков):

- **Избежание** – устранение причины риска или изменение плана проекта так, чтобы риск не мог возникнуть.
- **Передача** – перенос последствий риска на третью сторону (например, страхование, аутсорсинг).
- **Снижение** – уменьшение вероятности или влияния риска.
- **Принятие** – осознанное решение не менять планы, создать резерв на случай реализации риска.

Для возможностей (позитивных рисков):

- **Использование** – сделать всё, чтобы возможность реализовалась.
- **Усиление** – увеличить вероятность или влияние.
- **Разделение** – передача части ответственности за реализацию возможности партнёру.
- **Принятие** – использовать, если возможность появится.

6. Реестр рисков

Документ, в котором фиксируются все выявленные риски, их характеристики и планы реагирования. Обычно включает:

- ID риска
 - Описание риска
 - Причина
 - Вероятность
 - Влияние (по шкале)
 - Уровень риска (произведение вероятности и влияния или категория)
 - Владелец риска
 - Стратегия реагирования
 - План действий
 - Триггеры (признаки наступления)
-

Ход работы

Работа выполняется индивидуально или в группах по 2–3 человека. Для анализа используется проект «CollegeHelper» (мобильное приложение для студентов колледжа).

Этап 1. Идентификация рисков

Задание 1.1. Проведите мозговой штурм и составьте список не менее **10 рисков**, которые могут возникнуть при разработке и внедрении проекта «CollegeHelper».

Рассмотрите следующие категории рисков:

- Технические (сложность разработки, интеграция, производительность).
- Организационные (изменение требований, проблемы в команде, недостаток ресурсов).
- Внешние (конкуренты, изменение законодательства, экономическая ситуация).
- Риски управления (неправильная оценка сроков, бюджета).
- Риски безопасности (утечка данных, взлом).

Заполните таблицу:

ID	Категория	Описание риска	Причина
R1	Технический	...	...
R2	...	...	...
...	...	...	...

Этап 2. Качественный анализ рисков

Задание 2.1. Для каждого идентифицированного риска оцените:

- **Вероятность** (по шкале от 1 до 5, где 1 – очень низкая, 5 – очень высокая);
- **Влияние на проект** (по шкале от 1 до 5, где 1 – незначительное, 5 – критическое, может привести к срыву проекта).

Используйте шкалы:

Оценка	Вероятность	Влияние
1	менее 10%	Незначительные задержки / перерасход до 5%
2	10–30%	Задержки до 10% / перерасход до 10%
3	30–50%	Задержки до 20% / перерасход до 20%
4	50–70%	Задержки до 30% / перерасход до 30%
5	более 70%	Срыв сроков более чем на 30% / провал проекта

Задание 2.2. Постройте матрицу вероятности и влияния (5×5) и распределите риски по зонам (критическая, средняя, низкая). Для этого умножьте балл вероятности на балл влияния и определите приоритет:

- **Критические риски** (произведение 15–25) – требуют немедленного планирования реагирования.
- **Средние риски** (произведение 6–14) – требуют внимания, разработки мер.
- **Низкие риски** (произведение 1–5) – можно отслеживать, но не тратить ресурсы.

Заполните таблицу:

ID	Описание риска	Вероятность (1–5)	Влияние (1–5)	Произведение	Приоритет
R1	...	...	...	...	...
R2	...	...	...	...	...
...	...	...	...	...	...

Задание 2.3. Выберите **5 рисков** с наивысшим приоритетом (критическая зона и наиболее опасные из средней) – именно для них будете разрабатывать меры реагирования. Запишите их ID.

Этап 3. Планирование реагирования на риски

Задание 3.1. Для каждого из 5 выбранных рисков разработайте стратегию реагирования и конкретный план действий. Заполните таблицу.

ID	Стратегия	План действий	Владелец риска (роль)	Триггеры (когда начинать действовать)
R1	(избежание/передача/снижение/принятие)	...	...	...
R2	...	...	...	...
...	...	...	...	...

Примечание: если риск представляет собой возможность (позитивный), используйте стратегии для возможностей.

Пример:

- **Риск:** Уход ключевого разработчика.
- **Стратегия:** Снижение.
- **План:** Обеспечить кросс-обучение, документирование кода, иметь контакты фрилансеров на подмену.
- **Владелец:** Руководитель проекта.
- **Триггер:** Заявление об увольнении, длительное отсутствие.

Этап 4. Создание реестра рисков

Задание 4.1. Оформите итоговый реестр рисков проекта «CollegeHelper», включив в него все идентифицированные риски (минимум 10) с оценками и планы реагирования для топ-5. Используйте шаблон ниже. Реестр можно оформить в виде таблицы в текстовом документе или Excel.

Шаблон реестра рисков:

ID	Категория	Описание риска	Причина	Вероятность	Влияние	Произведение	Приоритет	Стратегия	План действий	Владелец	Триггеры	Статус
----	-----------	----------------	---------	-------------	---------	--------------	-----------	-----------	---------------	----------	----------	--------

Статус может быть: активный, реализовался, закрыт.

Этап 5. Мониторинг и контроль (ситуационный анализ)

Задание 5.1. Представьте, что один из ваших критических рисков реализовался. Опишите ситуацию:

- Какой риск произошёл?
- Каковы последствия (влияние на сроки, бюджет, качество)?
- Сработал ли ваш план реагирования? Что было сделано?
- Какие дополнительные меры потребовались?

Заполните:

- **Риск:** R___ (опишите)
- **Ситуация:**

- **Последствия:**

- **Реализация плана:**

- **Итог:**

Задание 5.2. Предложите, как можно улучшить процесс управления рисками в проекте на основе этого опыта.

Этап 6. Оформление отчёта

Отчёт по практической работе должен содержать:

1. Титульный лист (тема, ФИО студента/участников группы).
2. Результаты идентификации рисков (таблица из задания 1.1).
3. Результаты качественного анализа (таблица из задания 2.2) и список топ-5 рисков.
4. Планы реагирования (таблица из задания 3.1).
5. Итоговый реестр рисков (задание 4.1).
6. Анализ реализовавшегося риска (задание 5.1) и предложения по улучшению (5.2).
7. Вывод по работе: что нового узнали, какие навыки приобрели, с какими трудностями столкнулись.

Контрольные вопросы (для самопроверки)

1. Что такое риск проекта? Чем угроза отличается от возможности?
 2. Перечислите этапы управления рисками.
 3. Какие методы используются для идентификации рисков?
 4. Как проводится качественный анализ рисков? Что такое матрица вероятности и влияния?
 5. Назовите стратегии реагирования на негативные риски (угрозы). Приведите примеры.
 6. Какие стратегии применяются для позитивных рисков (возможностей)?
 7. Что включает в себя реестр рисков? Кто его ведёт?
 8. Почему важно назначать владельца риска?
-

Практическое занятие №11. Планирование задач в Scrum.

Цель работы

Научиться планировать работы в рамках Scrum-методологии: проводить оценку пользовательских историй в story points, определять velocity команды, формировать бэклог спринта, декомпозировать истории на задачи и организовывать ключевые события Scrum (планирование спринта, daily scrum, ретроспектива).

Краткие теоретические сведения

1. Основы Scrum

Scrum – это фреймворк для управления проектами, чаще всего используемый в разработке ПО. Он основан на итеративном подходе: работа ведётся короткими циклами (спринтами) длительностью 1–4 недели.

Роли в Scrum:

- **Владелец продукта (Product Owner)** – отвечает за бэклог продукта, определяет приоритеты, представляет интересы заказчика.
- **Scrum-мастер** – обеспечивает соблюдение принципов Scrum, помогает команде устранять препятствия.
- **Команда разработки** – кросс-функциональная группа, которая выполняет задачи (разработчики, тестировщики, аналитики).

Артефакты Scrum:

- **Бэклог продукта (Product Backlog)** – упорядоченный список всего, что может понадобиться в продукте (пользовательские истории, ошибки, технические задачи).
- **Бэклог спринта (Sprint Backlog)** – набор элементов бэклога продукта, выбранных для реализации в текущем спринте, плюс план их достижения.
- **Инкремент продукта** – сумма всех выполненных задач спринта, готовый к поставке фрагмент продукта.

События Scrum:

- **Планирование спринта (Sprint Planning)** – команда определяет, что будет сделано в спринте и как.
- **Daily Scrum (ежедневное совещание)** – 15-минутная встреча для синхронизации.
- **Обзор спринта (Sprint Review)** – демонстрация результатов заказчику.
- **Ретроспектива спринта (Sprint Retrospective)** – анализ прошедшего спринта для улучшения процессов.

2. Оценка задач в story points

Story points – относительная единица измерения трудоёмкости пользовательской истории. Вместо абсолютных часов команды оценивают сложность, объём и неопределённость задачи. Обычно используется последовательность Фибоначчи (1, 2, 3, 5, 8, 13, 21...), так как она отражает растущую неопределённость в больших оценках.

Методы оценки:

- **Покер планирования (Planning Poker)** – каждый член команды тайно даёт свою оценку, затем оценки обсуждаются до консенсуса.
- **Экспертная оценка** – на основе опыта.

3. Velocity (скорость команды)

Velocity – количество story points, которое команда в среднем выполняет за один спринт. Velocity используется для прогнозирования: сколько историй команда сможет взять в следующий спринт.

4. Планирование спринта

На входе: бэклог продукта с приоритетами, оценённый в story points, и velocity команды.

На выходе: цель спринта и бэклог спринта.

Процесс:

1. Владелец продукта представляет наиболее приоритетные истории.
2. Команда обсуждает, может ли реализовать их, учитывая свою скорость.
3. Команда выбирает истории, которые планирует выполнить.
4. Выбранные истории декомпозируются на конкретные задачи (обычно в часах) для ежедневного планирования.

5. Доска задач (Scrum board)

Визуализирует прогресс спринта. Обычно содержит колонки: «To Do» (запланировано), «In Progress» (в работе), «Done» (готово). Для каждой задачи указывается ответственный и статус.

Ход работы

Работа выполняется в группах по 3–4 человека (имитация Scrum-команды).
Используется проект «CollegeHelper» – мобильное приложение для студентов колледжа. Если у вас нет готового бэклога с предыдущих занятий, воспользуйтесь приведённым ниже списком пользовательских историй (можно дополнить).

Исходный бэклог продукта (Product Backlog) для проекта «CollegeHelper»

ID	Пользовательская история	Приоритет (MoSCoW)	Story points (пока не заполнено)
US1	As a student, I want to view my daily schedule so that I know which classes to attend	Must	
US2	As a student, I want to view weekly schedule so that I can plan my week	Should	
US3	As a student, I want to receive push notifications about schedule changes	Could	
US4	As a student, I want to view college news feed	Must	
US5	As a student, I want to submit a maintenance request (broken equipment)	Must	
US6	As a student, I want to track the status of my request	Should	
US7	As a student, I want to view my grades for each subject	Must	
US8	As a student, I want to see attendance statistics	Could	
US9	As a teacher, I want to post news to all students	Must	

ID	Пользовательская история	Приоритет (MoSCoW)	Story points (пока не заполнено)
US10	As a teacher, I want to upload grades	Should	
US11	As an administrator, I want to manage user accounts	Must	
US12	As a student, I want to log in securely with college credentials	Must	
US13	As a student, I want to reset my password	Should	
US14	As a student, I want to see a map of the college campus	Won't have (future)	
US15	As a student, I want to sync schedule with personal calendar	Could	

Этап 1. Оценка пользовательских историй в story points

Задание 1.1. Изучите список историй. Как команда, проведите оценку каждой истории в story points. Используйте последовательность Фибоначчи: 1, 2, 3, 5, 8, 13, 21. Договоритесь о единой шкале (например, 1 – очень простая задача, 5 – средняя, 13 – сложная).

Заполните столбец «Story points» в таблице выше. Результаты запишите в отчёт.

Методика: можно провести «покер планирования» внутри группы: каждый участник предлагает свою оценку, обсуждаете и приходите к консенсусу.

Этап 2. Определение velocity команды

Задание 2.1. Предположим, ваша команда имеет среднюю скорость (velocity) **20 story points** за двухнедельный спринт. Эта цифра основана на предыдущих спринтах (в реальности она вычисляется по факту). Если у вас другая оценка суммарного бэклога, можно скорректировать, но для простоты примем 20.

Вопрос: Почему важно знать velocity команды? Кратко поясните.

Этап 3. Планирование спринта (Sprint Planning)

Задание 3.1. Вы – команда, владелец продукта (можно распределить роли) предлагает начать разработку MVP (минимально жизнеспособного продукта). На основе приоритетов MoSCoW (Must have) и вашей скорости в 20 story points, выберите истории для первого спринта. Учитывайте, что сумма их story points не должна превышать velocity (можно чуть меньше для запаса).

Выбранные истории (укажите ID и названия):

1. _____
2. _____
3. _____
4. _____
5. _____

Задание 3.2. Сформулируйте **цель спринта** (Sprint Goal) – одно-два предложения, описывающие, что будет достигнуто в результате спринта. Например: «Реализовать

базовую функциональность для студентов: просмотр расписания, новостей и возможность подачи заявок».

Цель спринта:

Задание 3.3. Для каждой выбранной истории проведите декомпозицию на конкретные задачи (tasks). Задачи должны быть технически реализуемыми и обычно оцениваются в часах (или просто описываются). Заполните таблицу.

История	Задачи	Примерная трудоёмкость (часы)
US1 – Просмотр расписания на сегодня	1. Создать экран расписания 2. Получить данные с сервера 3. Отобразить список занятий	4 6 3
US2 – ...	...	...
...	...	...

Примечание: точная оценка в часах не обязательна, достаточно показать понимание декомпозиции.

Этап 4. Создание бэклога спринта (Sprint Backlog) и доски задач

Задание 4.1. На основе полученных задач составьте бэклог спринта – упорядоченный список всех задач, которые нужно выполнить. Можно оформить в виде таблицы:

ID задачи	Описание задачи	Ответственный (роль)	Статус (To Do / In Progress / Done)
T1	...	...	To Do
T2	...	...	To Do
...	...	...	...

Задание 4.2. Нарисуйте (или опишите) доску задач для первого спринта. Как будут распределены задачи по колонкам? Используйте классические колонки: «To Do», «In Progress», «Done». Можно добавить «Review» или «Testing».

Визуализация: можно изобразить схематично или описать словами.

Этап 5. Планирование Daily Scrum и других событий

Задание 5.1. Опишите, как будет проходить ежедневное совещание (Daily Scrum) в вашей команде:

- В какое время?
- Сколько длится?
- Какие три вопроса задаёт каждый участник? (Что сделано вчера? Что буду делать сегодня? Какие препятствия?)
- Кто его проводит?

Задание 5.2. Представьте, что спринт завершён. Опишите план проведения ретроспективы спринта. Какие вопросы вы будете обсуждать? Какой формат (Start/Stop/Continue, Sailboat и т.д.)?

Этап 6. Оформление отчёта

Отчёт должен содержать:

1. Титульный лист (тема, ФИО участников группы).
 2. Результаты оценки историй в story points (задание 1.1 – таблица с заполненным столбцом).
 3. Ответ на вопрос о velocity (задание 2.1).
 4. Список историй, выбранных в спринт, и цель спринта (задание 3.1–3.2).
 5. Декомпозиция выбранных историй на задачи (таблица из задания 3.3).
 6. Бэклог спринта (задание 4.1).
 7. Описание доски задач (задание 4.2).
 8. Описание Daily Scrum и ретроспективы (задание 5.1–5.2).
 9. Вывод по работе: что нового узнали, какие навыки приобрели, что показалось сложным.
-

Контрольные вопросы (для самопроверки)

1. Какие роли существуют в Scrum? Опишите обязанности каждой.
2. Что такое бэклог продукта и чем он отличается от бэклога спринта?
3. Зачем оценивать задачи в story points, а не в часах?
4. Что такое velocity команды и как он используется при планировании?
5. Какие вопросы обсуждаются на Daily Scrum?
6. Для чего проводится ретроспектива спринта?
7. Что такое цель спринта и почему она важна?

Практическое занятие №12. Ведение отчетности.

Цель работы

Научиться разрабатывать структуру отчётов по проекту, заполнять различные виды отчётов (статус-отчёты, отчёты по спринту, отчёты для заказчика), использовать визуализацию данных и анализировать информацию для принятия управленческих решений.

Краткие теоретические сведения

1. Зачем нужна отчётность в проекте?

Отчётность – это инструмент коммуникации и контроля. Она позволяет:

- Информировать заинтересованных лиц (стейкхолдеров) о ходе проекта;
- Фиксировать достигнутые результаты и отклонения;
- Обосновывать запросы на дополнительные ресурсы или изменение сроков;
- Анализировать эффективность работы команды;
- Обеспечивать прозрачность и доверие.

2. Виды отчётов в ИТ-проектах

Тип отчёта	Периодичность	Целевая аудитория	Содержание
Ежедневный статус (Daily Report)	Ежедневно	Руководитель проекта, команда	Что сделано вчера, планы на сегодня, блокеры
Еженедельный статус (Weekly Status Report)	Раз в неделю	Руководство, заказчик	Выполненные задачи, планы на следующую неделю, риски, проблемы, % завершения
Отчёт по спринту (Sprint Review)	По окончании спринта	Заказчик, Product Owner	Что сделано, демо результатов, отзывы, планы на следующий спринт
Финансовый отчёт	Ежемесячно/по этапам	Финансовый отдел, инвесторы	Фактические затраты, отклонения от бюджета, прогноз
Отчёт по рискам	Еженедельно/ежемесячно	Руководитель, стейкхолдеры	Текущие риски, их статус, меры реагирования
Итоговый отчёт (Project Closure Report)	По окончании проекта	Все стейкхолдеры	Итоги, достигнутые цели, уроки, статистика

3. Структура типового статус-отчёта

Большинство статус-отчётов включают следующие разделы:

- Общая информация:** название проекта, отчётный период, автор, дата.
 - Ключевые показатели:** % выполнения, статус (зелёный/жёлтый/красный), прогноз.
 - Выполненные задачи за период** (с указанием ответственных).
 - Планы на следующий период.**
 - Риски и проблемы** – текущие, с оценкой влияния и планом действий.
 - Отклонения** – от графика, бюджета, содержания.
 - Запросы/решения** – что требуется от руководства или заказчика.
 - Приложения** – детали, графики, диаграммы.
- ### 4. Правила визуализации данных
- Используйте графики (Ганта, Burndown chart) для иллюстрации прогресса.
 - Для статусов применяйте светофор: зелёный – всё хорошо, жёлтый – есть проблемы, красный – критическая ситуация.
 - Диаграммы сгорания задач (Burndown) показывают, сколько работы осталось до конца спринта или проекта.
 - Круговые диаграммы и гистограммы – для распределения задач или затрат.

5. Инструменты для ведения отчётности

- **Текстовые процессоры** (Word, Google Docs) – для формальных отчётов.
- **Электронные таблицы** (Excel, Google Sheets) – для данных и графиков.
- **Системы управления проектами** (Jira, Redmine, Trello, Asana) – автоматически генерируют отчёты.
- **Специализированные BI-инструменты** (Power BI, Tableau) – для продвинутой аналитики.
- **Почта и мессенджеры** – для ежедневных статусов (например, в Telegram-ботах).

Ход работы

Работа выполняется в группах по 2–3 человека (имитация проектной команды). Все задания основаны на сквозном проекте «CollegeHelper» – мобильном приложении для студентов колледжа. Используйте материалы предыдущих занятий (бэклог, диаграмму Ганта, бюджет, реестр рисков) для заполнения отчётов.

Этап 1. Разработка шаблона еженедельного статус-отчёта

Задание 1.1. Изучив теоретический материал, разработайте шаблон еженедельного статус-отчёта для проекта «CollegeHelper». Шаблон должен включать все необходимые разделы и быть удобным для заполнения. Оформите его в виде таблицы в текстовом документе или электронной таблице.

Требования к шаблону:

- Заголовок (название проекта, отчётная неделя, дата, автор).
- Раздел «Ключевые показатели» (процент выполнения, статус, прогноз по срокам и бюджету).
- Раздел «Выполненные задачи за неделю».
- Раздел «Планы на следующую неделю».
- Раздел «Риски и проблемы».
- Раздел «Отклонения».
- Раздел «Запросы к заказчику/руководству».

Результат: заполненный шаблон (можно пока без данных) в отчёте.

Этап 2. Заполнение еженедельного отчёта на основе данных

Задание 2.1. Представьте, что проект находится на 3-й неделе разработки. Используя информацию из предыдущих работ (например, из диаграммы Ганта, бэклога, реестра рисков), заполните созданный шаблон отчёта за прошедшую неделю. При необходимости придумайте недостающие данные (какие задачи были выполнены, какие возникли проблемы).

Исходные допущения:

- Команда работает по Scrum, спринт длится 2 недели.
- На текущей неделе завершается первый спринт.
- Были выполнены задачи: авторизация, просмотр расписания, новостная лента.
- Возникла проблема: задержка в интеграции с сервером колледжа (нужны дополнительные согласования).
- Бюджет пока в пределах плана, но есть риск увеличения расходов на хостинг.

Заполните все разделы шаблона максимально реалистично.

Этап 3. Создание отчёта по завершению спринта (Sprint Review Report)

Задание 3.1. После завершения первого спринта необходимо подготовить отчёт для демонстрации заказчику (администрации колледжа). Отчёт должен включать:

- Цель спринта.
- Список завершённых пользовательских историй (с указанием story points).
- Демонстрация результатов (краткое описание, можно добавить скриншоты экранов, если есть).
- Отзывы пользователей (придумайте, например, от студентов-тестировщиков).
- Метрики спринта: velocity, объём незавершённых задач (если есть), диаграмма сгорания (Burndown chart).
- Планы на следующий спринт.

Задание 3.2. Постройте **диаграмму сгорания задач (Burndown chart)** для первого спринта. Для этого:

1. Определите общее количество story points, запланированных на спринт (из задания по планированию спринта, занятие №11).
2. Придумайте, как в идеале (или с небольшими отклонениями) уменьшался объём оставшейся работы по дням.
3. Начертите график (можно в Excel или от руки, затем вставить скриншот).

Пример диаграммы: ось X – дни спринта, ось Y – оставшиеся story points.

Идеальная линия – прямая от максимума до нуля, реальная – ступенчатая.

Этап 4. Визуализация данных проекта

Задание 4.1. Используя данные из финансового бюджета (практическое занятие №7) и реестра рисков (занятие №10), создайте визуальные элементы для итогового отчёта:

- **Круговая диаграмма** – структура затрат (трудовые, оборудование, накладные, резерв).
- **Гистограмма** – распределение рисков по уровню (критические, средние, низкие).
- **График** – фактическое выполнение задач по сравнению с планом (можно использовать процент выполнения по неделям).

Достаточно одного-двух графиков, оформленных аккуратно с подписями.

Этап 5. Анализ отчётности и рекомендации

Задание 5.1. Проанализируйте полученные отчёты и ответьте на вопросы:

1. Насколько достоверно и полно отражено состояние проекта?
2. Какие разделы отчёта наиболее важны для разных стейкхолдеров (заказчика, команды, руководства)?
3. Какие дополнительные метрики вы бы включили в регулярную отчётность?
4. Как часто нужно обновлять отчёты, чтобы они оставались полезными, но не отнимали слишком много времени?

Задание 5.2. Сформулируйте три рекомендации по улучшению процесса отчётности в вашей команде.

Этап 6. Оформление отчёта

Итоговый отчёт по практической работе должен содержать:

1. Титульный лист (тема, ФИО участников группы).
2. Разработанный шаблон еженедельного статус-отчёта (задание 1.1).
3. Заполненный еженедельный отчёт (задание 2.1).
4. Отчёт по завершению спринта (задание 3.1) с диаграммой сгорания (3.2).
5. Визуализацию данных (графики из задания 4.1).
6. Анализ и рекомендации (задания 5.1 и 5.2).
7. Вывод по работе: что нового узнали, какие навыки приобрели, с какими трудностями столкнулись.

Контрольные вопросы (для самопроверки)

1. Какие виды отчётов используются в управлении ИТ-проектами? Для чего каждый из них предназначен?
2. Что такое диаграмма сгорания (Burndown chart) и как её строить?
3. Какие разделы обязательно должны быть в еженедельном статус-отчёте?
4. Почему важно визуализировать данные в отчётах? Какие типы визуализации наиболее эффективны?
5. Как часто следует готовить отчёты для заказчика? Для команды?
6. Какие ошибки чаще всего допускают при подготовке отчётов?
7. Как отчётность помогает управлять рисками и проблемами?

Раздел 3. Мониторинг, контроль и завершение проекта

Тема 3.1. Мониторинг прогресса и контроль качества

Практическое занятие №13. Анализ отклонений.

Цель работы

Научиться выявлять отклонения фактического хода проекта от планового по срокам, стоимости и содержанию, применять метод освоенного объема (EVM) для количественной оценки эффективности, разрабатывать корректирующие мероприятия для возврата проекта в допустимые рамки.

Краткие теоретические сведения

1. Понятие отклонения в проекте

Отклонение – это разница между запланированными и фактическими показателями проекта. Отклонения могут быть:

- **По срокам** – фактическая дата завершения работ отличается от плановой.
- **По стоимости** – фактические затраты отличаются от бюджетных.
- **По содержанию** – выполненные работы не соответствуют запланированным (недоделки, лишние работы).
- **По качеству** – результаты не удовлетворяют требованиям.

2. Метод освоенного объема (Earned Value Management – EVM)

EVM – это метод интегрированной оценки выполнения проекта по срокам и стоимости. Он позволяет ответить на вопросы: не отстаём ли мы от графика? укладываемся ли в бюджет?

Основные показатели EVM:

Показатель	Обозначение	Что означает	Формула
Плановый объем (Planned Value)	PV	Стоимость работ, которые должны быть выполнены на текущую дату по плану	—
Освоенный объем (Earned Value)	EV	Стоимость фактически выполненных работ на текущую дату	—
Фактическая стоимость (Actual Cost)	AC	Реальные затраты на выполненные работы на текущую дату	—
Отклонение по стоимости (Cost Variance)	CV	Перерасход (+) или экономия (–) бюджета	$CV = EV - AC$
Отклонение по срокам (Schedule Variance)	SV	Опережение (+) или отставание (–) от графика	$SV = EV - PV$
Индекс выполнения стоимости (Cost Performance Index)	CPI	Эффективность расходования средств (если $CPI < 1$ – перерасход)	$CPI = EV / AC$
Индекс выполнения сроков (Schedule Performance Index)	SPI	Эффективность соблюдения графика (если $SPI < 1$ – отставание)	$SPI = EV / PV$

Прогнозные показатели:

- **Оценка по завершении (Estimate at Completion – EAC)** – прогноз общей стоимости проекта. Может рассчитываться по-разному, например:

- $EAC = AC + (BAC - EV) / CPI$ (если тенденция сохранится).

- **Оценка до завершения (Estimate to Complete – ETC) = EAC – AC.**

3. Анализ причин отклонений

После выявления отклонений необходимо определить их причины:

- Внешние (задержки поставок, проблемы у заказчика).
- Внутренние (недооценка трудоёмкости, низкая производительность, ошибки).
- Технические (сложность реализации, неожиданные проблемы).

4. Корректирующие действия

В зависимости от результатов анализа могут быть предприняты:

- Перераспределение ресурсов.
 - Корректировка графика (например, параллельное выполнение задач).
 - Пересмотр бюджета (запрос дополнительных средств).
 - Изменение содержания (сокращение объёма, перенос части задач).
 - Усиление контроля и мотивации.
-

Ход работы

Работа выполняется индивидуально или в группах по 2–3 человека. Используется проект «CollegeHelper» (мобильное приложение для студентов). Для выполнения заданий вам понадобятся данные из предыдущих работ (диаграмма Ганта, бюджет), а также фактические данные, приведённые ниже.

Исходные данные

Проект «CollegeHelper» должен быть завершён за 4 месяца (16 недель). Бюджет (BAC – Budget at Completion) составляет **1 500 000 руб.** Плановые затраты распределены равномерно по неделям (примерно 93 750 руб. в неделю). На текущий момент (конец 8-й недели) должны быть выполнены следующие работы:

- Завершён этап анализа и проектирования (плановая стоимость 200 000 руб.).
- Завершён этап дизайна (плановая стоимость 150 000 руб.).
- Начата разработка – выполнено 40% работ по разработке (плановая стоимость всего этапа разработки 700 000 руб., т.е. должно быть выполнено на 280 000 руб.).

Итого **PV** на конец 8-й недели = 200 000 + 150 000 + 280 000 = **630 000 руб.**

Фактические данные на конец 8-й недели:

- Этап анализа и проектирования выполнен полностью, фактические затраты – 210 000 руб.
- Этап дизайна выполнен полностью, фактические затраты – 160 000 руб.
- Разработка выполнена только на 30% (фактически на 210 000 руб. в плановых ценах), при этом фактические затраты на разработку составили 250 000 руб.
- Фактические затраты на управление проектом (административные) за 8 недель – 80 000 руб. (эти затраты не были отдельно запланированы, но должны быть учтены в бюджете; будем считать, что они включены в общий BAC равномерно, т.е. по плану тоже должны быть затраты на управление, но мы их отдельно не выделяли; для простоты примем, что плановые затраты на управление уже включены в указанные суммы этапов, а фактические дополнительные затраты – это превышение).

Таким образом, **EV** (освоенный объем) = стоимость выполненных работ в плановых ценах:

- Анализ и проектирование: 200 000 руб. (выполнено полностью).
- Дизайн: 150 000 руб. (выполнено полностью).
- Разработка: 30% от 700 000 = 210 000 руб.

Итого **EV** = 200 000 + 150 000 + 210 000 = **560 000 руб.**

AC (фактические затраты):

- Анализ и проектирование: 210 000 руб.
- Дизайн: 160 000 руб.
- Разработка: 250 000 руб.
- Управление: 80 000 руб.

Итого **AC** = 210 000 + 160 000 + 250 000 + 80 000 = **700 000 руб.**

Этап 1. Анализ отклонений по срокам и стоимости

Задание 1.1. На основе приведённых данных рассчитайте:

- Отклонение по стоимости (CV) = EV – AC.
- Отклонение по срокам (SV) = EV – PV.
- Индекс выполнения стоимости (CPI) = EV / AC.

- Индекс выполнения сроков (SPI) = EV / PV.

Результаты занесите в таблицу:

Показатель	Значение
PV	630 000
EV	560 000
AC	700 000
CV	
SV	
CPI	
SPI	

Задание 1.2. Интерпретируйте полученные значения:

- Что означают положительные/отрицательные значения CV и SV?
- О чём говорят индексы CPI и SPI (больше 1, меньше 1)?
- Сформулируйте вывод о текущем состоянии проекта: укладывается ли он в бюджет? соблюдаются ли сроки?

Этап 2. Прогнозирование

Задание 2.1. Используя метод EVM, сделайте прогноз общей стоимости проекта (EAC) при условии, что текущая тенденция эффективности (CPI) сохранится до конца проекта. Формула: $EAC = AC + (BAC - EV) / CPI$, где BAC = 1 500 000 руб.

Рассчитайте:

- EAC = ?
- ETC (оценка до завершения) = EAC – AC.

Задание 2.2. Оцените, насколько прогнозируемая итоговая стоимость отличается от первоначального бюджета (BAC). Каков перерасход в процентах? Сделайте вывод.

Этап 3. Анализ причин отклонений

Задание 3.1. Изучите исходные данные и подумайте, какие причины могли привести к отклонениям. Заполните таблицу возможных причин:

Отклонение	Возможные причины (не менее двух)
Перерасход по этапу анализа и проектирования (210 тыс. вместо 200)	
Перерасход по дизайну (160 тыс. вместо 150)	
Значительный перерасход по разработке (250 тыс. факт против 210 тыс. освоенных)	
Дополнительные затраты на управление (80 тыс.)	
Отставание по разработке (30% вместо 40%)	

Задание 3.2. Какие из этих причин являются внешними, а какие – внутренними? Какие можно было предусмотреть на этапе планирования?

Этап 4. Разработка корректирующих действий

Задание 4.1. Предложите не менее трёх корректирующих мероприятий, которые помогут вернуть проект в приемлемые рамки (по срокам и/или бюджету). Для каждого мероприятия укажите ожидаемый эффект и возможные риски.

Мероприятие	Ожидаемый эффект	Риски
1. ...	...	...
2. ...	...	...
3. ...	...	...

Задание 4.2. Выберите одно мероприятие как наиболее приоритетное и обоснуйте свой выбор.

Этап 5. Анализ отклонений по содержанию (опционально)

Задание 5.1. Предположим, что заказчик запросил отчёт о выполненных функциях. Сравните запланированный состав функциональности MVP (из занятия №4 или №11) с тем, что реально сделано на 8-й неделе. Если есть несоответствия, опишите их.

Подсказка: используйте бэклог и данные о выполненных задачах из предыдущих занятий. Если точных данных нет, смоделируйте ситуацию: например, из 10 запланированных функций реализовано 8, две функции отложены.

Задание 5.2. Какие риски возникают из-за отклонений по содержанию? Как это может повлиять на удовлетворённость заказчика?

Этап 6. Оформление отчёта

Отчёт по практической работе должен содержать:

1. Титульный лист (тема, ФИО студента/участников группы).
2. Результаты расчётов (таблица из задания 1.1) и их интерпретацию (задание 1.2).
3. Прогнозные показатели (задание 2.1) и вывод (2.2).
4. Анализ причин отклонений (таблица 3.1 и ответы 3.2).
5. Таблицу корректирующих мероприятий (4.1) и обоснование приоритетного (4.2).
6. Анализ по содержанию (если выполняли, задание 5).
7. Общий вывод по работе: что нового узнали, какие навыки приобрели, как метод EVM помогает в управлении проектами.

Контрольные вопросы (для самопроверки)

1. Что такое отклонение по стоимости и по срокам? Как они рассчитываются?
2. Какие показатели метода освоенного объема вы знаете? Что они характеризуют?
3. Что означают значения CPI и SPI меньше 1? Больше 1?
4. Как можно спрогнозировать итоговую стоимость проекта с помощью EVM?
5. Какие корректирующие действия можно предпринять при отставании от графика?
6. Какие корректирующие действия возможны при перерасходе бюджета?
7. Почему важно анализировать не только численные отклонения, но и их причины?

Практическое занятие №14. Корректировка проекта.

Цель работы

Научиться разрабатывать план корректирующих действий на основе анализа отклонений, оформлять запросы на изменения, пересматривать базовые планы проекта (сроки, бюджет, содержание) и управлять их согласованием с заинтересованными сторонами.

Краткие теоретические сведения

1. Необходимость корректировки проекта

В ходе реализации любого ИТ-проекта возникают отклонения от первоначальных планов: задержки, перерасход бюджета, изменение требований, технические проблемы. Корректировка проекта – это процесс внесения изменений в планы, методы или ресурсы для приведения проекта в соответствие с целями или для адаптации к новым условиям.

2. Процесс управления изменениями

Управление изменениями – это формальная процедура, обеспечивающая, чтобы все изменения в проекте рассматривались, оценивались, утверждались и документировались.

Основные шаги процесса:

1. **Идентификация необходимости изменения** – выявление проблемы или возможности (например, по результатам анализа отклонений, новых требований заказчика).

2. **Подготовка запроса на изменение (Change Request)** – документ, описывающий суть изменения, причины, влияние на сроки, бюджет, качество, риски.

3. **Анализ влияния** – оценка последствий внесения изменения (может выполняться командой, руководителем).

4. **Согласование и утверждение** – решение принимается уполномоченным лицом (руководитель проекта, заказчик, комитет по изменениям).

5. **Внесение изменения в планы** – обновление базовых планов (расписания, бюджета, содержания, документации).

6. **Доведение до команды и контроль исполнения** – информирование всех участников и мониторинг реализации.

3. Виды корректировок

Тип корректировки	Описание	Пример
Корректирующее действие	Мероприятие, направленное на устранение уже возникшего отклонения и возврат к плану	Ускорение работ за счёт сверхурочных
Предупреждающее действие	Мероприятие, направленное на предотвращение возможного будущего отклонения	Дополнительное обучение команды новой технологии
Исправление дефекта	Устранение несоответствия в продукте	Исправление ошибки в коде
Изменение содержания	Добавление, удаление или изменение функциональности	Добавление нового модуля по требованию заказчика
Обновление планов	Пересмотр сроков, бюджета, ресурсов	Перенос даты релиза

4. Документирование изменений

Запрос на изменение (Change Request) обычно включает:

- Уникальный идентификатор;
- Дата подачи;
- Автор запроса;
- Описание изменения (что предлагается изменить и почему);

- Причина изменения (ссылка на отклонение, новое требование и т.д.);
- Влияние на цели проекта (сроки, стоимость, качество, содержание);
- Влияние на другие области (риски, ресурсы, контракты);
- Предлагаемое решение;
- Рекомендация (утвердить/отклонить);
- Подписи согласующих лиц.

После утверждения изменения обновляются **базовые планы** (baselines) проекта.

5. Коммуникация изменений

Все заинтересованные стороны должны быть проинформированы об утверждённых изменениях. Для этого используются:

- Обновлённые отчёты о статусе;
- Совещания;
- Электронные письма;
- Обновление проектной документации.

Ход работы

Работа выполняется индивидуально или в группах по 2–3 человека. Используется сквозной проект «CollegeHelper» и результаты анализа отклонений, полученные на занятии №13.

Исходные данные (напоминание)

- Проект: мобильное приложение для студентов колледжа.
- Бюджет (BAC): 1 500 000 руб.
- Длительность: 16 недель.
- На конец 8-й недели зафиксированы отклонения:
 - Отставание по срокам (SPI = 0,89).
 - Перерасход бюджета (CPI = 0,8).
 - Прогнозная стоимость (EAC) – около 1 875 000 руб.
 - Выполнено 8 из 10 запланированных функций MVP.

Вам предстоит разработать план корректировки проекта, чтобы завершить его с приемлемыми показателями.

Этап 1. Анализ текущей ситуации и выявление необходимости изменений

Задание 1.1. Используя данные анализа отклонений (занятие №13), определите основные проблемы проекта на текущий момент. Заполните таблицу:

Проблема	Причина (из анализа)	Степень критичности (высокая/средняя/низкая)
1. Отставание от графика		
2. Перерасход бюджета		
3. Недовыполнение объёма работ		
4. ... (добавьте, если есть)		

Задание 1.2. Какие из этих проблем требуют немедленной корректировки? Обоснуйте.

Этап 2. Разработка вариантов корректирующих действий

Задание 2.1. Предложите не менее трёх различных вариантов действий для возврата проекта в приемлемые рамки. Для каждого варианта опишите:

- Суть действий.
- Влияние на сроки (сократятся/увеличатся).
- Влияние на бюджет (потребуется дополнительные средства или удастся сэкономить).
- Влияние на содержание (какие функции пострадают или будут добавлены).

- Влияние на качество.
- Риски, связанные с этим вариантом.

Вариант А (жёсткая экономия): сократить объём тестирования, использовать менее дорогих специалистов, перенести часть функций в следующий релиз.

Вариант Б (ускорение): привлечь дополнительных разработчиков (фрилансеров), работать сверхурочно, оплатить ускорение работ.

Вариант В (пересмотр бюджета и сроков): согласовать с заказчиком увеличение бюджета и перенос даты завершения.

Вариант Г (ваш собственный): ... (предложите свой вариант).

Заполните таблицу для каждого варианта (можно использовать форму ниже).

Параметр	Вариант А	Вариант Б	Вариант В	Вариант Г
Суть действий				
Влияние на сроки				
Влияние на бюджет				
Влияние на содержание				
Влияние на качество				
Риски				

Этап 3. Выбор оптимального варианта и разработка плана корректировки

Задание 3.1. Проанализируйте все варианты и выберите один (или комбинацию), который, на ваш взгляд, является наиболее реалистичным и приемлемым для проекта «CollegeHelper». Обоснуйте свой выбор, учитывая интересы заказчика (администрация колледжа), ограниченность ресурсов и важность функциональности.

Задание 3.2. На основе выбранного варианта разработайте детальный план корректирующих действий. План должен включать:

- Конкретные шаги (что будет сделано).
- Ответственные за каждый шаг.
- Сроки выполнения.
- Необходимые ресурсы (деньги, люди, материалы).
- Ожидаемые результаты.

Форма для плана:

№	Действие	Ответственный	Срок	Ресурсы	Ожидаемый результат
1					
2					
...					

Этап 4. Оформление запроса на изменение (Change Request)

Задание 4.1. Оформите официальный запрос на изменение для утверждения выбранного варианта корректировки. Используйте шаблон, приведённый в приложении. Заполните все поля.

Этап 5. Пересмотр базовых планов проекта

Задание 5.1. Внесите изменения в следующие документы проекта (можно кратко, в виде новых значений или описаний):

- **Новый календарный план (диаграмма Ганта):** укажите новую дату окончания проекта, пересмотрите длительность оставшихся задач.
- **Новый бюджет:** рассчитайте обновлённый бюджет с учётом корректирующих действий (если требуется увеличение, укажите новую сумму).

- **Обновлённое содержание (Score):** если вы решили сократить или перенести часть функциональности, перечислите, какие именно функции останутся в текущем релизе, а какие будут реализованы позже.
 - **Обновлённый реестр рисков:** добавьте новые риски, связанные с изменениями.
-

Этап 6. Коммуникация изменений

Задание 6.1. Подготовьте краткое информационное сообщение для команды проекта о предстоящих изменениях. Сообщение должно быть понятным, мотивирующим и содержать основные пункты плана. (Напишите текст письма или объявления.)

Задание 6.2. Подготовьте краткую презентацию (3–4 слайда) для заказчика, в которой обоснуйте необходимость изменений, опишите новый план и запросите утверждение. Слайды можно описать текстом или сделать скриншоты, если работаете в электронном виде.

Этап 7. Оформление отчёта

Отчёт по практической работе должен содержать:

1. Титульный лист (тема, ФИО студента/участников группы).
 2. Таблицу проблем и их критичности (задание 1.1) с обоснованием (1.2).
 3. Таблицу с вариантами корректирующих действий (задание 2.1).
 4. Обоснование выбора оптимального варианта (задание 3.1).
 5. Детальный план корректировки (задание 3.2).
 6. Заполненный запрос на изменение (задание 4.1).
 7. Обновлённые базовые планы (задание 5.1) – в краткой форме.
 8. Текст информационного сообщения для команды (6.1) и описание презентации для заказчика (6.2).
 9. Вывод по работе: что нового узнали, какие навыки приобрели, как корректировка связана с управлением изменениями.
-

Контрольные вопросы (для самопроверки)

1. В каких случаях возникает необходимость корректировки проекта?
2. Опишите процесс управления изменениями (основные шаги).
3. Что такое запрос на изменение (Change Request) и какую информацию он содержит?
4. Какие виды корректирующих действий существуют? Приведите примеры.
5. Чем отличаются корректирующие, предупреждающие действия и исправление дефектов?
6. Почему важно обновлять базовые планы после утверждения изменений?
7. Как следует коммуницировать изменения команде и заказчику?

Практическое занятие №15. Разработка плана передачи проекта в эксплуатацию и миграции данных.

Цель работы

Научиться разрабатывать комплексный план передачи ИТ-проекта в эксплуатацию, включая миграцию данных, подготовку пользователей, проведение приемочных испытаний и оформление завершающей документации.

Краткие теоретические сведения

1. Этап передачи в эксплуатацию

Передача проекта в эксплуатацию – это завершающая фаза жизненного цикла разработки, в ходе которой созданный программный продукт передается заказчику (эксплуатирующей организации) для постоянного использования. Основные задачи этапа:

- Развертывание системы в продуктивной среде;
- Миграция данных из старых систем (если требуется);
- Обучение пользователей и подготовка документации;
- Проведение приемочных испытаний;
- Подписание акта о вводе в эксплуатацию;
- Планирование поддержки и сопровождения.

2. План передачи в эксплуатацию

Документ, описывающий все мероприятия, сроки, ответственных и ресурсы, необходимые для успешного перехода системы в продуктивную эксплуатацию. Обычно включает разделы:

- Цели и критерии готовности;
- Состав работ и график;
- План миграции данных;
- План обучения пользователей;
- План развертывания (деплой);
- План приемочных испытаний;
- План отката (rollback) на случай неудачи;
- Риски и мероприятия по их снижению.

3. Миграция данных

Миграция данных – процесс переноса информации из одной или нескольких исходных систем в целевую систему. Включает этапы:

1. **Анализ исходных данных** – оценка объема, качества, структуры.
2. **Очистка и преобразование** – исправление ошибок, приведение к формату целевой системы.
3. **Загрузка** – физический перенос данных.
4. **Верификация** – проверка полноты и корректности переноса.

4. Приемочные испытания

Приемочные испытания проводятся для подтверждения того, что система соответствует требованиям и готова к эксплуатации. Обычно включают:

- Проверку функциональных требований (по тест-кейсам);
- Проверку нефункциональных требований (производительность, безопасность);
- Демонстрацию ключевых сценариев заказчику;
- Оформление протокола испытаний и акта приемки.

5. Документация для передачи

- Руководство пользователя;
- Руководство администратора;
- Инструкции по установке и настройке;
- Описание API (если есть);
- Акт приема-передачи;
- План поддержки и сопровождения.

Ход работы

Работа выполняется индивидуально или в группах по 2–3 человека. Используется проект «CollegeHelper» – мобильное приложение для студентов колледжа. Предполагается, что разработка завершена, и необходимо спланировать передачу системы в эксплуатацию.

Исходные данные по проекту

- **Продукт:** мобильное приложение (iOS/Android) и серверная часть (API, админ-панель).
- **Пользователи:** студенты (≈ 2000 чел.), преподаватели (≈ 150 чел.), администраторы (5 чел. из учебной части).
- **Существующие данные:** расписание занятий хранится в Excel-файлах в учебной части; списки студентов и групп – в информационной системе колледжа (старая БД Access); новости публикуются на сайте вручную.
- **Необходимо:** развернуть сервер, настроить интеграцию с существующей базой студентов (или загрузить данные из Excel), обеспечить доступ пользователей, обучить преподавателей и администраторов.

Этап 1. Определение состава работ по передаче в эксплуатацию

Задание 1.1. Составьте перечень всех основных работ, которые необходимо выполнить для передачи проекта «CollegeHelper» в эксплуатацию. Сгруппируйте работы по категориям:

- Подготовка инфраструктуры (серверы, ПО, сеть);
- Развертывание (установка и настройка);
- Миграция данных;
- Обучение пользователей;
- Приемочные испытания;
- Документирование;
- Завершающие мероприятия (подписание актов, передача прав).

Используйте таблицу:

Категория	Работы (подробно)	Примерная длительность (дни)
Подготовка инфраструктуры		
Развертывание		
Миграция данных		
Обучение пользователей		
Приемочные испытания		
Документирование		
Завершение		

Задание 1.2. Определите критический путь для этапа передачи (какие работы должны быть выполнены строго последовательно, а какие могут идти параллельно). Постройте упрощённую диаграмму Ганта (в виде таблицы или схемы) для этого этапа.

Этап 2. Разработка плана миграции данных

Задание 2.1. Изучите описание исходных данных. Определите источники данных для каждого модуля CollegeHelper:

Модуль	Источник данных	Формат	Необходимые преобразования
Расписание	Excel-файлы учебной части	.xlsx	...
Студенты и группы	Старая БД Access	.mdb	...
Новости	Сайт (публикуются вручную)	HTML/текст	...
Пользователи (логины, пароли)	Нужно создать заново или интегрировать с доменом колледжа?	—	...

Заполните таблицу, указав для каждого источника предполагаемые преобразования (очистка, конвертация, обогащение).

Задание 2.2. Составьте детальный план миграции данных, включая:

- Этапы (анализ, очистка, пробный перенос, основной перенос, верификация).
- Инструменты и методы (скрипты, ETL-средства, ручная проверка).
- Ответственные (роли).
- Сроки.
- Критерии успешной миграции (например, количество перенесённых записей, отсутствие ошибок).

Оформите в виде таблицы или текстового плана.

Задание 2.3. Какие риски могут возникнуть при миграции данных? Заполните таблицу рисков (не менее 3) с оценкой вероятности и влияния, а также мерами предотвращения.

Риск	Вероятность (выс/ср/низ)	Влияние (выс/ср/низ)	Меры предотвращения
...			

Этап 3. Планирование обучения пользователей и подготовки документации

Задание 3.1. Для каждой группы пользователей (студенты, преподаватели, администраторы) определите:

- Какие навыки им необходимы для работы с системой.
- Какой формат обучения наиболее эффективен (инструкции, видео, вебинар, очный тренинг).
- Кто будет проводить обучение.
- Продолжительность обучения.
- Необходимые материалы.

Заполните таблицу:

Группа пользователей	Навыки	Формат обучения	Ответственный	Длительность	Материалы
Студенты					
Преподаватели					
Администраторы					

Задание 3.2. Перечислите, какую документацию необходимо подготовить до передачи системы в эксплуатацию. Для каждого документа укажите целевую аудиторию и краткое содержание.

Документ	Аудитория	Содержание
Руководство пользователя (студенты)	Студенты	...
Руководство пользователя (преподаватели)	Преподаватели	...
Руководство администратора	Администраторы	...
...		

Этап 4. Планирование приемочных испытаний

Задание 4.1. Разработайте программу приемочных испытаний для проекта «CollegeHelper». Определите:

- Цель испытаний.
- Состав приемочной комиссии (роли: заказчик, руководитель проекта, технический специалист, представитель пользователей).
- Перечень ключевых сценариев для проверки (не менее 5) с ожидаемыми результатами.
- Критерии успешного завершения испытаний (например, все критические сценарии выполнены, количество ошибок не более ...).
- Процедура оформления результатов (протокол, акт).

Оформите в виде структурированного плана.

Задание 4.2. Составьте чек-лист готовности системы к эксплуатации (Readiness Checklist). Включите пункты по следующим категориям:

- Техническая готовность (серверы, ПО, доступность).
- Данные (миграция завершена, проверена).
- Документация (все документы готовы).
- Обучение (проведено, материалы розданы).
- Организационные вопросы (назначены ответственные за поддержку, решены вопросы с подпиской и т.д.).

Этап 5. Разработка итогового плана передачи в эксплуатацию и анализ рисков

Задание 5.1. Сведите все полученные материалы в единый сводный план передачи в эксплуатацию. Используйте шаблон (см. Приложение) или создайте свой. План должен включать:

- Общие сведения (название проекта, дата, автор).
- Цели и критерии готовности.
- Состав работ с графиком (можно ссылаться на диаграмму Ганта).
- План миграции данных.
- План обучения.
- План приемочных испытаний.
- План отката (rollback) – что делаем, если что-то пойдёт не так.
- Реестр рисков этапа передачи.

Задание 5.2. Проанализируйте, какие риски могут возникнуть непосредственно во время передачи (например, сбой при миграции, недоступность сервера, неготовность

пользователей). Для трёх наиболее значимых рисков опишите конкретные действия по реагированию (план Б).

Этап 6. Оформление отчёта

Отчёт по практической работе должен содержать:

1. Титульный лист (тема, ФИО студента/участников группы).
2. Результаты этапа 1 (перечень работ и диаграмма).
3. Результаты этапа 2 (план миграции данных, риски).
4. Результаты этапа 3 (таблицы по обучению и документации).
5. Результаты этапа 4 (программа испытаний и чек-лист).
6. Результаты этапа 5 (сводный план передачи).
7. Вывод по работе: что нового узнали, какие навыки приобрели, с какими

трудностями столкнулись.

Контрольные вопросы (для самопроверки)

1. Какие основные задачи решаются на этапе передачи проекта в эксплуатацию?
2. Из каких этапов состоит процесс миграции данных?
3. Какие документы необходимо подготовить для передачи системы пользователям?
4. Кто входит в приёмочную комиссию и какова её роль?
5. Что такое план отката (rollback) и зачем он нужен?
6. Какие критерии готовности системы к эксплуатации вы знаете?
7. Почему важно проводить обучение пользователей до начала эксплуатации?

Практическое занятие №16. Контроль качества продукта.

Цель работы

Научиться планировать и выполнять мероприятия по контролю качества программного продукта: разрабатывать тест-кейсы, проводить тестирование, оформлять отчёты о дефектах, анализировать метрики качества и предлагать улучшения.

Краткие теоретические сведения

1. Понятие качества и контроль качества

Качество программного продукта – степень соответствия продукта установленным требованиям и ожиданиям пользователей.

Контроль качества (Quality Control, QC) – процесс проверки и тестирования продукта с целью выявления дефектов. Включает:

- анализ требований;
- разработку тестовой документации;
- выполнение тестов;
- регистрацию дефектов;
- анализ результатов.

Обеспечение качества (Quality Assurance, QA) – процесс создания системы, гарантирующей, что качество будет достигнуто (процессы, стандарты, аудиты). QC – это проверка продукта, QA – проверка процессов.

2. Виды тестирования

Классификация	Виды
По уровню	Модульное, интеграционное, системное, приёмочное
По знанию системы	Black box (без знания кода), White box (со знанием кода)
По степени автоматизации	Ручное, автоматизированное
По объекту тестирования	Функциональное, нефункциональное (нагрузочное, безопасности, юзабилити)
По позитивности	Позитивное (проверка ожидаемого поведения), негативное (проверка реакции на ошибки)

3. Техники тест-дизайна

- **Эквивалентное разбиение** – разделение входных данных на классы, поведение в пределах одного класса одинаково. Достаточно проверить одно значение из класса.
- **Анализ граничных значений** – проверка на границах допустимых диапазонов, где чаще всего возникают ошибки.
- **Попарное тестирование** – для проверки комбинаций параметров.
- **Таблица принятия решений** – для сложной бизнес-логики.

4. Тестовая документация

Тест-кейс (тестовый случай) – артефакт, описывающий:

- ID, название;
- предусловия;
- шаги;
- ожидаемый результат;
- пост-условия;
- приоритет.

Чек-лист – список проверок (без детальных шагов), используется для быстрой проверки.

Отчёт о дефекте (баг-репорт) – документ, описывающий найденную ошибку:

- ID, краткое описание;
- окружение;
- шаги воспроизведения;
- фактический и ожидаемый результат;
- серьёзность (Severity) и приоритет (Priority);
- приложения (скриншоты, логи).

5. Метрики качества

- Количество найденных дефектов (всего, по критичности, на модуль).
- Процент пройденных тестов (passed / total).
- Плотность дефектов (дефекты на 1000 строк кода или на функцию).
- Время исправления дефекта.

Ход работы

Работа выполняется индивидуально или в группах по 2 человека. Используется проект «CollegeHelper». В качестве объекта тестирования выберите одну из пользовательских историй (например, «Просмотр расписания на сегодня»).

Исходные данные

Пользовательская история US1 (из бэклога):

As a student, I want to view my daily schedule so that I know which classes to attend.

Критерии приемки (Acceptance Criteria):

1. При входе в приложение на главном экране отображается расписание на текущий день.
2. Если на сегодня нет занятий, показывается сообщение «Сегодня занятий нет».
3. Каждая запись расписания содержит: название предмета, время начала и конца, аудиторию, имя преподавателя.
4. При нажатии на запись открывается детальная информация (можно добавить описание).
5. Расписание обновляется автоматически при изменении данных на сервере (при наличии интернета).

Нефункциональные требования:

- Время загрузки расписания не более 2 секунд.
- Приложение должно корректно работать на устройствах с iOS 15+ и Android 10+.

Этап 1. Анализ требований и составление тест-кейсов

Задание 1.1. Изучите пользовательскую историю US1 и её критерии приемки.

Определите тестовые сценарии, используя техники эквивалентного разбиения и анализа граничных значений. Заполните таблицу тест-кейсов.

Примерные направления для тестов:

- Проверка отображения расписания при наличии занятий (разное количество занятий).
- Проверка при отсутствии занятий.
- Проверка корректности данных (название, время, аудитория).
- Проверка поведения при отсутствии интернета.
- Проверка обновления данных при изменении на сервере.
- Негативные проверки (некорректные данные с сервера, ошибки соединения).

Таблица тест-кейсов (минимум 5 тест-кейсов):

ID	Название тест-кейса	Предусловия	Шаги	Ожидаемый результат	Приоритет (High/Mid/Low)
ТС-001	Просмотр расписания при наличии занятий	Пользователь авторизован, на сервере есть расписание на сегодня	1. Открыть приложение 2. Перейти на главный экран	Отображается список занятий на сегодня с корректными данными	High
ТС-002	...	...	...	...	...

Задание 1.2. Для одного из тест-кейсов примените технику **граничных значений**.

Например, если расписание может содержать от 0 до 10 занятий (максимум), какие значения нужно проверить? Опишите в комментариях.

Этап 2. Проведение тестирования (симуляция)

Задание 2.1. Представьте, что вы выполнили тест-кейсы из этапа 1. В процессе тестирования вы обнаружили несколько дефектов. Ниже приведено описание ситуации (можно считать, что вы нашли эти дефекты).

Выявленные проблемы (дефекты):

1. При отсутствии интернета приложение показывает сообщение «Ошибка загрузки», но не предлагает повторить или показать кэшированную версию.
2. В одной из записей расписания время окончания отображается как 25:00 (некорректный формат).
3. При наличии 10 занятий список прокручивается, но после последнего элемента остаётся пустое пространство, и нижнее меню перекрывает последнюю запись.
4. При быстром переключении между днями (если реализована навигация) приложение иногда вылетает (crash) – воспроизводится нестабильно.

Добавьте к этому списку ещё **2 дефекта**, которые вы могли бы предположить (придумайте сами).

Задание 2.2. Заполните отчёты о дефектах для каждого из найденных (минимум 3 дефекта). Используйте шаблон из приложения.

ID	Краткое описание	Шаги воспроизведения	Фактический результат	Ожидаемый результат	Серьёзность (Critical/High/Mid/Low)	Приоритет
BUG-001	...	...	...	...	...	...

Серьёзность (Severity):

- Critical – система неработоспособна, потеря данных.
- High – важная функция не работает, нет обходного пути.
- Mid – функция работает с ошибками, но есть обходной путь.
- Low – косметические дефекты, опечатки.

Приоритет (Priority): определяется срочностью исправления (High – срочно, Mid – в текущем спринте, Low – в бэклог).

Этап 3. Анализ результатов тестирования

Задание 3.1. Подсчитайте метрики по результатам тестирования (условно, на основе ваших тест-кейсов и найденных дефектов):

- Всего тест-кейсов: _____ (из этапа 1)
- Тест-кейсов пройдено успешно: _____ (сколько, по вашему мнению, прошло бы без дефектов)
- Тест-кейсов с ошибками: _____ (на которых найдены дефекты)
- Всего найдено дефектов: _____
- Критических/высоких дефектов: _____

Задание 3.2. Постройте круговую диаграмму распределения дефектов по серьёзности (Critical, High, Mid, Low). Можно выполнить в Excel или от руки и вставить скриншот.

Задание 3.3. Сделайте вывод о текущем качестве продукта. Готов ли он к релизу? Какие риски остаются?

Этап 4. Предложения по улучшению качества

Задание 4.1. На основе выявленных дефектов предложите мероприятия по улучшению качества продукта и процесса разработки. Заполните таблицу:

Мероприятие	Цель	Ответственный (роль)	Срок
1. Добавить обработку ошибок сети и кэширование	Улучшить стабильность при отсутствии интернета	Разработчик	
2. ...	...	...	

Задание 4.2. Какие дополнительные виды тестирования вы бы рекомендовали провести перед релизом (например, нагрузочное, юзабилити, безопасности)? Почему?

Этап 5. Оформление отчёта

Отчёт по практической работе должен содержать:

1. Титульный лист (тема, ФИО студента/участников группы).
2. Таблицу тест-кейсов (задание 1.1).
3. Пояснения по применению граничных значений (задание 1.2).
4. Таблицу отчётов о дефектах (задание 2.2).
5. Метрики и диаграмму (задание 3.1–3.2) и вывод о качестве (3.3).
6. Предложения по улучшению (задание 4.1–4.2).
7. Общий вывод по работе: что нового узнали, какие навыки приобрели, с какими трудностями столкнулись.

Контрольные вопросы (для самопроверки)

1. В чём разница между контролем качества (QC) и обеспечением качества (QA)?
2. Какие виды тестирования вы знаете? Для чего используется каждый?
3. Что такое тест-кейс и из каких элементов он состоит?
4. Какие техники тест-дизайна помогают сократить количество тестов без потери качества?
5. Как правильно оформить отчёт о дефекте?
6. Какие метрики качества можно использовать для оценки готовности продукта?
7. Почему важно проводить тестирование на разных уровнях (модульное, интеграционное, системное)?

Тема 3.2. Завершение проекта

Практическое занятие №17. Подготовка итоговой документации.

Цель работы

Научиться формировать комплект итоговой документации по завершении ИТ-проекта: составлять отчёт о результатах проекта, готовить документацию для передачи заказчику (пользовательскую и техническую), оформлять акты приёмки и архив проекта.

Краткие теоретические сведения

1. Зачем нужна итоговая документация?

Итоговая документация завершает жизненный цикл проекта и выполняет следующие функции:

- Фиксирует фактические результаты и достигнутые цели;
- Служит основанием для подписания акта приёмки и закрытия контракта;
- Обеспечивает заказчика и пользователей всей необходимой информацией для эксплуатации продукта;
- Сохраняет знания и опыт для будущих проектов (извлечённые уроки);
- Является архивом на случай аудита или судебных разбирательств.

2. Состав итоговой документации

Типовой перечень документов, подготавливаемых на завершающем этапе:

Документ	Назначение	Аудитория
Итоговый отчёт о проекте	Сводка о выполнении целей, сроков, бюджета, качества; анализ отклонений и извлечённые уроки	Руководство, заказчик, команда
Акт приёмки-передачи работ	Юридический документ, подтверждающий завершение работ и передачу результата заказчику	Заказчик, исполнитель
Руководство пользователя	Инструкция по работе с системой для конечных пользователей	Пользователи
Руководство администратора	Описание установки, настройки, администрирования системы	Администраторы, техподдержка
Техническая документация	Описание архитектуры, базы данных, API, развёртывания	Разработчики (для поддержки)
План поддержки и сопровождения	Условия дальнейшей поддержки: сроки, каналы связи, ответственные, SLA	Заказчик, техподдержка
Архив проекта	Собрание всех артефактов: планы, отчёты, исходный код, документация	Руководитель, архив

3. Структура итогового отчёта о проекте

1. **Титульный лист** (название проекта, заказчик, исполнитель, дата).
2. **Краткое резюме** (1–2 абзаца об итогах).
3. **Достигнутые результаты** – что создано, функциональность, соответствие ТЗ.
4. **Анализ выполнения плановых показателей:**
 - Сроки (факт vs план, причины отклонений).
 - Бюджет (факт vs план, причины перерасхода/экономии).
 - Качество (количество дефектов, удовлетворённость заказчика).
5. **Анализ рисков** – какие риски реализовались, как сработали меры.
6. **Извлечённые уроки** – что удалось, что можно улучшить в будущих проектах.
7. **Заключение и рекомендации.**
8. **Приложения** (акты, отзывы, метрики).

4. Акт приёмки-передачи работ

Документ обычно содержит:

- Название и реквизиты сторон;
- Перечень выполненных работ/результатов;
- Ссылку на договор и ТЗ;
- Заявление о соответствии результата требованиям;
- Отсутствие претензий (или перечень недоделок сроками устранения);
- Подписи и печати.

5. Пользовательская документация

Должна быть:

- Понятной (минимум технического жаргона);
- Структурированной (оглавление, разделы по функциям);
- Содержать пошаговые инструкции с иллюстрациями (скриншотами);
- Включать ответы на частые вопросы (FAQ) и контакты поддержки.

6. Архив проекта

Формируется в электронном виде (папка с файлами) и включает:

- Управленческую документацию (устав, планы, отчёты);
- Техническую документацию (ТЗ, архитектура, инструкции);
- Исходный код (с тегами версий);
- Данные тестирования (тест-кейсы, отчёты о дефектах);
- Материалы приёмки (акты, протоколы).

Ход работы

Работа выполняется индивидуально или в группах по 2–3 человека. Используется сквозной проект «CollegeHelper». Предполагается, что проект успешно завершён, все работы выполнены, продукт протестирован и готов к передаче.

Исходные данные (обобщение результатов проекта):

- **Продолжительность проекта:** плановая – 16 недель, фактическая – 18 недель (отставание 2 недели).
- **Бюджет:** плановый – 1 500 000 руб., фактический – 1 620 000 руб. (перерасход 8%).
- **Результат:** разработано мобильное приложение для iOS/Android, реализованы все функции MVP (авторизация, расписание, новости, заявки в техподдержку, просмотр оценок). Дополнительно реализованы push-уведомления (по желанию заказчика, за счёт резерва).
- **Качество:** в процессе тестирования найдено 47 дефектов, из них 2 критических (исправлены), остальные закрыты. Удовлетворённость заказчика высокая.
- **Риски:** реализовался риск задержки интеграции с сервером колледжа (из-за бюрократии), сработал план Б – временное хранение данных на собственном сервере с последующей синхронизацией.
- **Извлечённые уроки:** необходимо закладывать больше времени на согласования с внешними службами; улучшить процесс оценки трудоёмкости.

Этап 1. Подготовка итогового отчёта о проекте

Задание 1.1. Используя приведённые исходные данные, составьте итоговый отчёт о проекте «CollegeHelper». Придерживайтесь структуры, описанной в теории. Недостающие данные (например, имена, должности, конкретные цифры) можно придумать.

Отчёт должен содержать все разделы:

- Титульный лист;
- Резюме;
- Достигнутые результаты;
- Анализ сроков, бюджета, качества;
- Анализ рисков;
- Извлечённые уроки;
- Заключение и рекомендации;
- Приложения (перечислить, что прилагается).

Объём – 2–3 страницы (можно сократить, но логика должна прослеживаться).

Задание 1.2. Оформите в отчёте раздел «Извлечённые уроки» в виде таблицы:

Что удалось	Что можно улучшить	Рекомендации для будущих проектов
...	...	...

Этап 2. Разработка акта приёмки-передачи

Задание 2.1. Составьте проект акта приёмки-передачи работ по проекту «CollegeHelper». Используйте шаблон из приложения. Заполните все поля, укажите перечень переданных результатов (приложение, документация, исходный код и т.д.). Предусмотрите пункт о том, что заказчик претензий не имеет (или укажите гарантийные обязательства).

Этап 3. Создание фрагмента пользовательской документации

Задание 3.1. Выберите одну из функций приложения CollegeHelper (например, «Просмотр расписания» или «Подача заявки в техподдержку») и напишите для неё раздел руководства пользователя. Раздел должен включать:

- Название функции;
- Краткое описание (зачем нужна);
- Пошаговую инструкцию с указанием, где находятся нужные элементы интерфейса;
- Скриншоты (можно схематичные или текстовые описания экранов, если нет реальных);
- Возможные проблемы и способы их решения (FAQ).

Оформите в виде текста с заголовками и списками.

Этап 4. Формирование состава архива проекта

Задание 4.1. Составьте перечень папок и файлов, которые должны войти в итоговый архив проекта «CollegeHelper». Опишите структуру архива (в виде дерева или списка с пояснениями). Например:

```
/CollegeHelper_archive
  /01_Management
    - Устав проекта.pdf
    - Планы (план управления, риски, бюджет).xlsx
    - Отчёты о статусе (папка с еженедельными отчётами)
    - Итоговый отчёт.pdf
  /02_Requirements
    - Техническое задание.pdf
```

- Бэклог продукта.xlsx
- User Stories.docx
- /03_Design
 - Макеты (Figma-ссылка или файлы)
 - UI-kit
- /04_Development
 - Исходный код (ссылка на репозиторий, тег v1.0)
 - Инструкция по сборке.md
- /05_Testing
 - Тест-кейсы.xlsx
 - Отчёты о дефектах.xlsx
 - Протоколы тестирования.pdf
- /06_Documentation
 - Руководство пользователя.pdf
 - Руководство администратора.pdf
 - Описание API.md
- /07_Closure
 - Акт приёмки-передачи.pdf
 - Отзыв заказчика.pdf
 - Извлечённые уроки.pptx

Задание 4.2. Какие из перечисленных документов были созданы вами в ходе предыдущих практических занятий? Укажите соответствие (например, «Техническое задание – занятие №5», «Бюджет – занятие №7»). Это поможет понять, что архив формируется из реально выполненных работ.

Этап 5. Подготовка презентации для заказчика (кратко)

Задание 5.1. Подготовьте структуру презентации (3–5 слайдов) для итоговой встречи с заказчиком, на которой вы представляете результаты проекта. Укажите, что будет на каждом слайде.

Пример структуры:

- Слайд 1: Титульный (название, команда, дата).
- Слайд 2: Что сделано (основные функции, достигнутые цели).
- Слайд 3: Ключевые цифры (сроки, бюджет, качество).
- Слайд 4: Демонстрация (скриншоты или видео работы).
- Слайд 5: Следующие шаги (поддержка, планы развития).

Этап 6. Оформление отчёта

Итоговый отчёт по практической работе должен содержать:

1. Титульный лист (тема, ФИО участников группы).
 2. Итоговый отчёт о проекте (задание 1.1) с таблицей извлечённых уроков (1.2).
 3. Проект акта приёмки-передачи (задание 2.1).
 4. Фрагмент пользовательской документации (задание 3.1).
 5. Структуру архива проекта (задание 4.1) и соответствие документации занятиям (4.2).
 6. Структуру презентации для заказчика (задание 5.1).
 7. Вывод по работе: что нового узнали, какие навыки приобрели, как итоговая документация помогает завершить проект.
-

Контрольные вопросы (для самопроверки)

1. Какие документы обязательно должны быть в итоговом комплекте документации?
2. Для чего нужен итоговый отчёт о проекте? Какие разделы он содержит?
3. Чем акт приёмки отличается от итогового отчёта?
4. Кто является целевой аудиторией руководства пользователя? Каким требованиям оно должно соответствовать?
5. Что такое архив проекта и зачем его сохранять?
6. Какие извлечённые уроки вы можете назвать из собственного опыта (или из примера)?
7. Почему важно фиксировать фактические показатели (сроки, бюджет) в итоговых документах?

Дополнительное задание

1. **Разработка полного руководства пользователя.** Для выбранной роли (студент, преподаватель, администратор) создайте полное руководство пользователя по всему приложению CollegeHelper (не менее 5–7 страниц) с подробными инструкциями и скриншотами (можно использовать макеты или рисунки).
2. **Создание руководства администратора.** Опишите процесс установки серверной части, настройки базы данных, развёртывания обновлений.
3. **Подготовка финальной презентации.** Создайте презентацию (до 10 слайдов) для защиты проекта перед комиссией, включив в неё все ключевые аспекты: цели, команда, процесс, результаты, демонстрация, уроки.
4. **Рефлексия по курсу.** Напишите эссе (1–2 страницы) о том, какие знания и навыки, полученные в ходе выполнения практических работ, были наиболее полезными и почему.

Приложение. Шаблон акта приёмки-передачи работ АКТ ПРИЁМКИ-ПЕРЕДАЧИ РАБОТ ПО ДОГОВОРУ

№ ____ от «_» _____ 20 г.

г. _____ «_» _____ 20 г.

Заказчик: _____ в лице
_____, действующего на основании
_____, с одной стороны, и

Исполнитель: _____ в лице
_____, действующего на основании
_____, с другой стороны, составили настоящий акт о нижеследующем:

1. Исполнитель выполнил, а Заказчик принял работы по разработке программного продукта «CollegeHelper» (далее – Продукт) в соответствии с Техническим заданием (Приложение №1 к Договору).
2. В результате выполнения работ создан следующий Продукт:
 - Мобильное приложение для платформ iOS и Android;
 - Серверная часть (API и админ-панель);
 - Комплект документации (руководство пользователя, руководство администратора, техническая документация);

- Исходный код программы.
- 3. Выполненные работы соответствуют требованиям Технического задания, приняты без замечаний (или: с замечаниями, которые Исполнитель обязуется устранить в срок до _____).
- 4. Стороны претензий друг к другу не имеют (или: претензии перечислены в приложении №2).
- 5. Настоящий акт составлен в двух экземплярах, имеющих равную юридическую силу, по одному для каждой из сторон.

ПОДПИСИ СТОРОН:

Заказчик: _____ //

Исполнитель: _____ //

Практическое занятие №18. Презентация проекта.

Цель работы

Представить результаты выполнения проекта «CollegeHelper» перед аудиторией (заказчиком, преподавателем, другими студентами), продемонстрировать навыки публичной защиты проекта, ответить на вопросы, получить обратную связь.

Задачи

- Подготовить презентацию, отражающую все ключевые аспекты проекта: цели, команда, процесс, результаты, управление, уроки.
- Выступить с защитой проекта (5–7 минут).
- Ответить на вопросы аудитории.
- Провести самооценку и взаимооценку выступлений.

Ход работы

Этап 1. Подготовка презентации (выполняется до занятия или в начале)

Задание 1.1. Подготовьте презентацию проекта «CollegeHelper» в формате PowerPoint, Google Slides или PDF. Презентация должна содержать следующие слайды (рекомендуемая структура):

№	Название слайда	Содержание
1	Титульный слайд	Название проекта, состав команды (ФИО), учебное заведение, дата
2	Цели и задачи	Для чего создавалось приложение, какие проблемы решает, целевая аудитория
3	Ключевая функциональность	Перечень основных возможностей (можно с иллюстрациями, скриншотами макетов)
4	Управление проектом	Выбранная методология (Scrum), роли в команде, основные этапы, длительность, инструменты управления
5	Бюджет и сроки	Плановые и фактические показатели, анализ отклонений
6	Риски и управление ими	Основные риски, как ими управляли, реализовавшиеся риски и реакция

№	Название слайда	Содержание
7	Качество и тестирование	Подход к тестированию, количество дефектов, метрики качества
8	Результаты и достижения	Что создано, что получилось особенно хорошо, отзывы (можно вымышленные)
9	Извлечённые уроки	Что узнали в процессе, что бы сделали иначе, рекомендации будущим командам
10	Демонстрация продукта	Показ работы приложения (видео, скриншоты, живая демо)
11	Заключение и перспективы	Планы по доработке, поддержка, внедрение
12	Заключительный слайд	Приглашение к вопросам, контакты

Требования к оформлению:

- Единый стиль, читаемый шрифт, не более 7 строк текста на слайде.
- Использование иллюстраций, схем, скриншотов.
- Соблюдение регламента – презентация рассчитана на 5–7 минут выступления.

Задание 1.2. Распределите роли для выступления: кто какой слайд представляет, или выступает один докладчик (по желанию группы). Подготовьте текст выступления (можно тезисно).

Этап 2. Защита проектов (50–60 минут)

Задание 2.1. Выступления групп. Каждая группа представляет свой проект.

Регламент:

- Выступление – 5–7 минут.
- Вопросы от аудитории – 2–3 минуты.

Задание 2.2. Во время выступления остальные студенты и преподаватель заполняют **оценочные листы** (см. Приложение 1). Оценка производится по следующим критериям:

Критерий	Описание	Баллы
Полнота раскрытия темы	Освещены все ключевые аспекты проекта	0–5
Логичность и структурированность	Выступление построено логично, слайды последовательны	0–5
Качество презентации	Дизайн, иллюстрации, читаемость, отсутствие ошибок	0–5
Убедительность и аргументированность	Докладчик уверен, аргументы обоснованы	0–5
Ответы на вопросы	Точность, полнота, умение держаться	0–5
Соблюдение регламента	Уложились в 5–7 минут	0–3
Командная работа	Равномерное участие всех членов группы (если заметно)	0–2
ИТОГО		30

Этап 3. Обсуждение и обратная связь (10 минут)

Задание 3.1. После всех выступлений проведите краткое обсуждение:

- Какие проекты показались наиболее интересными? Почему?
- Какие ошибки в презентациях вы заметили?
- Что можно улучшить в будущих выступлениях?

Задание 3.2. Каждая группа проводит **самоанализ** своего выступления (устно или письменно в отчёте):

- Что получилось хорошо?
- Что можно было сделать лучше?
- Какие трудности возникли при подготовке и защите?

Этап 4. Оформление отчёта

Отчёт по практической работе должен содержать:

1. Титульный лист (тема, ФИО участников группы).
2. Ссылку на презентацию (или скриншоты ключевых слайдов) и краткое описание выступления.
3. Самоанализ (что удалось, какие были трудности, оценка своего выступления по критериям).
4. Вывод по работе: чему научились в ходе подготовки и защиты, как навыки презентации пригодятся в профессиональной деятельности.

Контрольные вопросы

1. Какие навыки публичных выступлений вы развили в ходе подготовки?
2. Что было самым сложным при подготовке презентации?
3. Как вы оцениваете свою работу в команде?
4. Что бы вы сделали иначе, если бы пришлось готовить презентацию снова?

Тема 3.3. Постпроектный анализ

Практическое занятие №19. Анализ работы команды.

Цель работы

Научиться проводить ретроспективу команды, анализировать эффективность взаимодействия, выявлять проблемные области и разрабатывать рекомендации по улучшению командной работы в будущих проектах.

Краткие теоретические сведения

1. Зачем анализировать работу команды?

Анализ работы команды (ретроспектива) позволяет:

- оценить, насколько эффективно команда взаимодействовала;
- выявить сильные стороны, которые стоит сохранить;
- обнаружить проблемы и договориться об их решении;
- повысить мотивацию и сплочённость;
- извлечь уроки для следующих проектов.

2. Модель развития команды по Такману (Tuckman)

Брюс Такман выделил пять стадий развития команды:

1. **Forming (формирование)** – знакомство, вежливость, неопределённость.
2. **Storming (конфликты)** – столкновение мнений, борьба за лидерство, напряжённость.
3. **Norming (нормирование)** – выработка общих правил, распределение ролей, сплочение.
4. **Performing (работа)** – эффективная работа, взаимопомощь, достижение результатов.
5. **Adjourning (расставание)** – завершение проекта, подведение итогов.

Понимание стадии помогает выбрать правильные методы управления командой.

3. Методы проведения ретроспективы

Start/Stop/Continue:

- **Start** – что стоит начать делать (новые практики, инструменты).
- **Stop** – что стоит перестать делать (неэффективные действия, вредные привычки).
- **Continue** – что следует продолжать делать (успешные практики).

Sailboat (парусник):

- Визуализация: лодка (команда), ветер (что помогает двигаться), якорь (что тормозит), рифы (риски), остров (цель).

Glad/Sad/Mad:

- Участники пишут на стикерах события, вызвавшие радость, грусть или злость, затем группируют и обсуждают.

4. Командные метрики

- **Velocity** – скорость команды (количество story points за спринт) – позволяет отслеживать производительность.
- **Текущее состояние кадров** – уход членов команды (негативно влияет на проект).
- **Удовлетворённость команды** – можно измерить анонимным опросом.
- **Количество конфликтов** – частота и разрешённость.
- **Качество коммуникации** – скорость ответов, прозрачность.

5. Факторы успешной команды

- Чёткое понимание целей и ролей.
- Доверие и взаимопомощь.
- Открытая коммуникация.
- Способность к конструктивным конфликтам.
- Приверженность общим решениям.
- Регулярная ретроспектива и адаптация.

Ход работы

Работа выполняется в тех же группах, которые работали над проектом «CollegeHelper» на протяжении курса. Если по каким-то причинам группа не сформирована, можно выполнять индивидуально (анализируя свой опыт участия в учебных проектах или работу своей группы).

Этап 1. Оценка удовлетворённости работой в команде

Задание 1.1. Каждый участник группы индивидуально заполняет **опросник удовлетворённости** (см. Приложение 1). Будьте честны – это анонимно внутри группы, но результаты помогут в обсуждении.

Задание 1.2. Соберите ответы (можно устно обсудить) и заполните сводную таблицу. Выделите средние баллы по вопросам.

Вопрос	Средний балл
1. Я понимал(а) свою роль в проекте	
2. В команде были чётко распределены обязанности	
3. Коммуникация внутри команды была эффективной	
4. Конфликты решались конструктивно	
5. Я чувствовал(а) поддержку от коллег	
6. Мой вклад был оценён	
7. Я удовлетворён(а) итоговым результатом	
8. Я готов(а) работать с этой командой снова	

Этап 2. Проведение ретроспективы методом Start/Stop/Continue

Задание 2.1. В группе проведите обсуждение по методу Start/Stop/Continue. Каждый участник предлагает свои идеи, затем они обсуждаются и записываются в таблицу.

Start (что начать делать)	Stop (что перестать делать)	Continue (что продолжать делать)
1.	1.	1.
2.	2.	2.
3.	3.	3.
...	...	...

Примеры:

- Start: проводить ежедневные летучки, использовать общий чат для срочных вопросов, писать тесты до кода.
- Stop: опаздывать на встречи, менять требования без обсуждения, работать в выходные без необходимости.
- Continue: планирование спринта, code review, помощь коллегам.

Этап 3. Анализ командных метрик за весь проект

Задание 3.1. Если в процессе работы вы фиксировали какие-либо метрики (например, velocity по спринтам, количество дефектов, количество пропущенных дней), проанализируйте их динамику. Если нет – используйте условные данные, отражающие ваш опыт.

Заполните таблицу:

Метрика	Спринт 1	Спринт 2	Спринт 3	...	Вывод
Velocity (story points)					
Количество дефектов на спринт					
Количество конфликтов (субъективно)					

Метрика	Спринт 1	Спринт 2	Спринт 3	...	Вывод
Средняя удовлетворённость (по опросу)					

Задание 3.2. Сделайте вывод о динамике: росла ли эффективность, были ли спады, с чем это связано.

Этап 4. Разработка рекомендаций для будущих проектов

Задание 4.1. На основе результатов опроса, ретроспективы и анализа метрик сформулируйте **не менее трёх рекомендаций** для улучшения работы вашей команды в будущем. Рекомендации должны быть конкретными и реалистичными.

Рекомендация	Ожидаемый эффект	Кому адресована (роль)
1. ...	...	...
2. ...	...	...
3. ...	...	...

Задание 4.2. Выберите одну рекомендацию, которую вы считаете наиболее важной, и опишите план её внедрения (первые шаги, сроки, ответственные).

Этап 5. Оформление отчёта

Отчёт по практической работе должен содержать:

1. Титульный лист (тема, ФИО участников группы).
2. Результаты опроса удовлетворённости (сводная таблица и краткий комментарий).
3. Таблицу ретроспективы Start/Stop/Continue.
4. Анализ командных метрик (таблица и вывод).
5. Рекомендации для будущих проектов (таблица и план по одной рекомендации).
6. Вывод по работе: что нового узнали о своей команде, какие навыки анализа приобрели, как ретроспектива помогает улучшать командную работу.

Контрольные вопросы (для самопроверки)

1. Для чего нужна ретроспектива команды?
 2. Какие методы проведения ретроспективы вы знаете?
 3. Назовите стадии развития команды по Такману.
 4. Какие метрики можно использовать для оценки работы команды?
 5. Почему важно не только выявить проблемы, но и предложить конкретные действия?
 6. Как часто следует проводить ретроспективы в ходе проекта?
 7. Какие факторы делают команду успешной?
-

Дополнительное задание

1. **Применение другого метода.** Проведите ретроспективу вашей команды методом Sailboat или Glad/Sad/Mad (можно в сокращённом виде). Опишите результаты.
 2. **Индивидуальная рефлексия.** Напишите эссе «Мой вклад в командную работу и что я узнал о себе как о члене команды» (1 страница).
 3. **Сравнительный анализ.** Если вы работали в нескольких группах в течение обучения, сравните их работу. Какие подходы были эффективнее? Почему?
-

Приложение 1. Опросник удовлетворённости работой в команде

Оцените по шкале от 1 до 5, где:

1 – полностью не согласен, 2 – не согласен, 3 – нейтрально, 4 – согласен, 5 – полностью согласен.

№	Вопрос	Оценка
1	Я понимал(а) свою роль в проекте	
2	В команде были чётко распределены обязанности	
3	Коммуникация внутри команды была эффективной (своевременной, понятной)	
4	Конфликты решались конструктивно и быстро	
5	Я чувствовал(а) поддержку от коллег при возникновении трудностей	
6	Мой вклад в проект был оценён командой	
7	Я удовлетворён(а) итоговым результатом работы команды	
8	Я готов(а) работать с этой командой снова	

Приложение 2. Пример таблицы ретроспективы (заполненный фрагмент)

Start	Stop	Continue
Проводить ежедневные стендапы	Опаздывать на встречи	Использовать общий чат для обсуждения
Заранее согласовывать интерфейсы	Менять требования без оповещения	Делать code review
Оставлять время на рефакторинг	Работать в выходные без крайней нужды	Планировать спринт вместе

Практическое занятие №20. Защита итогового проекта.

Цель работы

Представить результаты выполнения комплексного проекта «CollegeHelper» перед аудиторией (преподаватель, приглашённые эксперты, другие студенты), продемонстрировать сформированные компетенции в области управления ИТ-проектами, защитить принятые проектные решения, ответить на вопросы и получить итоговую оценку.

Краткие теоретические сведения

1. Защита проекта как завершающий этап

Защита проекта – это публичное мероприятие, на котором команда представляет результаты своей работы, демонстрирует достижение поставленных целей, обосновывает принятые решения и отвечает на вопросы аудитории. Защита позволяет оценить не только качество полученного продукта, но и умение команды презентовать результаты, аргументировать выбор, работать с возражениями.

2. Структура успешной защиты

Типовая структура выступления (7–10 минут):

1. **Введение** – название проекта, состав команды, проблема, которую решает проект.
2. **Цели и задачи** – что планировали достичь, ключевые требования.
3. **Процесс работы** – выбранная методология, этапы, основные вехи, инструменты управления.
4. **Демонстрация продукта** – показ ключевой функциональности (скриншоты, видео, живая демо).
5. **Управление проектом** – сроки, бюджет, риски, качество (сравнение план/факт).
6. **Результаты и достижения** – что получилось, отзывы (если есть), метрики.
7. **Извлечённые уроки** – что узнали в процессе, рекомендации будущим командам.
8. **Заключение** – перспективы развития, ответы на вопросы.

3. Критерии оценки защиты

Обычно оцениваются:

- **Полнота и качество презентации** – охват всех ключевых аспектов, логика, оформление.
 - **Убедительность и аргументация** – обоснованность решений, уверенность докладчика.
 - **Качество продукта (демо)** – работоспособность, соответствие заявленным функциям.
 - **Ответы на вопросы** – точность, полнота, способность отстаивать свою точку зрения.
 - **Командная работа** – равномерность участия, синхронность.
 - **Соблюдение регламента.**
- ## 4. Подготовка к защите
- Распределить роли: кто за какой слайд отвечает, кто отвечает на вопросы по какой области.
 - Провести репетицию, уложиться в регламент.
 - Подготовить демо-сценарий: заранее проверить, что всё работает, иметь запасной вариант (видео).
 - Продумать возможные вопросы и подготовить ответы.

Ход работы

Работа выполняется в группах, сформированных для выполнения проекта «CollegeHelper». Занятие проводится в формате мини-конференции.

Этап 1. Финальная доработка презентации (15 минут)

Задание 1.1. Если презентация не была полностью готова заранее, используйте первые 15 минут для финальной настройки: проверьте слайды, исправьте ошибки, убедитесь, что демо-материалы работают. Синхронизируйте выступление между членами команды.

Этап 2. Защита проектов (60 минут)

Задание 2.1. Каждая группа по очереди представляет свой проект. Регламент:

- Выступление – до 7 минут.

- Вопросы от аудитории и преподавателя – **до 3 минут**.

Во время выступления остальные студенты и преподаватель заполняют **оценочные листы** (Приложение 1). Оценка производится по критериям.

Задание 2.2. Будьте внимательны к выступающим: фиксируйте сильные стороны и замечания – это пригодится для итогового обсуждения.

Этап 3. Подведение итогов и обратная связь (15 минут)

Задание 3.1. После всех выступлений проведите краткое обсуждение:

- Какие проекты запомнились больше всего? Почему?
- Какие приёмы презентации были наиболее удачными?
- Какие ошибки в защитах вы заметили?

Задание 3.2. Каждая группа проводит **самоанализ** своего выступления (можно письменно или устно в группе):

- Что получилось лучше всего?
- Что можно было улучшить?
- Какие вопросы оказались самыми сложными?
- Как вы оцениваете своё выступление по критериям?

Этап 4. Оформление отчёта

Отчёт по практической работе должен содержать:

1. Титульный лист (тема, ФИО участников группы).
2. Ссылку на итоговую презентацию (или скриншоты ключевых слайдов).
3. Самоанализ защиты (ответы на вопросы из задания 3.2).
4. Вывод по работе: чему научились в процессе подготовки и защиты, как навыки презентации и публичных выступлений пригодятся в будущей профессиональной деятельности.

Контрольные вопросы

1. Что было самым сложным при подготовке к защите?
 2. Как вы распределяли роли в команде для выступления?
 3. Какие вопросы аудитории оказались неожиданными?
 4. Что бы вы изменили в своём выступлении, если бы пришлось выступать снова?
 5. Как опыт защиты поможет вам в дальнейшей карьере?
-

Дополнительное задание

1. Доработать презентацию с учётом замечаний и сдать финальную версию в электронном виде.
 2. Написать краткое эссе «Мои впечатления от защиты итогового проекта» (1 страница).
-

Приложение 1. Оценочный лист для защиты проекта

Группа (название/номер)	Полнота и структура (0–5)	Качество презентации (0–5)	Убедительность (0–5)	Демонстрация продукта (0–5)	Ответы на вопросы (0–5)	Соблюдение регламента (0–3)	Командная работа (0–2)	ИТОГО (30)
Группа 1								
Группа 2								
...								

Критерии оценивания:

- **Полнота и структура (0–5):** освещены все необходимые разделы, логичное построение выступления.
- **Качество презентации (0–5):** дизайн, читаемость, иллюстрации, отсутствие ошибок.
- **Убедительность (0–5):** уверенность докладчика, аргументированность решений.
- **Демонстрация продукта (0–5):** работоспособность, понятность, соответствие ТЗ.
- **Ответы на вопросы (0–5):** точность, полнота, умение держаться.
- **Соблюдение регламента (0–3):** уложились в 7 минут.
- **Командная работа (0–2):** равномерное участие всех членов.

Итоговая оценка:

27–30 баллов – «отлично»

22–26 баллов – «хорошо»

15–21 балл – «удовлетворительно»

менее 15 баллов – «неудовлетворительно»

Приложение 2. Рекомендации для выступающих

- Начинайте с приветствия и представления команды.
- Говорите чётко, не слишком быстро, смотрите на аудиторию.
- Не читайте текст со слайдов – используйте их как опору.
- Если демонстрируете приложение, покажите ключевые сценарии.
- На вопросы отвечайте по существу, если не знаете ответа – признайтесь и предложите уточнить позже.
- Следите за временем: попросите кого-то в команде показывать тайминг.

7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

7.1.1. Методическая литература:

- методические указания для практических занятий;

7.1.2. Программное обеспечение:

Офисное программное обеспечение

- **Microsoft Excel** (или **LibreOffice Calc**, **Google Sheets**) – для табличных расчётов, построения диаграмм Ганта, ведения бюджетов, анализа метрик (используется в работах №6, 7, 10, 13, 16).
- **Microsoft Word** (или **LibreOffice Writer**, **Google Docs**) – для оформления отчётов, разработки технического задания, создания итоговой документации (используется во всех работах №1–20).
- **Microsoft PowerPoint** (или **LibreOffice Impress**, **Google Slides**) – для подготовки презентаций к защите проектов (используется в работах №18, 20).

Системы управления проектами (таск-трекеры)

- **WEEEK** – российский аналог Trello, бесплатный базовый функционал. Используется для ведения бэклога, Scrum-досок, отчётности (работы №2, 3, 4, 11, 12).
- **Kaiten** – российский аналог Jira/Trello, имеет бесплатный пробный период или базовые функции. Подходит для канбан-досок и процессного управления (работы №2, 3, 4, 11, 12).
- **YouGile** – российский аналог Trello/Asana, бесплатен до 10 человек. Простой интерфейс, чат по задачам (работы №2, 3, 4, 11).
- **Projecto** – универсальная платформа (аналог Jira + Confluence), есть 30-дневный тестовый период. Используется для знакомства с ПО и управления проектами (работы №3, 4, 11, 12).
- **Strive** – система для разработки (аналог Jira), бесплатный базовый функционал. Подходит для спринтов и метрик команды (работы №3, 11).
- **ЛидерТаск (LeaderTask)** – таск-трекер с офлайн-доступом, есть бесплатная версия. Используется для планирования (работы №3, 11).
- **Планфикс** – российская разработка, бесплатен до 5 человек. Модуль «Карточки» аналогичен Trello (работы №3, 11).

Инструменты для построения диаграммы Ганта и календарного планирования

- **GanttProject** – полностью бесплатное (open source) ПО для построения диаграмм Ганта, экспорт в PDF/PNG (работы №6, 15).
- **Microsoft Project** – профессиональное ПО (платное, используется при наличии лицензии в колледже) (работа №6, опционально).
- **Excel / Google Sheets** с шаблонами – для ручного построения диаграммы Ганта (работы №6, 15).

Инструменты для визуализации и совместной работы

- **Miro** (или аналоги: **Sboard**, **Pruffme**) – онлайн-доска для мозговых штурмов, построения карт, SWOT-анализа (бесплатный тариф с ограничениями) (работы №4, 9).
- **[Draw.io](https://draw.io) (diagrams.net)** – полностью бесплатный инструмент для создания схем, диаграмм, визуализации архитектуры (работы №5, 6, 9).

- **Paint** / любой графический редактор – для простой визуализации, создания скриншотов (встроенное ПО Windows) (работы №4, 9, 12, 16).

Браузер и доступ в интернет

- **Любой современный браузер** (Google Chrome, Яндекс.Браузер, Mozilla Firefox, Microsoft Edge) – для доступа к онлайн-сервисам, электронно-библиотечным системам, системам управления проектами (используется во всех работах, где требуются онлайн-инструменты).

7.1.3. Материально-техническое обеспечение дисциплины

Мультимедийное оборудование для чтения лекций. Компьютерный класс, оснащенный компьютерами, с установленными программами из п. 7.1.2.

7.1.4 Список используемых источников

Основная литература

1. **Баланов, А. Н.** Управление IT-проектами : учебное пособие для СПО / А. Н. Баланов. — Санкт-Петербург : Лань, 2024. — 136 с. — ISBN 978-5-507-49699-0. — Текст : электронный // Электронно-библиотечная система «Лань» : [сайт]. — URL: <https://e.lanbook.com/book/428084> (дата обращения: 23.02.2026). — Режим доступа: для авторизир. пользователей.

2. **Грекул, В. И.** Методические основы управления IT-проектами : учебник для СПО / В. И. Грекул, Н. Л. Коровкина, Ю. В. Куприянов. — 2-е изд. — Саратов : Профобразование, 2024. — 467 с. — ISBN 978-5-4488-1000-8. — Текст : электронный // Электронный ресурс цифровой образовательной среды СПО PROФобразование : [сайт]. — URL: <https://profspo.ru/books/139753> (дата обращения: 23.02.2026). — Режим доступа: для авторизир. пользователей.

3. **Зуб, А. Т.** Управление проектами : учебник и практикум для СПО / А. Т. Зуб. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 397 с. — (Профессиональное образование). — ISBN 978-5-534-17511-0. — Текст : электронный // Образовательная платформа Юрайт : [сайт]. — URL: <https://urait.ru/bcode/562264> (дата обращения: 23.02.2026).

4. **Царенко, А. С.** Управление проектами : учебное пособие для СПО / А. С. Царенко. — 3-е изд., стер. — Санкт-Петербург : Лань, 2025. — 236 с. — ISBN 978-5-507-54135-5. — Текст : электронный // Электронно-библиотечная система «Лань» : [сайт]. — URL: <https://e.lanbook.com/book/505415> (дата обращения: 23.02.2026). — Режим доступа: для авторизир. пользователей.

Дополнительная литература

5. **Баланов, А. Н.** Управление IT-проектами : учебное пособие для вузов / А. Н. Баланов. — Санкт-Петербург : Лань, 2024. — 616 с. — ISBN 978-5-507-49698-3. — Текст : электронный // Электронно-библиотечная система «Лань» : [сайт]. — URL: <https://e.lanbook.com/book/428081> (дата обращения: 23.02.2026). — Режим доступа: для авторизир. пользователей.

6. **Беликова, И. П.** Управление проектами : учебное пособие (краткий курс лекций) / И. П. Беликова. — Ставрополь : Ставропольский государственный аграрный университет, 2014. — 80 с. — ISBN 2227-8397. — Текст : электронный // Электронно-библиотечная система IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/47372.html> (дата обращения: 23.02.2026). — Режим доступа: для авторизир. пользователей.

7. **Киселев, А. А.** Основы управления проектами и организация проектной деятельности : учебник для СПО / А. А. Киселев. — Москва : КноРус, 2025. — ISBN 9785406144152. — Текст : электронный.

8. **Советов, Б. Я.** Информационные технологии : учебник для СПО / Б. Я. Советов, В. В. Цехановский. — 2025. — 414 с. — Текст : электронный.

9. **Яркова, С. А.** Управление ИТ-проектами : методические материалы и указания по изучению дисциплины / С. А. Яркова. — 2024. — 50 с. — Текст : электронный.

Нормативные документы и стандарты

10. **ГОСТ Р ИСО/МЭК 12207-2010** Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств : утв. приказом Федерального агентства по техническому регулированию и метрологии от 30 ноября 2010 г. № 631-ст. — 2024. — 99 с. — Текст : электронный.

11. **ГОСТ Р ИСО/МЭК ТО 15271-2002** Информационная технология. Руководство по применению ГОСТ Р ИСО/МЭК 12207 (Процессы жизненного цикла программных средств) : принят и введен в действие постановлением Госстандарта РФ от 5 июня 2002 г. № 227-ст. — 2024. — 44 с. — Текст : электронный.

12. **ГОСТ Р 59853-2021** Информационные технологии. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Термины и определения : утв. и введен в действие приказом Федерального агентства по техническому регулированию и метрологии от 19 ноября 2021 г. № 1520-ст. — 2024. — 10 с. — Текст : электронный.

Электронные ресурсы

13. **IPR SMART** : электронно-библиотечная система. — URL: <https://www.iprbookshop.ru/> (дата обращения: 23.02.2026). — Режим доступа: для авторизир. пользователей.

14. **Лань** : электронно-библиотечная система. — URL: <https://e.lanbook.com/> (дата обращения: 23.02.2026). — Режим доступа: для авторизир. пользователей.

15. **Образовательная платформа Юрайт**. — URL: <https://urait.ru/> (дата обращения: 23.02.2026). — Режим доступа: для авторизир. пользователей.

16. **PROFобразование** : электронный ресурс цифровой образовательной среды СПО. — URL: <https://profspo.ru/> (дата обращения: 23.02.2026). — Режим доступа: для авторизир. пользователей.

Интернет-источники

17. **Project Management Institute (PMI)**. — URL: <https://www.pmi.org/> (дата обращения: 23.02.2026). — Текст : электронный.

18. **Scrum Guides**. — URL: <https://scrumguides.org/> (дата обращения: 23.02.2026). — Текст : электронный.

19. **Habr.com** : сообщество ИТ-специалистов. — URL: <https://habr.com/> (дата обращения: 23.02.2026). — Текст : электронный.