Министерство образования и науки Самарской области

АССОЦИАЦИЯ ПРОФЕССИОНАЛЬНАЯ ОБРАЗОВАТЕЛЬНАЯ ОРГАНИЗАЦИЯ «ИНСТИТУТ НАПРАВО.РУ»

УТВЕРЖДАЮ
Приказ директора
ИНСТИТУТ НАПРАВО.РУ
от 08.06.2026 г № 022/с

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ВЫПОЛНЕНИЮ ПРАКТИЧЕСКИХ РАБОТ МДК.03.01 ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ

по программе подготовки специалистов среднего звена
09.02.11 Разработка и управление программным обеспечением

Тольятти, 2026

СОДЕРЖАНИЕ

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА 5
Практическая работа№1. Проведение интервью с «заказчиком» 10
Практическая работа№2. Составление списка требований 13
Практическая работа№3. Формализация требований с использованием таблиц 16
Практическая работа№4. Определение функциональных требований 19
Практическая работа№5. Уточнение нефункциональных требований 21
Практическая работа№6. Приоритизация требований методом MoSCoW . 24
Практическая работа№7. Разработка глоссария проекта 26
Практическая работа№8. Моделирование бизнес-процесса 28
Практическая работа№9. Построение карты заинтересованных сторон 30
Практическая работа№10. Анализ сценариев взаимодействия 32
Практическая работа№11. Проработка шаблона спецификации требований (SRS) 34
Практическая работа№12. Подготовка таблицы ограничений 36
Практическая работа№13. Идентификация рисков на этапе анализа 38
Практическая работа№14. Оценка полноты требований по чек-листу 41
Практическая работа№15. Формирование структуры ТЗ 43
Практическая работа№16. Построение модели прецедентов 45
Практическая работа№17. Анализ конфликта требований 47
Практическая работа№18. Составление матрицы соответствия 50
Практическая работа№19. Сопоставление требований и модулей 52
Практическая работа№20. Создание модели поведения системы 54
Практическая работа№21. Определение объемов данных на входе/выходе56
Практическая работа№22. Работа с гипотетическим заказчиком 58
Практическая работа№23. Сценарное моделирование 60
Практическая работа№24. Формализация альтернативных потоков данных 62
Практическая работа№25. Выявление точек интеграции 64
Практическая работа№26. Описание функциональных блоков 66

Практическая работа№27. Разработка и ревизия требований..... 68
Практическая работа№28. Работа с фреймами требований 70
Практическая работа№29. Подготовка презентации требований..... 72
Практическая работа№30. Составление и согласование технического задания..... 74
Практическая работа№31. Разработка диаграммы прецедентов 76
Практическая работа№32. Построение диаграммы активностей 78
Практическая работа№33. Создание диаграммы состояний для объекта ... 80
Практическая работа№34. Создание диаграммы классов 82
Практическая работа№35. Разработка модели данных 84
Практическая работа№36. Проектирование ER-диаграммы..... 86
Практическая работа№37. Нормализация таблиц до 3НФ..... 88
Практическая работа№38. Определение сущностей и связей..... 90
Практическая работа№39. Проектирование интерфейса пользователя..... 92
Практическая работа№40. Разработка прототипа интерфейса 94

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Данные методические указания предназначаются для оказания помощи студентам в выполнении практических работ по учебной МДК.03.01 «Проектирование информационных систем»

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 учебной дисциплины МДК.03.01 «Проектирование информационных систем» обучающийся должен знать и уметь:

ПК 3.1. Собирать исходные данные для разработки проектной документации на информационную систему.	**Навыки:**
	− сбора в соответствии с трудовым заданием документации заказчика касательно его запросов и потребностей применительно к типовой ИС; − анкетирования представителей заказчика в соответствии с трудовым заданием; − интервьюирования представителей заказчика в соответствии с трудовым заданием; − документирования собранных данных в соответствии с регламентами организации
	Умения:
	− проводить сбор и анализ исходных данных для разработки проектной документации на информационную систему; − определять требования и функциональность информационной системы на основе собранных данных; − организовывать и управлять процессом сбора исходных данных для разработки проектной документации; − проводить анкетирование; − проводить интервьюирование
	Знания:
	− основные принципы и методы сбора и анализа исходных данных для разработки проектной документации на информационную систему; − возможности типовой ИС;

	− предметную область автоматизации; − инструменты и методы выявления требований; − технологии межличностной и групповой коммуникации в деловом взаимодействии, основы конфликтологии; − архитектуру, устройство и функционирование вычислительных систем; − коммуникационное оборудование; − сетевые протоколы; − основы современных операционных систем; − основы современных систем управления базами данных; − устройство и функционирование современных ИС; − современные стандарты информационного взаимодействия систем; − программные средства и платформы инфраструктуры информационных технологий организаций; − системы классификации и кодирования информации, в том числе присвоение кодов документам и элементам справочников; − отраслевую нормативную техническую документацию; − источники информации, необходимой для профессиональной деятельности; − современный отечественный и зарубежный опыт в профессиональной деятельности; − основы бухгалтерского учета и отчетности организаций; − основы налогового законодательства российской федерации; − культуру речи; − правила деловой переписки
ПК 3.2. Разрабатывать проектную документацию на разработку информационной системы в соответствии с требованиями заказчика.	**Навыки:**
	− разработки проектной документации для информационных систем
	Умения:
	− выбирать оптимальные технологии для реализации проекта; − разрабатывать планы проекта и управлять процессом разработки; − документировать проектную документацию в соответствии со стандартами и нормативными документами; − оценивать риски и принимать меры по их управлению
	Знания:
	− методологию разработки информационных систем; − принципы и методы анализа требований заказчика; − методы проектирования информационных систем и их компонентов; − принципы и методы выбора технологий для реализации проекта; − методы оценки рисков и управления проектом; − методы документирования проектной документации; − стандарты и нормативные документов в области разработки информационных систем; − принципы и методы обеспечения безопасности информационных систем; − принципы и методы управления изменениями в информационных системах
ПК 3.3. Разрабатывать подсистемы безопасности информационной системы в соответствии с техническим заданием.	**Навыки:**
	− разработки подсистем безопасности информационных систем; − применения современных методов и технологий в области безопасности информационных систем; − оптимизации подсистем безопасности информационных систем
	Умения:
	− анализировать требований безопасности информационных систем; − разрабатывать и реализовывать подсистемы безопасности информационных систем; − тестировать и проводить отладку подсистем безопасности информационных

	систем
	Знания:
	− принципы безопасности информационных систем; − современные методы и технологии в области безопасности информационных систем; − законодательных и нормативных актов в области безопасности информационных систем
ПК 3.4. Производить разработку модулей информационной системы в соответствии с техническим заданием.	**Навыки:**
	− разработки кода ИС и баз данных ИС в соответствии с техническим заданием; − верификации кода ИС и баз данных ИС относительно дизайна ИС и структуры баз данных ИС в соответствии с трудовым заданием; − устранения обнаруженных несоответствий в соответствии с трудовым заданием
	Умения:
	− разрабатывать модули информационной системы с использованием выбранного языка программирования; − разрабатывать модули информационной системы в соответствии с требованиями, описанными в техническом задании; − разрабатывать API; − организовывать взаимодействие модулей информационной системы
	Знания:
	− языки программирования и работы с базами данных; − инструменты и методы модульного тестирования; − основы современных операционных систем; − основы современных систем управления базами данных; − устройство и функционирование современных ИС; − теорию баз данных; − системы хранения и анализа баз данных; − основы программирования; − современные объектно-ориентированные языки программирования; − современные структурные языки программирования; − языки современных бизнес-приложений; − современные методики тестирования разрабатываемых ИС; − современные стандарты информационного взаимодействия систем; − программные средства и платформы инфраструктуры информационных технологий организаций; − системы классификации и кодирования информации, в том числе присвоение кодов документам и элементам справочников; − отраслевую нормативную техническую документацию; − источники информации, необходимой для профессиональной деятельности; − основные языки программирования, такие как понимание принципов работы и особенностей выбранного языка программирования; − методологии разработки модулей информационной системы; − основные инструменты разработки, такие как среды разработки, системы контроля версий; − структуру и содержание технического задания
ПК 3.5. Интегрировать информационную систему с существующими	**Навыки:**
	− интеграции информационной системы с существующими системами заказчика; − разработки API для интеграции информационной системы; − тестирования и отладки интеграции информационной системы;

информационными системами заказчика.	− проектирования интерфейсов обмена данными в соответствии с трудовым заданием; − разработки интерфейсов обмена данными в соответствии с трудовым заданием
	Умения:
	− работать в команде над интеграцией модулей в информационную систему; − выполнять интеграцию программный модулей в программный продукт; − кодировать на языках программирования; − находить и анализировать ключевые понятия и термины в сторонней документации для интеграции, а также разбираться в их контексте и использовании в рамках проекта.
	Знания:
	− принципы интеграции информационной системы с другими системами; − современные технологии и инструменты для разработки интеграции информационной системы; − принципы тестирования и отладки интеграции информационной системы; − форматы обмена данных; − интерфейсы обмена данных
ПК 3.6. Осуществлять модульное и интеграционное тестирование информационной системы.	**Навыки:**
	− выделения классов эквивалентности значений каждого типа входных данных; − составления списка комбинаций значений из различных классов эквивалентности; − построения тестовых случаев, в которых сочетаются одна перестановка значений с необходимыми внешними ограничениями; − написания/настройки программ для автоматизированного тестирования ПО; − разработки рабочих заданий по подготовке тестовых данных и выполнению тестовых процедур ПО; − описания тестовых случаев; − разработки автоматизированных тестов, в том числе для проверки информационной безопасности разрабатываемого ПО
	Умения:
	− документировать тесты в соответствии с требованиями организации; − разрабатывать скрипты и/или программные модули для автоматизации; тестирования по, в том числе для проверки информационной безопасности разрабатываемого ПО; − оформлять тестовые случаи; − применять различные техники проектирования тестов (тест-дизайна); − применять универсальные языки моделирования (сценариев); − применять языки программирования для написания программного кода; − применять специализированное ПО для создания автотестов; − применять стандарты оформления кода; − анализировать тестовые случаи на предмет полноты учета покрытия
	Знания:
	− нормативно-технические материалов по вопросам испытания и тестирования ПО; − основные понятия о качестве ПО; − виды технической документации; − российские и международные стандарты тестирования информационных систем; − требования по обеспечению безопасности аппаратных и программных средств автоматизированных систем, используемых при выполнении тестовых

	процедур, включая вопросы антивирусной защиты; − основы работы в операционной системе, в которой производится тестирование, на уровне, необходимом для тестирования разработанного ПО; − классификация видов и типов тестирования ПО; − техники проектирования и комбинаторики тестов; − основы работы необходимых приложений; − системы автоматизированного тестирования ПО; − языки программирования; − тестовые данные, обеспечивающие проверку безопасности ПО
ПК 3.7. Разрабатывать техническую документацию на эксплуатацию информационной системы.	**Навыки:**
	− разработки технической документации на эксплуатацию информационной системы для компании; − участия в проекте по внедрению новой информационной системы в компанию, включая разработку соответствующей документации; − проведения обучения пользователей по использованию информационной системы на основе разработанной документации
	Умения:
	− собирать и анализировать информацию о системе; − описывать процедуры установки и настройки системы; − описывать основные функции и возможности системы; − описывать процедуры обслуживания и регулярного обновления системы − разрабатывать руководство пользователя
	Знания:
	− принципы работы информационных систем; − процедуры установки и настройки системы; − типы, виды и содержание документации на информационные системы в соответствии с ISO и ГОСТ на каждом этапе жизненного цикла информационных систем
ПК 3.8. Производить оценку информационной системы для выявления возможности ее модернизации.	**Навыки:**
	− участия в проекте по модернизации информационной системы компании; − разработки плана модернизации информационной системы для компании; − участия в проекте по внедрению новых технологий в информационную систему компании
	Умения:
	− анализировать текущее состояние информационной системы и выявить ее слабые места; − предлагать меры по улучшению информационной системы и оценивать их эффективность; − анализировать совместимость новых технологий с текущей информационной системой и предлагать меры по их интеграции
	Знания:
	− принципы работы информационных систем; − основные проблемы, с которыми может столкнуться информационная система; − современные технологий и методы модернизации информационных систем; − принципы оценки эффективности мер по модернизации информационной системы

Практическая работа№1. Проведение интервью с «заказчиком»

Цель

Научиться планировать и проводить структурированное интервью для выявления требований к информационной системе.

Задачи

1. Подготовить список вопросов для интервью
2. Провести интервью (реальное или ролевое)
3. Задокументировать ответы

Теоретический материал

Интервью — основной метод выявления требований, при котором аналитик задаёт вопросы заказчику и фиксирует ответы.

Типы вопросов:

Тип	Описание	Пример	Когда использовать
Открытые	Развёрнутый ответ	«Какие проблемы вы хотите решить?»	В начале интервью
Закрытые	Да/Нет или выбор	«Система должна работать онлайн?»	Для уточнения
Уточняющие	Детализация	«Что именно вы имеете в виду под "быстро"?»	При неоднозначности
Зеркальные	Повтор ответа	«Вы сказали, что важно удобство. Что это значит?»	Для проверки понимания

Правила хорошего интервью:

1. Готовьте вопросы заранее (15-20 штук)
2. Начинайте с общих целей, затем детали
3. Не задавайте наводящих вопросов («Вам ведь нужна система с отчётами?»)
4. Записывайте (диктофон + заметки)

5. Спрашивайте «почему» 2-3 раза подряд для поиска истинной потребности

Структура интервью:

1. Вступление (5 мин): представление, цель, регламент
2. Общие цели (10 мин): зачем нужна система
3. Пользователи и роли (10 мин): кто будет работать
4. Функциональность (15 мин): что должна делать
5. Ограничения (10 мин): бюджет, сроки, технологии
6. Заключение (5 мин): итоги, следующие шаги

Пример вопроса и фиксации:

Вопрос: «Опишите типичный день пользователя вашей системы»

Ответ: «Утром я захожу, смотрю список новых заказов, распределяю их по курьерам...»

Фиксация: «Пользователь (менеджер) → просмотр списка → распределение заказов»

Типичные ошибки:

- Задавать слишком общие вопросы («Что вам нужно?»)
- Перебивать заказчика
- Не уточнять неоднозначные термины

Задание

Подготовить и провести интервью с заказчиком для системы по вашему варианту.

Пошаговое выполнение

Шаг 1. Составьте список из 15-20 вопросов (разделите на блоки):

- Блок 1: Общие цели (3 вопроса)
- Блок 2: Пользователи и роли (3 вопроса)
- Блок 3: Функциональность (5 вопросов)
- Блок 4: Технические ограничения (4 вопроса)
- Блок 5: Бюджет и сроки (3 вопроса)

Шаг 2. Найдите «заказчика» (одногруппник, преподаватель или сыграйте роль сами).

Шаг 3. Проведите интервью (20-30 минут), записывая ответы.

Шаг 4. Расшифруйте записи в структурированный документ.

Результат

- Список вопросов
- Протокол интервью (1-2 страницы)

Критерии оценки

Критерий	**Баллы**
Вопросы покрывают все ключевые области	4
Вопросы не наводящие, открытого типа	3
Протокол структурирован, читаем	2
Есть уточняющие вопросы по ходу	1
Итого	10

Практическая работа№2. Составление списка требований

Цель

Научиться преобразовывать сырую информацию из интервью в структурированный список требований.

Задачи

1. Извлечь требования из протокола интервью
2. Присвоить каждому требованию уникальный ID
3. Удалить дубликаты

Теоретический материал

Требование — это условие или возможность, необходимая пользователю для решения задачи, которой должна соответствовать система.

Формулировка хорошего требования:

«Система должна [глагол] + [объект] + [условие]»

Плохо: «Система должна быть удобной» (неизмеримо)

Хорошо: «Система должна позволять пользователю найти задачу по названию за 3 секунды»

Правила формулировки:

1. Одно требование — один глагол
2. Не использовать «и», «также» (разбивайте на несколько)
3. Использовать однозначные термины (из глоссария)
4. Избегать слов «удобный», «быстрый», «надёжный» без цифр

Процесс составления списка:

1. Извлеките все «хотелки» из протокола интервью
2. Переформулируйте каждую в формальное требование
3. Присвойте уникальный ID (R-001, R-002...)
4. Удалите дубликаты
5. Сгруппируйте по темам (Авторизация, Отчёты, Интеграции)

Пример извлечения:

Слова заказчика: «Хочу, чтобы при регистрации приходило письмо с подтверждением»

Требование: «R-001: Система должна отправлять email с ссылкой для подтверждения регистрации»

Типичные ошибки:

- Смешивание функциональных и нефункциональных требований
- Слишком длинные требования (более 25 слов)
- Отсутствие ID

Задание

На основе протокола интервью (Работа №1) составить список требований.

Пошаговое выполнение

Шаг 1. Прочитайте протокол интервью и выпишите все «хотелки» заказчика.

Шаг 2. Сформулируйте каждую как требование по шаблону.

Шаг 3. Присвойте ID: R-001, R-002...

Шаг 4. Удалите дубликаты (одинаковые требования).

Шаг 5. Сгруппируйте по темам (Авторизация, Работа с данными, Отчёты...).

Результат

Таблица:

ID	Требование	Тема
R-001	Система должна ...	

Критерии оценки

Критерий	Баллы
Минимум 15 требований	4
Каждое сформулировано по шаблону	3

Критерий	Баллы
Уникальные ID	1
Группировка по темам	2
Итого	10

Практическая работа№3. Формализация требований с использованием таблиц

Цель

Научиться представлять требования в формате, пригодном для дальнейшего анализа и трассировки.

Задачи

1. Создать таблицу требований с атрибутами
2. Заполнить все поля
3. Настроить фильтрацию

Теоретический материал

Формализация — приведение требований к единому структурированному формату для дальнейшего анализа.

Атрибуты требования:

Атрибут	Описание	Пример
ID	Уникальный номер	R-001
Название	Краткое (до 10 слов)	Регистрация пользователя
Описание	Полный текст	Система должна позволять пользователю создать учётную запись по email
Тип	F (функциональное) / NFR (нефункциональное)	F
Источник	Откуда взято	Интервью от 15.01
Приоритет	Must/Should/Could/Won't	Must
Статус	Предложено / Утверждено / Отклонено	Предложено

Преимущества табличной формализации:

- Возможность фильтрации и сортировки
- Легко найти требования по любому атрибуту
- Основа для трассировки

Пример таблицы

ID	Название	Описание	Тип	Источник	Приоритет	Статус
R-001	Регистрация	Система должна позволять регистрацию по email	F	Интервью	Must	Предложено
R-002	Производительность	Время отклика < 200 мс	NFR	Интервью	Should	Предложено

Правила:

- Каждое требование должно иметь все атрибуты
- Не оставлять пустых ячеек
- Тип NFR требует указания единицы измерения

Типичные ошибки:

- Пропуск атрибута «Источник»
- Нет единообразия в названиях

Задание

Формализовать требования из Работы №2 в таблицу Excel.

Пошаговое выполнение

Шаг 1. Создайте Excel/Google Sheets.

Шаг 2. Добавьте колонки: ID, Название, Описание, Тип, Приоритет, Источник, Статус.

Шаг 3. Перенесите все требования из Работы №2.

Шаг 4. Для каждого требования укажите тип (F или NFR).

Шаг 5. Заполните источник (например, «Интервью от 01.01.2025»).

Шаг 6. Настройте фильтры (Данные → Фильтр) и условное форматирование.

Результат

Файл Excel с таблицей требований (минимум 15 строк).

Критерии оценки

Критерий	Баллы
Все колонки заполнены	4
Тип указан верно	2
Источник указан для каждого	2
Настроены фильтры	2
Итого	10

Практическая работа№4. Определение функциональных требований

Цель

Научиться выделять функциональные требования из общего списка и формулировать их в тестируемом виде.

Задачи

1. Отфильтровать функциональные требования
2. Проверить каждое на тестируемость
3. Переформулировать «плохие» требования

Теоретический материал

Функциональное требование (FR) — описывает **что** система ДЕЛАЕТ: действие, операцию, функцию.

Признаки FR:

- Начинается с глагола (отправить, сохранить, отобразить, рассчитать)
- Можно проверить тестом (Да/Нет)
- Описывает поведение, а не качество

Примеры FR:

- «Система должна сохранять данные заказа в базе данных»
- «Система должна отображать список активных задач на главном экране»

Тестируемость FR:

Вопрос: «Можно ли написать тест, который однозначно покажет, выполнено требование или нет?»

Требование	Тестируемо?	Почему?
«Система должна отправлять email»	Да	Можно проверить — пришёл email или нет
«Система должна работать быстро»	Нет	Нет критерия «быстро»

Как исправить нетестируемое FR:

- Добавить измерение: «быстро» → «менее чем за 2 секунды»
- Добавить условие: «надёжно» → «с вероятностью 99.9%»

Типичные ошибки:

- Смешение с NFR («Система должна быть безопасной» — это NFR)
- Отсутствие глагола («Пользователь и заказ»)

Задание

Из общего списка требований выделить функциональные и проверить их на тестируемость.

Пошаговое выполнение

Шаг 1. Отфильтруйте в таблице требования с типом «F».

Шаг 2. Для каждого ответьте на вопрос: «Можно ли написать тест, который проверит это требование?»

Шаг 3. Если нет — переформулируйте.

Пример переформулировки:

- Плохо: «Система должна быть надёжной»
- Хорошо: «Система должна обрабатывать 1000 одновременных запросов без падения»

Результат. Список функциональных требований (минимум 10) с отметкой о тестируемости (Да/Нет).

Критерии оценки

Критерий	Баллы
Выделены все FR из списка	3
Каждое FR проверено на тестируемость	3
«Плохие» FR переформулированы	4
Итого	10

Практическая работа№5. Уточнение нефункциональных требований

Цель

Научиться выявлять и формулировать измеримые нефункциональные требования.

Задачи

1. Выделить нефункциональные требования
2. Классифицировать по FURPS+
3. Сделать каждое измеримым

Теоретический материал

Нефункциональное требование (NFR) — описывает как система работает: качественные характеристики.

Классификация NFR (FURPS+):

Категория	Что описывает	Пример
U — Usability	Удобство использования	«Экран входа загружается < 2 сек»
R — Reliability	Надёжность	«Время безотказной работы 99.9%»
P — Performance	Производительность	«API отвечает < 200 мс при 100 RPS»
S — Supportability	Поддерживаемость	«Логи хранятся 30 дней»
+	Дополнительно	Безопасность, юридические, физические

Правила хорошего NFR:

1. Должно быть измеримым (содержать число/единицу)
2. Должно быть проверяемым (тест или измерение)
3. Должно быть реалистичным (не 0 мс время отклика)

Плохо vs Хорошо:

Плохо	Хорошо

Плохо	Хорошо
Система должна быть быстрой	Время отклика API < 200 мс при 100 одновременных запросах
Система должна быть надёжной	Доступность системы 99.9% в месяц
Система должна быть безопасной	Шифрование паролей bcrypt, блокировка после 3 неудачных попыток

Типичные ошибки:

- NFR без цифр («высокая производительность»)
- NFR, которые нельзя проверить до релиза («лёгкость сопровождения»)

Задание

Сформулировать измеримые нефункциональные требования для вашей системы.

Пошаговое выполнение

Шаг 1. Отфильтруйте требования с типом «NFR».

Шаг 2. Для каждого определите категорию (U/R/P/S).

Шаг 3. Проверьте, есть ли у требования число/измерение.

Шаг 4. Если нет — добавьте.

Пример:

- Было: «Система должна работать быстро»
- Стало: «Время отклика API при 100 RPS не превышает 200 мс»

Результат

Таблица:

ID	NFR	Категория	Измеримый критерий

Критерии оценки

Критерий	**Баллы**
Минимум 5 NFR	3

Критерий	Баллы
Каждое отнесено к категории FURPS	3
Каждое имеет измеримый критерий	4
Итого	10

Практическая работа№6. Приоритизация требований методом MoSCoW

Цель

Научиться расставлять приоритеты требований совместно с заказчиком.

Задачи

1. Подготовить стикеры требований
2. Провести workshop с заказчиком
3. Распределить требования по 4 группам

Теоретический материал

MoSCoW — метод расстановки приоритетов, где требования делятся на 4 группы.

Группа	Значение	% от общего
M — Must have	Без этого система бесполезна	~60%
S — Should have	Важно, но есть обходной путь	~20%
C — Could have	Приятно иметь	~20%
W — Won't have	Не делаем в этом релизе	—

Как определить Must have:

«Что случится, если этого не будет?» → «Система не сможет работать»

«Можем ли мы запуститься без этого?» → «Нет»

Как определить Won't have:

Заказчик сказал «не надо» или «потом»

Ресурсов (бюджет/время) заведомо не хватит

Пример распределения для интернет-магазина:

Must: Каталог товаров, Корзина, Оформление заказа

Should: Отзывы, Рекомендации

Could: Чат с поддержкой, Сравнение товаров

Won't: Мобильное приложение (в первом релизе)

Правила проведения workshop:

Разложите стикеры всех требований на столе

Попросите заказчика разложить их на 4 группы

Если группа Must > 60% — попросите пересмотреть

Фиксируйте результат

Типичные ошибки:

Всё подряд в Must (обычно 80-90% у новичков)

Нет Won't (значит, не отсекли лишнее)

Задание

Провести приоритизацию всех требований методом MoSCoW.

Пошаговое выполнение

Шаг 1. Распечатайте или нарисуйте стикеры для каждого требования.

Шаг 2. Пригласите «заказчика» (можно в роли).

Шаг 3. Попросите разложить стикеры на 4 группы.

Шаг 4. Задавайте уточняющие вопросы:

- «Что случится, если это не сделать?»
- «Можно ли это сделать позже?»

Шаг 5. Зафиксируйте результат в таблице требований.

Результат

Таблица требований с заполненной колонкой «Приоритет» (M/S/C/W).

Критерии оценки

Критерий	Баллы
Все требования имеют приоритет	3
Must have не более 60%	4
Won't have минимум 1 требование	3
Итого	10

Практическая работа№7. Разработка глоссария проекта

Цель

Научиться создавать единый словарь терминов для устранения неоднозначности.

Задачи

1. Выявить ключевые термины предметной области
2. Дать каждому чёткое определение
3. Указать запрещённые синонимы

Теоретический материал

Глоссарий — словарь терминов проекта, который устраняет неоднозначность понимания между заказчиком и командой.

Зачем нужен глоссарий:

- Заказчик говорит «клиент», разработчик — «пользователь», а это одно и то же (или нет?)
- Термин «заказ» может означать «счёт», «накладную», «резерв»

Правила хорошего определения:

1. Не использовать сам термин в определении («Заказ — это заказ товара»)
2. Определение должно быть полным, но кратким (2-3 предложения)
3. Указать, что НЕ является этим термином (запрещённые синонимы)

Структура глоссария**:**

Термин	Определение	Запрещённые синонимы
Заказ	Запись о намерении пользователя приобрести товар(ы) с фиксацией цены, количества и срока доставки	Покупка, Транзакция, Чек
Пользователь	Физическое лицо, имеющее учётную запись в системе	Клиент, Юзер, Оператор

Откуда брать термины:

- Из интервью (неоднозначные слова заказчика)
- Из требований (существительные)
- Из предметной области (специфические термины)

Типичные ошибки:

- Определение длиннее самого термина
- Отсутствие запрещённых синонимов (самая частая ошибка)

Правила:

- Один термин — одно определение
- Определение должно отличаться от самого термина

Задание

Разработать глоссарий из 10-15 терминов для вашей системы.

Пошаговое выполнение

Шаг 1. Выпишите все специфические слова из интервью и требований.

Шаг 2. Для каждого напишите определение своими словами.

Шаг 3. Укажите синонимы, которые НЕЛЬЗЯ использовать.

Шаг 4. Согласуйте с заказчиком.

Результат. Таблица:

Термин	Определение	Запрещённые синонимы
Заказ	...	«Покупка», «Транзакция»

Критерии оценки

Критерий	Баллы
Минимум 10 терминов	4
Определения чёткие, без круговых ссылок	3
Указаны запрещённые синонимы	3
Итого	10

Практическая работа№8. Моделирование бизнес-процесса

Цель

Научиться визуализировать текущий бизнес-процесс в нотации BPMN.

Задачи

1. Собрать информацию о текущем процессе
2. Нарисовать диаграмму «как есть» (AS-IS)
3. При необходимости — диаграмму «как будет» (TO-BE)

Теоретический материал

Бизнес-процесс — последовательность действий, выполняемых сотрудниками (или системой) для достижения результата.

Основные элементы BPMN:

Элемент	Обозначение	Назначение
Пул (Pool)	Прямоугольник	Организация или система
Дорожка (Lane)	Полоса внутри пула	Роль или отдел
Задача (Task)	Скруглённый прямоугольник	Действие
Событие (Event)	Круг	Начало/конец/пауза
Шлюз (Gateway)	Ромб	Ветвление

AS-IS vs TO-BE:

AS-IS («как есть») — текущий процесс, со всеми проблемами

TO-BE («как будет») — целевой процесс после внедрения системы

Задание

Смоделировать бизнес-процесс для выбранной системы.

Пошаговое выполнение

Шаг 1. Опишите текстом текущий процесс (5-10 шагов).

Шаг 2. Определите роли участников.

Шаг 3. В Draw.io или Visio нарисуйте диаграмму.

Шаг 4. Обозначьте все ветвления (да/нет).

Шаг 5. Сохраните в PNG.

Результат. Скриншот BPMN-диаграммы.

Критерии оценки

Критерий	Баллы
Отражены все роли (минимум 2 дорожки)	3
Есть ветвления (шлюзы)	3
Диаграмма читается без пояснений	4
Итого	10

Практическая работа№9. Построение карты заинтересованных сторон

Цель

Научиться выявлять всех стейкхолдеров проекта и определять стратегию работы с ними.

Задачи

1. Составить список заинтересованных сторон
2. Оценить власть и интерес каждого
3. Построить матрицу и определить стратегию

Теоретический материал

Заинтересованная сторона (стейкхолдер) — любой человек или группа, которые влияют на проект или зависят от него.

Задание

Построить карту заинтересованных сторон для вашей системы.

Пошаговое выполнение

Шаг 1. Составьте список всех, кто влияет на проект или зависит от него.

Шаг 2. Каждому поставьте оценку «Власть» (1-5) и «Интерес» (1-5).

Шаг 3. Нанесите точки на матрицу (в Excel или Miro).

Шаг 4. Для каждого квадранта пропишите стратегию.

Результат

- Список стейкхолдеров (минимум 6)
- Матрица «власть-интерес» (скриншот)
- Таблица стратегий

Критерии оценки

Критерий	Баллы
Минимум 6 стейкхолдеров	3

Критерий	Баллы
Корректное распределение по квадрантам	4
Стратегии соответствуют квадрантам	3
Итого	10

Практическая работа№10. Анализ сценариев взаимодействия

Цель

Научиться описывать пошаговый диалог между пользователем и системой.

Задачи

1. Выбрать 5 ключевых сценариев
2. Описать каждый в формате «шаг за шагом»
3. Указать входные и выходные данные

Теоретический материал

Сценарий взаимодействия — пошаговое описание диалога между пользователем и системой.

Структура сценария взаимодействия:

Сценарий: [Название]

Предусловие: [Что уже сделано]

Шаг 1: Пользователь делает [действие]

Шаг 2: Система отвечает [реакция]

...

Постусловие: [Состояние системы после]

Предусловие — что должно быть true до начала сценария. **Постусловие** — что изменится в системе после успешного выполнения.

Типы шагов:

- **Действие пользователя:** «Пользователь нажимает кнопку "Войти"»
- **Реакция системы:** «Система проверяет логин и пароль»
- **Решение:** «Если данные верны → переход к шагу 4, если нет → показать ошибку»

Ключевые сценарии для любой системы:

1. Вход/регистрация
2. Создание основного объекта (заказ, задача, бронь)

3. Просмотр списка
4. Редактирование/удаление
5. Формирование отчёта

Типичные ошибки:

- Нет предусловия или постусловия
- Шаги слишком крупные («пользователь работает с системой»)

Задание

Описать 5 сценариев взаимодействия для вашей системы.

Пошаговое выполнение

Шаг 1. Выберите 5 ключевых задач пользователя (например, «Зарегистрироваться», «Создать заказ»).

Шаг 2. Для каждого распишите пошагово.

Шаг 3. Укажите, что должно быть в системе до начала (предусловие).

Шаг 4. Укажите, что изменится в системе после (постусловие).

Результат

Таблица из 5 сценариев.

Критерии оценки

Критерий	Баллы
5 сценариев	3
Каждый имеет предусловие и постусловие	3
Шаги логичны и воспроизводимы	4
Итого	10

Практическая работа№11. Проработка шаблона спецификации требований (SRS)

Цель

Научиться использовать стандартный шаблон для документирования требований.

Задачи

1. Взять шаблон SRS (IEEE 830 или ГОСТ 34)
2. Заполнить основные разделы
3. Подготовить структуру документа

Теоретический материал

SRS (Software Requirements Specification) — основной документ, содержащий все требования к системе.

Стандарт IEEE 830 — международный стандарт для SRS.

Структура SRS по IEEE 830:

Раздел	Содержание
1. Введение	Цель документа, область действия, определения
2. Общее описание	Контекст системы, пользователи, ограничения
3. Функциональные требования	Что система делает (FR)
4. Нефункциональные требования	Как система работает (NFR)
5. Требования к интерфейсам	UI, API, интеграции
6. Модели данных	ER-диаграмма, нормализация
Приложения	Глоссарий, диаграммы, use case'ы

Правила оформления:

- Каждое требование должно быть трассируемым (иметь ID)
- Использовать ссылки между разделами
- Версионировать документ

Что НЕ должно быть в SRS:

- Детали реализации (алгоритмы, структуры данных)
- План проекта (сроки, бюджет)
- Дизайн интерфейса (это отдельный документ)

Типичные ошибки:

- Путаница между SRS и ТЗ (ТЗ — более широкий документ, включает план работ)
- Отсутствие версий

Задание

Создать шаблон SRS для вашей системы.

Пошаговое выполнение

Шаг 1. Создайте документ Word/Google Docs.

Шаг 2. Оформите титульный лист.

Шаг 3. Создайте разделы по структуре IEEE 830.

Шаг 4. Заполните вводную часть (цель, область).

Шаг 5. Оставьте место для требований (вставите из Работы №3).

Результат

Документ SRS (минимум 5 страниц с заголовками и пустыми таблицами).

Критерии оценки

Критерий	Баллы
Соответствие IEEE 830	4
Заполнены вводные разделы	3
Титульный лист оформлен	3
Итого	10

Практическая работа№12. Подготовка таблицы ограничений

Цель

Научиться выявлять и документировать ограничения проекта.

Задачи

1. Выявить технические, бюджетные, временные, юридические ограничения
2. Оформить в таблицу
3. Указать источник и влияние

Теоретический материал

Ограничение — условие или правило, которое сужает пространство решений.

Типы ограничений:

Тип	Описание	Пример
Бюджетные	Фиксированная сумма	«Бюджет проекта — 5 млн руб.»
Временные	Дедлайны	«Система готова через 6 месяцев»
Технические	Технологии, платформы	«Использовать PostgreSQL, не MySQL»
Юридические	Законы, регуляторы	«Персональные данные — на территории РФ»
Организационные	Доступность экспертов	«Только сотрудники компании X»

Формат записи ограничения:

О-001: [Тип] [Описание] — Источник: [откуда] — Влияние: [высокое/среднее/низкое]

Пример:

О-003: Бюджетное — Бюджет проекта фиксирован и составляет 5 млн рублей — Источник: контракт — Влияние: высокое

Зачем нужна таблица ограничений:

- Не проектировать то, что заведомо невозможно
- Обосновывать решения перед заказчиком
- Оценивать риски

Типичные ошибки:

- Путаница ограничений и рисков (риск — возможное событие, ограничение — факт)
- Ограничения, которые не являются жёсткими («желательно использовать...»)

Задание

Составить таблицу ограничений для вашей системы.

Пошаговое выполнение

Шаг 1. Просмотрите протокол интервью и найдите упоминания ограничений.

Шаг 2. Добавьте стандартные (бюджет, сроки, технологии).

Шаг 3. Для каждого укажите источник.

Шаг 4. Оцените влияние на проект (высокое/среднее/низкое).

Результат

Таблица:

ID	Ограничение	Тип	Источник	Влияние
C-001	...	...	...	...

Критерии оценки

Критерий	Баллы
Минимум 5 ограничений	4
Все типы представлены	3
Указаны источники	3
Итого	10

Практическая работа№13. Идентификация рисков на этапе анализа

Цель

Научиться выявлять риски, связанные с требованиями, и разрабатывать меры снижения.

Задачи

1. Выявить минимум 5 рисков
2. Оценить вероятность и влияние
3. Предложить меры снижения

Теоретический материал

Риск — возможное событие, которое может негативно повлиять на проект.

Формула оценки риска:

Уровень риска = Вероятность × Влияние

Шкала вероятности (1-5):

Значение	Вероятность	Описание
1	<5%	Почти невозможно
2	5-20%	Маловероятно
3	20-50%	Возможно
4	50-80%	Вероятно
5	>80%	Почти наверняка

Шкала влияния (1-5):

Значение	Описание	Пример
1	Незначительное	Микро-корректировка
2	Низкое	Небольшие доработки
3	Среднее	Сдвиг сроков на 10-20%

Значение	Описание	Пример
4	Высокое	Сдвиг сроков на 30-50%
5	Критическое	Проект под угрозой

Типовые риски анализа требований:

Риск	Причина
Заказчик меняет требования	Нечёткое ТЗ
Не все стейкхолдеры опрошены	Плохая коммуникация
Требования противоречат друг другу	Отсутствие анализа

Формат записи риска:

RISK-001: [Описание] — Причина: [почему] — Вероятность: [1-5] — Влияние: [1-5] — Меры снижения: [что делать]

Типичные ошибки:

- Описание риска как уже случившейся проблемы
- Отсутствие мер снижения

Задание

Идентифицировать риски, связанные с процессом сбора и анализа требований.

Пошаговое выполнение

Шаг 1. Составьте список из 5-7 рисков.

Шаг 2. Каждому присвойте вероятность (1-5) и влияние (1-5).

Шаг 3. Рассчитайте уровень риска (вероятность × влияние).

Шаг 4. Для рисков с уровнем >10 предложите меры снижения.

Результат

Таблица:

Риск	Причина	Вероятность	Влияние	Уровень	Меры снижения

Риск	Причина	Вероятность	Влияние	Уровень	Меры снижения

Критерии оценки

Критерий	Баллы
Минимум 5 рисков	4
Реалистичные меры снижения	3
Корректный расчёт уровня	3
Итого	10

Практическая работа№14. Оценка полноты требований по чек-листу

Цель

Научиться проверять качество требований по формальным критериям.

Задачи

1. Взять готовый чек-лист
2. Проверить каждое требование
3. Отметить проблемные

Теоретический материал

Чек-лист — список критериев для проверки качества требований.

Критерии INVEST (для гибких методологий):

Буква	Критерий	Вопрос
I	Independent	Можно реализовать отдельно от других?
N	Negotiable	Есть возможность уточнить?
V	Valuable	Даёт ценность пользователю?
E	Estimable	Можно оценить трудоёмкость?
S	Small	Можно сделать за итерацию (2-5 дней)?
T	Testable	Есть объективный критерий проверки?

Дополнительные критерии (IEEE 830):

№	Критерий	Вопрос
1	Корректность	Требование технически реализуемо?
2	Однозначность	Нет двусмысленных слов?
3	Полнота	Все условия описаны?
4	Непротиворечивость	Не конфликтует с другими?
5	Трассируемость	Есть ID и источник?

№	Критерий	Вопрос
6	Тестируемость	Можно написать тест?

Пример проверки:

Требование: «Система должна быть удобной»

- Однозначность: НЕТ (что значит «удобной»?)
- Тестируемость: НЕТ (как проверить?)
- Вывод: требование нужно переформулировать

Типичные ошибки:

- Проверка «на глаз», без чек-листа
- Игнорирование требований с «Нет» (надо исправлять)

Задание

Проверить требования вашей системы по чек-листу.

Пошаговое выполнение

Шаг 1. Скопируйте чек-лист в Excel.

Шаг 2. Для каждого требования (из Работы №3) пройдите по всем пунктам чек-листа.

Шаг 3. Ставьте «Да» или «Нет».

Шаг 4. Для требований с хотя бы одним «Нет» напишите замечание.

Результат. Таблица «Требование → Результаты проверки» + список замечаний.

Критерии оценки

Критерий	Баллы
Проверены все требования	4
Замечания конкретны	3
Чек-лист применён корректно	3
Итого	10

Практическая работа№15. Формирование структуры ТЗ

Цель

Научиться создавать структуру технического задания.

Задачи

1. Изучить типовую структуру ТЗ
2. Создать документ с разделами
3. Подготовить содержание

Теоретический материал

Техническое задание (ТЗ) — документ, который регламентирует разработку системы и является юридически значимым.

Структура ТЗ по ГОСТ 34 (стандарт на автоматизированные системы):

Раздел	Содержание
1. Основания для разработки	Документ-основание, заказчик, исполнитель
2. Назначение и цели	Зачем нужна система, какие задачи решает
3. Требования к системе	Функциональные, производительность, надёжность, безопасность
4. Требования к функционалу	Детализация по модулям
5. Состав и содержание работ	Этапы, сроки, результаты
6. Порядок контроля и приёмки	Критерии, процедура сдачи
7. Требования к документации	Какие документы должны быть
Приложения	Глоссарий, диаграммы, SRS (может быть отдельно)

Отличие ТЗ от SRS:

- SRS — только требования к системе (технический документ)
- ТЗ — требования + план работ + юридическая часть (договорной документ)

Правила оформления:

- Каждый раздел начинается с новой страницы

- Обязательно оглавление
- Версия документа и дата на каждом листе

Задание

Создать структуру ТЗ для вашей системы.

Пошаговое выполнение

Шаг 1. Создайте документ Word/Google Docs.

Шаг 2. Сформируйте разделы по ГОСТ 34.

Шаг 3. Внутри разделов создайте подразделы.

Шаг 4. Добавьте автонумерацию и оглавление.

Результат

Документ ТЗ с пустыми разделами (5-7 страниц).

Критерии оценки

Критерий	Баллы
Соответствие ГОСТ 34	4
Есть все основные разделы	3
Оглавление работает (гиперссылки)	3
Итого	10

Практическая работа№16. Построение модели прецедентов

Цель

Научиться создавать текстовое описание use case'ов на основе диаграммы.

Задачи

1. Выделить все прецеденты
2. Для каждого составить карточку
3. Описать основной и альтернативные потоки

Теоретический материал

Модель прецедентов (Use Case Model) — текстовое описание всех use case'ов с детализацией шагов.

Структура карточки use case'а:

Поле	Описание	Пример
ID	Уникальный номер	UC-001
Название	Глагол + существительное	Создать заказ
Актёр	Кто инициирует	Покупатель
Предусловие	Что должно быть true до начала	Пользователь авторизован
Основной поток	Нумерованный список шагов	1. ... 2. ...
Альтернативные потоки	Ветвления и исключения	3а. Недостаточно средств → ...
Постусловие	Состояние после успеха	Заказ сохранён

Основной поток — всегда успешный сценарий (без ошибок).

Альтернативные потоки:

- Альтернатива (Alt): Пользователь выбирает другой путь (например, оплата картой вместо наличных)
- Исключение (Exc): Что-то пошло не так (ошибка, таймаут)

Пример альтернативного потока:

3а. Недостаточно средств на счёте

1. Система отображает сообщение «Недостаточно средств»
2. Пользователь выбирает другой способ оплаты
3. Use case возвращается к шагу 3

Правила:

- Один use case — одна карточка
- Основной поток — 5-10 шагов
- У каждого use case минимум 2 альтернативы

Типичные ошибки:

- Слишком длинный основной поток (>15 шагов) — разбивайте на подпрецеденты (include)
- Отсутствие альтернатив (значит, не продуманы ошибки)

Задание

Построить модель прецедентов (текстовое описание) для всех прецедентов из диаграммы.

Пошаговое выполнение

Шаг 1. Возьмите диаграмму прецедентов (из темы 1.2).

Шаг 2. Для каждого эллипса создайте карточку.

Шаг 3. Опишите основной поток (5-10 шагов).

Шаг 4. Опишите альтернативные потоки (минимум 2 на use case).

Результат. Таблица карточек use case'ов (минимум 5 штук).

Критерии оценки

Критерий	Баллы
Все прецеденты описаны	3
Основной поток детализирован (5+ шагов)	3
Есть альтернативные потоки	4
Итого	10

Практическая работа№17. Анализ конфликта требований

Цель

Научиться находить противоречия между требованиями и предлагать решения.

Задачи

1. Найти минимум 3 конфликта
2. Зафиксировать причину
3. Предложить варианты разрешения

Теоретический материал

Конфликт требований — ситуация, когда два (или более) требования противоречат друг другу.

Типы конфликтов:

Тип	Описание	Пример
Ресурсные	Не хватает бюджета/времени/людей	«Сделать 50 функций за месяц»
Технические	Противоречащие технологии	«Работать оффлайн» vs «Данные в облаке»
Функциональные	Логическое противоречие	«Автосохранение» vs «Подтверждение действия»
Стейкхолдеров	Разные желания заказчиков	«Менеджер хочет отчёты» vs «Программист не хочет их делать»

Как найти конфликт:

1. Прочитайте все требования
2. Ищите противоположные глаголы (сохранять vs не сохранять, показывать vs скрывать)
3. Ищите количественные противоречия («время отклика < 100 мс» vs «БД в другом городе»)

Пример анализа конфликта:

Требование A: «Система должна работать в оффлайн-режиме» Требование B: «Все данные синхронизируются с облаком в реальном времени»

Причина конфликта: Оффлайн-режим означает локальное хранение, а реальная синхронизация требует постоянного интернета

Варианты решения:

1. Убрать оффлайн-режим
2. Сделать отложенную синхронизацию (не реальное время)
3. Разделить на два режима работы

Типичные ошибки:

- Находить конфликты там, где их нет (ложные срабатывания)
- Не предлагать варианты решения

Задание

Найти и проанализировать конфликты в требованиях вашей системы.

Пошаговое выполнение

Шаг 1. Просмотрите все пары требований (особенно Must have).

Шаг 2. Найдите противоречия (логические или ресурсные).

Шаг 3. Для каждого конфликта опишите причину.

Шаг 4. Предложите 2-3 варианта разрешения.

Шаг 5. Выберите лучший вариант.

Результат

Таблица конфликтов:

Конфликт	Требования	Причина	Варианты решения	Выбранное

Критерии оценки

Критерий	Баллы
Минимум 3 конфликта	4
Причина чётко сформулирована	2

Критерий	Баллы
Варианты решения реалистичны	4
Итого	10

Практическая работа№18. Составление матрицы соответствия

Цель

Научиться проверять, что каждое требование имеет источник.

Задачи

1. Создать матрицу «Требования — Источники»
2. Отметить соответствия
3. Найти «висящие» требования

Теоретический материал

Матрица соответствия (Traceability Matrix) — таблица, показывающая связь между требованиями и их источниками.

Зачем нужна:

- **Проверить, что у каждого требования есть источник (не «с потолка»)**
- **Проверить, что каждый источник породил требования (не зря проводили интервью)**
- **При изменении требования — понять, откуда оно взялось**

Структура матрицы:

ID требования	Интервью	Use Case	User Story	Бизнес-процесс	SRS
R-001	X				X
R-002		X	X		
R-003	X	X		X	

Правила:

- У каждого требования должен быть хотя бы один X
- У каждого источника должно быть хотя бы одно требование

Что делать, если требование без X («висячее»):

1. Вспомнить, откуда оно взялось
2. Если не вспомнили — возможно, требование лишнее (удалить)
3. Или добавить источник постфактум

Что делать, если источник без требований:

1. Проверить, правильно ли извлекли требования из этого источника
2. Если нет требований — возможно, источник не нужен

Типичные ошибки:

- Матрица заполнена, но никто её не использует
- Слишком сложная (20+ источников) — нечитаема

Задание

Составить матрицу соответствия для всех требований.

Пошаговое выполнение

Шаг 1. Создайте Excel с колонками: ID требования, Интервью, Use Case, User Story, Бизнес-процесс.

Шаг 2. Для каждого требования поставьте X в тех источниках, откуда оно взято.

Шаг 3. Проверьте: есть ли требования без X?

Шаг 4. Для «висячих» требований найдите или добавьте источник.

Результат. Матрица соответствия (Excel).

Критерии оценки

Критерий	Баллы
Все требования в матрице	4
У каждого требования есть X	3
У каждого источника есть хотя бы одно требование	3
Итого	10

Практическая работа№19. Сопоставление требований и модулей

Цель

Научиться распределять требования по функциональным модулям системы.

Задачи

1. Определить функциональные модули
2. Сопоставить каждое требование с модулем
3. Найти несопоставленные

Теоретический материал

Что такое функциональный модуль:

- Логически связанная группа функций
- Имеет чёткую границу (входы/выходы)
- Может разрабатываться отдельно

Примеры модулей: Модуль авторизации, модуль каталога, модуль оплаты.

Задание

Сопоставить требования с функциональными модулями.

Пошаговое выполнение

Шаг 1. Определите 5-7 функциональных модулей.

Шаг 2. Создайте матрицу: строки — требования, столбцы — модули.

Шаг 3. Для каждого требования поставьте X в модуле, который его реализует.

Шаг 4. Проверьте: есть ли требования без модуля? Есть ли модули без требований?

Результат

Матрица соответствия «Требование → Модуль».

Критерии оценки

Критерий	Баллы

Критерий	Баллы
5-7 модулей	3
Все требования распределены	4
Нет модулей без требований	3
Итого	10

Практическая работа№20. Создание модели поведения системы

Цель

Научиться интегрировать поведенческие диаграммы в единую модель.

Задачи

1. Собрать все поведенческие диаграммы
2. Описать реакцию системы на события
3. Создать таблицу «Событие → Реакция»

Теоретический материал

Модель поведения системы описывает, как система реагирует на внешние и внутренние события.

Таблица «Событие → Реакция»:

Событие	Реакция системы	Постусловие
Пользователь нажал «Войти»	Проверка логина/пароля	Сессия создана

Задание

Создать модель поведения системы на основе собранных сценариев и диаграмм.

Пошаговое выполнение

Шаг 1. Составьте список всех событий, на которые реагирует система.

Шаг 2. Для каждого события опишите реакцию системы.

Шаг 3. Укажите, что меняется в системе после реакции.

Шаг 4. Убедитесь, что все use case'ы покрыты.

Результат

Таблица «Событие — Реакция — Постусловие» (минимум 15 строк).

Критерии оценки

Критерий	Баллы
Минимум 15 событий	4
Реакции конкретны	3

Критерий	Баллы
Постусловия измеримы	3
Итого	10

Практическая работа№21. Определение объемов данных на входе/выходе

Цель

Научиться оценивать объёмы данных для каждого интерфейса системы.

Задачи

1. Выделить все точки ввода/вывода данных
2. Для каждой определить формат и объём
3. Оценить частоту вызовов

Теоретический материал

Характеристики интерфейса:

Параметр	Описание	Пример
Тип	REST / SOAP / gRPC / файл	REST
Формат	JSON / XML / CSV / Protobuf	JSON
Размер	Ожидаемый размер данных	< 1 КБ
Частота	RPS или раз в день	10 RPS
Время ответа	Максимальное время	< 200 мс

Задание

Определить объёмы данных для всех внешних интерфейсов системы.

Пошаговое выполнение

Шаг 1. Составьте список всех интерфейсов (API, файловый обмен, UI).

Шаг 2. Для каждого заполните таблицу характеристик.

Шаг 3. Если данные приблизительные — укажите допущение.

Шаг 4. Отметьте критичные интерфейсы (высокая частота или большой объём).

Результат

Таблица интерфейсов с объёмами данных (минимум 5 интерфейсов).

Критерии оценки

Критерий	Баллы
Минимум 5 интерфейсов	4
Заполнены все колонки	3
Критичные интерфейсы отмечены	3
Итого	10

Практическая работа№22. Работа с гипотетическим заказчиком

Цель

Научиться имитировать взаимодействие с заказчиком в условиях неполной информации.

Задачи

1. Создать образ гипотетического заказчика
2. Провести ролевое интервью
3. Зафиксировать требования

Теоретический материал

Когда применяется:

- Нет доступа к реальному заказчику
- Тренировка навыков интервьюирования
- Быстрое прототипирование требований

Задание

Провести ролевое интервью с «гипотетическим заказчиком».

Пошаговое выполнение

Шаг 1. Придумайте персонажа заказчика (должность, характер, цели).

Шаг 2. Напишите краткий бриф (что нужно системе).

Шаг 3. Проведите интервью, задавая вопросы из Работы №1.

Шаг 4. Запишите ответы как если бы они были реальными.

Результат

- Описание гипотетического заказчика
- Протокол интервью

Критерии оценки

Критерий	Баллы
Образ заказчика проработан	3
Интервью логично	4
Требования реалистичны	3

Критерий	Баллы
Итого	10

Практическая работа№23. Сценарное моделирование

Цель

Научиться создавать диаграммы последовательности для ключевых сценариев.

Задачи

1. Выбрать 3 ключевых сценария
2. Построить Sequence Diagram для каждого
3. Добавить альтернативные ветвления (alt, loop)

Теоретический материал

Sequence Diagram (диаграмма последовательности):

Элемент	Обозначение
Объект	Прямоугольник с названием
Линия жизни	Пунктирная вертикальная линия
Синхронный вызов	Сплошная стрелка →
Ответ	Пунктирная стрелка →
Альтернатива	Фрагмент alt
Цикл	Фрагмент loop

Задание

Построить диаграммы последовательности для 3 ключевых сценариев.

Пошаговое выполнение

Шаг 1. Выберите 3 сценария из Работы №10.

Шаг 2. Определите участвующие объекты (Пользователь, Контроллер, Сервис, БД).

Шаг 3. В Draw.io нарисуйте Sequence Diagram.

Шаг 4. Добавьте ветвления (alt) и циклы (loop), где нужно.

Результат

3 скриншота Sequence Diagram.

Критерии оценки

Критерий	Баллы
3 диаграммы	3
Корректные объекты и линии жизни	2
Есть alt и/или loop	3
Соответствует сценарию	2
Итого	10

Практическая работа№24. Формализация альтернативных потоков данных

Цель

Научиться описывать исключительные ситуации и ветвления в сценариях.

Задачи

1. Для каждого основного сценария выявить альтернативы
2. Описать действия системы при ошибках
3. Оформить в таблицу

Теоретический материал

Типы альтернативных потоков:

Тип	Пример
Ошибка пользователя	Ввод неверного пароля
Системная ошибка	Таймаут БД
Бизнес-исключение	Недостаточно средств на счете
Ветвление	Пользователь нажал «Отмена»

Задание

Формализовать альтернативные потоки для всех сценариев.

Пошаговое выполнение

Шаг 1. Возьмите основные сценарии из Работы №10.

Шаг 2. Для каждого шага подумайте: «Что может пойти не так?»

Шаг 3. Опишите альтернативу в формате «Если X, то система делает Y».

Шаг 4. Оформите в таблицу.

Результат

Таблица:

Сценарий	Альтернатива	Действие системы	Ожидаемый результат

Критерии оценки

Критерий	Баллы
Для каждого сценария минимум 2 альтернативы	4
Альтернативы реалистичны	3
Действия системы описаны чётко	3
Итого	10

Практическая работа№25. Выявление точек интеграции

Цель

Научиться определять, с какими внешними системами должна взаимодействовать ваша ИС.

Задачи

1. Составить список внешних систем
2. Для каждой определить тип интеграции
3. Указать протокол и формат данных

Теоретический материал

Точки интеграции — внешние системы, с которыми общается ваша система:

Внешняя система	Тип интеграции	Протокол	Формат
Платёжный шлюз	Синхронный	REST HTTPS	JSON
СМС-провайдер	Асинхронный	HTTP	XML
1С	Пакетный	Файловый обмен	CSV

Задание

Выявить все точки интеграции для вашей системы.

Пошаговое выполнение

Шаг 1. Просмотрите требования и сценарии — найдите упоминания внешних систем.

Шаг 2. Добавьте типовые интеграции (email, СМС, оплата, логирование).

Шаг 3. Для каждой определите тип (синхронный/асинхронный).

Шаг 4. Укажите протокол и формат (предположительно).

Результат

Таблица точек интеграции (минимум 4).

Критерии оценки

Критерий	Баллы

Критерий	Баллы
Минимум 4 точки	4
Указаны протоколы	3
Тип интеграции корректен	3
Итого	10

Практическая работа№26. Описание функциональных блоков

Цель

Научиться декомпозировать систему на функциональные блоки.

Задачи

1. Разбить систему на 5-7 модулей
2. Для каждого описать входы и выходы
3. Указать зависимости между блоками

Теоретический материал

Функциональный блок (модуль):

Характеристика	Описание
Название	Модуль авторизации
Входные данные	Логин, пароль
Выходные данные	Токен доступа, ошибка
Зависимости	БД пользователей

Задание

Выполнить функциональную декомпозицию вашей системы.

Пошаговое выполнение

Шаг 1. Выделите 5-7 крупных модулей.

Шаг 2. Для каждого напишите краткое описание.

Шаг 3. Определите, какие данные поступают на вход модуля.

Шаг 4. Определите, какие данные выдаёт модуль.

Шаг 5. Укажите, от каких других модулей зависит этот.

Результат

Таблица функциональных блоков.

Критерии оценки

Критерий	Баллы
5-7 модулей	4

Критерий	Баллы
Входы и выходы описаны	3
Зависимости указаны	3
Итого	10

Практическая работа№27. Разработка и ревизия требований

Цель

Научиться проводить ревизию требований и управлять версиями.

Задачи

1. Внести исправления по итогам проверки
2. Обновить версию документа
3. Составить список изменений

Теоретический материал

Управление версиями требований:

Версия	Дата	Автор	Изменения
0.1	01.01	Иванов	Создан черновик
1.0	15.01	Иванов	Согласовано с заказчиком
1.1	20.01	Петров	Исправлены конфликты

Задание

Провести ревизию требований и обновить версию.

Пошаговое выполнение

Шаг 1. Соберите все замечания из чек-листа (Работа №14) и инспекции.

Шаг 2. Внесите исправления в таблицу требований.

Шаг 3. Обновите версию документа (например, 1.0 → 1.1).

Шаг 4. Создайте список изменений (Change Log).

Результат

- Обновлённая таблица требований
- Change Log

Критерии оценки

Критерий	Баллы
Все замечания исправлены	4

Критерий	Баллы
Версия обновлена	3
Change Log заполнен	3
Итого	10

Практическая работа№28. Работа с фреймами требований

Цель

Научиться использовать готовые шаблоны (фреймы) для структурирования требований.

Задачи

1. Взять фрейм Volere или FURPS+
2. Заполнить все атрибуты для каждого требования
3. Проверить полноту

Теоретический материал

Фрейм требований Volere (12 атрибутов):

1. ID
2. Название
3. Описание
4. Рациональность (зачем это нужно)
5. Источник
6. Приоритет (MoSCoW)
7. Статус
8. Критерий завершения (тест)
9. Автор
10. Дата создания
11. Версия
12. Согласование

Задание

Применить фрейм Volere к требованиям вашей системы.

Пошаговое выполнение

Шаг 1. Скопируйте список требований в новую таблицу.

Шаг 2. Добавьте 12 колонок по фрейму Volere.

Шаг 3. Заполните для каждого требования все поля.

Шаг 4. Отметьте требования, у которых пропущены важные атрибуты.

Результат

Таблица требований с 12 атрибутами Volere.

Критерии оценки

Критерий	Баллы
Использованы все 12 атрибутов	5
Заполнены для всех требований	5
Итого	10

Практическая работа№29. Подготовка презентации требований

Цель

Научиться готовить презентацию требований для заказчика и команды.

Задачи

1. Создать 10-15 слайдов
2. Выделить ключевые требования
3. Подготовить речь для защиты

Теоретический материал

Структура презентации требований:

1. Титульный слайд
2. Цель и границы проекта
3. Карта стейкхолдеров
4. Ключевые функциональные требования (Must have)
5. Диаграмма прецедентов (упрощённая)
6. План релизов (MoSCoW)
7. Риски и ограничения
8. Следующие шаги

Задание

Подготовить презентацию требований (10-15 слайдов).

Пошаговое выполнение

Шаг 1. Создайте PowerPoint/Google Slides.

Шаг 2. Сделайте титульный слайд.

Шаг 3. Добавьте слайд с целью и границами.

Шаг 4. Добавьте карту стейкхолдеров (из Работы №9).

Шаг 5. Добавьте список Must have (5-7 требований).

Шаг 6. Добавьте упрощённую диаграмму прецедентов.

Шаг 7. Добавьте план релизов (MoSCoW).

Шаг 8. Добавьте риски и следующие шаги.

Результат

Файл презентации (PPTX/**PDF).**

Критерии оценки

Критерий	Баллы
10-15 слайдов	3
Все ключевые разделы есть	4
Визуальное оформление читаемое	3
Итого	10

Практическая работа№30. Составление и согласование технического задания

Цель

Научиться оформлять итоговый документ ТЗ и проводить его согласование.

Задачи

1. Заполнить ТЗ по структуре
2. Вставить ссылки на SRS и диаграммы
3. Провести процедуру согласования

Теоретический материал

Процедура согласования ТЗ:

1. Отправить черновик заказчику
2. Получить замечания
3. Внести правки
4. Повторить (обычно 1-2 итерации)
5. Получить подпись

Задание

Составить ТЗ и провести его согласование (имитация).

Пошаговое выполнение

Шаг 1. Возьмите структуру ТЗ из Работы №15.

Шаг 2. Заполните все разделы (текст + ссылки на ранее созданные артефакты).

Шаг 3. Отправьте «заказчику» на согласование.

Шаг 4. Получите 3-5 замечаний (придумайте).

Шаг 5. Внесите правки.

Шаг 6. Получите «подпись» (скан или отметку в документе).

Результат

- Финальный ТЗ (Word/PDF)
- Протокол согласования (список замечаний и исправлений)

Критерии оценки

Критерий	Баллы
ТЗ заполнен полностью	4
Есть ссылки на SRS и диаграммы	2
Процесс согласования имитирован	2
Есть «подпись» заказчика	2
Итого	10

Практическая работа№31. Разработка диаграммы прецедентов

Цель

Научиться создавать Use Case Diagram для визуализации границ системы.

Задачи

1. Выделить актёров системы
2. Определить прецеденты
3. Построить диаграмму с include/extend

Теоретический материал

Use Case Diagram — элементы:

Элемент	Обозначение
Система	Прямоугольник с названием
Актёр	Человечек
Прецедент	Эллипс
Ассоциация	Линия между актёром и прецедентом
Include	Пунктирная стрелка с <<include>>
Extend	Пунктирная стрелка с <<extend>>

Задание

Построить диаграмму прецедентов для вашей системы.

Пошаговое выполнение

Шаг 1. Откройте Draw.io / Lucidchart / StarUML.

Шаг 2. Нарисуйте границу системы (прямоугольник).

Шаг 3. Добавьте актёров (минимум 3: пользователь, администратор, внешняя система).

Шаг 4. Добавьте прецеденты (минимум 8).

Шаг 5. Соедините актёров с прецедентами.

Шаг 6. Добавьте связи include (для общего функционала, например,

«Авторизация») и extend (для расширений).

Результат

Скриншот диаграммы прецедентов.

Критерии оценки

Критерий	Баллы
Минимум 3 актёра	2
Минимум 8 прецедентов	3
Корректные связи include/extend	3
Читаемая компоновка	2
Итого	10

Практическая работа№32. Построение диаграммы активностей

Цель

Научиться моделировать алгоритмы работы системы с помощью Activity Diagram.

Задачи

1. Выбрать ключевой сценарий
2. Построить блок-схему с ветвлениями
3. Разделить по ролям (swimlanes)

Теоретический материал

Activity Diagram — элементы:

Элемент	Обозначение
Начальный узел	●
Действие	Скруглённый прямоугольник
Решение (ветвление)	Ромб
Разделитель (swimlane)	Горизонтальная полоса
Конечный узел	● с ободком

Задание

Построить диаграмму активностей для одного ключевого сценария.

Пошаговое выполнение

Шаг 1. Выберите сценарий (например, «Оформление заказа»).

Шаг 2. Определите роли (Пользователь, Система, Внешняя система).

Шаг 3. Нарисуйте разделители для каждой роли.

Шаг 4. Расположите действия в хронологическом порядке.

Шаг 5. Добавьте ромбы для ветвлений (да/нет).

Шаг 6. Добавьте параллельные потоки (fork/join), если нужно.

Результат

Скриншот Activity Diagram.

Критерии оценки

Критерий	Баллы
Есть разделители по ролям (2+)	3
Есть ветвления (минимум 2 ромба)	3
Поток логичен	2
Есть начальный и конечный узлы	2
Итого	10

Практическая работа№33. Создание диаграммы состояний для объекта

Цель

Научиться моделировать жизненный цикл объекта с помощью State Machine Diagram.

Задачи

1. Выбрать ключевой объект (например, «Заказ»)
2. Определить все возможные состояния
3. Нарисовать переходы с событиями

Теоретический материал

State Machine Diagram — элементы:

Элемент	Обозначение
Стартовое состояние	●
Состояние	Скруглённый прямоугольник
Переход	Стрелка с подписью события
Действие при входе	entry / ...
Конечное состояние	● с ободком

Задание

Построить диаграмму состояний для ключевого объекта вашей системы.

Пошаговое выполнение

Шаг 1. Выберите объект (Заказ, Пользователь, Платёж, Бронирование).

Шаг 2. Составьте список всех состояний (минимум 5).

Шаг 3. Нарисуйте стартовое состояние.

Шаг 4. Для каждого перехода укажите событие (например, «оплатить»).

Шаг 5. Добавьте конечное состояние.

Шаг 6. При необходимости добавьте entry/exit действия.

Результат

Скриншот State Machine Diagram.

Критерии оценки

Критерий	Баллы
Минимум 5 состояний	4
Все переходы подписаны событиями	3
Нет недостижимых состояний	3
Итого	10

Практическая работа№34. Создание диаграммы классов

Цель

Научиться моделировать статическую структуру системы с помощью Class Diagram.

Задачи

1. Выделить классы из предметной области
2. Определить атрибуты и методы
3. Нарисовать связи между классами

Теоретический материал

Class Diagram — элементы:

Типы связей:

- Ассоциация ———
- Агрегация ◇———
- Композиция ◆———
- Наследование ▷———

Задание

Построить диаграмму классов для вашей системы.

Пошаговое выполнение

Шаг 1. На основе ER-диаграммы выделите классы.

Шаг 2. Для каждого класса добавьте атрибуты с типами.

Шаг 3. Добавьте методы (глаголы: создать, изменить, удалить).

Шаг 4. Нарисуйте связи между классами.

Шаг 5. Укажите множественность (1, 0.., *1*..).

Результат

Скриншот Class Diagram.

Критерии оценки

Критерий	Баллы
Минимум 5 классов	3

Критерий	Баллы
Атрибуты с типами	2
Методы соответствуют поведению	2
Связи и множественность указаны	3
Итого	10

Практическая работа№35. Разработка модели данных

Цель

Научиться создавать логическую модель данных для ИС.

Задачи

1. Определить сущности
2. Описать атрибуты и ключи
3. Указать ограничения

Теоретический материал

Модель данных — описание сущностей в табличном виде:

Сущность	Атрибут	Тип	Ключ	Ограничение
Пользователь	id	int	PK	NOT NULL
Пользователь	email	varchar(255)		UNIQUE

Задание

Разработать модель данных для вашей системы.

Пошаговое выполнение

Шаг 1. На основе ER-диаграммы (Работа №36) выпишите все сущности.

Шаг 2. Для каждой сущности перечислите атрибуты.

Шаг 3. Укажите типы данных (int, varchar, date, decimal, boolean).

Шаг 4. Определите первичные ключи (PK) и внешние (FK).

Шаг 5. Добавьте ограничения (NOT NULL, UNIQUE, CHECK).

Результат

Таблица модели данных (Excel или Word).

Критерии оценки

Критерий	Баллы
Минимум 5 сущностей	4
У каждой сущности есть PK	2

Критерий	Баллы
Типы данных указаны	2
Есть внешние ключи	2
Итого	10

Практическая работа№36. Проектирование ER-диаграммы

Цель

Научиться визуализировать модель данных в нотации «сущность-связь».

Задачи

1. Нарисовать сущности и атрибуты
2. Обозначить связи (1:1, 1:M, M:N)
3. Использовать нотацию Crow's Foot

Теоретический материал

Нотация Crow's Foot (птичья лапка):

Связь	Обозначение
Один и только один	|——|
Ноль или один	○——|
Один или много	|——<
Ноль или много	○——<

Задание

Построить ER-диаграмму в нотации Crow's Foot.

Пошаговое выполнение

Шаг 1. Откройте Draw.io / dbdiagram.io / Vertabelo.

Шаг 2. Нарисуйте прямоугольники для каждой сущности.

Шаг 3. Внутри прямоугольников перечислите атрибуты.

Шаг 4. Нарисуйте линии между сущностями.

Шаг 5. Оформите концы линий по нотации Crow's Foot.

Шаг 6. Подпишите связи глаголами («имеет», «состоит из»).

Результат

Скриншот ER-диаграммы.

Критерии оценки

Критерий	Баллы
Нотация Crow's Foot соблюдена	4
Все связи подписаны	3
Атрибуты перечислены	3
Итого	10

Практическая работа№37. Нормализация таблиц до 3НФ

Цель

Научиться проверять структуру БД на нормальные формы и устранять нарушения.

Задачи

1. Проверить таблицы на 1НФ
2. Проверить на 2НФ
3. Проверить на 3НФ
4. Выполнить декомпозицию при нарушениях

Теоретический материал

Нормальные формы:

Форма	Проверка	Пример нарушения
1НФ	Нет повторяющихся групп	телефоны: "123,456" → вынести в отдельную таблицу
2НФ	Нет частичных зависимостей	В таблице с составным ключом атрибут зависит от части ключа
3НФ	Нет транзитивных зависимостей	город → индекс (индекс зависит от города, а не от ключа)

Задание

Выполнить нормализацию таблиц до 3НФ.

Пошаговое выполнение

Шаг 1. Возьмите ER-диаграмму из Работы №36.

Шаг 2. Проверьте каждую таблицу на 1НФ (нет ли столбцов с перечислением).

Шаг 3. Проверьте на 2НФ (нет ли составных ключей, где часть не используется).

Шаг 4. Проверьте на 3НФ (нет ли атрибутов, зависящих от другого неключевого атрибута).

Шаг 5. При нарушениях — декомпозируйте таблицу (разделите на 2-3

таблицы).

Результат

- Таблица с результатами проверки каждой формы
- Итоговая ER-диаграмма после нормализации (если были изменения)

Критерии оценки

Критерий	Баллы
Проверены все 3 формы	4
Нарушения найдены (если есть)	3
Декомпозиция выполнена корректно	3
Итого	10

Практическая работа№38. Определение сущностей и связей

Цель

Научиться выделять сущности предметной области из требований и сценариев.

Задачи

1. Извлечь все существительные из требований
2. Отобрать значимые сущности
3. Определить связи между ними

Теоретический материал

Как найти сущности:

- Прочитайте требования и сценарии
- Выпишите все существительные
- Уберите: системы, интерфейсы, роли пользователей (это актёры)
- Оставьте: то, о чём система хранит данные

Пример: «Пользователь создаёт задачу» → сущности: Пользователь, Задача.

Задание

Определить сущности и связи для вашей системы.

Пошаговое выполнение

Шаг 1. Просмотрите требования (Работа №3).

Шаг 2. Выпишите все существительные.

Шаг 3. Оставьте только те, о которых система должна хранить данные.

Шаг 4. Для каждой пары сущностей определите тип связи (1:1, 1:M, M:N).

Шаг 5. Оформите в таблицу.

Результат

- Список сущностей (минимум 6)
- Таблица связей

Критерии оценки

Критерий	Баллы
Минимум 6 сущностей	4
Типы связей определены верно	3
Нет лишних сущностей	3
Итого	10

Практическая работа№39. Проектирование интерфейса пользователя

Цель

Научиться создавать wireframe (черновые макеты) интерфейса.

Задачи

1. Определить ключевые экраны
2. Нарисовать wireframe каждого экрана
3. Обозначить элементы и навигацию

Теоретический материал

Wireframe — это чёрно-белая схема интерфейса без дизайна, показывающая:

- Расположение элементов (кнопки, поля, таблицы)
- Иерархию блоков
- Навигацию между экранами

Инструменты: Figma, Balsamiq, Draw.io, бумага + карандаш.

Задание

Спроектировать интерфейс пользователя (wireframe) для вашей системы.

Пошаговое выполнение

Шаг 1. Определите 3-5 ключевых экранов:

- Экран входа / регистрации
- Дашборд / главный экран
- Форма создания объекта
- Список объектов с фильтрами
- Экран деталей

Шаг 2. Для каждого экрана нарисуйте wireframe.

Шаг 3. Обозначьте все элементы (прямоугольники с подписями).

Шаг 4. Стрелками укажите навигацию между экранами.

Результат

3-5 скриншотов wireframe (или фото, если рисовали от руки).

Критерии оценки

Критерий	Баллы
Минимум 3 экрана	4
Все элементы обозначены	3
Навигация понятна	3
Итого	10

Практическая работа№40. Разработка прототипа интерфейса

Цель

Научиться создавать интерактивный прототип интерфейса (кликабельный).

Задачи

1. Перевести wireframe в цифровой вид
2. Добавить интерактивность (переходы)
3. Показать прототип «заказчику»

Теоретический материал

Интерактивный прототип — это модель интерфейса, по которой можно кликать, имитируя работу системы.

Инструменты: Figma (бесплатно), Axure, Proto.io.

Задание

Создать интерактивный прототип интерфейса в Figma.

Пошаговое выполнение

Шаг 1. Зарегистрируйтесь в Figma (бесплатно).

Шаг 2. Создайте новый файл.

Шаг 3. Нарисуйте экраны (или импортируйте wireframe из Работы №39).

Шаг 4. Для каждой кнопки настройте переход (Prototype → Interaction → Navigate to).

Шаг 5. Настройте переходы для всех ключевых сценариев.

Шаг 6. Нажмите «Present» и проверьте работу.

Результат

Ссылка на прототип в Figma (или скриншоты с режимом прототипирования).

Критерии оценки

Критерий	Баллы

Критерий	Баллы
Прототип интерактивный	4
Переходы работают	3
Покрыты ключевые сценарии	3
Итого	10

ПРИЛОЖЕНИЕ А