

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
 ФЕДЕРАЦИИ
 федеральное государственное автономное образовательное учреждение высшего
 образования
 "САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
 АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ"

Кафедра № 43

УТВЕРЖДАЮ

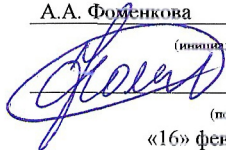
Руководитель образовательной программы

доц. к.т.н. _____

(должность, уч. степень, звание)

А.А. Фоменкова _____

(инициалы, фамилия)



(подпись)

«16» февраля 2026 г

Лист согласования рабочей программы дисциплины

Программу составил (а)

Ст. преподаватель _____ 16.02.26 _____
 (должность, уч. степень, звание) (подпись, дата)

Е.В. Павлов
 (инициалы, фамилия)

Программа одобрена на заседании кафедры № 43
 «16» февраля 2026 г, протокол № 02/2026

Заведующий кафедрой № 43

д.т.н., проф. _____ 16.02.26 _____
 (уч. степень, звание) (подпись, дата)

М.Ю. Охтилев
 (инициалы, фамилия)

Заместитель директора института №4 по методической работе

доц. к.т.н. _____ 16.02.26 _____
 (должность, уч. степень, звание) (подпись, дата)

А.А. Фоменкова
 (инициалы, фамилия)

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Основы проектной деятельности в профессии»
 (Наименование дисциплины)

Код направления подготовки/ специальности	09.03.04
Наименование направления подготовки/ специальности	Программная инженерия
Наименование направленности/ специализации	Проектирование программных систем
Форма обучения	заочная
Год приема	2026

Аннотация

Дисциплина «Основы проектной деятельности в профессии» входит в образовательную программу высшего образования – программу бакалавриата по направлению подготовки/ специальности 09.03.04 «Программная инженерия» направленности/специализации «Проектирование программных систем». Дисциплина реализуется кафедрой «№43».

Дисциплина нацелена на формирование у выпускника следующих компетенций:

УК-1 «Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач»

УК-2 «Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений»

УК-3 «Способен осуществлять социальное взаимодействие и реализовывать свою роль в команде»

УК-5 «Способен воспринимать межкультурное разнообразие общества в социально-историческом, этическом и философском контекстах»

УК-6 «Способен управлять своим временем, выстраивать и реализовывать траекторию саморазвития на основе принципов образования в течение всей жизни»

ПК-0 «Способен выстраивать и реализовывать траекторию профессионального саморазвития»

Содержание дисциплины охватывает круг вопросов, связанных с полным жизненным циклом создания программного обеспечения: от инициации, сбора требований и технико-экономического обоснования до планирования ресурсов, управления рисками и контроля качества продукта. Особое внимание уделяется изучению гибких методологий разработки (Agile, Scrum, Kanban), освоению инструментальных средств коллективной работы (таск-трекеров, систем контроля версий), а также развитию навыков командного взаимодействия, распределения профессиональных ролей и презентации результатов проектной деятельности.

Преподавание дисциплины предусматривает следующие формы организации учебного процесса: лекции, практические занятия, самостоятельная работа обучающегося.

Программой дисциплины предусмотрены следующие виды контроля: текущий контроль успеваемости, промежуточная аттестация в форме зачета (1 семестр).

Общая трудоемкость освоения дисциплины составляет 2 зачетных единицы, 72 часа.

Язык обучения по дисциплине «русский»

1. Перечень планируемых результатов обучения по дисциплине

1.1. Цели преподавания дисциплины

Целью преподавания дисциплины является формирование у студентов проектного мышления, теоретических знаний и практических умений, позволяющих успешно участвовать в реализации проектов разработки программного обеспечения средней степени сложности в условиях ограниченных ресурсов. Реализация дисциплины направлена на ознакомление обучающихся с фундаментальными принципами ведения проектной деятельности, основами планирования, а также методологией и инструментами решения проектных задач на всех стадиях жизненного цикла программного обеспечения.

1.2. Дисциплина входит в состав части, формируемой участниками образовательных отношений, образовательной программы высшего образования (далее – ОП ВО).

1.3. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения ОП ВО.

В результате изучения дисциплины обучающийся должен обладать следующими компетенциями или их частями. Компетенции и индикаторы их достижения приведены в таблице 1.

Таблица 1 – Перечень компетенций и индикаторов их достижения

Категория (группа) компетенции	Код и наименование компетенции	Код и наименование индикатора достижения компетенции
Универсальные компетенции	УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.Д.1 осуществляет анализ ситуации в реальных социальных условиях для выявления актуальной социально-значимой задачи/проблемы, требующей решения УК-1.Д.2 производит постановку проблемы путем фиксации ее содержания, выявления субъекта проблемы, а также всех заинтересованных сторон в данной ситуации УК-1.Д.3 определяет требования и ожидания заинтересованных сторон с учетом социального контекста
Универсальные компетенции	УК-2 Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений	УК-2.Д.1 вырабатывает гипотезу решения в целях реализации проекта в условиях ресурсных, нормативных и этических ограничений, регулярного проведения рефлексивных мероприятий для развития гражданственности и профессионализма участников проекта УК-2.Д.2 разрабатывает паспорт проекта с учетом компетенций студенческой команды, имеющихся ресурсов, а также самоопределения участников проекта по отношению к решаемой проблеме УК-2.Д.3 целенаправленно использует академические знания и умения для достижения целей социально-ориентированного проекта и общественного развития

<p>Универсальные компетенции</p>	<p>УК-3 Способен осуществлять социальное взаимодействие и реализовывать свою роль в команде</p>	<p>УК-3.3.1 знать основы социального взаимодействия УК-3.У.1 уметь применять нормы социального взаимодействия для реализации своей роли в команде, в том числе использовать технологии цифровой коммуникации УК-3.В.1 владеть навыками эффективного социального взаимодействия УК-3.Д.1 определяет свою позицию по отношению к поставленной в проекте проблеме, осознанно выбирает свою роль в команде УК-3.Д.2 проявляет в своем поведении способность к совместной проектной деятельности на благо общества, отдельных сообществ и граждан УК-3.Д.3 учитывает в рамках реализации проекта социальный контекст и действует с учетом своей роли в команде для достижения целей общественного развития</p>
<p>Универсальные компетенции</p>	<p>УК-5 Способен воспринимать межкультурное разнообразие общества в социально-историческом, этическом и философском контекстах</p>	<p>УК-5.Д.5 выражает свою гражданскую идентичность – принадлежность к государству, обществу, культурному и языковому пространству страны, осознает принятие на себя ответственности за будущее страны УК-5.Д.6 выражает приверженность традиционным российским ценностям, проявляет активную гражданскую позицию и гражданскую солидарность УК-5.Д.7 эффективно применяет рефлексивные практики для осмысления результатов и присвоения опыта реализации социально-ориентированных проектов; осознания взаимосвязей между академическими знаниями, гражданственности и позитивными социальными изменениями</p>
<p>Универсальные компетенции</p>	<p>УК-6 Способен управлять своим временем, выстраивать и реализовывать траекторию саморазвития на основе принципов образования в течение всей жизни</p>	<p>УК-6.3.1 знать основные приемы эффективного управления собственным временем; основные методики самоконтроля, саморазвития и самообразования УК-6.У.1 уметь управлять своим временем; ставить себе образовательные цели под возникающие жизненные задачи</p>
<p>Профессиональные компетенции</p>	<p>ПК-0 Способен выстраивать и реализовывать</p>	<p>ПК-0.3.1 знать направления профессионального развития, в том числе инновационные</p>

	траекторию профессионального саморазвития	ПК-0.У.1 уметь ставить себе образовательные цели под возникающие профессиональные задачи ПК-0.В.1 владеть инструментами различных направлений профессионального развития, в том числе цифровыми
--	---	--

2. Место дисциплины в структуре ОП

Дисциплина может базироваться на знаниях и навыках, ранее приобретенных обучающимися при освоении программ среднего образования, а также при изучении дисциплин настоящей программы бакалавриата, предусматривающих формирование универсальных компетенций в 1-2 семестрах, в том числе:

– «Основы программирования»,

Знания, полученные при изучении материала данной дисциплины, имеют как самостоятельное значение, так и используются при изучении других дисциплин:

– «Проектирование программных систем»,

– «Проектирование баз данных»,

– «Объектно-ориентированное программирование»,

– «Экономическое обоснование программных проектов»,

– «Проектирование человеко-машинного интерфейса»,

– «Производственная преддипломная практика».

3. Объем и трудоемкость дисциплины

Данные об общем объеме дисциплины, трудоемкости отдельных видов учебной работы по дисциплине (и распределение этой трудоемкости по семестрам) представлены в таблице 2.

Таблица 2 – Объем и трудоемкость дисциплины

Вид учебной работы	Всего	Трудоемкость по семестрам
		№1
1	2	3
Общая трудоемкость дисциплины, ЗЕ/ (час)	2 / 72	2 / 72
Из них часов практической подготовки		
Аудиторные занятия, всего час.	8	8
в том числе:		
лекции (Л), (час)	2	2
практические/семинарские занятия (ПЗ), (час)	6	6
лабораторные работы (ЛР), (час)		
курсовой проект (работа) (КП, КР), (час)		
экзамен, (час)		
Самостоятельная работа, всего (час)	64	64
Вид промежуточной аттестации: зачет, дифф. зачет, экзамен (Зачет, Дифф. зач, Экз.)	Зачет,	Зачет,

4. Содержание дисциплины

4.1. Распределение трудоемкости дисциплины по разделам и видам занятий.

Разделы, темы дисциплины и их трудоемкость приведены в таблице 3.

Таблица 3 – Разделы, темы дисциплины, их трудоемкость

Разделы, темы дисциплины	Лекции (час)	ПЗ (СЗ) (час)	ЛР (час)	КП/КР (час)	СР (час)
Семестр 1					
Раздел 1. Введение в ИТ-проектирование и жизненный цикл ПО Тема 1.1. Роль проектной деятельности в ИТ-индустрии; Тема 1.2. Определение ИТ-проекта, его специфика и характеристики; Тема 1.3. Полный жизненный цикл разработки программного обеспечения (SDLC); Тема 1.4. Классификация ИТ-проектов и модели разработки (Waterfall vs. Agile).	1	2	–	–	4
Раздел 2. Инициация ИТ-проекта и управление требованиями Тема 2.1. Инициация проекта и технико-экономическое обоснование (ТЕО) ИТ-продукта. Тема 2.2. Методы сбора, анализа и документирования требований к ПО; Тема 2.3. Артефакты и модели требований SRS (Software Requirements Specification): диаграммы Use Case, описание вариантов использования, ERD, Data Dictionary, et cetera; Тема 2.4. Декомпозиция целей: создание иерархической структуры работ (WBS) и бэклога продукта; Тема 2.5. Критерии успешности и причины неудач проектов по разработке ПО.	1	3	–	–	20
Раздел 3. Гибкие методологии управления ИТ-проектами (Agile) Тема 3.1. Манифест Agile: ценности и принципы гибкой разработки; Тема 3.2. Описание взаимодействия пользователя и системы: пользовательские истории (User Story) и критерии приёмки; Тема 3.3. Фреймворк Scrum: роли, артефакты, события и процессы; Тема 3.4. Метод Kanban: визуализация потока работ, управление ограничениями (WIP-лимиты); Тема 3.5. Инструментальные средства коллективной работы: настройка и использование таск-трекеров (Jira, Kaiten, Битрикс24, et cetera).	–	1	–	–	12

Раздел 4. Проектная команда и инженерные инструменты Тема 4.1. Профессиональные роли в ИТ-команде (Product Owner, Scrum Master, Developers, QA); Тема 4.2. Психология командного взаимодействия и методы разрешения конфликтов в ИТ; Тема 4.3. Инструменты контроля версий (Git, GitHub, GitLab) в совместной работе; Тема 4.4. Планирование ресурсов и оценка трудозатрат на разработку.	-	-	-	-	12
Раздел 5. Управление рисками и контроль качества в ИТ-проектах Тема 5.1. Понятие ИТ-рисков, их классификация и методы минимизации; Тема 5.2. Практика применения SWOT- и PEST-анализа для ИТ-продуктов; Тема 5.3. Основы обеспечения и контроля качества продукта (QA/QC); Тема 5.4. Выработка и проверка продуктовых гипотез (MVP, CustDev).	-	-	-	-	8
Раздел 6. Оценка результатов и презентация ИТ-проекта Тема 6.1. Порядок проведения экспертизы и приемки программных продуктов; Тема 6.2. Техническая и проектная отчетность как форма фиксации результатов; Тема 6.3. Презентация (питчинг) ИТ-проекта и демонстрация прототипов (Demo).	-	-	-	-	8
Итого в семестре:	2	6	-	-	64
Итого	2	6	0	0	64

Практическая подготовка заключается в непосредственном выполнении обучающимися определенных трудовых функций, связанных с будущей профессиональной деятельностью.

4.2. Содержание разделов и тем лекционных занятий.

Содержание разделов и тем лекционных занятий приведено в таблице 4.

Таблица 4 – Содержание разделов и тем лекционного цикла

Номер раздела	Название и содержание разделов и тем лекционных занятий
Раздел 1. Введение в ИТ-проектирование и жизненный цикл ПО	Тема 1.1. Роль проектной деятельности в ИТ-индустрии Место проектного управления в современной цифровой экономике. Отличие проектной деятельности от операционной (Run vs. Change). Специфика ИТ-отрасли: высокая неопределенность, скорость изменений, интеллектуальная емкость. Профессиональные стандарты в

	<p>области управления проектами (PMBOK, IPMA, PRINCE2).</p> <p>Тема 1.2. Определение ИТ-проекта, его специфика и характеристики</p> <p>Классический «проектный треугольник» (рамки, бюджет, время) и его адаптация под ИТ-сферу (качество, риски, ценность). Уникальность результатов ИТ-проекта. Критерии классификации: масштаб, сложность, тип конечного продукта (программное обеспечение, инфраструктура, сервис).</p> <p>Тема 1.3. Полный жизненный цикл разработки программного обеспечения (SDLC)</p> <p>Фазы SDLC: анализ требований, проектирование (архитектура), конструирование (кодинг), тестирование и обеспечение качества, развертывание (деплой), сопровождение и утилизация. Взаимосвязь между фазами, роли и зоны ответственности на каждом этапе.</p> <p>Тема 1.4. Классификация ИТ-проектов и модели разработки (Waterfall vs. Agile)</p> <p>Сравнительный анализ предиктивного (каскадного/Waterfall) и адаптивного (гибкого/Agile) подходов. Итеративная и инкрементальная модели разработки. Критерии выбора модели в зависимости от определенности требований, стабильности технологий и готовности заказчика к вовлечению.</p>
<p>Раздел 2. Инициация ИТ-проекта и управление требованиями</p>	<p>Тема 2.1. Инициация проекта и технико-экономическое обоснование (ТЭО) ИТ-продукта</p> <p>Процесс запуска проекта. Разработка Устава проекта (Project Charter). Понятие ТЭО и бизнес-кейса (Business Case). Оценка экономической целесообразности: расчет показателей ROI, NPV, PP. Определение границ проекта (Project Scope) и ключевых стейкхолдеров.</p> <p>Тема 2.2. Методы сбора, анализа и документирования требований к ПО</p> <p>Уровни требований (бизнес-требования, пользовательские, системные). Классификация требований на функциональные и нефункциональные (NFR). Техники выявления требований:</p>

	<p>интервьюирование, анкетирование, воркшопы, метод фокус-групп, анализ документации, бенчмаркинг.</p> <p>Тема 2.3. Артефакты и модели требований SRS (Software Requirements Specification)</p> <p>Содержание: Структура и назначение стандарта SRS (IEEE 830 / ISO 29148). Инструменты моделирования требований. Визуализация: диаграммы вариантов использования (Use Case Diagrams), ER-диаграммы (модели данных). Текстовые артефакты: спецификации Use Case, словари данных (Data Dictionary), матрицы трассировки требований (RTM).</p> <p>Тема 2.4. Декомпозиция целей: создание иерархической структуры работ (WBS) и бэклога продукта</p> <p>Принципы декомпозиции «сверху вниз». Построение Иерархической структуры работ (WBS / ИСР) до уровня рабочих пакетов. Особенности декомпозиции в гибких подходах: формирование бэклога продукта (Product Backlog), декомпозиция от Эпиков (Epics) к Пользовательским историям (User Stories).</p> <p>Тема 2.5. Критерии успешности и причины неудач проектов по разработке ПО</p> <p>Метрики успешности ИТ-проекта (соблюдение дедлайнов, бюджета, соответствие ТЗ, удовлетворенность пользователей). Анализ отраслевых отчетов (например, Standish Group CHAOS Report). Типичные ошибки: «расползание» рамок (Scope Creep), коммуникационные разрывы, технологические риски.</p>
<p>Раздел 3. Гибкие методологии управления ИТ-проектами (Agile)</p>	<p>Тема 3.1. Манифест Agile: ценности и принципы гибкой разработки</p> <p>История появления и философия Agile. Разбор 4 ключевых ценностей и 12 принципов Agile Manifesto. Ограничения применения Agile. Различие между понятиями «быть Agile» (мышление/Mindset) и «делать Agile» (инструменты и практики).</p> <p>Тема 3.2. Описание взаимодействия пользователя и системы: пользовательские истории (User Story) и критерии приёмки</p>

	<p>Формат User Story (Роль — Действие — Ценность). Критерии качества историй (инвестиционный подход INVEST). Разработка критериев приемки (Acceptance Criteria) по шаблону BDD (Given-When-Then). Понятие Definition of Done (DoD) и Definition of Ready (DoR).</p> <p>Тема 3.3. Фреймворк Scrum: роли, артефакты, события и процессы</p> <p>Теория Scrum (эмпиризм, прозрачность, инспекция, адаптация). Зоны ответственности трех ролей: Владелец продукта, Scrum-мастер, Команда разработки. Спринт как базовая итерация. События Scrum: Планирование спринта, Ежедневный Scrum, Обзор спринта (Демо), Ретроспектива. Артефакты: Бэклог продукта, Бэклог спринта, Инкремент продукта.</p> <p>Тема 3.4. Метод Kanban: визуализация потока работ, управление ограничениями (WIP-лимиты)</p> <p>Философия «бережливого производства» (Lean) в ИТ. Ключевые практики Kanban: визуализация (Kanban-доска), ограничение незавершенной работы (WIP-лимиты), управление потоком (Lead Time, Cycle Time). Накопительная диаграмма потока (CFD) как аналитический инструмент.</p> <p>Тема 3.5. Инструментальные средства коллективной работы: настройка и использование таск-трекеров</p> <p>Сравнительный обзор современных систем управления задачами (Jira, Kaiten, Битрикс24, Яндекс Трекер). Настройка цифрового рабочего пространства: создание проектов, кастомизация воркфлоу (статусов задач), управление бэклогом, построение спринтов, настройка дашбордов и автоматических отчетов (Burndown Chart).</p>
<p>Раздел 4. Проектная команда и инженерные инструменты</p>	<p>Тема 4.1. Профессиональные роли в ИТ-команде</p> <p>Матрица компетенций современной ИТ-команды. Обязанности, навыки и фокус внимания ролей: Product Owner (бизнес-ценность), Scrum Master / Project Manager (процессы и фасилитация), Архитектор и Разработчики (техническая реализация), QA/QC инженеры (качество), Системные/Бизнес-аналитики (требования). Cross-functional и T-shaped специалисты.</p>

	<p>Тема 4.2. Психология командного взаимодействия и методы разрешения конфликтов в ИТ</p> <p>Модель развития группы по Такмену (формирование, бурление, нормирование, функционирование). Принципы построения самоорганизующихся команд. Пять пороков команды по Ленсиони. Классификация конфликтов в ИТ (технические споры, межличностные трения) и стратегии их разрешения по Томасу-Килманну.</p> <p>Тема 4.3. Инструменты контроля версий (Git, GitHub, GitLab) в совместной работе</p> <p>Архитектура распределенных систем контроля версий. Базовые операции Git (init, clone, add, commit, push, pull). Разветвление (branching) и слияние (merging) кода. Популярные модели ветвления (Git Flow, GitHub Flow). Процесс рецензирования кода через Pull/Merge Requests. Интеграция Git с системами управления задачами.</p> <p>Тема 4.4. Планирование ресурсов и оценка трудозатрат на разработку</p> <p>Проблема неточных оценок в ИТ. Методы экспертных оценок (Дельфи, Planning Poker в Story Points). Оценка по аналогам и параметрическая оценка (PERT-анализ трех точек). Оценка «снизу вверх» (Bottom-Up). Понятие скорости команды (Velocity) и ее использование для прогнозирования релизов.</p>
<p>Раздел 5. Управление рисками и контроль качества в ИТ-проектах</p>	<p>Тема 5.1. Понятие ИТ-рисков, их классификация и методы минимизации</p> <p>Определение проектного риска. Категории ИТ-рисков: технологические (смена стека, баги), человеческие (уход ключевых сотрудников), управленческие (срыв сроков), коммерческие. Процесс управления рисками: идентификация, качественная и количественная оценка (матрица вероятности и влияния). Стратегии реагирования: уклонение, снижение, передача, принятие. Реестр рисков.</p> <p>Тема 5.2. Практика применения SWOT- и PEST-анализа для ИТ-продуктов</p> <p>Стратегическое планирование ИТ-решений. PEST-анализ: оценка влияния внешних факторов (политических,</p>

	<p>экономических, социальных, технологических) на жизнеспособность проекта. SWOT-анализ: матричное сопоставление внутренних сильных/слабых сторон проекта с внешними возможностями/угрозами.</p> <p>Тема 5.3. Основы обеспечения и контроля качества продукта (QA/QC)</p> <p>Различие между Quality Assurance (обеспечение процессов) и Quality Control (проверка продукта). Пирамида тестирования (Unit, Integration, UI-тесты). Виды тестирования: функциональное, нагрузочное, регрессионное, юзабилити-тестирование. Жизненный цикл дефекта (баг-трекинг).</p> <p>Тема 5.4. Выработка и проверка продуктовых гипотез (MVP, CustDev)</p> <p>Концепция Lean Startup (Бережливый стартап). Формулирование продуктовых гипотез. Разработка минимально жизнеспособного продукта (MVP): типы MVP, проверка востребованности с минимальными затратами. Методология развития клиентов (Customer Development): проведение проблемных и «решенческих» интервью.</p>
<p>Раздел 6. Оценка результатов и презентация ИТ-проекта</p>	<p>Тема 6.1. Порядок проведения экспертизы и приемки программных продуктов</p> <p>Организация этапа приемо-сдаточных испытаний (ПСИ). Разработка программы и методики испытаний (ПМИ). Проведение пользовательского приемочного тестирования (UAT — User Acceptance Testing). Критерии подписания акта приемки-передачи.</p> <p>Тема 6.2. Техническая и проектная отчетность как форма фиксации результатов</p> <p>Документальное закрытие проекта. Составление итогового отчета по проекту (Project Closure Report). Фиксация извлеченных уроков (Lessons Learned). Сбор и архивация проектных артефактов (техническая документация, руководства пользователя, архитектурные схемы).</p> <p>Тема 6.3. Презентация (питчинг) ИТ-проекта и демонстрация прототипов (Demo)</p> <p>Правила подготовки эффективной презентации ИТ-решения</p>

	для разных целевых аудиторий (инвесторы, бизнес-заказчики, технические эксперты). Структура инвестиционного питча (Pitch Deck). Искусство проведения публичной демонстрации работающего продукта (Demo) и управление рисками живого показа (Live Demo).
--	---

4.3. Практические (семинарские) занятия

Темы практических занятий и их трудоемкость приведены в таблице 5.

Таблица 5 – Практические занятия и их трудоемкость

№ п/п	Темы практических занятий	Формы практических занятий	Трудоемкость, (час)	Из них практической подготовки, (час)	№ раздела дисциплины
Семестр 1					
1	Введение в ИТ-проектную деятельность и жизненный цикл ПО (SDLC)	Семинар	2	–	1
2	Документирование требований: SRS, Use Case, User Stories и критерии приёмки	Выполнение практических заданий в группах	2	–	2-3
3	Создание моделей требований: ERD, Data Dictionary, User Story Mapping	Практическое занятие	2	–	2
Всего			6		

4.4. Лабораторные занятия

Темы лабораторных занятий и их трудоемкость приведены в таблице 6.

Таблица 6 – Лабораторные занятия и их трудоемкость

№ п/п	Наименование лабораторных работ	Трудоемкость, (час)	Из них практической подготовки, (час)	№ раздела дисциплины
Учебным планом не предусмотрено				
Всего				

4.5. Выполнение курсового проекта/ курсовой работы

Учебным планом не предусмотрено

4.6. Самостоятельная работа обучающихся

Виды самостоятельной работы и ее трудоемкость приведены в таблице 7.

Таблица 7 – Виды самостоятельной работы и ее трудоемкость

Вид самостоятельной работы	Всего, час	Семестр 1, час
1	2	3
Изучение теоретического материала дисциплины (ТО)	22	22
Домашнее задание, в том числе подготовка к практическим занятиям (ДЗ)	26	26
Контрольные работы заочников (КРЗ)	12	12
Подготовка к промежуточной аттестации (ПА)	4	4
Всего:	64	64

5. Перечень учебно-методического обеспечения

для самостоятельной работы обучающихся по дисциплине (модулю)

Учебно-методические материалы для самостоятельной работы обучающихся указаны в п.п. разделов 6-11.

6. Перечень печатных и электронных учебных изданий

Перечень печатных и электронных учебных изданий приведен в таблице 8.

Таблица 8 – Перечень печатных и электронных учебных изданий

Шифр / URL адрес	Библиографическая ссылка	Количество экземпляров в библиотеке (кроме электронных экземпляров)
https://urait.ru/bcode/583111	Зуб, А. Т. Управление проектами : учебник и практикум для вузов / А. Т. Зуб. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2026. — 397 с. — (Высшее образование). — ISBN 978-5-534-17500-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/583111 (дата обращения: 24.05.2026)	—
https://urait.ru/bcode/582619	Управление проектами : учебник и практикум для вузов / А. И. Балашов, Е. М. Рогова, М. В. Тихонова, Е. А. Ткаченко ; под редакцией Е. М. Роговой. — Москва : Издательство Юрайт, 2026. — 302 с. — (Высшее образование). — ISBN 978-5-534-21476-5. — Текст : электронный // Образовательная платформа Юрайт [сайт].	—

	— URL: https://urait.ru/bcode/582619 (дата обращения: 24.05.2026)	
https://urait.ru/bcode/540420	Шкурко, В. Е. Управление рисками проекта : учебник для вузов / В. Е. Шкурко ; под научной редакцией А. В. Гребенкина. — 2-е изд. — Москва : Издательство Юрайт, 2024. — 163 с. — (Высшее образование). — ISBN 978-5-534-16836-5. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/540420 (дата обращения: 24.05.2026)	—
https://urait.ru/bcode/536478	Поляков, Н. А. Управление инновационными проектами : учебник и практикум для вузов / Н. А. Поляков, О. В. Мотовилов, Н. В. Лукашов. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2024. — 384 с. — (Высшее образование). — ISBN 978-5-534-15534-1. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/536478 (дата обращения: 24.05.2026)	—
https://urait.ru/bcode/582402	Зараменских, Е. П. Управление жизненным циклом информационных систем : учебник и практикум для вузов / Е. П. Зараменских. — 4-е изд. — Москва : Издательство Юрайт, 2026. — 458 с. — (Высшее образование). — ISBN 978-5-534-21900-5. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/582402 (дата обращения: 24.05.2026)	—
https://urait.ru/bcode/590554	Грекул, В. И. Проектирование информационных систем : учебник и практикум для вузов / В. И. Грекул, Н. Л. Коровкина, Г. А. Левочкина. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2026. — 404 с. — (Высшее образование). — ISBN 978-5-534-19505-7. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/590554 (дата обращения: 24.05.2026)	—
УДК 004.415.2 ББК 30.2-5-05 П12	Проектирование программных систем: лаб. Практикум / Е.В. Павлов, Ю. Бабюк — СПб.: ГУАП, 2024. — 90 с.	50

7. Перечень электронных образовательных ресурсов информационно-телекоммуникационной сети «Интернет»

Перечень электронных образовательных ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины приведен в таблице 9.

Таблица 9 – Перечень электронных образовательных ресурсов информационно-телекоммуникационной сети «Интернет»

URL адрес	Наименование
https://elibrary.ru/defaultx.asp?	Электронная научная библиотека
https://scholar.google.ru	Google Академия
https://rospatent.gov.ru/ru	Роспатент. Федеральная служба по интеллектуальной собственности
https://urait.ru/	Образовательная платформа Юрайт
http://lms.guap.ru	Система дистанционного обучения ГУАП
http://www.edu.ru/	Федеральный портал «Российское образование»
https://www.pmssoft.ru/knowledgebase/	Группа компаний ПМСОФТ. База знаний

8. Перечень информационных технологий

8.1. Перечень программного обеспечения, используемого при осуществлении образовательного процесса по дисциплине.

Перечень используемого программного обеспечения представлен в таблице 10.

Таблица 10 – Перечень программного обеспечения

№ п/п	Наименование
	Не предусмотрено

8.2. Перечень информационно-справочных систем, используемых при осуществлении образовательного процесса по дисциплине

Перечень используемых информационно-справочных систем представлен в таблице 11.

Таблица 11 – Перечень информационно-справочных систем

№ п/п	Наименование
	Не предусмотрено

9. Материально-техническая база

Состав материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине, представлен в таблице 12.

Таблица 12 – Состав материально-технической базы

№ п/п	Наименование составной части материально-технической базы	Номер аудитории (при необходимости)
1	Учебная аудитория для проведения занятий лекционного типа — укомплектована специализированной (учебной) мебелью, набором демонстрационного оборудования и учебно-наглядными пособиями, обеспечивающими тематические иллюстрации, соответствующие рабочим учебным программам дисциплин (модулей).	—
2	Учебная аудитория для проведения практических работ — укомплектована специализированной мебелью, оснащена компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечена доступом в электронную информационно-образовательную среду ГУАП	—
3	Помещение для самостоятельной работы — укомплектовано специализированной (учебной) мебелью, оснащено компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечено доступом в электронную информационно-образовательную среду организации	—
4	Учебная аудитория для текущего контроля и промежуточной аттестации — укомплектована специализированной (учебной) мебелью, техническими средствами обучения, служащими для представления учебной информации	—

10. Оценочные средства для проведения промежуточной аттестации

10.1. Состав оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине приведен в таблице 13.

Таблица 13 – Состав оценочных средств для проведения промежуточной аттестации

Вид промежуточной аттестации	Перечень оценочных средств
Зачет	Список вопросов; Тесты;

10.2. В качестве критериев оценки уровня сформированности (освоения) компетенций обучающимися применяется 5-балльная шкала оценки сформированности компетенций, которая приведена в таблице 14. В течение семестра может использоваться 100-балльная шкала модульно-рейтинговой системы Университета, правила использования которой, установлены соответствующим локальным нормативным актом ГУАП.

Таблица 14 –Критерии оценки уровня сформированности компетенций

Оценка компетенции	Характеристика сформированных компетенций
5-балльная шкала	
«отлично» «зачтено»	Обучающийся: – глубоко и всесторонне усвоил программный материал; – уверенно, логично, последовательно и грамотно его излагает; – опираясь на знания основной и дополнительной литературы, тесно связывает усвоенные научные положения с практической деятельностью направления; – умело обосновывает и аргументирует выдвигаемые им идеи; – делает выводы и обобщения; – свободно владеет системой специализированных понятий. – правильно выполнил от 90% до 100% тестовых заданий** .
«хорошо» «зачтено»	Обучающийся: – твердо усвоил программный материал, грамотно и по существу излагает его, опираясь на знания основной литературы; – не допускает существенных неточностей; – увязывает усвоенные знания с практической деятельностью направления; – аргументирует научные положения; – делает выводы и обобщения; – владеет системой специализированных понятий. – правильно выполнил от 70% до 89% тестовых заданий** .
«удовлетворительно» «зачтено»	– обучающийся усвоил только основной программный материал, по существу излагает его, опираясь на знания только основной литературы; – допускает несущественные ошибки и неточности; – испытывает затруднения в практическом применении знаний направления; – слабо аргументирует научные положения; – затрудняется в формулировании выводов и обобщений; – частично владеет системой специализированных понятий. – правильно выполнил от 51% до 69% тестовых заданий** .
«неудовлетворительно» «не зачтено»	– обучающийся не усвоил значительной части программного материала; – допускает существенные ошибки и неточности при рассмотрении проблем в конкретном направлении; – испытывает трудности в практическом применении знаний; – не может аргументировать научные положения; – не формулирует выводов и обобщений. – правильно выполнил менее 51% тестовых заданий** .

10.3. Типовые контрольные задания или иные материалы.

Вопросы (задачи) для экзамена представлены в таблице 15.

Таблица 15 – Вопросы (задачи) для экзамена

№ п/п	Перечень вопросов (задач) для экзамена	Код индикатора
	Учебным планом не предусмотрено	

Вопросы (задачи) для зачета / дифф. зачета представлены в таблице 16.

Таблица 16 – Вопросы (задачи) для зачета / дифф. зачета

№ п/п	Перечень вопросов (задач) для зачета / дифф. зачета	Код индикатора
1	<p>УК-1.Д.1 (Анализ проблемных ситуаций на основе системного подхода)</p> <p>1) Заказчик требует внедрить новую функцию в середине спринта. Проанализируйте данный запрос с точки зрения системных связей фреймворка Scrum и опишите последствия для бэклога и фокуса команды.</p> <p>2) Объясните взаимосвязь между фазами жизненного цикла ПО (SDLC). Почему ошибка, допущенная на этапе сбора требований (SRS), системно масштабирует затраты на этапе тестирования (QA)?</p>	УК-1.Д.1
2	<p>УК-1.Д.2 (Выделение базовых составляющих системы и связей между ними)</p> <p>1) Выделите ключевые сущности для проектирования базы данных интернет-магазина и составьте логические связи между ними в формате ER-диаграммы (ERD).</p> <p>2) Какие элементы входят в структуру документа SRS (стандарт IEEE 830 / ISO 29148) и как функциональные требования связаны со словарем данных (Data Dictionary)?</p> <p>3) Опишите структуру и составные элементы пользовательской истории (User Story), и ее связь с критериями приемки (Acceptance Criteria)</p>	УК-1.Д.2
3	<p>УК-1.Д.3 (Разработка стратегии действий для решения практических задач)</p> <p>1) Сформулируйте алгоритм (стратегию) действий руководителя проекта при обнаружении критического риска ухода ведущего разработчика со стеком технологий проекта.</p> <p>2) Опишите стратегию перевода команды с каскадной модели разработки (Waterfall) на гибкую (Scrum). Какие основные барьеры придется преодолеть?</p>	УК-1.Д.3
4	<p>УК-2.Д.1 (Формулирование проектной задачи на основе поставленной цели)</p> <p>1) На основе бизнес-цели «Оптимизировать процесс технической поддержки клиентов» сформулируйте 3 функциональных требования для технического задания (SRS).</p> <p>2) Заказчик поставил цель: «Сделать приложение безопасным». Сформулируйте на основе этой цели 3 конкретных нефункциональных требования (NFR).</p>	УК-2.Д.1
5	<p>УК-2.Д.2 (Разработка концепции проекта в рамках заданных ограничений)</p> <p>1) Какие ограничения (Constraints) чаще всего накладываются на концепцию ИТ-проекта со стороны архитектуры, законодательства (например, ФЗ о персональных данных) и бюджета?</p> <p>2) Используя метод SWOT-анализа, разработайте матрицу первичной концепции для игрового мобильного приложения с высокой рыночной конкуренцией.</p>	УК-2.Д.2
6	<p>УК-2.Д.3 (Планирование ресурсов, управление рисками и контроль результатов)</p>	УК-2.Д.3

	<p>1) Составьте фрагмент Реестра рисков для ИТ-проекта (не менее 3 рисков), оценив их вероятность и влияние с помощью матрицы рисков, и укажите стратегии реагирования.</p> <p>2) Спроектируйте процесс контроля результатов спринта в Scrum. Опишите входные и выходные артефакты событий «Обзор спринта» (Демо) и «Ретроспектива».</p>	
7	<p>УК-3.3.1 (Знание ролей, методов формирования и динамики развития команды)</p> <p>1) Опишите специфику распределения ответственности между ролями Product Owner, Scrum Master и Командой разработки (Developers) во фреймворке Scrum.</p> <p>2) Раскройте суть стадий динамики развития команды по модели Б. Такмена (формирование, бурление, нормирование, функционирование) применительно к ИТ-команде.</p> <p>3) Что такое матрицы компетенций, а также кросс-функциональность и T-shaped профиль специалистов в современных продуктовых ИТ-командах?</p> <p>4) Назовите «Пять пороков команды» по Патрику Ленсиони. Какой порок является базовым и как он блокирует эффективность совместной разработки ПО?</p>	УК-3.3.1
8	<p>УК-3.У.1 (Умение эффективно взаимодействовать с членами команды и разрешать конфликты)</p> <p>1) В ИТ-команде возник технический конфликт между архитектором и ведущим разработчиком по поводу выбора СУБД. Какую стратегию по Томасу-Килманну должен применить РМ/Scrum-мастер?</p> <p>2) Разработчик регулярно срывает сроки выполнения задач в спринте, что злит команду QA. Опишите сценарий беседы с ним, используя техники конструктивной обратной связи.</p> <p>3) На этапе планирования спринта команда разделилась во мнениях по оценке сложности User Story. Продемонстрируйте, как с помощью фасилитации и Planning Poker прийти к консенсусу.</p>	УК-3.У.1
9	<p>УК-3.В.1 (Владение навыками командного управления и инструментами фасилитации)</p> <p>1) Смоделируйте проведение ежедневного Scrum-митинга (Daily Scrum). Сформулируйте три классических вопроса для каждого участника и правила тайм-боксинга.</p> <p>2) Продемонстрируйте владение процессом коллективного код-ревью (Code Review). Опишите регламент взаимодействия разработчиков через Pull/Merge Requests в GitLab/GitHub.</p>	УК-3.В.1
10	<p>УК-3.Д.1 (Демонстрация способности делегировать полномочия и распределять задачи)</p> <p>1) Как концепция самоорганизующейся команды в Agile влияет на процессы распределения задач и ответственности по сравнению с директивным стилем управления Waterfall?</p> <p>2) На основе бэклога продукта распределите задачи на ближайший спринт, учитывая показатель Velocity (скорость команды) и индивидуальные матрицы компетенций.</p>	УК-3.Д.1
11	<p>УК-3.Д.2 (Обеспечение координации действий участников команды для достижения цели)</p> <p>1) Опишите правила и культуру работы с распределенной системой контроля версий Git (правила ветвления Git Flow), которые</p>	УК-3.Д.2

	<p>координируют одновременную работу программистов над кодом.</p> <p>2) На основе накопительной диаграммы потока (CFD) в task-трекере обнаружено резкое увеличение задач в статусе «In Testing». Какие координационные действия необходимо предпринять?</p>	
12	<p>УК-3.Д.3 (Принятие ответственности за результат командной работы)</p> <p>1) Кто несет конечную ответственность перед бизнес-заказчиком за провал ИТ-проекта в Waterfall-модели и кто отвечает за недостижение цели спринта в Scrum? Разъясните разницу.</p> <p>2) Команда выпустила релиз с критическим багом, из-за чего упал продакшн-сервер клиента. Сформулируйте план разбора инцидента на Ретроспективе, исключая поиск «козла отпущения».</p> <p>3) Как понятие «Definition of Done» (DoD) формирует у ИТ-команды культуру разделяемой (коллективной) ответственности за качество поставляемого инкремента программного обеспечения?</p> <p>4) На Demo клиент остался недоволен реализованным интерфейсом. Защитите позицию команды, если требования совпали с согласованным бэклогом, но признайте зоны ответственности за UX/UI</p>	УК-3.Д.3
13	<p>УК-5.Д.5 (Учет социокультурных особенностей в ИТ-проектировании)</p> <p>1) Как культурные и национальные особенности целевой аудитории влияют на проектирование пользовательского интерфейса (UI/UX) и локализацию ИТ-продукта (на примере CustDev-интервью)?</p> <p>2) ИТ-продукт масштабируется на рынок Ближнего Востока. Какие изменения необходимо внести в требования SRS (раздел интерфейсов и макетов), учитывая социокультурную специфику региона?</p> <p>3) Сформулируйте этические и социокультурные требования к ИТ-продуктам, использующим искусственный интеллект и обработку больших пользовательских данных.</p> <p>4) При проведении проблемных интервью (CustDev) с респондентами из разных возрастных когорт (зумеры vs. бумеры) выявите различия в их паттернах взаимодействия с task-трекерами.</p>	УК-5.Д.5
14	<p>УК-5.Д.6 (Организация взаимодействия в кросс-культурных и распределенных ИТ-командах)</p> <p>1) С какими основными барьерами (языковыми, ментальными, временными зонами) сталкивается РМ при управлении распределенной ИТ-командой (офшорная разработка) и как их минимизировать?</p> <p>2) Напишите регламент коммуникаций для интернациональной команды разработки, использующей Scrum, с целью сглаживания разницы в часовых поясах и праздничных календарях.</p> <p>3) Как особенности высококонтекстных и низкоконтекстных культур (по Э. Холлу) влияют на ведение проектной переписки и постановку задач разработчикам?</p>	УК-5.Д.6
15	<p>УК-5.Д.7 (Разрешение конфликтов, обусловленных культурными различиями в проекте)</p> <p>1) Рассмотрите ситуацию, когда иностранный заказчик считает прямую критику архитектурного решения со стороны тимлида проявлением неуважения.</p>	УК-5.Д.7

	<p>Как РМ должен урегулировать данный конфликт?</p> <p>2) В распределенной команде возникло недопонимание: инженеры из одной страны строго следуют тексту ТЗ, а аналитики из другой — ждали гибкости. Сформулируйте компромиссное решение.</p> <p>3) Какие инструменты фасилитации Scrum-событий помогают нивелировать культурные различия участников в отношении публичных высказываний (например, на Ретроспективе)?</p> <p>4) Из-за разного отношения к дедлайнам в разных культурах сорван срок поставки компонента API. Сформулируйте текст письма для урегулирования ситуации без перекалывания вины.</p>	
16	<p>УК-6.3.1 (Знание принципов тайм-менеджмента и планирования профессионального роста в ИТ)</p> <p>1) Как принципы тайм-менеджмента интегрированы внутрь фреймворка Scrum (понятие тайм-боксинга событий, лимиты времени на спринт)?</p> <p>2) Какие профессиональные стандарты и квалификационные требования предъявляются к позициям Системного аналитика и Руководителя ИТ-проектов (РМ)?</p> <p>3) Что такое «технический долг» (Technical Debt) разработчика и как планирование времени на его устранение влияет на личную эффективность инженера?</p> <p>4) Раскройте понятие «выгорания» в ИТ-индустрии. Какие практики бережливой разработки (Lean/Kanban) направлены на поддержание стабильного темпа работы команды?</p>	УК-6.3.1
17	<p>УК-6.У.1 (Умение критически оценивать свои навыки и выстраивать траекторию саморазвития)</p> <p>1) На основе матрицы ролей в ИТ-команде (Тема 4.1) проведите самооценку своих текущих Hard и Soft skills. Сформируйте личный план развития (ИПР) для достижения роли Product Owner.</p> <p>2) Проанализируйте свои ошибки, допущенные в ходе командного учебного ИТ-проекта (используя подход Lessons Learned из Темы 6.2). Сформулируйте 3 вывода для будущих проектов.</p> <p>3) Задача: Вы — разработчик, застрявший на рутинных задачах. Используя технику SWOT-анализа применительно к самому себе, определите свои скрытые зоны роста в сфере инженерии (например, изучение Git/CI-CD).</p> <p>4) Защитите перед преподавателем выбор своей роли в командном проекте дисциплины, аргументировав ее соответствие вашим карьерным целям и пробелам в знаниях.</p>	УК-6.У.1
18	<p>ПК-0.3.1 (Знание методов инженерии требований, архитектурных паттернов и инструментов ИТ-проектирования)</p> <p>1) Назовите и подробно опишите три уровня требований к ПО. В чем ключевое различие между бизнес-требованиями и системными требованиями?</p> <p>2) Сравните текстовую спецификацию Use Case и Пользовательскую историю (User Story) с критериями приемки. В каких моделях жизненного цикла ПО применяется каждый из этих артефактов?</p> <p>3) Каково назначение Словаря данных (Data Dictionary) в документе SRS и из каких атрибутов состоит описание элемента данных?</p> <p>4) Какие виды тестирования входят в классическую «пирамиду</p>	ПК-0.3.1

	тестирования» и в чем различие между процессами Quality Assurance (QA) и Quality Control (QC)?	
19	ПК-0.У.1 (Умение собирать требования, проектировать модели данных и вести таск-трекинг) 1) Проведите симуляцию сбора требований для ИТ-системы «Умная парковка». Напишите 2 функциональных требования и 2 нефункциональных требования (производительность и безопасность). 2) На основе текстового описания процесса «Регистрация пользователя на вебинар» разработайте UML-диаграмму вариантов использования (Use Case Diagram). 3) Спроектируйте структуру бэклога продукта в таск-трекере (Jira/Kaiten) для CRM-системы: создайте 1 Эпик (Epic), декомпозируйте его на 3 Пользовательские истории (User Stories). 4) Напишите сценарий проведения Пользовательского приемочного тестирования (UAT) для модуля «Корзина покупок» интернет-магазина.	ПК-0.У.1
20	ПК-0.В.1 (Владение инструментами документирования требований, контроля версий и презентации ИТ-продуктов) 1) Разработайте готовый фрагмент документа SRS (раздел «Словарь данных») для сущностей «Пользователь», «Заказ» и «Платежная транзакция». 2) Продемонстрируйте владение консольными командами Git для разрешения конфликта слияния (Merge Conflict), возникшего при одновременном редактировании файла SRS двумя аналитиками. 3) Разработайте структуру и слайды презентации (Pitch Deck, не менее 5 слайдов) для защиты разработанного MVP ИТ-продукта перед инвесторами	ПК-0.В.1

Перечень тем для выполнения курсового проекта/ курсовой работы представлены в таблице 17.

Таблица 17 – Перечень тем для выполнения курсового проекта / курсовой работы

№ п/п	Примерный перечень тем для выполнения курсового проекта / курсовой работы
	Учебным планом не предусмотрено

Вопросы для проведения промежуточной аттестации в виде тестирования представлены в таблице 18.

Таблица 18 – Примерный перечень вопросов для тестов

№ п/п	Примерный перечень вопросов для тестов	Код индикатора
1	1) Проектная команда столкнулась с системной проблемой: исправление багов в коде приводит к появлению новых ошибок в смежных модулях. Что является наиболее вероятной первопричиной с точки зрения системного анализа ИТ-архитектуры? А) Отсутствие в команде роли Scrum-мастера	УК-1.Д.1

	<p>Б) Высокая зацепленность (coupling) модулей системы и отсутствие регрессионного тестирования</p> <p>В) Использование Kanban вместо Waterfall</p> <p>Г) Неправильный выбор шкалы оценки в Story Points на планировании</p> <p>2) Если на этапе сбора требований (SRS) допущена критическая ошибка в логике бизнес-процесса, на какой фазе SDLC её исправление будет стоить дороже всего согласно системной теории ЖЦ ПО?</p> <p>А) На этапе архитектурного проектирования</p> <p>Б) На этапе написания исходного кода</p> <p>В) На этапе интеграционного тестирования</p> <p>Г) На этапе промышленной эксплуатации и сопровождения</p>	
2	<p>1) При анализе структуры требований к ПО, какой элемент связывает функциональное требование системы с конкретными физическими полями в базе данных?</p> <p>А) Устав проекта (Project Charter)</p> <p>Б) Карта переходов (User Flow)</p> <p>В) Словарь данных (Data Dictionary) в SRS</p> <p>Г) Манифест Agile</p> <p>2) Изучается Use Case диаграмма. Что из перечисленного является базовой составляющей этой графической модели требований?</p> <p>А) Таблица условий и действий (Decision Table)</p> <p>Б) Актер (роль), прецедент (вариант использования) и связь между ними</p> <p>В) Класс, метод и модификатор доступа</p> <p>Г) Коммит, ветка и пулл-реквест</p>	УК-1.Д.2
3	<p>1) Какую первоочередную стратегию действий должен выбрать менеджер проекта при подтверждении риска внезапного ухода ведущего разработчика?</p> <p>А) Закрывать проект и зафиксировать убытки перед инвесторами</p> <p>Б) Активировать план реагирования: провести аудит кода, актуализировать документацию и перераспределить задачи</p> <p>В) Запретить разработчику увольняться под угрозой штрафа</p> <p>Г) Срочно перевести проект с методологии Scrum на Waterfall</p> <p>2) Какую модель разработки ПО стратегически выгоднее выбрать, если требования к продукту со стороны государственного заказчика жестко зафиксированы законодательно, а бюджет и сроки строго ограничены?</p> <p>А) Чистый Scrum с недельными спринтами</p> <p>Б) Каскадную модель (Waterfall)</p> <p>В) Экстремальное программирование (XP)</p> <p>Г) Метод Kanban без жестких дедлайнов</p>	УК-1.Д.3
4	<p>1) Какая из формулировок задач ИТ-проекта является корректно поставленной на основе бизнес-цели повышения безопасности приложения?</p> <p>А) Сделать так, чтобы хакеры не смогли нас взломать</p> <p>Б) Реализовать двухфакторную аутентификацию пользователей по SMS в соответствии с требованиями безопасности SRS к 15 ноября</p> <p>В) Улучшить код в модуле авторизации как можно скорее</p>	УК-2.Д.1

	<p>Г) Провести совещание на тему «Безопасность — наш главный приоритет»</p> <p>2) В чем заключается ключевое отличие задачи из бэклога продукта типа «User Story» от задачи из Иерархической структуры работ (WBS)?</p> <p>А) User Story описывает ценность для пользователя, а WBS — иерархический объем работ по проекту</p> <p>Б) User Story пишется только на английском языке, а WBS — на русском</p> <p>В) WBS создается программистами, а User Story — исключительно генеральным директором</p> <p>Г) WBS не имеет отношения к срокам, а User Story всегда длится ровно один день</p>	
5	<p>1) К какому квадранту PEST-анализа относится риск изменения требований к хранению персональных данных граждан в рамках разработки медицинского ИТ-приложения?</p> <p>А) P (Political / Политический или Правовой/Юридический контекст)</p> <p>Б) E (Economic / Экономический)</p> <p>В) S (Social / Социальный)</p> <p>Г) T (Technological / Технологический)</p> <p>2) При разработке концепции корпоративной системы в рамках заданных ограничений, к какому типу относится требование «Продукт должен использовать только open-source базы данных из-за санкционных рисков»?</p> <p>А) Бизнес-правило (Business Rule)</p> <p>Б) Технологическое ограничение (Constraint)</p> <p>В) Функциональное требование</p> <p>Г) Пользовательская история (User Story)</p>	УК-2.Д.2
6	<p>1) Что является главным источником данных для контроля темпов выполнения работ в Scrum-спринте в режиме реального времени?</p> <p>А) Реестр рисков проекта</p> <p>Б) Диаграмма сгорания задач (Burndown Chart)</p> <p>В) Текстовый документ SRS</p> <p>Г) SWOT-матрица продукта</p>	УК-2.Д.3
7	<p>1) Какая роль во фреймворке Scrum несет единоличную ответственность за максимизацию ценности продукта и управление бэклогом продукта (Product Backlog)?</p> <p>А) Scrum-мастер</p> <p>Б) Владелец продукта (Product Owner)</p> <p>В) Ведущий QA-инженер</p> <p>Г) Тимлид команды разработки (Lead Developer)</p> <p>2) Как называется специалист, обладающий глубокими экспертными знаниями в одной узкой технической области (например, iOS-разработка) и широким пониманием смежных дисциплин (аналитика, тестирование, дизайн)?</p> <p>А) I-shaped специалист</p> <p>Б) T-shaped специалист</p> <p>В) Проектный менеджер общего профиля</p> <p>Г) Стейкхолдер проекта</p>	УК-3.3.1

	<p>3) Согласно Патрику Ленсиони, какой порок лежит в самом основании пирамиды дисфункций команды и блокирует все остальные уровни взаимодействия?</p> <p>А) Боязнь конфликта Б) Неопределенность обязательств В) Недостаток доверия Г) Безответственность</p>	
8	<p>1) В ИТ-команде возник жесткий конфликт между разработчиком и аналитиком по поводу интерпретации требований в ТЗ. Какое действие Scrum-мастера будет наиболее конструктивным для разрешения ситуации?</p> <p>А) Приказать обоим замолчать и оштрафовать за нарушение дисциплины Б) Выступить медиатором: организовать совместную встречу, выслушать аргументы сторон и помочь найти объективное компромиссное решение В) Самостоятельно переписать код и ТЗ, не спрашивая участников конфликта Г) Игнорировать конфликт в надежде, что они разберутся сами</p> <p>2) Разработчик систематически сдает задачи тестировщикам в последний день спринта, из-за чего QA-инженеры вынуждены работать сверхурочно. Как правильно дать обратную связь разработчику на Scrum-ретроспективе?</p> <p>А) Сказать: «Ты вечно все портишь и срываешь нам спринты!» Б) Сказать через «Я-сообщение»: «Когда задачи приходят на тест в самый последний день, команда QA испытывает сильный стресс и возрастает риск пропустить баги. Давайте подумаем, как распределить нагрузку равномерно» В) Молча пожаловаться высшему руководству компании Г) Начать игнорировать разработчика на ежедневных встречах</p>	УК-3.У.1
9	<p>1) При настройке доски в таск-трекере (Jira, Kaiten) для ИТ-команды, для чего вводится инструмент WIP-лимитов (Work in Progress)?</p> <p>А) Для ограничения максимального количества задач, находящихся в конкретном статусе одновременно, с целью устранения заторов в потоке Б) Для контроля времени прихода сотрудников на работу В) Для автоматического подсчета заработной платы программистов Г) Для запрета добавления новых багов в систему</p> <p>2) Какое из правил проведения Ежедневного Scrum-митинга (Daily Scrum) является обязательным для удержания фокуса и тайм-бокса встречи?</p> <p>А) Каждый участник должен детально показать весь написанный за день код Б) Встреча длится строго не более 15 минут, участники кратко отвечают на 3 вопроса о прогрессе к Цели Спринта и возникших блокерах В) На встрече обязательно присутствует конечный заказчик и вносит правки в дизайн Г) Встреча проводится раз в неделю в конце рабочего дня</p>	УК-3.В.1

	<p>3) В рамках инженерной практики Code Review через Pull Request в Git, какое поведение проверяющего считается профессиональным и эффективным?</p> <p>А) Написание токсичных комментариев вроде «Кто это писал? Переделай всё»</p> <p>Б) Оставление аргументированных замечаний по архитектуре и стилю кода со ссылками на стандарты команды и предложением вариантов оптимизации</p> <p>В) Автоматическое одобрение (Approve) всех пулл-реквистов без чтения кода</p> <p>Г) Удаление ветки коллеги без объяснения причин</p>	
10	<p>1) Руководитель ИТ-проекта решает делегировать часть своих функций. Какую из задач эффективнее всего полностью передать лиду команды тестирования (QA Lead)?</p> <p>А) Проведение переговоров с инвесторами о раунде финансирования</p> <p>Б) Разработку стратегии тестирования продукта, выбор инструментов автоматизации и распределение задач внутри отдела QA</p> <p>В) Написание кода ядра базы данных системы</p> <p>Г) Увольнение неэффективных backend-разработчиков</p> <p>2) Как в гибких методологиях (Agile) регулируется объем задач, который команда берет в работу на итерацию (спринт)?</p> <p>А) Объем задач жестко спускается сверху генеральным директором компании</p> <p>Б) Команда самостоятельно оценивает свои возможности, опираясь на историческую скорость (Velocity), и «втягивает» задачи из бэклога</p> <p>В) На каждый спринт всегда берется ровно 50 задач независимо от их сложности</p> <p>Г) Задачи распределяются случайным образом с помощью жеребьевки</p>	УК-3.Д.1
11	<p>1) Какой инструмент координации действий команды позволяет наглядно увидеть, укладывается ли команда в план спринта или отстает от графика?</p> <p>А) ER-диаграмма данных</p> <p>Б) Диаграмма сгорания задач (Burndown Chart)</p> <p>В) Словарь данных в SRS</p> <p>Г) Устав проекта</p> <p>2) Какое правило работы с репозиторием Git (модель ветвления Git Flow) обеспечивает координацию совместных инженерных действий программистов и предотвращает затирание чужого кода?</p> <p>А) Все разработчики должны писать код одновременно прямо в ветку main (master) без создания персональных веток</p> <p>Б) Разработка ведется в изолированных ветках feature/*, а слияние в стабильные ветки происходит только после успешного Code Review и прохождения тестов</p> <p>В) Запрет на использование команды git commit чаще одного раза в месяц</p> <p>Г) Передача исходного кода друг другу на флеш-картах</p>	УК-3.Д.2

	<p>3) На Накопительной диаграмме потока (Cumulative Flow Diagram) зафиксировано резкое расширение полосы статуса «In Code Review». Какое координационное действие должна предпринять команда?</p> <p>А) Начать писать еще больше нового кода, игнорируя старый</p> <p>Б) Приостановить взятие новых задач в разработку и совместно сфокусироваться на проверке и закрытии накопившихся Pull Request'ов</p> <p>В) Удалить таск-трекер и перейти в чаты</p> <p>Г) Объявить выговор отделу тестирования</p>	
12	<p>1) В релиз на продакшн попал критический баг, заблокировавший оплату на сайте. Какое поведение лидера команды на Ретроспективе демонстрирует принятие ответственности и зрелость?</p> <p>А) Публичный поиск и увольнение виновного программиста перед лицом заказчика</p> <p>Б) Смещение фокуса на поиск системной ошибки в процессе QA: проведение анализа первопричин (Root Cause Analysis), выработка мер по автоматизации тестов и недопущению проблемы впредь без перехода на личности</p> <p>В) Игнорирование инцидента и перевод разговора на другие темы</p> <p>Г) Обвинение пользователей в том, что они неправильно нажимали кнопки</p> <p>2) Что такое «Definition of Done» (Критерии готовности) с точки зрения ответственности команды за ИТ-продукт?</p> <p>А) Список пожеланий заказчика, который можно не выполнять</p> <p>Б) Согласованный командой регламент и стандарт качества, гарантирующий, что инкремент ПО полностью протестирован, документирован и готов к деплою, за соблюдение которого команда отвечает коллективно</p> <p>В) Инструкция по включению компьютера для конечного пользователя</p> <p>Г) Документ, содержащий только подписи директоров компании</p>	УК-3.Д.3
13	<p>1) Какой аспект необходимо обязательно учесть в требованиях к интерфейсу (SRS) ИТ-продукта, если он проектируется для мусульманских стран Ближнего Востока (например, Саудовская Аравия)?</p> <p>А) Поддержку чтения и расположения элементов интерфейса справа налево (RTL — Right-to-Left)</p> <p>Б) Использование исключительно ярко-красных агрессивных цветов в оформлении кнопок</p> <p>В) Полный отказ от использования текстовой информации на экранах</p> <p>Г) Ограничение работы приложения в ночное время</p> <p>2) Какая ошибка проектирования UI/UX ИТ-платформы свидетельствует об игнорировании глобальных социокультурных стандартов доступности (Accessibility / WCAG)?</p> <p>А) Наличие англоязычной версии сайта</p> <p>Б) Отсутствие поддержки контрастной темы и экранных дикторов для слабовидящих пользователей</p> <p>В) Использование реляционных баз данных вместо нереляционных</p>	УК-5.Д.5

	<p>Г) Автоматическое сохранение черновиков писем</p> <p>3) Вы проектируете финтех-приложение для рынка Индии. На этапе анализа требований выяснилось, что значительная часть целевой аудитории пользуется бюджетными смартфонами с низкой пропускной способностью интернета. В какой раздел SRS должно быть занесено это ограничение?</p> <p>А) Бизнес-требования к окупаемости проекта Б) Нефункциональные требования (NFR) к производительности и размеру дистрибутива приложения В) Диаграмма вариантов использования (Use Case) Г) Словарь данных (Data Dictionary)</p>	
14	<p>1) Руководитель проекта управляет распределенной ИТ-командой, где backend-разработчики находятся в Новосибирске, а frontend-разработчики — в Берлине. Какое организационное решение является наиболее эффективным для синхронизации их работы?</p> <p>А) Заставить новосибирскую команду работать строго по ночам по берлинскому времени Б) Выделить фиксированное «окно пересечения» (например, с 14:00 до 16:00 по Томску/Новосибирску) для проведения общих Scrum-встреч и зафиксировать регламент асинхронного общения в Jira/Slack В) Отменить все совместные созвоны и общаться только через электронную почту раз в месяц Г) Переписать весь код проекта на один язык программирования</p> <p>2) При планировании календаря релизов в международном ИТ-проекте, что обязан учесть РМ для предотвращения срыва сроков разработки?</p> <p>А) Разницу в климатических зонах стран-участников Б) Различия в национальных и государственных праздниках и производственных календарях вовлеченных регионов В) Курс акций компании на Нью-Йоркской бирже Г) Скорость обновления браузеров у пользователей</p>	УК-5.Д.6
15	<p>1) Разработчик из низкоконтекстной культуры (привыкший к прямой критике) оставил жесткий комментарий в Pull Request коллеги из высококонтекстной культуры, который воспринял это как личное оскорбление и заблокировал работу. Каковы действия РМ по урегулированию конфликта?</p> <p>А) Уволить обоих инженеров за срыв спринта Б) Провести беседу с каждым, объяснить разницу в культурных стилях коммуникации, помочь переформулировать технические замечания в конструктивное русло и внедрить единый гайдлайн по ведению Code Review в команде В) Встать на сторону того, чей код объективно лучше написан технически Г) Запретить комментировать код в Git навсегда</p>	УК-5.Д.7
16	<p>1) Что из перечисленного является примером «технического долга» (Technical Debt), который со временем ухудшает самоорганизацию команды и замедляет разработку ПО?</p> <p>А) Кредит компании в коммерческом банке на покупку ноутбуков Б) Написание кода наспех, без соблюдения архитектурной структуры и без автотестов ради быстрого релиза, требующее</p>	УК-6.3.1

	<p>масштабного рефакторинга в будущем</p> <p>В) Отсутствие кофемашины в офисе ИТ-компании</p> <p>Г) Большое количество оплачиваемых отпусков у сотрудников</p>	
17	<p>1) Проводя личный SWOT-анализ своих навыков, к какому разделу студент должен отнести полное отсутствие практического опыта работы с системами контроля версий (Git)?</p> <p>А) S (Strengths / Сильные стороны)</p> <p>Б) W (Weaknesses / Слабые стороны)</p> <p>В) O (Opportunities / Возможности)</p> <p>Г) T (Threats / Угрозы)</p> <p>2) Студент готовит отчет о самооценке выполнения роли Системного аналитика в командном проекте. Какой показатель будет главным критерием успешности его личной работы в рамках дисциплины?</p> <p>А) Количество написанных им строк кода на языке Python</p> <p>Б) Качество, полнота и трассируемость разработанного им документа SRS (включая Use Case, ERD и Data Dictionary)</p> <p>В) Количество отправленных им сообщений в рабочий чат команды</p> <p>Г) Дизайн финальной презентации проекта для инвесторов</p>	УК-6.У.1
18	<p>1) Какое из следующих утверждений точнее всего описывает фундаментальное различие между каскадной (Waterfall) и гибкой (Agile) моделями жизненного цикла ПО?</p> <p>А) Waterfall полностью запрещает использование документации, в то время как Agile требует детального описания требований (SRS) на начальном этапе.</p> <p>Б) Waterfall предполагает последовательное прохождение фиксированных фаз с фиксацией требований, тогда как Agile ориентирован на итеративную поставку инкрементов продукта и адаптивность к изменениям.</p> <p>В) В модели Waterfall риски проекта оцениваются только на этапе закрытия, а в Agile управление рисками не требуется вовсе из-за коротких спринтов.</p> <p>Г) Модель Waterfall используется исключительно для разработки веб-сайтов, а Agile — для создания сложных наукоемких систем.</p> <p>2) Какая базовая зона ответственности закреплена за ролью Scrum Master (Скрам-мастер) во фреймворке Scrum?</p> <p>А) Управление бюджетом проекта, начисление премий и единоличное распределение задач между разработчиками.</p> <p>Б) Формирование бизнес-требований к ИТ-продукту, определение его видения и финальная приёмка готовых фич.</p> <p>В) Написание автоматизированных тестов, развертывание кода в Git и проектирование архитектуры баз данных.</p> <p>Г) Фасилитация процессов, устранение препятствий в работе команды (impediments) и помощь в понимании теории и практики Scrum.</p> <p>3) Что из перечисленного является обязательным компонентом документа «Спецификация требований к программному обеспечению» (SRS) согласно классическим стандартам системного анализа?</p> <p>А) Иерархическая структура работ (WBS) с указанием фамилий</p>	ПК-0.3.1

	<p>разработчиков и точных сроков сдачи.</p> <p>Б) Диаграммы вариантов использования (Use Case), описание функциональных требований, ER-диаграммы и словарь данных (Data Dictionary).</p> <p>В) Итоговый финансовый отчет проекта и протокол распределения прибыли (ТЕО).</p> <p>Г) Исходный код ключевых модулей системы и конфигурационные файлы CI/CD.</p>	
19	<p>1) При настройке таск-трекера (Jira, Kaiten) для ИТ-команды возникла необходимость описать требования к новой функциональности в формате Пользовательской истории (User Story). Какая из формулировок составлена верно с точки зрения проектной практики?</p> <p>А) «Запрограммировать метод POST в модуле авторизации и связать его с таблицей users в PostgreSQL».</p> <p>Б) «Я как зарегистрированный покупатель хочу восстановить пароль по email, чтобы вернуть доступ к личному кабинету в случае его утери».</p> <p>В) «Система должна отправлять технический отчет об ошибках QA-инженеру каждые 15 минут в формате JSON».</p> <p>Г) «Провести MVP и CustDev среди пользователей мобильного приложения для минимизации ИТ-рисков».</p>	ПК-0.У.1
20	<p>1) В ИТ-команде начался скрытый межличностный конфликт между ведущим разработчиком и QA-инженером. Разработчик считает, что тестировщик пишет слишком «придирчивые» баг-репорты, а тестировщик обвиняет разработчика в игнорировании критериев приемки (Acceptance Criteria). Из-за этого падает скорость поставки (Velocity). Какое управленческое решение должен принять проектный лидер?</p> <p>А) Уволить обоих сотрудников и нанять новых, чтобы не тратить время на психологию командного взаимодействия.</p> <p>Б) Организовать совместную встречу (или поднять вопрос на Ретроспективе спринта), объективно разобрать критерии готовности задач (Definition of Done) и совместно зафиксировать прозрачные правила оформления дефектов.</p> <p>В) Ввести штрафы за каждый заведенный баг и каждый просроченный коммит в GitLab.</p> <p>Г) Перевести проект на каскадную модель (Waterfall), исключив любые личные контакты между разработкой и тестированием.</p>	ПК-0.В.1

Перечень тем контрольных работ по дисциплине обучающихся заочной формы обучения, представлены в таблице 19.

Таблица 19 – Перечень контрольных работ

№ п/п	Перечень контрольных работ
1	<p>Инициация, требования и декомпозиция ИТ-проекта</p> <p>Постановка задачи: Необходимо выбрать одну предметную область из предложенного списка. Выступая в роли системного аналитика / руководителя проекта, выполнить этап инициации:</p> <ol style="list-style-type: none"> 1) Сформулировать бизнес-цели и описать краткое ТЭО (в чем ценность продукта); 2) Разработать спецификации требований (SRS): составить таблицу функциональных требований и построить диаграмму Use Case (не менее 5 прецедентов) с детальным текстовым описанием одного основного сценария. 3) Спроектировать структуру данных: построить ER-диаграмму (ERD) сущностей будущей системы и составить словарь данных. 4) Выполнить декомпозицию: построить Иерархическую структуру работ (WBS) для этапа разработки (не менее 3 уровней вложенности).
2	<p>Управление проектом по методологии Agile и проектирование команды</p> <p>Постановка задачи: В соответствии с выбранной предметной областью и результатами работы 1 необходимо перевести управление проектом на гибкую методологию (Scrum/Kanban):</p> <ol style="list-style-type: none"> 1) Сформировать Бэклог продукта (Product Backlog): составить не менее 7 User Stories с четкими критериями приемки (Acceptance Criteria) для разных ролей пользователей; 2) Распределить роли в гипотетической команде: описать, какие задачи на проекте будут выполнять Product Owner, Scrum Master, Developers и QA; 3) Создать доску проекта в любом бесплатном таск-трекере (Kaiten, Битрикс24, Яндекс Трекер, et cetera). Настроить этапы (Backlog, Todo, In Progress, QA, Done). 4) Завести созданные User Stories в трекер, декомпонировать их на технические подзадачи и настроить WIP-лимиты для этапа разработки; 5) В качестве отчета предоставить скриншоты настроенной и заполненной доски.
3	<p>Управление рисками, качеством и подготовка к защите ИТ-проекта</p> <p>Постановка задачи: Завершить проработку ИТ-проекта, подготовив его к защите и минимизировав проектные угрозы:</p> <ol style="list-style-type: none"> 1) Провести SWOT-анализ ИТ-продукта на рынке; 2) Составить Реестр рисков проекта (не менее 5 рисков, включая технические и командные). Для каждого риска определить вероятность, критичность и прописать конкретные методы минимизации; 3) Описать концепцию MVP (минимально жизнеспособного продукта): какие функции из бэклога войдут в первую версию, а какие будут отложены. Сформулировать 2 продуктовые гипотезы для проверки через CustDev; 4) Разработать презентацию (питч-дек) проекта для инвесторов или заказчиков (8-10 слайдов), включающую описание проблемы, решения, архитектуры (из KP1), плана спринтов (из KP2) и матрицы рисков.

10.4. Методические материалы, определяющие процедуры оценивания индикаторов, характеризующих этапы формирования компетенций, содержатся в локальных нормативных актах ГУАП, регламентирующих порядок и процедуру проведения текущего контроля успеваемости и промежуточной аттестации обучающихся ГУАП.

11. Методические указания для обучающихся по освоению дисциплины

11.1. Методические указания для обучающихся по освоению лекционного материала.

Основное назначение лекционного материала – логически стройное, системное, глубокое и ясное изложение учебного материала. Назначение современной лекции в рамках дисциплины не в том, чтобы получить всю информацию по теме, а в освоении фундаментальных проблем дисциплины, методов научного познания, новейших достижений научной мысли. В учебном процессе лекция выполняет методологическую, организационную и информационную функции. Лекция раскрывает понятийный аппарат конкретной области знания, её проблемы, дает цельное представление о дисциплине, показывает взаимосвязь с другими дисциплинами.

Планируемые результаты при освоении обучающимися лекционного материала:

- получение современных, целостных, взаимосвязанных знаний, уровень которых определяется целевой установкой к каждой конкретной теме;
- получение опыта творческой работы совместно с преподавателем;
- развитие профессионально-деловых качеств, любви к предмету и самостоятельного творческого мышления.
- появление необходимого интереса, необходимого для самостоятельной работы;
- получение знаний о современном уровне развития науки и техники и о прогнозе их развития на ближайшие годы;
- научиться методически обрабатывать материал (выделять главные мысли и положения, приходить к конкретным выводам, повторять их в различных формулировках);
- получение точного понимания всех необходимых терминов и понятий.

Лекционный материал может сопровождаться демонстрацией слайдов и использованием раздаточного материала при проведении коротких дискуссий об особенностях применения отдельных тематик по дисциплине.

Структура предоставления лекционного материала:

- Ответы на вопросы, возникшие при самостоятельной работе;
- Рассмотрение плана лекции;
- Устное изложение материала лекции, сопровождаемое демонстрацией презентационных материалов;
- Дискуссия с участием преподавателя и студентов по ключевым вопросам по теме лекции;
- Подведение итогов лекции и представление рекомендаций для самостоятельного изучения материала.

11.2. Методические указания для обучающихся по участию в семинарах

Основной целью для обучающегося является систематизация и обобщение знаний по изучаемой теме, разделу, формирование умения работать с дополнительными источниками информации, сопоставлять и сравнивать точки зрения, конспектировать прочитанное, высказывать свою точку зрения и т.п. В соответствии с ведущей дидактической целью содержанием семинарских занятий являются узловые, наиболее трудные для понимания и усвоения темы, разделы дисциплины. Спецификой данной

формы занятий является совместная работа преподавателя и обучающегося над решением поставленной проблемы, а поиск верного ответа строится на основе чередования индивидуальной и коллективной деятельности.

При подготовке к семинарскому занятию по теме лекции необходимо ознакомиться с планом его проведения, с литературой и научными публикациями по теме семинара.

Требования к проведению семинаров

Для прохождения курса семинарских занятий студент должен:

- знакомиться с планом проведения каждого занятия;
- перед занятием изучать необходимый теоретический материал и подготовить вопросы по теме занятия;
- принимать участие в коллективном обсуждении, заслушивании и защите докладов, разборе кейсов и дискуссии.

11.3. Методические указания для обучающихся по прохождению практических занятий

Практическое занятие является одной из основных форм организации учебного процесса, заключающаяся в выполнении обучающимися под руководством преподавателя комплекса учебных заданий с целью усвоения научно-теоретических основ учебной дисциплины, приобретения умений и навыков, опыта творческой деятельности.

Целью практического занятия для обучающегося является привитие обучающимся умений и навыков практической деятельности по изучаемой дисциплине.

Планируемые результаты при освоении обучающимся практических занятий:

- закрепление, углубление, расширение и детализация знаний при решении конкретных задач;
- развитие познавательных способностей, самостоятельности мышления, творческой активности;
- овладение новыми методами и методиками изучения конкретной учебной дисциплины;
- выработка способности логического осмысления полученных знаний для выполнения заданий;
- обеспечение рационального сочетания коллективной и индивидуальной форм обучения.

Требования к проведению практических занятий

Для прохождения курса практических занятий студент должен:

- знакомиться с планом проведения каждого занятия;
- перед каждым занятием изучать теоретический материал, необходимый для выполнения предусмотренных планом заданий, анализировать исследуемые проблемы и готовить вопросы по теме занятия;
- в установленные сроки выполнять индивидуальные практические задания и участвовать в дискуссиях и коллективном решении поставленных задач;
- следовать ходу управляемой дискуссии и указаниям преподавателя.

Задание к выполнению практической работы выдается преподавателем в соответствии с планом занятий. Темы практических работ приведены в табл. 5 данной программы.

Отчет о практической работе, если его подготовка предусмотрена в конкретной работе, должен содержать: титульный лист, введение, основную часть, заключение, список использованных источников.

На титульном листе должны быть указаны: название дисциплины, название практической работы, фамилия и инициалы преподавателя, фамилия и инициалы студента, номер его учебной группы и дата защиты работы.

Раздел «Введение» должен содержать короткое описание актуальности работы, формулировка цели и постановка задачи, индивидуальный вариант задания;

Основная часть должна содержать задание, результаты решения задач работы, расчетно-аналитические материалы (при необходимости), листинг кода/скрин экрана (при необходимости). Во всех изображениях в отчете должен быть светлый или прозрачный фон (актуально в основном для диаграмм, если таковые имеются в отчете).

Раздел «Заключение» должны содержать основные выводы по работе.

Список использованных источников должен содержать не менее 4 наименований.

Титульный лист отчета должен соответствовать шаблону, приведенному в секторе нормативной документации ГУАП <https://guap.ru/regdocs/docs/uch>

Оформление основной части отчета должно быть выполнено в соответствии с ГОСТ 7.32-2017. Требования приведены в секторе нормативной документации ГУАП <https://guap.ru/regdocs/docs/uch>

При формировании списка источников студентам необходимо руководствоваться требованиями стандарта ГОСТ 7.0.100-2018. Примеры оформления списка источников приведены в секторе нормативной документации ГУАП. <https://guap.ru/regdocs/docs/uch>

11.4. Методические указания для обучающихся по прохождению самостоятельной работы

В ходе выполнения самостоятельной работы, обучающийся выполняет работу по заданию и при методическом руководстве преподавателя, но без его непосредственного участия.

Для обучающихся по заочной форме обучения выполнение контрольных работ является элементом текущего контроля успеваемости и самостоятельной работы

В процессе выполнения самостоятельной работы у обучающегося формируется целесообразное планирование рабочего времени, которое позволяет ему развивать умения и навыки в усвоении и систематизации приобретаемых знаний, обеспечивает высокий уровень успеваемости в период обучения, помогает получить навыки повышения профессионального уровня.

Методическими материалами, направляющими самостоятельную работу обучающихся являются:

- учебно-методический материал по дисциплине;
- методические указания по выполнению контрольных работ (для обучающихся по заочной форме обучения).

Требования к структуре и оформлению отчетов по контрольным работам аналогичны требованиям к практическим работам (см. 11.3).

11.5. Методические указания для обучающихся по прохождению текущего контроля успеваемости.

Текущий контроль успеваемости предусматривает контроль качества знаний обучающихся, осуществляемого в течение семестра с целью оценивания хода освоения дисциплины.

Текущий контроль успеваемости осуществляется путем тестирования или теоретического опроса, на который отводится время на одном из занятий в середине учебного семестра, а также посредством оценки результатов выполнения практических работ. Удовлетворительным результатом прохождения контроля считается при получении не менее 50% от максимального количества баллов, которые может набрать обучающийся за отчетный период (половину семестра). Результаты текущего контроля позволяют выявить отставание от плана подготовки, но напрямую не влияют на результаты промежуточной аттестации.

11.6. Методические указания для обучающихся по прохождению промежуточной аттестации.

Промежуточная аттестация обучающихся предусматривает оценивание промежуточных и окончательных результатов обучения по дисциплине в форме зачета.

Подготовка обучающихся к зачету предполагает, как самостоятельную работу в течение семестра, так и систематизацию и закрепление знаний в дни, предшествующие зачету.

В начале освоения курса студент на основе рекомендаций преподавателя отбирает источники, которые в наибольшей степени освещают вопросы, изучение которых предусмотрено учебной программой. Рекомендуется использовать при подготовке не менее двух учебников или учебных пособий, написанных разными группами авторов. При подготовке к зачету в течение семестра студент самостоятельно изучает материал избранных источников и повторяет теорию. Ключевые вопросы, возникшие при изучении материала и подготовке к зачету, выносятся на обсуждение в часы занятий, отведенные на повторение материала и консультации. Конспекты учебного материала, подготовленные в течение семестра в ходе самостоятельной работы, используются для систематизации и закрепления знаний. Обязательным этапом подготовки к зачету является самоконтроль знаний, полученных в ходе изучения дисциплины.

В течение семестра для допуска к зачету студенту необходимо сдать не менее 50% заданий (работ). Далее студент допускается к итоговому тестированию или собеседованию. Зачет выставляется на основании выполненных в течение семестра всех заданий (работ) и прохождения тестирования/собеседования.

– Система оценок при проведении текущего контроля и промежуточной аттестации осуществляется в соответствии с руководящим документом организации РДО ГУАП. СМК 3.76 «Положение о текущем контроле успеваемости и промежуточной аттестации студентов и аспирантов, обучающихся по образовательным программам высшего образования в ГУАП» https://docs.guap.ru/guap/2020/sto_smk-3-76.pdf

Лист внесения изменений в рабочую программу дисциплины

Дата внесения изменений и дополнений. Подпись внесшего изменения	Содержание изменений и дополнений	Дата и № протокола заседания кафедры	Подпись зав. кафедрой