

Министерство науки и высшего образования Российской Федерации
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Санкт-Петербургский государственный университет промышленных технологий и дизайна»
(СПбГУПТД)

УТВЕРЖДАЮ
Первый проректор, проректор
по УР

_____ А.Е. Рудин

Рабочая программа дисциплины

Б1.В.07

Инженерия программного обеспечения

Учебный план: 2026-2027 09.03.04 ИИТА Программ инженерии №1-1-195.plx

Кафедра: **33** Цифровых и аддитивных технологий

Направление подготовки:
(специальность) 09.03.04 Программная инженерия

Профиль подготовки:
(специализация) Программная инженерия

Уровень образования: бакалавриат

Форма обучения: очная

План учебного процесса

Семестр (курс для ЗАО)		Контактная работа обучающихся		Сам. работа	Контроль, час.	Трудоёмкость, ЗЕТ	Форма промежуточной аттестации
		Лекции	Практ. занятия				
6	УП	17	34	56,75	0,25	3	Зачет
	РПД	17	34	56,75	0,25	3	
7	УП	16	32	59,75	0,25	3	Зачет
	РПД	16	32	59,75	0,25	3	
8	УП	18	36	51,75	2,25	3	Зачет, Курсовой проект
	РПД	18	36	51,75	2,25	3	
Итого	УП	51	102	168,25	2,75	9	
	РПД	51	102	168,25	2,75	9	

Рабочая программа дисциплины составлена в соответствии с федеральным государственным образовательным стандартом высшего образования по направлению подготовки 09.03.04 Программная инженерия, утверждённым приказом Минобрнауки России от 19.09.2017 г. № 920

Составитель (и):

к.т.н., доцент

Якуничева Елена
Николаевна

От кафедры составителя:

Заведующий кафедрой цифровых и аддитивных
технологий

Сошников Антон
Владимирович

От выпускающей кафедры:

Заведующий кафедрой

Сошников Антон
Владимирович

Методический отдел:

1 ВВЕДЕНИЕ К РАБОЧЕЙ ПРОГРАММЕ ДИСЦИПЛИНЫ

1.1 Цель дисциплины: Формирование у обучающихся целостного представления о процессе систематической разработки, сопровождения и эволюции программного обеспечения на основе современных методологий, стандартов и инженерных практик, обеспечивающих соответствие программных продуктов функциональным и нефункциональным требованиям, а также требованиям качества, безопасности и сопровождаемости.

1.2 Задачи дисциплины:

Сформировать знания о жизненном цикле программного обеспечения, его моделях и процессах, соответствующих международным и национальным стандартам (ISO/IEC, IEEE, ГОСТ).

Развить навыки анализа и структурирования требований заинтересованных сторон, включая функциональные и нефункциональные аспекты.

Обеспечить владение методами проектирования архитектуры программного продукта и создания технической документации (ТЗ, спецификации, диаграммы).

Отработать практики интеграции верификации и валидации в процессы разработки, включая планирование и методическое сопровождение испытаний.

Освоить применение гибких и традиционных методологий управления проектами ПО в контексте реальных бизнес-задач.

Сформировать компетенции по документированию, управлению конфигурациями и обеспечению качества в течение всего жизненного цикла ПО

1.3 Требования к предварительной подготовке обучающегося:

Предварительная подготовка предполагает создание основы для формирования компетенций, указанных в п. 2, при изучении дисциплин:

Алгоритмизация и программирование

Теория систем и системный анализ

Информационные системы и технологии

Прикладная статистика

Архитектура бизнес-системы

DevOps и производственная инфраструктура

Теория принятия решений

Учебная практика (технологическая (проектно-технологическая) практика)

Учебная практика (научно-исследовательская работа (получение первичных навыков научно-исследовательской работы))

2 КОМПЕТЕНЦИИ ОБУЧАЮЩЕГОСЯ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ

ПК-1: Способен осуществлять разработку технических спецификаций на программные компоненты и их взаимодействие
Знать: Жизненный цикл программного обеспечения, его модели и процессы; методологии разработки программного обеспечения (традиционные и гибкие); принципы управления требованиями, конфигурациями и качеством программного обеспечения.
Уметь: Применять инженерные подходы к анализу, проектированию, разработке и сопровождению программного обеспечения в соответствии с выбранной моделью жизненного цикла; осуществлять верификацию и валидацию требований на разных этапах разработки.
Владеть: Практиками документирования требований и проектирования артефактов программного обеспечения (спецификации, диаграммы, сценарии).
ПК-3: Способен осуществлять концептуально-логическое проектирование программного продукта
Знать: Методологии и процессы жизненного цикла программного обеспечения; принципы сбора, анализа и документирования требований; подходы к функциональному и нефункциональному моделированию; основы архитектурного проектирования и управления качеством на ранних этапах разработки.
Уметь: Выявлять и структурировать требования заинтересованных сторон; применять стандарты (IEEE, ISO/IEC) при составлении технических заданий; анализировать противоречия в требованиях и предлагать пути их разрешения; проектировать логическую структуру программного продукта на основе бизнес-потребностей.
Владеть: Навыками разработки концептуальных моделей программных продуктов с использованием диаграмм, построенных на универсальном языке моделирования; методами отслеживания связей между целями, требованиями и проектными решениями.
ПК-4: Способен осуществлять разработку технического задания программного продукта
Знать: Нормативные и методологические основы разработки технического задания на программные продукты, включая требования стандартов (ГОСТ 19.201, ГОСТ 34.602 и др.); этапы жизненного цикла программного обеспечения и роль технического задания в управлении проектом; принципы классификации и структурирования функциональных и нефункциональных требований.
Уметь: Выявлять, анализировать и формализовать требования к программному продукту с учетом интересов заказчика и ограничений реализации; проектировать структуру технического задания, адекватную масштабу и типу ИТ-продукта.
Владеть: Методиками документирования требований и составления разделов технического задания, обеспечивающими однозначную интерпретацию на всех этапах разработки.

ПК-5: Способен осуществлять методическое сопровождение испытаний программного продукта

Знать: Процессы жизненного цикла программного обеспечения (по ISO/IEC 12207); методологии разработки (водопадная, итеративная, гибкие); роли и артефакты, связанные с верификацией и валидацией.

Уметь: Интегрировать мероприятия по тестированию и оценке качества в общий процесс разработки; выявлять точки контроля и критические этапы, влияющие на достоверность результатов испытаний.

Владеть: Навыками проектирования процессов обеспечения качества в рамках жизненного цикла ПО; подходами к согласованию требований и критериев приемки с заинтересованными сторонами.

3 РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Наименование и содержание разделов, тем и учебных занятий	Семестр (курс для ЗАО)	Контактная работа		СР (часы)	Инновац. формы занятий	Форма текущего контроля
		Лек. (часы)	Пр. (часы)			
Раздел 1. Основы инженерии программного обеспечения.	6					О
Тема 1. Модели жизненного цикла ПО: водопадная, спиральная, итеративная. Практические занятия: Сравнение моделей жизненного цикла на примере реальных проектов.		2	4	7		
Тема 2. Стандарты инженерии ПО: ISO/IEC 12207, IEEE 830, ГОСТ 19.201, ГОСТ 34.602. Практические занятия: Классификация требований в кейсах: выделение функциональных и нефункциональных.		2	4	7		
Тема 3. Работа с аудио- и видео-моделями. Озвучка презентаций, генерация фоновой музыки для проектов. Практические занятия: Анализ стандартов: составление матрицы соответствия ГОСТ и ISO.		2	4	7		
Тема 4. Управление требованиями: сбор, анализ, приоритизация, отслеживаемость. Практические занятия: Управление требованиями в Jira/Rational DOORS/ReqView.		2	4	7		
Раздел 2. Проектирование и документирование требований						О
Тема 5. Методы моделирования требований: use-case, user stories, сценарии. Практические занятия: Разработка use-case и user stories для веб-приложения.		2	4	8		
Тема 6. Диаграммы UML для описания требований и архитектуры. Практические занятия: Построение диаграмм UML в инструментах.		2	4	8		
Тема 7. Верификация и валидация требований: методы и роли. Практические занятия: Портфолио отчёт.		2	4	7		
Тема 8. Принципы создания технического задания на ПО. Практические занятия: Составление ТЗ по шаблону ГОСТ 34.602		3	6	5,75		
Итого в семестре (на курсе для ЗАО)		17	34	56,75		
Консультации и промежуточная аттестация (Зачет)		0,25				
Раздел 3. Углублённые архитектуры	7					О

Тема 9. Архитектурные стили и паттерны: MVC, микросервисы, слоистая архитектура. Практические занятия: Применение архитектурных паттернов к заданному домену.		2	4	7		
Тема 10. Концептуальное и логическое проектирование ПО. Практические занятия: Построение логической модели на основе бизнес-процессов.		2	4	7		
Тема 11. Управление качеством на этапах проектирования: атрибуты качества.. Практические занятия: Анализ качества артефактов проектирования по ISO/IEC 25010.		2	4	7		
Тема 12. Архитектурные виды и представления. Практические занятия: Создание архитектурной документации по модели 4+1.		2	4	7		
Раздел 4. Техническое задание и управление проектом						
Тема 13. Декомпозиция требований и проектирование интерфейсов. Практические занятия: Декомпозиция функциональных требований и проектирование API.		2	4	7		О
Тема 14. Управление конфигурациями ПО: версионирование, трассировка требований. Практические занятия: Настройка системы контроля версий и отслеживания требований.		2	4	8		
Тема 15. Согласование ТЗ с заказчиком: техники коммуникации и фасилитации. Практические занятия: Моделирование диалога с заказчиком: техники выявления скрытых требований.		2	4	8		
Тема 16. Роль ТЗ в управлении проектом и контроль исполнения. Практические занятия: Согласование и защита ТЗ в формате ревью.		2	4	8,75		
Итого в семестре (на курсе для ЗАО)		16	32	59,75		
Консультации и промежуточная аттестация (Зачет)		0,25				
Раздел 5. Верификация, валидация и испытания ПО	8					О
Тема 17. Процессы верификации и валидации в жизненном цикле (по ISO/IEC 12207). Практические занятия: Построение V-модели верификации и валидации.		2	4	4		
Тема 18. Типы тестирования: функциональное, нагрузочное, регрессионное, приёмочное. Практические занятия: Разработка тест-кейсов на основе требований.		2	4	4		
Тема 19. Критерии и методики приемки ПО. Практические занятия: Составление плана приемочного тестирования.		2	4	4		

Тема 20. Планирование и организация испытаний программных продуктов. Практические занятия: Моделирование процесса испытаний: роли, артефакты, отчёты		2	4	4		
Раздел 6. Современные практики и метрики качества						0
Тема 21. Гибкие методологии (Scrum, Kanban) и их интеграция с инженерией требований. Практические занятия: Применение Scrum-артефактов к управлению требованиями.		2	4	8		
Тема 22. DevOps-подходы и непрерывная интеграция/доставка (CI/CD). Практические занятия: Настройка CI/CD pipeline с автоматизацией тестов.		2	4	8		
Тема 23. Метрики качества ПО: технические и бизнес-ориентированные Практические занятия: Расчёт метрик качества.		2	4	8		
Тема 24. Анализ и управление техническим долгом. Практические занятия: Анализ технического долга в реальном open-source проекте		4	8	11,75		
Итого в семестре (на курсе для ЗАО)		18	36	51,75		
Консультации и промежуточная аттестация (Зачет, Курсовой проект)		2,25				
Всего контактная работа и СР по дисциплине		155,75		168,25		

4 КУРСОВОЕ ПРОЕКТИРОВАНИЕ

4.1 Цели и задачи курсовой работы (проекта): Целью курсовой работы является формирование практических компетенций в области полного цикла инженерии программного обеспечения: от анализа предметной области и формализации требований до проектирования архитектуры, разработки технического задания (ТЗ) по ГОСТ, планирования верификации/валидации и интеграции современных практик (Agile, CI/CD, управление качеством).

Задачами курсового проекта являются:

- анализ моделей жизненного цикла ПО и выбор обоснованной модели для заданного проекта;
- освоение методов сбора, анализа, приоритизации и трассировки требований в соответствии со стандартами ISO/IEC 12207, IEEE 830, ГОСТ 34.602;
- применение нотаций UML и архитектурных паттернов (MVC, микросервисы, слоистая архитектура) для концептуального и логического проектирования;
- формирование полного пакета проектной документации, включая ТЗ по ГОСТ 34.602 и архитектурные представления по модели 4+1;
- проектирование стратегии верификации и валидации (V-модель), разработка тест-кейсов и плана приёмочных испытаний;
- интеграция современных практик разработки (Scrum/Kanban, CI/CD-пайплайны), расчёт метрик качества и анализ технического долга.

4.2 Тематика курсовой работы (проекта): Веб-приложение для мониторинга логистических операций в морском порту

Система управления записями и клиентской базой для сети салонов красоты
Платформа автоматизации тестирования и отчётов по качеству ПО
ИС учёта заказов, диагностики и ремонта бытовой техники
Система электронного документооборота и канцелярского учёта предприятия
Архивная ИС конструкторского бюро с системой версионирования проектной документации
Транспортно-экспедиционная система управления грузопотоками и интеграцией с внешними API
WMS-система складского учёта с поддержкой сканирования и IoT-датчиков
CRM-система автосервиса с модулем онлайн-записи, калькулятором услуг и аналитикой
Система инвентаризации, учёта и планирования обслуживания IT-оборудования
Модуль автоматизации паспортно-визового учёта и миграционных процессов
SRM-платформа управления закупками, тендерами и поставщиками
Туристическая платформа бронирования, управления турами и интеграции с платёжными системами
Система онлайн-продажи билетов, управления расписанием и пассажиропотоком автотранспорта
Клиентский портал и аналитическая подсистема коммерческого банка
Система поддержки учебного процесса, документооборота и взаимодействия кафедры ВУЗа
Лабораторная информационная система (LIS) для автоматизации медицинских анализов
Телемедицинская платформа защищённого психологического консультирования и ведения электронных карт

4.3 Требования к выполнению и представлению результатов курсовой работы (проекта):

Инструментарий: допускается использование профессиональных средств моделирования и управления проектами (PlantUML, Draw.io, Lucidchart, Enterprise Architect, Jira/YouTrack, ReqView, Git/GitLab/GitHub, Postman,

JMeter или аналоги). Презентационные материалы могут включать прототипы интерфейсов и мультимедийные элементы (при необходимости демонстрации интерактивных сценариев).

Объём и оформление: 35–40 страниц текста (без учёта приложений), шрифт Times New Roman 14 пт, межстрочный интервал 1,5, поля и оформление по ГОСТ 7.32-2017. Каждая модель, таблица и артефакт обязаны иметь номер, название, ссылку в тексте и краткое инженерное пояснение.

Обязательные артефакты в работе:

- ▶ Обоснованная модель ЖЦ и классификация требований;
- ▶ Матрица трассировки требований;
- ▶ Use-case / User Stories + UML-диаграммы (минимум 3 вида);
- ▶ Архитектурная схема по модели 4+1 с выбранным паттерном;
- ▶ ТЗ, структурированное по ГОСТ 34.602-89;
- ▶ План верификации/валидации и набор тест-кейсов (минимум 5);
- ▶ Описание CI/CD-процесса, расчёт метрик качества и анализ технического долга.

Процедура защиты:

- ▶ Доклад с использованием презентации (акцент на трассировке требований, архитектурных решениях, обосновании ТЗ по ГОСТ, стратегии тестирования и интеграции DevOps/Agile) – 10 мин;
- ▶ Ответы на вопросы по стандартам, нотациям, методам валидации, метрикам качества и управлению изменениями – 10 мин.

Критерии оценки: соответствие артефактов стандартам и нотациям РПД, логическая связность требований и архитектуры, полнота ТЗ по ГОСТ, обоснованность тестовой стратегии и практик CI/CD, качество презентации и аргументации

5. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

5.1 Описание показателей, критериев и системы оценивания результатов обучения

5.1.1 Показатели оценивания

Код компетенции	Показатели оценивания результатов обучения	Наименование оценочного средства
ПК-1	Раскрывает особенности применения моделей ЖЦ в проектных решениях. Формирует актуальную документацию по компонентам и их взаимодействию. Демонстрирует результат верификации требований на всех этапах ЖЦ программного продукта.	
ПК-4	Раскрывает структуру технического задания, соответствующую ГОСТ 34.602 и адекватное масштабу проекта. Систематизирует перечень требований на функциональные и нефункциональные. Демонстрирует шаблон протокола по результатам согласование ТЗ с заказчиком.	
ПК-5	Раскрывает алгоритм проведения испытаний программного продукта. Использует систему критериев приёмки, подтверждённую заинтересованными сторонами. Демонстрирует отчёты по тестированию программного продукта с метриками качества.	

ПК-3	Формулирует полные и структурированные требования, соответствующие IEEE/ISO. Строит концептуальные модели, отражающие бизнес-цели. Моделирует логическую архитектуру, поддерживаемую диаграмму UML	
------	--	--

5.1.2 Система и критерии оценивания

Шкала оценивания	Критерии оценивания сформированности компетенций	
	Устное собеседование	Письменная работа
5 (отлично)	Полный, исчерпывающий ответ, явно демонстрирующий глубокое понимание предмета и широкую эрудицию в оцениваемой области. Критический, оригинальный подход к материалу. Учитываются баллы, накопленные в течение семестра.	Качество исполнения всех элементов задания полностью соответствует всем требованиям. Учитываются баллы, накопленные в течение семестра.
4 (хорошо)	Ответ полный, основанный на проработке всех обязательных источников информации. Подход к материалу ответственный, но стандартный. Учитываются баллы, накопленные в течение семестра.	Задание выполнено в достаточном объеме, но ограничивается только основными подходами. Учитываются баллы, накопленные в течение семестра.
3 (удовлетворительно)	Ответ стандартный, в целом качественный, основан на всех обязательных источниках информации. Присутствуют небольшие пробелы в знаниях или несущественные ошибки. Учитываются баллы, накопленные в течение семестра.	Задание выполнено в соответствии с заданием. Имеются отдельные несущественные ошибки или отступления от правил оформления работы. Учитываются баллы, накопленные в течение семестра.
2 (неудовлетворительно)	Неспособность ответить на вопрос без помощи экзаменатора. Незнание значительной части принципиально важных элементов дисциплины. Многочисленные грубые ошибки. Не учитываются баллы, накопленные в течение семестра.	Отсутствие одного или нескольких обязательных элементов задания, либо многочисленные грубые ошибки в работе, либо грубое нарушение правил оформления или сроков представления работы. Не учитываются баллы, накопленные в течение семестра.
Зачтено	Ответ на теоретический вопрос по материалам лекций полный, с возможными несущественными ошибками. Качество исполнения всех элементов практико-ориентированного задания полностью соответствует всем требованиям. Учитываются баллы, накопленные в течение семестра.	Обучающийся своевременно выполнил практические задания. Учитываются баллы, накопленные в течение семестра.
Не зачтено	Ответ на теоретический вопрос не полный, с существенными ошибками. Не учитываются баллы, накопленные в течение семестра.	Своевременно не выполняет (выполнил частично) практические задания. Не учитываются баллы, накопленные в течение семестра.

5.2 Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности

5.2.1 Перечень контрольных вопросов

№ п/п	Формулировки вопросов
Семестр 6	
1	Что такое инженерия программного обеспечения и чем она отличается от программирования?
2	Какие цели преследует управление жизненным циклом ПО?
3	Перечислите основные этапы классической водопадной модели ЖЦ.
4	В чём заключаются преимущества и недостатки спиральной модели ЖЦ?
5	Как итеративные модели ЖЦ помогают снижать риски проекта?
6	Какие процессы определены в стандарте ISO/IEC 12207?
7	Чем отличается основной процесс от вспомогательного в ISO/IEC 12207?
8	Какие национальные стандарты регулируют разработку ПО в России?
9	Как ГОСТ 19.101 связан с этапами жизненного цикла ПО?

10	Почему важна стандартизация в инженерии ПО?
11	Какие роли участвуют в процессе разработки по модели ЖЦ?
12	Какие риски характерны для ранних этапов ЖЦ?
13	В чём разница между моделью ЖЦ и методологией разработки?
14	Как изменение требований влияет на выбор модели ЖЦ?
15	Какие артефакты создаются на этапе анализа требований?
16	Какие инструменты помогают визуализировать этапы ЖЦ?
17	Как жизненный цикл связан с управлением качеством ПО?
18	Почему важно соблюдать последовательность этапов ЖЦ даже в гибких методологиях?
19	Какие критерии используются для выбора модели ЖЦ под конкретный проект?
20	Какие проблемы возникают при отсутствии формализованной модели ЖЦ?
21	Дайте определение программным требованиям.
22	Чем функциональные требования отличаются от нефункциональных?
23	Приведите 5 примеров нефункциональных требований (производительность, безопасность и др.).
24	Какие источники могут использоваться для сбора требований?
25	В чём суть техники «интервью с заинтересованными сторонами»?
26	Какие проблемы возникают при неоднозначной формулировке требований?
27	Что такое «прослеживаемость требований» и зачем она нужна?
28	Какие инструменты позволяют управлять требованиями (перечислите 3)?
29	Какие ошибки чаще всего допускаются при документировании требований?
30	В чём преимущество use-case-подхода перед текстовым описанием?
31	Как user stories соотносятся с традиционными требованиями?
32	Какие элементы включает полноценный use-case?
33	Что такое сценарий использования и как он связан с требованиями?
34	Как диаграмма активности помогает в анализе требований?
35	Как диаграмма последовательности отражает взаимодействие с системой?
36	Какие правила записи требований предлагает стандарт IEEE 830?
37	Какие разделы содержит ТЗ по ГОСТ 34.602?
38	Как согласовать противоречивые требования от разных заинтересованных сторон?
39	Как проверить полноту требований перед переходом к проектированию?
40	Какие метрики можно использовать для оценки качества требований?
Семестр 7	
41	Какова роль технического задания в жизненном цикле ПО?
42	Как формулировать требования к внешним интерфейсам в ТЗ?
43	Какие инструменты позволяют управлять версиями ТЗ?
44	Как ТЗ влияет на оценку трудозатрат и сроков?
45	Какие последствия имеет неполное или неточное ТЗ?
46	Какие разделы ТЗ отвечают за описание нефункциональных требований?
47	Какие разделы ТЗ обязательны по ГОСТ 34.602?
48	Как ТЗ связано с договором на разработку ПО?
49	Какие ограничения должны быть отражены в ТЗ?
50	Какие методы используются для приоритизации требований в ТЗ?
51	Как декомпонировать сложные требования на подтребования?
52	Как обеспечить трассируемость от ТЗ к реализации?
53	Как согласовать ТЗ с заказчиком в условиях неопределённости?
54	Какие требования должны быть проверяемыми?
55	Как избежать двусмысленности при описании функций в ТЗ?
56	Какие артефакты помогают визуализировать требования из ТЗ?
57	Как учесть требования законодательства (например, ФЗ-152) в ТЗ?
Семестр 8	
58	Чем верификация отличается от валидации? Приведите примеры. Какие процессы верификации и валидации определены в ISO/IEC 12207?
59	Что такое V-модель разработки и как она связана с тестированием? Какие цели преследует функциональное тестирование?
60	Какие виды нефункционального тестирования вы знаете? Что включает в себя приёмочное тестирование (UAT)?

61	Какие критерии приёмки используются в agile-проектах? Как связать каждое требование с соответствующим тест-кейсом?
62	Что такое матрица трассируемости требований и тестов? Какие метрики применяются для оценки полноты тестирования?
63	Как организовать процесс ревью требований с участием тестировщиков? Какие роли участвуют в проведении испытаний ПО?
64	Какие артефакты создаются в ходе испытаний? Как документировать дефекты и их устранение?
65	Какие инструменты автоматизации тестирования вы знаете? Как планировать нагрузочное тестирование?
66	Как обеспечить воспроизводимость результатов испытаний? Какие риски возникают при отсутствии формализованного процесса валидации?
67	Как валидация связана с удовлетворённостью заказчика? Как интегрировать тестирование в непрерывный процесс разработки (CI)?
68	Какие принципы лежат в основе гибких методологий (Agile Manifesto)? Как Scrum интегрируется с инженерией требований?
69	Какие артефакты Scrum используются для управления требованиями? Как Kanban помогает визуализировать поток требований?
70	Что такое DevOps и как он влияет на инженерию ПО? Какие практики CI/CD способствуют повышению качества ПО?
71	Как автоматизация тестирования вписывается в CI/CD pipeline? Какие метрики качества относятся к уровню кода?
72	Какие метрики качества ориентированы на пользователя (SLO/SLI)? Что такое технический долг и как он влияет на развитие ПО?
73	Какие инструменты позволяют измерять технический долг? Какие стратегии используются для снижения технического долга?
74	Как метрики производительности интегрируются в мониторинг? Какие метрики важны на этапе эксплуатации ПО?
75	Как обеспечить соответствие ПО стандартам в условиях быстрой разработки? Какие практики документирования актуальны в DevOps-среде?
76	Как культура непрерывного улучшения влияет на качество ПО?
77	Какие метрики используются для оценки зрелости процессов разработки (CMMI)?
78	Как управлять качеством в распределённых командах?

5.2.2 Типовые тестовые задания

5.2.3 Типовые практико-ориентированные задания (задачи, кейсы)

Составить матрицу соответствия требований ГОСТ 19.201 и ISO/IEC/IEEE 29148.

Построить диаграмму последовательности для сценария авторизации.

Составить спецификацию интерфейсов пользователя и внешних систем.

Смоделировать сценарий приемочного тестирования с участием заказчика

5.3 Методические материалы, определяющие процедуры оценивания знаний, умений, владений (навыков и (или) практического опыта деятельности)

5.3.1 Условия допуска обучающегося к промежуточной аттестации и порядок ликвидации академической задолженности

Проведение промежуточной аттестации регламентировано локальным нормативным актом СПбГУПТД «Положение о проведении текущего контроля успеваемости и промежуточной аттестации обучающихся»

5.3.2 Форма проведения промежуточной аттестации по дисциплине

Устная

☐ +

Письменная

☐

Компьютерное тестирование

☐

Иная

☐

5.3.3 Особенности проведения промежуточной аттестации по дисциплине

Время на подготовку к устному собеседованию составляет 20 минут.

Защита курсового проекта:

1. Доклад - 8 минут

2. Ответы на вопросы - 2 минуты

6. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

6.1 Учебная литература

Автор	Заглавие	Издательство	Год издания	Ссылка
6.1.1 Основная учебная литература				

Сузи, Р. А.	Язык программирования Python	Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа	2020	https://www.iprbooks.hop.ru/97589.html
Доррер, Г. А.	Методология программной инженерии	Красноярск: Сибирский государственный университет науки и технологий имени академика М.Ф. Решетнева	2021	https://www.iprbooks.hop.ru/116641.html
Немков, Р. М.	Генетические алгоритмы	Ставрополь: Северо-Кавказский федеральный университет	2021	https://www.iprbooks.hop.ru/135677.html
Мейер, Б.	Объектно-ориентированное программирование и программная инженерия	Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа	2024	https://www.iprbooks.hop.ru/133956.html
Николюкин, М. С., Конкина, В. В., Патулин, К. И.	Тестирование программного обеспечения	Тамбов: Тамбовский государственный технический университет, ЭБС АСВ	2025	https://www.iprbooks.hop.ru/154963.html
Самочадин, А. В., Самочадина, Т. Н.	Оценка трудоемкости разработки программного обеспечения	Санкт-Петербург: Санкт-Петербургский политехнический университет Петра Великого	2023	https://www.iprbooks.hop.ru/147730.html
Воронина, В. В., Эгов, Е. Н.	Отладка программных решений	Ульяновск: Ульяновский государственный технический университет	2024	https://www.iprbooks.hop.ru/149282.html

6.1.2 Дополнительная учебная литература

Плас Дж. Вандер	Python для сложных задач: наука о данных и машинное обучение	Санкт-Петербург: Питер	2018	https://ibooks.ru/read ing.php? short=1&productid=356721
Элбон Крис	Машинное обучение с использованием Python. Сборник рецептов: Пер. с англ.	Санкт-Петербург: БХВ-Петербург	2019	https://ibooks.ru/read ing.php? short=1&productid=366635

6.2 Перечень профессиональных баз данных и информационно-справочных систем

Электронно-библиотечная система iBooks.ru [Электронный ресурс]. URL: <https://ibooks.ru/>
 Электронно-библиотечная система IPRbooks [Электронный ресурс]. URL: <https://www.iprbookshop.ru/>
 Платформа для проведения соревнований по Data Science [Электронный ресурс]. URL: <https://www.kaggle.com>

6.3 Перечень лицензионного и свободно распространяемого программного обеспечения

MicrosoftOfficeProfessional
 Microsoft Windows
 Microsoft Visual Studio Code
 Microsoft Visual C++ 2010 Express
 Notepad++
 Python
 Figma

6.4 Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

Аудитория	Оснащение
Компьютерный класс	Мультимедийное оборудование, компьютерная техника с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду
Лекционная аудитория	Мультимедийное оборудование, специализированная мебель, доска