

Федеральное государственное образовательное бюджетное учреждение
высшего образования

**«ФИНАНСОВЫЙ УНИВЕРСИТЕТ ПРИ ПРАВИТЕЛЬСТВЕ
РОССИЙСКОЙ ФЕДЕРАЦИИ»
(Финансовый университет)**

**Кафедра информационных технологий
Факультет информационных технологий и анализа больших данных**

Документ подписан усиленной неквалифицированной электронной подписью
Организация: Финансовый университет при Правительстве РФ
Утверждено: Проректор по учебной и методической работе Е.А. Каменева
Сертификат: 2L2J6Nafkj6PO8O0tOGBG6Wv9ZeJLkYr
Дата: 13.10.2025 г.

И. И. Хасанов

Программная инженерия

Рабочая программа дисциплины

для студентов, обучающихся по направлению подготовки

09.03.04 - Программная инженерия

Образовательная программа:

Инженер-разработчик программного обеспечения

Профиль

Инженер-разработчик программного обеспечения

Рекомендовано

*Факультет информационных технологий и анализа больших данных
(протокол №2 от 18.11.2025)*

Одобрено

*Кафедра информационных технологий
(протокол №11 от 12.11.2025)*

Москва 2025

Содержание

Наименование разделов РПД		Стр.
1.	Наименование дисциплины	5
2.	Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине	5
3.	Место дисциплины в структуре образовательной программы	10
4.	Объем дисциплины(модуля) в зачетных единицах и в академических часах с выделением объема аудиторной (лекции, семинары) и самостоятельной работы обучающихся	10
5.	Содержание дисциплины, структурированное по темам (разделам) дисциплины с указанием их объемов (в академических часах) и видов учебных занятий	11
5.1.	Содержание дисциплины	11
5.2.	Учебно-тематический план	14
5.3.	Содержание семинаров, практических занятий	16
6.	Перечень учебно-методического обеспечения для	19

	самостоятельной работы обучающихся по дисциплине	
6.1.	Перечень вопросов, отводимых на самостоятельное освоение дисциплины, формы внеаудиторной самостоятельной работы	19
6.2.	Перечень вопросов, заданий, тем для подготовки к текущему контролю	22
7.	Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине	24
8.	Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины	36
9.	Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины	38
10.	Методические указания для обучающихся по освоению дисциплины	39
11.	Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень необходимого программного обеспечения и информационных	41

	справочных систем	
12.	Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине	42

1. Наименование дисциплины

«Программная инженерия».

2. Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине

Код компетенции	Наименование компетенции	Индикатор достижения компетенции	Результаты обучения
ОПК-4	Способен участвовать в разработке стандартов, норм и правил, а также технической документации, связанной с профессиональной деятельностью	Демонстрирует знания основных стандартов ведения технической документации, как отечественных, так и зарубежных	Знать: различные подходы к формированию требований к программному обеспечению Уметь: выбирать подходящую методологию формирования требования, адаптировать под предметную область
		Создает комплект программной и проектной документации к разрабатываемой информационно й системе	Знать: типы документов, их назначение и содержание Уметь: выбирать номенклатуру документации соответствующую типу проекта, формировать документы в соответствии с отраслевыми стандартами
		Критически анализирует программную и проектную	Знать: основные методы критического

		документацию, строит на ее основе логические выводы	анализа программной и проектной документации Уметь: демонстрировать умение критически оценивать и дорабатывать программную и проектную документацию
ПКП-3	Способность собирать, анализировать и управлять требованиями к программной системе	Понимает виды требований к программному обеспечению, составляет формализованное описание требований по нетехническому описанию функционала программной системы	Знать: методики работы с требованиями позволяющие из требований формировать задачи на разработку Уметь: планировать и вести работу с заинтересованными лицами, выделять архитектурно значимые требования и исходя из них формировать и обосновывать архитектурные решения
		Анализирует программную систему на предмет соответствия требованиям, выявляет проблемные участки, модифицирует архитектуру и дизайн	Знать: виды требований к программному обеспечению и их особенности Уметь: анализировать и модифицировать архитектуру и дизайн программной системы в

		программной системы для полного соответствия набору требований	соответствии с требованиями
		Анализирует описанную систему требований к программной системе - выделяет требования высшего и низшего уровня, проводит декомпозицию требований, соотносит их с архитектурным и решениями программной системы, приоретизирует требования, оценивает реализуемость и тестируемость требований, составляет по системе требований протокол тестирования программной системы	Знать: методы анализа программной системы на соответствие требованиям и выявление проблемных участков. Уметь: выделять, декомпозировать, приоретизировать требования, оценивать их реализуемость и тестируемость
		Критически относится к сформулированным требованиям к программной системе - выявляет внутренние противоречия, риски, связанные с реализацией тех	Знать: критическое отношение к сформулированным требованиям, выявление противоречий, рисков и недостающих требований Уметь: находить

		или иных требований, воспринимает требования в более широком контексте постановки задачи, находит недостающие требования	внутренние противоречия, риски и недостающие требования в сформулированных требованиях к программной системе
		Самостоятельно формулирует подробную, формальную, полную и реализуемую систему требований к программной системе и описывает ее в технической документации	Знать: подробную, формальную, полную и реализуемую систему требований к программной системе Уметь: описывать ее в технической документации
ПКП-5	Способность проектировать и реализовывать интеллектуальные информационные системы	Демонстрирует знания основных методов машинного обучения и интеллектуального анализа данных, применяет готовые инструменты для создания интеллектуальных алгоритмов	Знать: основные методы машинного обучения, включая обучение с учителем, обучение без учителя, обучение с подкреплением, а также основы интеллектуального анализа данных, таких как обработка больших данных, статистический анализ и т.д. Уметь: применять методы машинного обучения и интеллектуальн

			ого анализа данных на практике
		Понимает особенности интеллектуальн ых информационны х систем в части операций разработки, развертывания и сопровождения	Знать: операции разработки, развертывания и сопровождения интеллектуальн ых информационны х систем Уметь: разрабатывать, развертывать и обслуживать интеллектуальн ые информационны е системы
		Адаптирует практики создания программных продуктов, в том числе командные, для интеллектуальн ых информационны х систем	Знать: практики создания программных продуктов, включая командные методы, применяемые в интеллектуальн ых информационны х системах Уметь: адаптировать практики создания программных продуктов для интеллектуальн ых информационны х систем
		Организовывает сбор и подготовку данных для	Знать: гибкие методологии управления разработкой

		систем машинного обучения, в том числе поточковых, онлайн обучения	Уметь: участвовать в процессе управления работами, планировать работы по гибким методологиям и контролировать ход их выполнения
--	--	--	---

3. Место дисциплины в структуре образовательной программы

Дисциплина «Программная инженерия» относится к «Профилю "Инженер-разработчик программного обеспечения"»

4. Объем дисциплины(модуля) в зачетных единицах и в академических часах с выделением объема аудиторной (лекции, семинары) и самостоятельной работы обучающихся

очно-заочная форма обучения

Вид учебной работы по дисциплине	Всего (в з/е и часах)	Семестр 5 (в часах)	Семестр 6 (в часах)
Общая трудоемкость дисциплины	8/288	144	144
Контактная работа - Аудиторные занятия	52	26	26
<i>Лекции</i>	16	8	8
<i>Семинары, практические занятия</i>	36	18	18
<i>В том числе курсовая работа/курсовый проект</i>	24	-	-
Самостоятельная работа	236	118	118
Вид текущего	Контрольная	Контрольная	

контроля	работа; ;	работа	
Вид промежуточной аттестации	Зачет; Экзамен;	Зачет	Экзамен

5. Содержание дисциплины, структурированное по темам (разделам) дисциплины с указанием их объемов (в академических часах) и видов учебных занятий

5.1. Содержание дисциплины

Тема 1. Управление разработкой ПО. Гибкие методологии.

Введение в "Свод знаний по инженерии программного обеспечения". Идеи гибкой разработки. Принципы гибкой разработки. Основные роли в Скраме. События Скрам. Формирование требований в гибкой разработке. Основные практики Канбана. Роли в Канбане.

Тема 2. Формирование требований при разработке ПО.

Требования к качеству ПО. Контекст системы. Категории требований. Методы сбора требований. Карта пользовательских историй. Типы документации. Критерии качества для требований. Способы моделирования требований. Типы конфликтов. Приоритизация требований. Управление изменениями требований.

Тема 3. Управление разработкой ПО. Проектное управление.

Что такое проект. Методология управления проектами. Группы процессов. Области знаний. Ограничения проекта. Жизненный цикл проекта. Устав проекта. Содержание проекта. План управления проектом. Риски проекта. Обзор методологии PRINCE2. Принципы. Ограничения. Процессы управления. Компоненты методологии.

Тема 4. Масштабирование управления разработкой ПО.

Масштабирование проектного управления по PMBOK. Портфель и программы проектов. Типы проектных офисов. Различные организационные структуры. Масштабирование гибких методологий на примере Scaled Agile Framework. События.Arteфакты. Роли. Критичные факторы успеха.

Тема 5. Тестирование ПО.

Цели тестирования. Принципы тестирования. Контекст тестирования. Уровни тестирования. Типы тестирования. Тестирование изменений. Статическое тестирование. Категории методов проектирования тестов. Стратегия тестирования. Дефекты. Инструменты тестирования. Автоматизация тестирования.

Тема 6. Сопровождение ПО.

Библиотека инфраструктуры информационных технологий. Модель управления услугами. Цепочка создания ценности. Принципы. Управление инцидентами. Управление проблемами. Управление изменениями. Соглашение об уровне услуг. Управление конфигурациями. Управление релизами.

Тема 7. Методология разработки и эксплуатации программных систем и её расширения.

Цели. Что такое разработка и эксплуатация программных систем. Сочетание с гибкими методологиями. Культура взаимодействия. Базовые инструменты. Непрерывная интеграция. Запросы на слияние. Сборка. Модульное тестирование. Разработка через тестирование. Статический анализ кода. Среды разработки и тестирования. Динамические среды. Непрерывная поставка. Непрерывное развертывание. Управление репозиторием. Управление конфигурациями. Тесты развертывания. Разработка, безопасность и эксплуатация. Основные точки встраивания. Непрерывные измерения. Процесс измерений. Метрики. Контейнеризация. Оркестрация контейнеров. Шаблоны и антишаблоны формирования команд.

Тема 8. Управление качеством производства ПО.

Основные аспекты управления качеством: качество результата, качество системы, качество процессов. Основные стандарты, регламентирующие качество ПО, модели качества ПО, процессы оценки качества, модели зрелости организаций.

Тема 9. Основные идеи современной программной инженерии.

Рассматриваются современные идеи программной инженерии, применение элементов научного подхода, идея о том, что две основные проблемы, которые пытается решить программная

инженерия, это проблема познания в эволюционном процессе создания программных систем и проблема управления сложностью. Рассматриваются самые базовые подходы применимые с вариациями на различных стадиях инженерного процесса (анализ, проектирование, конструирование, поставка, эксплуатация).

Тема 10. Бизнес-архитектура и техническая архитектура.

Бизнес-архитектура, как основа для программной архитектуры решения. Переход от требований к драйверам и архитектурно значимым артефактам. Создание программной архитектуры на основе бизнес-архитектуры. Архитектурные стили и шаблоны.

Тема 11. Современные варианты распределенных приложений.

Более подробно рассматриваются (относительно) новые варианты построения приложений - микросервисная архитектура, а также даётся представление о подходе «облако-нативных приложений», который в настоящее время рассматривается как одна из ключевых технологий и фактически стал отраслевым стандартом разработки программных решений. Большинство современных систем проектируются на основе микросервисной архитектуры, контейнеризации и оркестрации, что обеспечивает их масштабируемость, надёжность и гибкость. Одновременно наблюдается переход к использованию интеллектуальных инструментов и автоматизации процессов разработки, которые расширяют возможности облако-нативных технологий и формируют новое поколение цифровых решений.

Тема 12. Атрибуты качества и сценарии атрибутов качества.

Рассматривается понятие атрибутов качества, атрибутов, наличие которых в предполагаемой (разрабатываемой) системе должна обеспечить создаваемая архитектура системы. Сценарии атрибутов качества как способы подтверждения свойств системы. Практические вопросы применения атрибутов качества. Тактики достижения показателей качества.

Тема 13. Паттерны проектирования.

Паттерны проектирования, как подход к повторному использованию удачных решений. Принципы применения паттернов. Классификация паттернов. Каталог паттернов.

Тема 14. Рефакторинг.

Рефакторинг, как подход к улучшению существующего кода. Принципы рефакторинга. Каталог методов рефакторинга. Типовые ошибки.

Тема 15. Мониторинг.

Мониторинг, как средство обеспечения качества функционирующей системы. Типы мониторинга, основные подходы и инструменты реализации.

Тема 16. Метрики программного кода.

Способы и методы оценки параметров исходного кода. Основные группы метрик, применимость их в различных сценариях.

5.2. Учебно-тематический план

очно-заочная форма обучения

№ п/п	Наименование тем(разделов) дисциплины	Трудоемкость в часах				
		Всего	Контактная работа - Аудиторная работа			Самостоятельная работа
			Общая, в т.ч.:	Лекции	Семинары, практические занятия	
1	Управление разработкой ПО. Гибкие методологии.	18	3	1	2	15
2	Формирование требований при разработке ПО.	18	3	1	2	15

3	Управление разработкой ПО. Проектное управление.	18	3	1	2	15
4	Масштабирование управления разработкой ПО.	20	5	1	4	15
5	Тестирование ПО.	20	5	1	4	15
6	Сопровождение ПО.	18	3	1	2	15
7	Методология разработки и эксплуатации программных систем и её расширения.	18	3	1	2	15
8	Управление качеством производства ПО.	18	3	1	2	15
9	Основные идеи современной программной инженерии.	18	3	1	2	15

10	Бизнес-архитектура и техническая архитектура.	18	3	1	2	15
11	Современные варианты распределенных приложений.	18	3	1	2	15
12	Атрибуты качества и сценарии атрибутов качества.	18	3	1	2	15
13	Паттерны проектирования.	18	3	1	2	15
14	Рефакторинг.	16	3	1	2	13
15	Мониторинг.	16	3	1	2	13
16	Метрики программного кода.	18	3	1	2	15
	Итого	288	52	16	36	236

5.3. Содержание семинаров, практических занятий

Наименование тем (разделов) дисциплины	Перечень вопросов для обсуждения на семинарах, практических занятиях	Формы проведения занятий
Управление	Формирование	На основе пилотного

разработкой ПО. Гибкие методологии.	проектных команд (принципы, методы). Средства совместной работы в группах. Практики эффективного взаимодействия участников проекта.	проекта. Разработка сценариев тестирования.
Формирование требований при разработке ПО.	Методы и инструменты сбора требований. Систематизация и оформление итогового перечня требований.	Групповая работа. Подготовка итогового списка требований, соответствующих критериям качества.
Управление разработкой ПО. Проектное управление.	Виды проектных офисов. Основные принципы организации проектной деятельности.	Устный опрос.
Масштабирование управления разработкой ПО.	Подходы к масштабированию разработки - управление портфелем проектов (традиционный подход) и масштабная гибкая структура (адаптивный подход).	Устный опрос.
Тестирование ПО.	Разработка сценариев ручного тестирования на основе пользовательских историй. Анализ и фиксация дефектов по результатам тестирования.	На основе пилотного проекта. Разработка и оформление сценариев тестирования.
Сопровождение ПО.	Разработка соглашения об уровне услуг для программного продукта.	Подготовка базового соглашения об уровне услуг для созданного программного решения.
Методология разработки и эксплуатации программных систем и её расширения.	Настройка конвейеров поставки (непрерывная интеграция и поставка). Используемые инструменты, типовые	Групповая работа. Разработка регламента поставки применительно к конкретному проекту.

	этапы построения конвейера.	
Управление качеством производства ПО.	Работа с нормативными документами. Разработка стратегии обеспечения качества программного продукта.	Работа над проектом.
Основные идеи современной программной инженерии.	Отличительные особенности программной инженерии от производственных процессов. Современные методы управления сложностью и адаптации к изменяющимся требованиям.	Дополнительное обсуждение материала лекции/семинара.
Бизнес-архитектура и техническая архитектура.	Работа с требованиями для формирования проектных решений. Определение и формализация нефункциональных требований, влияющих на качество системы. Планирование взаимодействия с заинтересованными сторонами на этапе реализации проекта.	Работа в команде. Определение заинтересованных сторон, построение диаграмм «заинтересованность – влияние», разработка планов коммуникаций.
Современные варианты распределенных приложений.	Контейнеризация как универсальный способ развертывания программных систем. Docker и Kubernetes - примеры промышленной реализации данного подхода.	Работа с проектом. Упаковка приложения в контейнеры.
Атрибуты качества и сценарии атрибутов качества.	Атрибуты качества как средство формализации нефункциональных требований.	Командная работа. Создание карт атрибутов качества для проекта.

	Принципы формирования и подходы к обеспечению.	
Паттерны проектирования.	Шаблоны как решения для типовых задач разработки программных систем. Применение шаблонов на практике.	Командная работа (использование шаблонов в коде проекта).
Рефакторинг.	Рефакторинг как средство повышения качества кода. Основные инструменты и принципы.	Командная работа (рефакторинг кода проекта).
Мониторинг.	Средства мониторинга. Обеспечение стабильности и качества на этапе эксплуатации программного комплекса.	Работа с проектом. Подключение и настройка инструментов мониторинга.
Метрики программного кода.	Использование метрик для оценки и поддержания качества программного продукта.	Работа с проектом. Подключение инструментов анализа и расчёта метрик в процессе сборки приложения.

6. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

6.1. Перечень вопросов, отводимых на самостоятельное освоение дисциплины, формы внеаудиторной самостоятельной работы

Наименование тем (разделов) дисциплины	Перечень вопросов, отводимых на самостоятельное освоение	Формы внеаудиторной самостоятельной работы
Управление разработкой ПО. Гибкие методологии.	Основные методы гибкой разработки: экстремальное программирование,	Изучение учебных материалов (статьи, стандарты). Изучение литературы.

	бережливый подход, Kanban, Scrum. Принципы гибких методологий и их области применения.	
Формирование требований при разработке ПО.	Изучение нормативных и методических документов по описанию требований. Основные подходы к формализации требований.	Изучение документации.
Управление разработкой ПО. Проектное управление.	Нормативные основы проектного управления. Основные процессы и документы в управлении проектами.	Изучение литературы.
Масштабирование управления разработкой ПО.	Подходы к масштабированию проектов. Отличия проектного и гибкого подходов.	Изучение литературы.
Тестирование ПО.	Основные методы и принципы тестирования. Разработка через тестирование как подход к повышению качества.	Изучение литературы.
Сопровождение ПО.	Организация передачи программных решений в эксплуатацию. Работа с изменениями и поддержкой.	Изучение литературы.
Методология разработки и эксплуатации программных систем и её расширения.	Изучение практик взаимодействия разработки и эксплуатации. Современные инструменты автоматизации поставки и развертывания.	Изучение литературы.
Управление	Модели зрелости	Изучение литературы.

качеством производства ПО.	процессов и оценки уровня качества разработки.	
Основные идеи современной программной инженерии.	Подходы к управлению сложностью и изменчивостью требований. Эволюция инженерных принципов в разработке программных систем.	Изучение литературы.
Бизнес-архитектура и техническая архитектура.	Основные элементы архитектуры предприятия и их связь с программными решениями. Роль архитектуры в достижении стратегических целей организации.	Изучение литературы.
Современные варианты распределенных приложений.	Основные подходы к созданию распределённых систем. Микросервисы, облачные и контейнерные технологии.	Изучение материалов (статьи).
Атрибуты качества и сценарии атрибутов качества.	Основные подходы к формированию и описанию сценариев атрибутов качества. Методы обеспечения и оценки соответствия требованиям к качеству.	Изучение литературы.
Паттерны проектирования.	Применение шаблонов проектирования для решения типовых задач разработки. Анализ и выбор шаблонов под конкретные архитектурные решения.	Изучение литературы.
Рефакторинг.	Принципы и методы рефакторинга	Изучение литературы.

	программного кода. Влияние рефакторинга на качество и сопровождаемость программных решений.	
Мониторинг.	Современные инструменты мониторинга программных систем. Применение открытых решений для отслеживания состояния сервисов и инфраструктуры.	Изучение документации.
Метрики программного кода.	Использование инструментов для сбора и анализа метрик в процессе сборки и поставки программных продуктов. Значение метрик для оценки качества и производительности кода.	Изучение материалов (статьи).

6.2. Перечень вопросов, заданий, тем для подготовки к текущему контролю

Примерные задания для контрольной работы

1. Сформулировать пятнадцать требований к системе дистанционного обучения. Разделить их на функциональные и качественные, дополнительно классифицировать по модели удовлетворенности пользователей. Требования должны соответствовать критериям качества.
2. Сформулировать пятнадцать требований к корпоративной системе учета рабочего времени. Разделить их на функциональные и качественные, дополнительно классифицировать по модели удовлетворенности пользователей. Требования должны соответствовать критериям качества.

3. Разработать диаграмму «сущность-связь» для системы дистанционного обучения. Диаграмма должна содержать не менее шести сущностей, общий состав атрибутов не менее сорока. Связи «многие ко многим» не допускаются, должен быть приведен пример связи «один к одному».
4. Разработать диаграмму «сущность-связь» для корпоративной системы учета рабочего времени. Диаграмма должна содержать не менее шести сущностей, общий состав атрибутов не менее сорока. Связи «многие ко многим» не допускаются, должен быть приведен пример связи «один к одному».
5. Составить пятнадцать пользовательских историй для системы дистанционного обучения с указанием критериев приемки. Необходимо предусмотреть не менее двух ролей пользователей, например преподаватель и обучающийся.
6. Составить пятнадцать пользовательских историй для корпоративной системы учета рабочего времени с указанием критериев приемки. Необходимо предусмотреть не менее двух ролей пользователей, например сотрудник и руководитель.
7. Разработать шесть сценариев тестирования для системы дистанционного обучения, общее количество шагов не менее сорока. В сценариях должны быть использованы методы анализа граничных значений и таблицы альтернатив.
8. Разработать шесть сценариев тестирования для корпоративной системы учета рабочего времени, общее количество шагов не менее сорока. В сценариях должны быть использованы методы анализа граничных значений и таблицы альтернатив.
9. Составить основные положения соглашения об уровне обслуживания для системы дистанционного обучения, предоставляемой внешним исполнителем. Определить три уровня приоритета, критерии их установления и временные характеристики реагирования и устранения неисправностей.
10. Составить основные положения соглашения об уровне обслуживания для корпоративной системы учета рабочего времени, предоставляемой внешним исполнителем. Определить три

уровня приоритета, критерии их установления и временные характеристики реагирования и устранения неисправностей.

Критерии балльной оценки различных форм текущего контроля успеваемости содержатся в соответствующих методических рекомендациях Кафедры информационных технологий Факультета информационных технологий и анализа больших данных.

7. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине

Перечень компетенций с указанием индикаторов их достижения в процессе освоения образовательной программы содержится в разделе 2. *Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине.*

Типовые контрольные задания или иные материалы, необходимые для оценки индикаторов достижения компетенций, умений и знаний

ОПК-4 Способен участвовать в разработке стандартов, норм и правил, а также технической документации, связанной с профессиональной деятельностью

1) Демонстрирует знания основных стандартов ведения технической документации, как отечественных, так и зарубежных

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: различные подходы к формированию требований к программному обеспечению

Уметь: выбирать подходящую методологию формирования требования, адаптировать под предметную область

Типовые контрольные задания

Разработать комплект документов для программного комплекса (выбрать методологию, состав документов, содержание и нотацию, адресацию).

2) Создает комплект программной и проектной документации к разрабатываемой информационной системе

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: типы документов, их назначение и содержание

Уметь: выбирать номенклатуру документации соответствующую типу проекта, формировать документы в соответствии с отраслевыми стандартами

Типовые контрольные задания

Подготовить пакет документации в соответствии с требованиями ГОСТ 34 или в соответствии с требованиями методологией гибкого управления проектами.

3) Критически анализирует программную и проектную документацию, строит на ее основе логические выводы

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: основные методы критического анализа программной и проектной документации

Уметь: демонстрировать умение критически оценивать и дорабатывать программную и проектную документацию

Типовые контрольные задания

Разработать план проекта с учетом: стоимости, времени, не менее пяти исполнителей, не менее трех уровней вложенности. Составить реестр рисков и разработать план управления этими рисками.

ПКП-3 Способность собирать, анализировать и управлять требованиями к программной системе

1) Понимает виды требований к программному обеспечению, составляет формализованное описание требований по нетехническому описанию функционала программной системы

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: методики работы с требованиями позволяющие из требований формировать задачи на разработку

Уметь: планировать и вести работу с заинтересованными лицами, выделять архитектурно значимые требования и исходя из них формировать и обосновывать архитектурные решения

Типовые контрольные задания

Представить планирование на 3 спринта с 15 пользовательскими историями. На историях должны быть названия, оценка в сторипойнтах согласно ряду Фибоначчи. У спринтов должны быть емкость и скорость. Расчет емкости должен быть обоснован. Тематика разработки - портал ИТ-поддержки.

2) Анализирует программную систему на предмет соответствия требованиям, выявляет проблемные участки, модифицирует архитектуру и дизайн программной системы для полного соответствия набору требований

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: виды требований к программному обеспечению и их особенности

Уметь: анализировать и модифицировать архитектуру и дизайн программной системы в соответствии с требованиями

Типовые контрольные задания

Определить и классифицировать требования к программной системе на функциональные и нефункциональные.

3) Анализирует описанную систему требований к программной системе - выделяет требования высшего и низшего уровня, проводит декомпозицию требований, соотносит их с архитектурными решениями программной системы, приоритизирует требования, оценивает реализуемость и

тестируемость требований, составляет по системе требований протокол тестирования программной системы

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: методы анализа программной системы на соответствие требованиям и выявление проблемных участков.

Уметь: выделять, декомпозировать, приоритизировать требования, оценивать их реализуемость и тестируемость

Типовые контрольные задания

Провести анализ системы требований к программной системе и выделить требования высшего и низшего уровня.

Оценить реализуемость и тестируемость каждого требования программной системы.

4) Критически относится к сформулированным требованиям к программной системе - выявляет внутренние противоречия, риски, связанные с реализацией тех или иных требований, воспринимает требования в более широком контексте постановки задачи, находит недостающие требования

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: критическое отношение к сформулированным требованиям, выявление противоречий, рисков и недостающих требований

Уметь: находить внутренние противоречия, риски и недостающие требования в сформулированных требованиях к программной системе

Типовые контрольные задания

Провести критический анализ сформулированных требований к программной системе, выявить внутренние противоречия и риски их реализации и выполнить анализ рисков, связанных с реализацией противоречивых или неполных требований.

5) Самостоятельно формулирует подробную, формальную, полную и реализуемую систему требований к программной системе и описывает ее в технической документации

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: подробную, формальную, полную и реализуемую систему требований к программной системе

Уметь: описывать ее в технической документации

Типовые контрольные задания

Разработать формализованное техническое задание (ТЗ) на разработку программной системы, включающее цели, ограничения, функциональные и нефункциональные требования, а также критерии приёмки.

ПКП-5 Способность проектировать и реализовывать интеллектуальные информационные системы

1) Демонстрирует знания основных методов машинного обучения и интеллектуального анализа данных, применяет готовые инструменты для создания интеллектуальных алгоритмов

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: основные методы машинного обучения, включая обучение с учителем, обучение без учителя, обучение с подкреплением, а также основы интеллектуального анализа данных, таких как обработка больших данных, статистический анализ и т.д.

Уметь: применять методы машинного обучения и интеллектуального анализа данных на практике

Типовые контрольные задания

Провести анализ предоставленного набора данных, выполнить предварительную обработку (очистку, нормализацию, кодирование категориальных признаков) и визуализацию данных.

2) Понимает особенности интеллектуальных информационных систем в части операций разработки, развертывания и сопровождения

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: операции разработки, развертывания и сопровождения интеллектуальных информационных систем

Уметь: разрабатывать, развертывать и обслуживать интеллектуальные информационные системы

Типовые контрольные задания

Создать схему инфраструктуры развертывания ИИС (серверы, базы данных, API, контейнеризация, облачные сервисы).

3) Адаптирует практики создания программных продуктов, в том числе командные, для интеллектуальных информационных систем

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: практики создания программных продуктов, включая командные методы, применяемые в интеллектуальных информационных системах

Уметь: адаптировать практики создания программных продуктов для интеллектуальных информационных систем

Типовые контрольные задания

Сформировать структуру проектной команды для разработки ИИС, распределить роли и зоны ответственности.

Определить инструменты совместной работы и управления проектом и обосновать их выбор.

4) Организует сбор и подготовку данных для систем машинного обучения, в том числе потоковых, онлайн обучения

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: гибкие методологии управления разработкой

Уметь: участвовать в процессе управления работами, планировать работы по гибким методологиям и контролировать ход их выполнения

Типовые контрольные задания

Определить источники данных (внутренние и внешние), разработать схему интеграции данных для обучения модели, а

также смоделировать процесс организации потокового сбора данных для системы онлайн-обучения.

Примеры практико-ориентированных заданий

1. Разработать прототип веб-приложения для бронирования столиков в ресторане с возможностью выбора времени, количества гостей и автоматическим подтверждением заявки.
2. Смоделировать процесс технической поддержки пользователей в корпоративной системе, определить роли участников, маршруты обращений и показатели эффективности обслуживания.
3. Создать набор тестовых сценариев для мобильного приложения интернет-магазина, охватывающих основные пользовательские действия: поиск, оформление заказа, оплата и получение уведомлений.
4. Подготовить соглашение об уровне обслуживания для службы технической поддержки компании, включая описание приоритетов обращений и временные нормативы реагирования.
5. Разработать диаграмму архитектуры микросервисного приложения для онлайн-библиотеки, указав взаимодействие между сервисами, базами данных и внешними интерфейсами.

Примерные вопросы для подготовки к Зачету, Экзамену

Примерные вопросы для подготовки к зачету

1. Вам необходимо собрать требования по автоматизации операционного участка предприятия, выполняющего рутинную работу. Документация отсутствует, а работники не горят желанием рассказывать про свою работу, так как понимают, что автоматизация приведет к сокращениям. Какой способ сбора требований подходит больше всего.
2. Ваша компания делает стандартные сайты-визитки для небольших компаний. Какой способ сбора требований подходит больше всего.

3. Вы собираетесь в команде делать пилот нового приложения, аналоги которого практически отсутствуют. Выберите наиболее подходящие способы сбора требований для вашего приложения.
4. Вы собираетесь переделывать текущее приложение на новый технологический стек и с вами готовы работать продвинутые пользователи, объясняя текущую работу приложения и все особенности его работы. Выберите наиболее подходящий способ сбора требований для нового приложения.
5. Вам необходимо внедрить новую систему мониторинга приложений. Выберите фреймворк управления.
6. Вы собираетесь разработать новое мобильное приложение, состоящее внутри из различных модулей. Выберите фреймворк управления разработкой.
7. Вам поручено управление по разворачиванию новой инфраструктуры по виртуализации. Выберите фреймворк по управлению.
8. Вам поручено управлению по внедрению целого ряда процессов ITIL – управление инцидентами, управлением проблемами, управление запросами на обслуживание, управление запросами на изменение и управление конфигурациями. Выберите фреймворк по управлению.
9. Вам поручено управление разработкой набора микросервисов. Выберите фреймворк по управлению разработкой.
10. Вам поставили задачу внедрить в разработке софта DevOps pipeline . С каких базовых типов инструментов вы начнете?
11. Вы внедряете DevOps pipeline в разработке софта. К вам пришла информационная безопасность и потребовала предоставить им возможности контроля, чтобы сделать разработку безопасной. Какие варианты вы предложите информационной безопасности?
12. Вы внедряете DevOps pipeline в разработке софта. К вам пришли разработчики и тестировщики с тем, что они тратят кучу времени для поиска причин дефектов в работе внешнего сайта из-за меняющихся и различных настроек операционной системы и веб-сервера на тестовых виртуалках. Что вы им предложите?

13. Назовите основные практики DevOps и чем они концептуально друг от друга отличаются.

14. Система состоит из 5 сильно взаимосвязанных модулей, по запросу заказчика были внесены изменения в один из модулей. Какие типы тестирования обязательно применить перед выкладкой изменения на промышленную среду.

15. Безопасность постоянно ругается с разработчиками из-за регулярного использования в коде паролей в открытом виде. Какой вид тестирования может помочь безопасности избавиться от такого грубейшего нарушения в принципе?

16. К вам предъявили требования организовать тестирование на как можно более ранних этапах жизненного цикла разработки. Какой способ тестирования обеспечивает его на самом раннем этапе?

17. Ваша компания занимается разработкой стандартных сайтов-визиток по заданным шаблонам. Вам поставили задачу организовать процесс тестирования. Какой тип тестирования лучше всего подойдет с учетом требований по высокой скорости обратной связи?

Примерные вопросы для подготовки к экзамену

1. Жизненный цикл ПО. Понятие жизненного цикла ПО, этапы, их описание в терминологии различных стандартов (примеры).

2. Процессы создания ПО. Управление процессами, 5 групп процессов, их характерные особенности.

3. Проект. Управление проектами. Понятие проектного офиса, типы проектных офисов. Программы и портфели.

4. Требования к ПО. Виды требований. Техники сбора и описания.

5. Проектное управление. Ограничения и жизненный цикл проекта

6. Проектное управление. Методологии. PMBOOK и PRINCE2

7. Гибкие принципы управление. Agile – идеи и принципы.

8. Гибкие технологии. Scrum

9. Гибкие технологии. Kanban
10. Масштабирование гибкой разработки. Проблемы и решения. SAFe как пример масштабируемого фреймворка
11. Гибкое управление. Особенности организации работы дистанционно работающих команд
12. Поток поставки ценности. DevOps – понятия и основные принципы
13. DevOps. Компоненты и процессы. Понятие безопасной разработки
14. Документация. Типы, назначение. Принципы создания, группы заинтересованных лиц, методики формирования пакетов документации.
15. Тестирование ПО. Цели и принципы
16. Тестирование. Уровни и типы тестирования.
17. Тестирование. Методы черного и белого ящика. Статическое и динамическое тестирование
18. Тестирование. Методы и инструменты. Классификация дефектов
19. Сопровождение и поддержка ПО. ITSM и ITIL общие понятия и принципы
20. Сопровождение и поддержка. ITIL. Базовые процессы
21. Управление изменениями. ITIL и процесс управления изменениями
22. Процесс разработки. Архитектура решения. Назначение, принципы выбора. Анализ и оценка
23. Процесс разработки. Архитектура. Документирование и распространение сведений. Группы заинтересованных лиц.
24. Атрибуты качества систем и их связь с функциональными требованиями. Сценарии атрибутов качества
25. Атрибуты качества и сценарии - подробное описание на примере одного из типа сценариев (Модифицируемость, Ремонтопригодность, Производительность, Безопасность...)

26. Тактики обеспечения атрибутов качества (на примере одного из типов Модифицируемость, Ремонтопригодность, Производительность, Безопасность...)
27. Документирование программных систем. Заинтересованные лица, представления и нотации (с примерами).
28. Документирование программных систем. Неформальные нотации. Плюсы и минусы, области применения
29. Документирование программных систем. Частично формальные и формальные нотации. Плюсы и минусы, области применения
30. Нотации. Рассмотреть пример одной из нотаций на выбор. Основные типы диаграмм, области применения
31. Стили и шаблоны проектирования. Понятие архитектурного стиля и архитектурного шаблона
32. Архитектурные принципы проектирования. Основные принципы и подробно на примере 1 – 2 выбранных
33. Архитектурные стили. Уровни применения, типичные представители (на примере выбранного - монолит, SOA, Микросервисы...)
34. Архитектурные стили на уровне компонентов. Слоистая и модульная архитектура
35. Паттерны проектирования. Общие понятия, назначение, плюсы и минусы.
36. Паттерны проектирования. Порождающие паттерны (на выбранном примере)
37. Паттерны проектирования. Структурные паттерны (на выбранном примере)
38. Паттерны проектирования. Поведенческие паттерны (на выбранном примере)
39. Управление качеством. Основные понятия, процессы и практики
40. Управление качеством. Стандарты и модели

- 41. Управление качеством. Эталонная модель
 - 42. Методологии управления качеством. Классификация
 - 43. Рефакторинг. Назначение и принципы
 - 44. Рефакторинг. Техники рефакторинга (на примере одной из техник)
 - 45. Метрики программного кода. Основные принципы и подходы, инструменты, практика применения.
- *Промежуточная аттестация обучающихся проводится в формате тестирования.*

Примерные темы курсовой работы/курсового проекта

- 1. Процесс оценки качества программного обеспечения. Метрики, подходы на примере проекта. Предметная область: Сбор команд для совместных проектов.
- 2. Разработка архитектуры решения. Предметная область: Сбор команд для совместных проектов.
- 3. Разработка системы комплексной оценки качества программного продукта. Предметная область: Станция технического обслуживания автомобилей.
- 4. Разработка архитектуры решения. Предметная область: Мониторинг курсов криптовалют на нескольких биржах.
- 5. Разработка стратегии тестирования программного продукта. Предметная область: Расписание Финансового университета.
- 6. Сравнительный анализ гибких методологий Scrum и Экстремальное программирование на примере конкретного проекта. Предметная область: Расписание университета.

7. Разработка системы комплексной оценки качества программного продукта. Предметная область: Расписание сеансов сети кинотеатров.
8. Разработка архитектуры решения. Предметная область: Выбор автомобиля по каталогу.
9. Разработка системы комплексной оценки качества программного продукта. Предметная область: Мониторинг курсов криптовалют на нескольких биржах.
10. Разработка архитектуры решения. Предметная область: Выбор автомобиля по каталогу.

8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

Нормативно-правовые акты:

1. Федеральный закон «Об информации, информационных технологиях и о защите информации»
2. Федеральный закон «О персональных данных»
3. Федеральный закон «О безопасности критической информационной инфраструктуры Российской Федерации»
4. Федеральный закон «Об электронной подписи»
5. ГОСТ Р ISO/IEC/IEEE 12207-2020 «Системная и программная инженерия. Процессы жизненного цикла программного обеспечения»
6. ГОСТ Р ISO/IEC 15288-2020 «Системная и программная инженерия. Процессы жизненного цикла систем»
7. ГОСТ Р ISO/IEC 25010-2015 «Системы и программное обеспечение. Модели качества»
8. ГОСТ Р ISO/IEC 29110-2018 «Инженерия программного обеспечения. Руководство для малых предприятий»
9. ГОСТ Р 56939-2016 «Информационные технологии. Руководство по управлению ИТ-проектами»

10. ГОСТ Р ISO/IEC 9126-93 «Характеристики качества программного обеспечения»
11. ГОСТ 19781-90 «Программные средства. Термины и определения»
12. ГОСТ Р 57580-2017 «Защита информации финансовых организаций. Общие положения»
13. ГОСТ Р ISO/IEC 27001-2021 «Системы менеджмента информационной безопасности. Требования»
14. Методические рекомендации Минцифры России по созданию и сопровождению государственных информационных систем
15. Приказы ФСТЭК и ФСБ России в области защиты информации и безопасности программного обеспечения

Основная литература:

1. Лаврищева, Екатерина Михайловна Программная инженерия и технологии программирования сложных систем : учебник для вузов / Е. М. Лаврищева. 2-е изд., испр. и доп Электрон. дан. Москва : Юрайт, 2025 432 с (Высшее образование) URL: <https://urait.ru/bcode/561885> (дата обращения: 01.09.2025). Режим доступа: Электронно-библиотечная система Юрайт, для авториз. пользователей <https://urait.ru/bcode/561885> ISBN 978-5-534-07604-2 : 2089.00. [БИК ID: RU2fURAIT2f561885]
2. Зубкова, Т. М. Технология разработки программного обеспечения [Электронный ресурс] : учебное пособие / Зубкова Т. М. Санкт-Петербург : Лань, 2022 324 с. Книга из коллекции Лань - Информатика <https://e.lanbook.com/book/206882> ISBN 978-5-8114-3842-6. [БИК ID: RU-LAN-BOOK-206882]

Дополнительная литература:

1. Рыбалова, Е. А. Управление проектами [Электронный ресурс] : учебное методическое пособие / Рыбалова Е. А. Москва : ТУСУР, 2015 149 с. Книга из коллекции ТУСУР - Информатика <https://e.lanbook.com/book/110293>. [БИК ID: RU-LAN-BOOK-110293]
2. Антамошкин, О. А. Программная инженерия. Теория и практика [Электронный ресурс] : учебник / Антамошкин О. А. Красноярск : СФУ, 2012 247 с. Рекомендовано Учебно-методическим

объединением по образованию в области прикладной информатики в качестве учебника для студентов высших учебных заведений, обучающихся по специальности 080801 «Прикладная информатика (по областям)» и другим экономическим специальностям. Книга из коллекции СФУ - Инженерно-технические науки http://e.lanbook.com/books/element.php?pl1_id=45709 ISBN 978-5-7638-2511-4. [БИК ID: 45709]

3. Черткова, Елена Александровна Программная инженерия. Визуальное моделирование программных систем : учебник для вузов / Е. А. Черткова. 3-е изд., пер. и доп Электрон. дан. Москва : Юрайт, 2025 146 с (Высшее образование) URL: <https://urait.ru/bcode/562413> (дата обращения: 01.09.2025). Режим доступа: Электронно-библиотечная система Юрайт, для авториз. пользователей <https://urait.ru/bcode/562413> ISBN 978-5-534-18197-5 : 729.00. [БИК ID: RU2fURAIT2f562413]

9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

1. Электронная библиотека Финансового университета (ЭБ) <http://elib.fa.ru/> (<http://library.fa.ru/files/elibfa.pdf>)
2. Электронно-библиотечная система BOOK.RU <http://www.book.ru>
3. Образовательная платформа "ЮРАЙТ" <https://urait.ru/>
4. Научная электронная библиотека eLibrary.ru <http://elibrary.ru>
5. Информационно-образовательный портал Финуниверситета: <https://org.fa.ru>
6. •Электронно-библиотечная система издательства «ЮРАЙТ» <https://urait.ru/>
7. •Электронно-библиотечная система Znanium <http://www.znanium.ru/>
8. •Электронно-библиотечная система «Университетская библиотека ОНЛАЙН» <http://biblioclub.ru/>

9. •Электронно-библиотечная система издательства Лань
<https://e.lanbook.com/>
10. •Национальная электронная библиотека <http://нэб.рф/>
11. •Электронные продукты издательства Elsevier
<http://www.sciencedirect.com>
12. •Электронные коллекции книг и журналов издательства Springer:
<http://link.springer.com/>
13. СПАРК <https://spark-interfax.ru/>
14. Портал корпоративного управления. Раздел «Информационные технологии» - www.iteam.ru/publications/it/
15. www.government.ru - Официальный сайт Правительства Российской Федерации
16. www.cbr.ru - Официальный сайт Банка России

10. Методические указания для обучающихся по освоению дисциплины

Основные этапы работы студента по дисциплине Программная инженерия

1. Предварительная ориентировка в подлежащем изучению учебном материале по программе.
2. Ознакомление с рекомендованной учебной литературой.
3. Слушание и конспектирование лекций, а также выполнение других видов учебной работы.
4. Планирование самостоятельной работы.
5. Обобщение и систематизация информации, почерпнутой из лекций и прочитанной литературы.
6. Выполнение контрольной работы.

Рекомендации по работе с учебным материалом

1. Осознавайте наличный уровень полученных вами знаний.

2. В ситуации непонимания нужно выявить тот первичный уровень и факторы непонимания, которые стали препятствием понимания последующего.

3. Задавайте сами себе вопросы и пытайтесь ответить на них.

Рекомендации по работе на лекции и с лекционным материалом

1. Основная задача на лекции – осмысление излагаемого в ней материала. Для этого необходимо слушать лекцию с самого начала, не упуская общих, ориентирующих в материале рассуждений и установок лектора.

2. Ведение записей на лекции важно и полезно для лучшего осмысливания материала, для сохранения информации, с целью ее дальнейшего использования.

3. Для облегчения записи рекомендуется применять сокращения повторяющихся терминов или хорошо известных понятий.

Рекомендации по работе с литературой

1. Если возникли затруднения при разыскивании материала по какому-либо конкретному вопросу, следует обратиться к предметному указателю, напечатанному, как правило, в конце каждого литературного источника.

2. Предметный указатель – это алфавитный список основных научных понятий (терминов), содержание которых раскрыто в книге, рядом с термином стоят числа, обозначающие номера страниц, на которых изложен материал, относящийся к данному понятию.

Рекомендации по выполнению контрольной работы

1. Перед выполнением контрольной работы студент должен изучить соответствующие разделы учебной литературы.

2. Контрольную работу студент должен выполнять самостоятельно, используя те навыки и умения, которые получил на лекциях и практических занятиях.

3. При затруднениях, возникших при выполнении контрольной работы, студент может получить консультацию преподавателя.

Рекомендации по выполнению курсовой работы

1. Перед выполнением курсовой работы студент должен изучить соответствующие разделы учебной литературы.
2. Курсовую работу студент должен выполнять самостоятельно, используя те навыки и умения, которые получил на лекциях и практических занятиях.
3. При затруднениях, возникших при выполнении курсовой работы, студент может получить консультацию преподавателя.

11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень необходимого программного обеспечения и информационных справочных систем

- Комплект лицензионного программного обеспечения:

1. Kaspersky
2. Пакет офисных программ

- Современные профессиональные базы данных и информационные справочные системы:

1. Информационно-правовая система «Гарант»
2. Информационно-правовая система «Консультант Плюс»
3. Электронная энциклопедия: <http://ru.wikipedia.org/wiki/Wiki>
4. Размещение ЭУК: <https://www.campus.fa.ru/>
5. Создание тестов, система Moodle, осуществление прокторинга (асинхронный, синхронный): <https://edu.fa.ru/>
6. Система комплексного раскрытия информации «СКРИН» - <http://www.skrin.ru/>

- Сертифицированные программные и аппаратные средства защиты информации

1. не предусмотрены

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

1. **Учебная аудитория** для проведения учебных занятий, предусмотренных программой, в том числе групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, оснащенная оборудованием и техническими средствами обучения: мебель аудиторная (столы, стулья, доска аудиторная), набор демонстрационного оборудования (проектор, экран)

2. **Помещение для самостоятельной работы** обучающихся оснащенное компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечением доступа к электронной информационно-образовательной среде Университета.

3. **Компьютерный класс** для проведения учебных занятий, предусмотренных программой, в том числе групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, оснащенный оборудованием и техническими средствами обучения: мебель аудиторная (столы, стулья, доска аудиторная), персональные компьютеры, набор демонстрационного оборудования (проектор, экран)

4. Библиотека, читальный зал