

**Федеральное государственное образовательное бюджетное
учреждение высшего образования
«Финансовый университет при Правительстве Российской Федерации»
(Финуниверситет)**



Владикавказский филиал Финуниверситета

Кафедра «Математика и информатика»

СОГЛАСОВАНО

ООО «Экспертно-аналитические
системы»

Генеральный директор


И.В. Игнатович
«27» апреля 2026 г.


УТВЕРЖДАЮ

Директор филиала


Т.А. Хубаев
«27» апреля 2026 г.


А.М. Кумаритов

Программная инженерия

Рабочая программа дисциплины

для студентов, обучающихся по направлению подготовки:
09.03.04 – Программная инженерия,
ОП «Технологии разработки программного обеспечения»

*Рекомендовано Ученым советом Владикавказского филиала
Финуниверситета*

(протокол от «15» апреля 2026 г. № 30)

*Одобрено на заседании кафедры «Математика и информатика»
(протокол от «10» апреля 2026 г. № 8)*

Владикавказ 2026

Содержание

1. Наименование дисциплины.....	2
2. Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине	2
3. Место дисциплины в структуре образовательной программы.....	5
4. Объем дисциплины (модуля) в зачетных единицах и в академических часах с выделением объема аудиторной (лекции, семинары) и самостоятельной работы обучающихся	5
5. Содержание дисциплины, структурированное по темам (разделам) дисциплины с указанием их объемов (в академических часах) и видов учебных занятий.....	5
5.1. Содержание дисциплины.....	5
5.2. Учебно–тематический план.....	8
5.3. Содержание семинаров, практических занятий.....	11
6. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине.....	11
6.1. Перечень вопросов, отводимых на самостоятельное освоение дисциплины, формы внеаудиторной самостоятельной работы.....	15
6.2. Перечень вопросов, заданий, тем для подготовки к текущему контролю	15
7. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине.....	26
8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины	38
9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины.....	39
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 справочных систем	49
11.1. Комплект лицензионного программного обеспечения.....	49
11.2. Современные профессиональные базы данных и информационные справочные системы.....	49
11.3. Сертифицированные программные и аппаратные средства защиты информации.....	49
12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине.....	49

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	Индикаторы достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции
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 интеллектуального анализа данных на практике
		Понимает особенности интеллектуальных информационных систем в части операций разработки, развертывания и сопровождения.	Знать: операции разработки, развертывания и сопровождения интеллектуальных информационных систем. Уметь: разрабатывать, развертывать и обслуживать интеллектуальные информационные системы.
		Адаптирует практики создания программных продуктов, в том числе командные, для интеллектуальных информационных систем.	Знать: практики создания программных продуктов, включая командные методы, применяемые в интеллектуальных информационных системах. Уметь: адаптировать практики создания программных продуктов для интеллектуальных информационных систем.

		Организовывает сбор и подготовку данных для систем машинного обучения, в том числе потоковых, онлайн обучения.	Знать: гибкие методологии управления разработкой Уметь: участвовать в процессе управления работами, планировать работы по гибким методологиям и контролировать ход их выполнения
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 программной системы для полного соответствия набору требований.	Знать: виды требований к программному обеспечению и их особенности. Уметь: анализировать и модифицировать архитектуру и дизайн программной системы в соответствии с требованиями
		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.	Знать: методы анализа программной системы на соответствие требованиям и выявление проблемных участков. Уметь: выделять, декомпозировать, приоритизировать требования, оценивать их реализуемость и тестируемость.

		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.	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.
		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
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 ПО Уметь: выбирать подходящую методологию формирования требования, адаптировать под предметную область
		Создает комплект программной и проектной документации к разрабатываемой информационной 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 документацию, строит на ее основе логические выводы.	Знать: основные методы критического анализа программной и проектной документации. Уметь: демонстрировать умение критически оценивать и дорабатывать программную и проектную документацию.

3. Место дисциплины в структуре образовательной программы

Дисциплина «Программная инженерия» является дисциплиной профиля части, формируемой участниками образовательных отношений, учебного плана образовательной программы «Технологии разработки программного обеспечения» по направлению подготовки 09.03.04 Программная инженерия, профиль «Технологии разработки программного обеспечения».

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

Вид учебной работы по дисциплине	Всего (в з/е и часах)	Семестр 5 (в часах)	Семестр 6 (в часах)
Общая трудоемкость дисциплины	8/288	126	162
<i>Контактная работа – Аудиторные занятия</i>	<i>118</i>	<i>50</i>	<i>68</i>
<i>Лекции</i>	<i>32</i>	<i>16</i>	<i>16</i>
<i>Семинары, практические занятия</i>	<i>86</i>	<i>34</i>	<i>52</i>
<i>Самостоятельная работа</i>	<i>170</i>	<i>76</i>	<i>94</i>
Вид текущего контроля	контрольная работа, курсовая работа	контрольная работа	курсовая 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 (Agile)

Введение в стандарт SWEBOK. Идеи Agile. Принципы Agile. Основные роли в Скрам. События Скрам. Формирование требований в Agile. Основные практики Канбан. Роли в Канбан.

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 PMBOK. 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. Артефакты. 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.

ITIL. Модель управления услугами. Цепочка создания ценности. Принципы. Управление инцидентами. Управление проблемами. Управление изменениями. SLA. Управление конфигурациями. Управление релизами.

Тема 7. Методология DevOps и ее расширения.

Цели. Что такое DevOps. Сочетание с Agile. Культура взаимодействия. Базовые инструменты. Непрерывная интеграция. Запросы на слияние. Сборка. Модульное тестирование. TDD. Статический анализ кода. Среды разработки и тестирования. Динамические среды. Непрерывная поставка. Непрерывное развертывание. Управление репозиторием. Управление конфигурациями. Тесты развертывания.

DevSecOps. Основные точки встраивания. Непрерывные измерения. Процесс измерений. Метрики. Контейнеризация. Оркестрация контейнеров. Паттерны и антипаттерны образования 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 и дается представление о подходе Cloud Native Apps (в трендах Гартнер рассматривается как топ-3 технология на 2023-2024 годы).

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

№ п/п	Наименование тем (разделов) дисциплины	Трудоемкость в часах					Формы текущего контроля успеваемости
		Всего	Контактная работа - Аудиторная работа			Самостоят ельная работа	
			Общая, в т.ч.:	Лекции	Семинары, практическ ие занятия		
1	Управление разработкой ПО. Гибкие методологии (Agile)	16	6	2	4	10	Опрос, собеседование по домашним заданиям самостоятельной работы, решение практико- ориентированн ых задач.
2	Формирование требований при разработке ПО.	16	6	2	4	10	Опрос, собеседование по домашним заданиям самостоятельной работы, решение

							практико-ориентированных задач.
3	Управление разработкой ПО. Проектное управление.	16	6	2	4	10	Опрос, собеседование по домашним заданиям самостоятельной работы, решение практико-ориентированных задач.
4	Масштабирование управления разработкой ПО.	16	6	2	4	10	Опрос, собеседование по домашним заданиям самостоятельной работы, решение практико-ориентированных задач.
5	Тестирование ПО.	22	8	2	6	14	Опрос, собеседование по домашним заданиям самостоятельной работы, решение практико-ориентированных задач.
6	Сопровождение ПО.	16	6	2	4	10	Опрос, собеседование по домашним заданиям самостоятельной работы, решение практико-ориентированных задач.
7	Методология DevOps и ее расширения	16	6	2	4	10	Опрос, собеседование по домашним заданиям самостоятельной работы, решение практико-ориентированных задач.
8	Управление качеством производства ПО	20	8	2	6	12	Опрос, собеседование по домашним заданиям самостоятельной

							работы, решение практико-ориентированных задач.
9	Основные идеи современной программной инженерии	22	8	2	6	14	Опрос, собеседование по домашним заданиям самостоятельной работы, решение практико-ориентированных задач.
10	Бизнес архитектура и техническая архитектура	16	8	2	6	8	Опрос, собеседование по домашним заданиям самостоятельной работы, решение практико-ориентированных задач.
11	Современные варианты распределенных приложений	20	8	2	6	12	Опрос, собеседование по домашним заданиям самостоятельной работы, решение практико-ориентированных задач.
12	Атрибуты качества и сценарии атрибутов качества	22	8	2	6	14	Опрос, собеседование по домашним заданиям самостоятельной работы, решение практико-ориентированных задач.
13	Паттерны проектирования	22	10	2	8	12	Опрос, собеседование по домашним заданиям самостоятельной работы, решение практико-ориентированных задач.
14	Рефакторинг	16	8	2	6	8	Опрос, собеседование по домашним заданиям

							самостоятельной работы, решение практико-ориентированных задач.
15	Мониторинг	14	8	2	6	6	Опрос, собеседование по домашним заданиям самостоятельной работы, решение практико-ориентированных задач.
16	Метрики программного кода	18	8	2	6	10	Опрос, собеседование по домашним заданиям самостоятельной работы, решение практико-ориентированных задач.
	В целом по дисциплине	288	118	32	86	170	Согласно учебному плану: контрольная работа, курсовая работа
	Итого в %		41	27	73	59	

5.3. Содержание семинаров, практических занятий

Наименование тем (разделов) дисциплины	Перечень вопросов для обсуждения на семинарах, практических занятиях	Формы проведения занятий
Управление разработкой ПО. Гибкие методологии (Agile)	Формирование команд (принципы, методика). Инструменты коллаборации в командах. Практики взаимодействия	Интерактивная форма: опрос, собеседование по домашним заданиям самостоятельной работы, решение практико-ориентированных задач с последующим коллективным обсуждением их результатов

Формирование требований при разработке ПО.	Сбор требований различными способами. Формирование итоговых требований.	Интерактивная форма: опрос, собеседование по домашним заданиям самостоятельной работы, решение практико-ориентированных задач с последующим коллективным обсуждением их результатов
Управление разработкой ПО. Проектное управление.	Типы проектных офисов. Основы организации работ	Интерактивная форма: опрос, собеседование по домашним заданиям самостоятельной работы, решение практико-ориентированных задач с последующим коллективным обсуждением их результатов
Масштабирование управления разработкой ПО.	Походы к масштабированию – портфели проектов (проектный подход) и SAFE (гибкий подход)	Интерактивная форма: опрос, собеседование по домашним заданиям самостоятельной работы, решение практико-ориентированных задач с последующим коллективным обсуждением их результатов
Тестирование ПО.	Формирование сценариев ручного тестирования на основе User Stories Формирование списка дефектов по результатам тестирования.	Интерактивная форма: опрос, собеседование по домашним заданиям самостоятельной работы, решение практико-ориентированных задач с последующим коллективным обсуждением их результатов
Сопровождение ПО.	Составление SLA.	Интерактивная форма: опрос, собеседование по домашним заданиям самостоятельной работы, решение практико-ориентированных задач с последующим коллективным обсуждением их результатов

Методология DevOps и ее расширения	Настройка конвейеров поставки (ci/cd pipelines). Используемые инструменты, типовые стадии для построения конвейера	Интерактивная форма: опрос, собеседование по домашним заданиям самостоятельной работы, решение практико-ориентированных задач с последующим коллективным обсуждением их результатов
Управление качеством производства ПО.	Работа с регламентными документами, проработка стратегии обеспечения качества программного продукта	Интерактивная форма: опрос, собеседование по домашним заданиям самостоятельной работы, решение практико-ориентированных задач с последующим коллективным обсуждением их результатов. защита контрольной работы
Основные идеи современной программной инженерии	Отличительные особенности программной инженерии от производственной. Современные подходы к управлению сложностью и к работе с меняющимися требованиями	Дополнительное обсуждение материала лекции Интерактивная форма: опрос, собеседование по домашним заданиям самостоятельной работы, решение практико-ориентированных задач с последующим коллективным обсуждением их результатов
Бизнес архитектура и техническая архитектура	Работа с требованиями для выработки проектных решений. Выявление и формализация нефункциональных требований, определяющих качество системы. Планирование взаимодействия с заинтересованными лицами на стадии реализации проекта	Интерактивная форма: опрос, собеседование по домашним заданиям самостоятельной работы, решение практико-ориентированных задач с последующим коллективным обсуждением их результатов

Современные варианты распределенных приложений	Контейнеризация приложений как универсальный подход для развертывания приложений. Docker и Kubernetes как примеры промышленной реализации	Работа с проектом. Упаковка приложения в контейнеры Интерактивная форма: опрос, собеседование по домашним заданиям самостоятельной работы, решение практико-ориентированных задач с последующим коллективным обсуждением их результатов
Атрибуты качества и сценарии атрибутов качества	Атрибуты качества как средство формализации нефункциональных требований. Принципы формирования, стратегии обеспечения	Интерактивная форма: опрос, собеседование по домашним заданиям самостоятельной работы, решение практико-ориентированных задач с последующим коллективным обсуждением их результатов
Паттерны проектирования	Паттерны как решение для стандартных проблем разработки ПО. Практика применения	Интерактивная форма: опрос, собеседование по домашним заданиям самостоятельной работы, решение практико-ориентированных задач с последующим коллективным обсуждением их результатов
Рефакторинг	Рефакторинг приложений как инструмент повышения качества. Инструменты и принципы	Интерактивная форма: опрос, собеседование по домашним заданиям самостоятельной работы, решение практико-ориентированных задач с последующим коллективным обсуждением их результатов

Мониторинг	Инструменты мониторинга. Обеспечение качества на стадии эксплуатации программного комплекса	Интерактивная форма: опрос, собеседование по домашним заданиям самостоятельной работы, решение практико-ориентированных задач с последующим коллективным обсуждением их результатов
Метрики программного кода	Работа с метриками для обеспечения стабильного качества проекта	Интерактивная форма: опрос, собеседование по домашним заданиям самостоятельной работы, решение практико-ориентированных задач с последующим коллективным обсуждением их результатов, защита курсовой работы

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 (Agile)	Различные методики гибкой разработки – XP, Kanban, Lean, Scrum	Изучение материалов лекций и литературы, предложенной преподавателем, поиск и анализ информации, содержащейся в Интернет-ресурсах. Разбор вопросов, отводимых на самостоятельное освоение
Формирование требований при разработке ПО	Изучение нормативных документов (ГОСТы)	Изучение материалов лекций и литературы, предложенной преподавателем, поиск и анализ информации, содержащейся в Интернет-ресурсах. Разбор вопросов, отводимых на самостоятельное

		освоение, выполнение домашних заданий самостоятельной работы
Управление разработкой ПО. Проектное управление.	Изучение нормативных документов по проектному управлению (PMBOOK)	Изучение материалов лекций и литературы, предложенной преподавателем, поиск и анализ информации, содержащейся в Интернет-ресурсах. Разбор вопросов, отводимых на самостоятельное освоение, выполнение домашних заданий самостоятельной работы
Масштабирование управления разработкой ПО	Подходы к масштабированию при проектном и гибком подходах	Изучение материалов лекций и литературы, предложенной преподавателем, поиск и анализ информации, содержащейся в Интернет-ресурсах. Разбор вопросов, отводимых на самостоятельное освоение, выполнение домашних заданий самостоятельной работы
Тестирование ПО	Подход TDD (Test Driven Development)	Изучение материалов лекций и литературы, предложенной преподавателем, поиск и анализ информации, содержащейся в Интернет-ресурсах. Разбор вопросов, отводимых на самостоятельное освоение, выполнение домашних заданий самостоятельной работы
Сопровождение ПО.	Передача в поддержку, работа с изменениями и реорганизация команды проекта	Изучение материалов лекций и литературы, предложенной преподавателем, поиск и анализ информации, содержащейся в Интернет-ресурсах. Разбор вопросов, отводимых на самостоятельное освоение, выполнение домашних заданий самостоятельной работы
Методология DevOps и ее расширения	Изучение практик DevOps	Изучение материалов лекций и литературы, предложенной преподавателем, поиск и анализ информации, содержащейся в Интернет-ресурсах. Разбор вопросов, отводимых на самостоятельное освоение, выполнение домашних заданий самостоятельной работы
Управление качеством производства ПО.	Модели зрелости – CMM и CMMI	Изучение материалов лекций и литературы, предложенной преподавателем, поиск и анализ

		информации, содержащейся в Интернет-ресурсах. Разбор вопросов, отводимых на самостоятельное освоение, выполнение домашних заданий самостоятельной работы. Выполнение контрольной работы.
Основные идеи современной программной инженерии	Современные подходы к управлению сложностью и неопределенности требований	Изучение материалов лекций и литературы, предложенной преподавателем, поиск и анализ информации, содержащейся в Интернет-ресурсах. Разбор вопросов, отводимых на самостоятельное освоение, выполнение домашних заданий самостоятельной работы
Бизнес архитектура и техническая архитектура	Основные элементы бизнес-архитектуры применительно к программным продуктам	Изучение материалов лекций и литературы, предложенной преподавателем, поиск и анализ информации, содержащейся в Интернет-ресурсах. Разбор вопросов, отводимых на самостоятельное освоение, выполнение домашних заданий самостоятельной работы
Современные варианты распределенных приложений	Современные подходы к организации распределенных приложений	Изучение материалов лекций и литературы, предложенной преподавателем, поиск и анализ информации, содержащейся в Интернет-ресурсах. Разбор вопросов, отводимых на самостоятельное освоение, выполнение домашних заданий самостоятельной работы
Атрибуты качества и сценарии атрибутов качества	Стандартные подходы к формированию сценариев атрибутов качества. Стратегии обеспечения качества	Изучение материалов лекций и литературы, предложенной преподавателем, поиск и анализ информации, содержащейся в Интернет-ресурсах. Разбор вопросов, отводимых на самостоятельное освоение, выполнение домашних заданий самостоятельной работы
Паттерны проектирования	Работа с паттернами	Изучение материалов лекций и литературы, предложенной преподавателем, поиск и анализ информации, содержащейся в Интернет-ресурсах. Разбор вопросов, отводимых на самостоятельное

		освоение, выполнение домашних заданий самостоятельной работы
Рефакторинг	Рефакторинг программного кода	Изучение материалов лекций и литературы, предложенной преподавателем, поиск и анализ информации, содержащейся в Интернет-ресурсах. Разбор вопросов, отводимых на самостоятельное освоение, выполнение домашних заданий самостоятельной работы
Мониторинг	Open-source инструменты мониторинга Zabbix, Elasticsearch, Prometheus	Изучение материалов лекций и литературы, предложенной преподавателем, поиск и анализ информации, содержащейся в Интернет-ресурсах. Разбор вопросов, отводимых на самостоятельное освоение, выполнение домашних заданий самостоятельной работы
Метрики программного кода	Внедрение инструментов сбора метрик в конвейеры сборки и поставки программного кода	Изучение материалов лекций и литературы, предложенной преподавателем, поиск и анализ информации, содержащейся в Интернет-ресурсах. Разбор вопросов, отводимых на самостоятельное освоение, выполнение домашних заданий самостоятельной работы. Выполнение курсовой работы.

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

Примерный перечень тем для подготовки к опросу, подготовки докладов, презентаций

Тема 1. Управление разработкой ПО. Гибкие методологии (Agile)

Введение в стандарт SWEBOOK. Идеи Agile. Принципы Agile. Основные роли в Скрам. События Скрам. Формирование требований в Agile. Основные практики Канбан. Роли в Канбан.

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 PMBOK. 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. Артефакты. 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.

ITIL. Модель управления услугами. Цепочка создания ценности. Принципы. Управление инцидентами. Управление проблемами. Управление изменениями. SLA. Управление конфигурациями. Управление релизами.

Тема 7. Методология DevOps и ее расширения.

Цели. Что такое DevOps. Сочетание с Agile. Культура взаимодействия. Базовые инструменты. Непрерывная интеграция. Запросы на слияние. Сборка. Модульное тестирование. TDD. Статический анализ кода. Среды разработки и тестирования. Динамические среды. Непрерывная поставка. Непрерывное развертывание. Управление репозиторием. Управление конфигурациями. Тесты развертывания. DevSecOps. Основные точки встраивания. Непрерывные измерения. Процесс измерений. Метрики. Контейнеризация. Оркестрация контейнеров. Паттерны и антипаттерны образования 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 и дается представление о подходе Cloud Native Apps (в трендах Гартнер рассматривается как топ-3 технология на 2023-2024 годы).

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.

Примеры практико-ориентированных задач

Практико-ориентированная задача 1. Инициация и гибкое планирование.

Спонсор проекта (СЕО компании) провел встречу и набросал общее видение: «Сделайте приложение, чтобы сотрудники видели свой баланс отпусков, задачи со склада, новости компании и могли заказать пропуск на машину. Через 3 месяца нам нужен хоть какой-то результат, покажите его на совете директоров. Бюджет выделен, команда сформирована». Как тимлид команды, использующей гибкий подход (Scrum), опишите ваши первые действия:

1. Разработайте Vision Statement или ключевые цели проекта, определив границы (Scope) первого релиза, учитывая временные ограничения.
2. Составьте стартовый Product Backlog: сформулируйте не менее 5 User Stories для MVP, используя стандартный шаблон. Проведите их первичную приоритизацию, применив одну из изученных техник (например, MoSCoW).
3. Какие разделы Устава проекта (Project Charter) необходимо согласовать со спонсором до начала первого спринта, чтобы легитимизировать гибкий процесс в рамках жестких временных ограничений?

Практико-ориентированная задача 2. Анализ требований и проектирование архитектуры

После первой встречи со стейкхолдерами выяснилось, что у "заказа пропуска на машину" есть жесткое требование безопасности и производительности: пропуск должен генерироваться за 2 секунды, а система заказа должна быть независима от сбоев HR-портала, где хранятся данные сотрудников. Аналитик команды опросил будущих пользователей и выяснил, что им нужно видеть актуальное состояние пропуска офлайн.

1. Классифицируйте выявленные требования: какие из них являются функциональными, а какие — атрибутами качества (нефункциональными)?

2. Предложите архитектурный стиль (например, монолит, микросервисы, клиент-сервер) для модуля "Заказ пропусков", который обеспечит независимость от сбоев HR-системы.

3. Опишите один сценарий атрибута качества (Quality Attribute Scenario) для требования "генерация пропуска за 2 секунды". Укажите: *стимул, источник стимула, окружение, артефакт, реакция, мера реакции*.

Практико-ориентированная задача 3. Управление качеством и тестирование в процессе разработки

Команда завершила проектирование и приступила к разработке модуля "Профиль сотрудника". Разработчик Петров, вдохновляясь принципами TDD, написал код, но тестировщица Анна обнаружила, что метод расчета налога на заработную плату выдает ошибку при отрицательном значении премии (хотя в реальности премий меньше нуля не бывает). Кроме того, анализатор кода (SonarQube) показал высокую сложность одного из методов (Cyclomatic complexity > 15).

1. Является ли ошибка с отрицательной премией дефектом, который нужно фиксировать? Ответ обоснуйте с точки зрения принципов тестирования и контекста использования.

2. Предложите стратегию тестирования для данного модуля: какие уровни (модульное, интеграционное) и типы (функциональное, нагрузочное) тестирования вы бы применили в первую очередь?

3. Какую метрику кода "подсветил" анализатор и почему её высокое значение - это сигнал к рефакторингу? Какое действие вы предложите разработчику?

Задание 4. Сборка, поставка и эксплуатация (Связка DevOps и ITIL)

Модуль "Заказ пропусков" готов и протестирован в среде разработки. Необходимо развернуть его на тестовом полигоне (Staging) для приёмочного тестирования заказчиком, а затем выпустить в промышленную эксплуатацию. Параллельно администраторы сообщают, что на продуктивном сервере базы данных упало место из-за логов старого модуля.

1. Опишите процесс непрерывной поставки (Continuous Delivery) для модуля: какие этапы должен пройти артефакт (код) от момента коммита в репозиторий (Git) до развертывания на Staging-среде?
2. Если в продуктивной среде возникнет инцидент (например, модуль перестал открываться), опишите разницу между Управлением инцидентами и Управлением проблемами согласно ITIL. Что вы сделаете в первую очередь?
3. Предложите два ключевых метрики мониторинга для модуля "Заказ пропусков" после его запуска, которые позволят отслеживать доступность сервиса и качество данных.

Задание 5. Рефакторинг, цифровой долг и управление изменениями

Через полгода успешной эксплуатации, команде поставлена новая задача: интегрировать в "Ассистент" функцию записи в корпоративную библиотеку. При изучении кодовой базы модуля "Профиль сотрудника" разработчики обнаружили, что класс User (используемый для профиля) содержит также методы для генерации пропусков и расчета налогов. Изменение этого класса приводит к ошибкам в других, несвязанных частях системы. На очередном код-ревью прозвучал термин "божественный объект" (God Object).

1. Какую классификацию паттернов проектирования (порождающие, структурные, поведенческие) и какой конкретный паттерн вы бы применили для разделения User на несколько специализированных классов с четкими обязанностями?
2. Опишите последовательность действий по рефакторингу для устранения "объекта", гарантируя, что мы не сломаем существующую функциональность.

3. Данная ситуация - пример цифрового долга. Как тимлид, какие шаги вы предпримете, чтобы включить работу над этим долгом в ближайший спринт, сбалансировав его с разработкой новой функции для библиотеки? Какие риски проекта (по PMBOK) здесь реализовались?

Примерные задания контрольной работы (семестр 5)

1. Написать 15 требований к веб-сайту заказа билетов в кино. Разделить на функциональные и качественные, дополнительно разделить в соответствии с моделью Кано. Требования должны удовлетворять критериям качества.

2. Написать 15 требований к portalу ИТ-поддержки. Разделить на функциональные и качественные, дополнительно разделить в соответствии с моделью Кано. Требования должны удовлетворять критериям качества.

3. Составить ER-диаграмму по веб-сайту заказа билетов. Не менее 6 сущностей, общий состав атрибутов не менее 40, связи много ко много не допускаются, минимум один пример связи один к одному.

4. Составить ER-диаграмму по portalу ИТ-поддержки. Не менее 6 сущностей, общий состав атрибутов не менее 40, связи много ко много не допускаются, минимум один пример связи один к одному.

5. Написать 15 пользовательский историй к веб-сайту заказа билетов в кино с указанием критериев приемки и разделением минимум на две роли.

6. Написать 15 пользовательский историй к portalу ИТ-поддержки с указанием критериев приемки и разделением минимум на две роли.

7. Написать 6 сценариев тестирования по веб-сайту заказа билетов с общим количеством шагов не менее 40. Обязательно должны быть использованы методы анализа граничных значений и таблицы альтернатив (сложно).

8. Написать 6 сценариев тестирования к portalу ИТ-поддержки с общим количеством шагов не менее 40. Обязательно должны быть использованы методы анализа граничных значений и таблицы альтернатив (сложно).

9. Написать базовые положения соглашения об уровне обслуживания веб-сайта заказа билетов со стороны внешнего подрядчика. Выделить три уровня приоритета, критерии определения приоритета и временные характеристики.

10. Написать базовые положения соглашения об уровне обслуживания портала ИТ-поддержки со стороны внешнего подрядчика. Выделить три уровня приоритета, критерии определения приоритета и временные характеристики.

Примерная тематика курсовой работы (семестр 6)

1. Процесс оценки качества ПО. 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 Agile методологий Scrum и XP 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

Критерии балльной оценки различных форм текущего контроля успеваемости содержатся в соответствующих методических рекомендациях кафедры «Математика и информатика».

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

Примеры тестовых заданий

1. Выберите из нижеперечисленных требований те, которые не удовлетворяют критериям качества требований:
 - a. Хранение данных о моделях автомобилей, их комплектации и ценах
 - b. Возможность сравнения автомобилей между собой по заданным параметрам
 - c. Уникальный дизайн приложения
 - d. Минимальное использование приложением ресурсов телефона.
 - e. Возможность авторизации в приложении по логину/паролю
2. Выберите из нижеперечисленных требований те, которые не удовлетворяют критериям качества требований:
 - a. Наличие рецептов различных коктейлей в приложении
 - b. Отсутствие рекламы в платной версии приложения
 - c. Дружелюбный дизайн приложения
 - d. Обеспечение надежности хранения информации
 - e. Возможность авторизации в приложении по логину/паролю
3. 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:

- a. Интервью
- b. Анкетирование
- c. Мозговой штурм
- d. Работа в «поле»
- e. Диаграммы использования

4. Ваша компания делает стандартные сайты-визитки для небольших компаний.

Какой способ сбора требований подходит больше всего:

- a. Интервью
- b. Анкетирование
- c. Мозговой штурм
- d. Работа в «поле»
- e. Диаграммы использования

5. Вы собираетесь в команде делать пилот нового приложения, аналоги которого практически отсутствуют. Выберите наиболее подходящие способы сбора требований для вашего приложения:

- a. Интервью
- b. Анкетирование
- c. Мозговой штурм
- d. Работа в «поле»
- e. Диаграммы использования

6. Вы собираетесь переделывать текущее приложение 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:

- a. Интервью
- b. Анкетирование
- c. Мозговой штурм
- d. Работа в «поле»

е. Диаграммы использования

7. Вам дано задание составить тестовые сценарии с обязательной проверкой граничных значений. Калькулятор рассчитывает тариф отдельно для людей старше 60 лет. Какие значения вы будете проверять для проверки границ:
- а. 60, 61, 62
 - б. 59, 60
 - с. 58, 59, 60, 61, 62
 - д. 58, 59, 60
 - е. 59, 60, 61
8. Вам дано задание внедрить статический анализатор кода. В какое место pipeline вы поставите этот анализатор:
- а. После автотестов UI
 - б. После компиляции приложения
 - с. Перед компиляцией приложения
 - д. Перед выкладкой в промышленную среду
 - е. Автоматическая проверка при написании кода
9. Вам дано задание внедрить динамический анализатор кода. В какое место pipeline вы поставите этот анализатор:
- а. После автотестов UI
 - б. После компиляции приложения
 - с. Перед компиляцией приложения
 - д. Перед выкладкой в промышленную среду
 - е. Автоматическая проверка при написании кода
10. Ваша команда собирается перейти на полностью дистанционную работу, выберите типы приложений, которые обязательно понадобятся вашей команде:
- а. Мессенджер
 - б. Видео-связь
 - с. Задачник

- d. Мониторинг работы
- e. База знаний

Примерные вопросы для подготовки к зачету (семестр 5)

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 тестировщиками 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 (семестр 6)

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.

Пример экзаменационного билета

**Федеральное государственное образовательное бюджетное
учреждение высшего образования
«Финансовый университет при Правительстве Российской Федерации»**

Кафедра: **Математика и информатика**

Дисциплина: **Программная инженерия**

Филиал: **Владикавказский**; Форма обучения: **Очная**

Семестр: **6** Направление: **09.03.04 Программная инженерия**

Профиль: **Технологии разработки программного обеспечения**

Экзаменационный билет №

1 вопрос (30 баллов): написать 15 требований к веб-сайту заказа билетов в кино.

Разделить на функциональные и качественные, дополнительно разделить согласно модели Кано. Требования должны удовлетворять критериям качества.

2 вопрос (15 баллов): опишите отличия между проектным управлением pmbook и работой по agile (scrum)

3 вопрос (15 баллов): паттерны проектирования. Порождающие паттерны (на выбранном примере). Описать компоненты паттерна, их взаимодействие, привести собственный пример. Желательно проиллюстрировать схемой вызова.

Подготовил: _____

На основе перечня теоретических вопросов и практико-ориентированных заданий, утвержденного на заседании кафедры «Математика и информатика» протокол № ____ от _____.2026 г.

Утверждаю:

Заведующий кафедрой _____

Дата _____.2026г.

Примеры оценочных средств для проверки индикаторов достижения компетенций, формируемых дисциплиной

Наименование компетенции	Наименование индикаторов достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции	Типовые контрольные задания
ПКП-5 Способность проектировать и реализовывать интеллектуальные	Демонстрирует знания основных методов машинного обучения и интеллектуального анализа данных,	Знать: основные методы машинного обучения, включая обучение с учителем, обучение без учителя, обучение с подкреплением, а также	Для заданной задачи выбрать адекватные инструменты моделирования, разработать план сбора и очистки

информационные системы	применяет готовые инструменты для создания интеллектуальных алгоритмов.	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	данных для моделирования, применить методы машинного обучения и проанализировать полученный результат
	Понимает особенности интеллектуальных информационных систем в части операций разработки, развертывания и сопровождения.	Знать: операции разработки, развертывания и сопровождения интеллектуальных информационных систем. Уметь: разрабатывать, развертывать и обслуживать интеллектуальные информационные системы.	Разработать план развертывания для программного комплекса, создать комплект документации для передачи в эксплуатацию
	Адаптирует практики создания программных продуктов, в том числе командные, для интеллектуальных информационных систем.	Знать: практики создания программных продуктов, включая командные методы, применяемые в интеллектуальных информационных системах. Уметь: адаптировать практики создания программных продуктов для интеллектуальных информационных систем.	Выбрать наиболее подходящий для данного продукта метод управления процессом создания ПО
	Организовывает сбор и подготовку данных для систем машинного обучения, в том числе потоковых, онлайн обучения.	Знать: гибкие методологии управления разработкой Уметь: участвовать в процессе управления работами, планировать работы по гибким методологиям и контролировать ход их выполнения	Разработать процесс (pipeline) сбора и подготовки (очистка, нормализация) данных под заданную задачу
	Понимает виды требований к программному обеспечению, составляет формализованное описание требований по нетехническому описанию функционала	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	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.
ПКП-3 Способность собирать, анализировать и управлять требованиями к программной системе			

	программной системы.	формировать и обосновывать архитектурные решения	Расчет емкости должен быть обоснован. Тематика разработки - портал ИТ-поддержки.
	Анализирует программную систему на предмет соответствия требованиям, выявляет проблемные участки, модифицирует архитектуру и дизайн программной системы для полного соответствия набору требований.	Знать: виды требований к программному обеспечению и их особенности. Уметь: анализировать и модифицировать архитектуру и дизайн программной системы в соответствии с требованиями	Провести анализ программной системы используя сценарии атрибутов качества
	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.	Знать: методы анализа программной системы на соответствие требованиям и выявление проблемных участков. Уметь: выделять, декомпозировать, приоритизировать требования, оценивать их реализуемость и тестируемость.	Выбрать и обосновать выбор архитектуры программного решения исходя из предоставленных требований
	Критически относится к сформулированным требованиям к программной	Знать: критическое отношение к сформулированным требованиям, выявление	Составить матрицу заинтересованных лиц, разработать план взаимодействия, провести анализ

	системе - выявляет внутренние противоречия, риски, связанные с реализацией тех или иных требований, воспринимает требования в более широком контексте постановки задачи, находит недостающие требования.	противоречий, рисков и недостающих требований. Уметь: находить внутренние противоречия, риски и недостающие требования в сформулированных требованиях к программной системе.	соответствия целей и заявленных требований, сформировать список противоречий, разработать план коммуникаций и финализации требований
	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	Для готового программного модуля, создать руководство пользователя программного продукта. Документация должна содержать необходимые сведения по установке, обеспечению надёжной работы продукта, справочное пособие для пользователя, демонстрационные версии, примеры документов, создаваемых при помощи данного программного продукта, обучающие программы.
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 ПО Уметь: выбирать подходящую методологию формирования требования, адаптировать под предметную область	Разработать комплект документов для программного комплекса (выбрать методологию, состав документов, содержание и нотацию, адресатов)
	Создает комплект программной и проектной документации к разрабатываемой	Знать: типы документов, их назначение и содержание Уметь: выбирать номенклатуру документации соответствующую типу проекта, формировать	Подготовить пакет документации в соответствии с требованиями ГОСТ34 или в

	информационной системе.	документы в соответствии с отраслевыми стандартами	соответствии с требованиями Scrum
	Критически анализирует программную и проектную документацию, строит на ее основе логические выводы.	Знать: основные методы критического анализа программной и проектной документации. Уметь: демонстрировать умение критически оценивать и дорабатывать программную и проектную документацию.	Разработка плана проекта с учётом стоимости, времени, не менее 5 исполнителей, не менее 3-х уровней вложенности. Составить реестр рисков не менее трёх разных типов и разработать план управления этими рисками. Для примера брать проект по разработке мобильного приложения по покупке страховок

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

Основная литература:

1. Гагарина, Л. Г. Технология разработки программного обеспечения: учебное пособие / Л.Г. Гагарина, Е.В. Кокорева, Б.Д. Сидорова-Виснадул; под ред. Л.Г. Гагариной. — Москва: ФОРУМ: ИНФРА-М, 2025. — 400 с. — ISBN 978-5-8199-0812-9. - URL: <https://znanium.ru/catalog/product/2183867> – Режим доступа: Электронно-библиотечная система Znanium.com – Текст: электронный.

Дополнительная литература:

2. Митус, К. Н., Программное обеспечение как услуга. Модели и инструменты оценки экономической эффективности: монография / К. Н. Митус, И. А. Кацко, А. А. Митус. — Москва: Русайнс, 2020. — 203 с. — ISBN 978-5-4365-4666-7. — URL: <https://book.ru/book/936005> – Режим доступа: Электронно-библиотечная система Book.ru. – Текст: электронный.

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

1. Электронная библиотека Финансового университета (ЭБ) <http://elib.fa.ru/>
2. Электронно-библиотечная система BOOK.RU <http://www.book.ru>
3. Электронно-библиотечная система «Университетская библиотека ОНЛАЙН» <http://biblioclub.ru/>
4. Электронно-библиотечная система Znanium <http://www.znaniy.com>
5. Электронно-библиотечная система издательства «ЮРАЙТ» <https://urait.ru/>
7. Электронно-библиотечная система издательства Лань <https://e.lanbook.com/>
9. Электронная библиотека Издательского дома «Гребенников» <https://grebennikon.ru/>

10. Методические указания для обучающихся по освоению дисциплины.

Методика освоения дисциплины предусматривает подготовку обучающихся к лекциям, семинарам и практическим занятиям, выполнение студентами самостоятельной внеаудиторной работы, в том числе – контрольной работы.

Рекомендации по подготовке к лекционным занятиям.

Для наиболее полного освоения дисциплины студентам необходимо:

перед каждой лекцией просматривать рабочую программу дисциплины, ее основные вопросы и рекомендуемую литературу. Это позволит сэкономить время на записывание основных вопросов темы;

перед очередной лекцией просматривать материалы предыдущих, чтобы освоение материала не оставляло пробелов.

Рекомендации по подготовке к семинарам, практическим занятиям.

Студентам следует:

проработать теоретический материал к занятию по рекомендованным литературным источникам и лекциям;

использовать при подготовке к занятию нормативно-правовые документы, научные публикации, информационный материал, рекомендуемый преподавателем;

перед занятиями задать вопросы по невыясненным в ходе самостоятельной подготовки темам или отдельным положениям темы;

в ходе занятия давать четкие и исчерпывающие ответы на вопросы;

на занятии демонстрировать понимание обсуждаемых тем и вопросов.

Студентам, пропустившим занятия по различным причинам, необходимо перед очередным занятием отработать пропущенный материал, подготовив его самостоятельно.

Методические рекомендации по выполнению различных форм самостоятельной работы

Студентам при организации самостоятельной работы следует руководствоваться Приказом Финансового университета № 1040/о от 11.05.2021г. «Об утверждении методических рекомендаций по планированию и организации внеаудиторной самостоятельной работы студентов по образовательным программам бакалавриата и магистратуры в Финансовом университете».

Самостоятельная работа содержит в себе различные виды и формы работ. Самостоятельная работа студентов включает в себя выполнение различного рода заданий, которые ориентированы на более глубокое усвоение материала изучаемой дисциплины. По теме учебной дисциплины студентам предлагается перечень заданий для самостоятельной работы.

В ходе изучения дисциплины предусмотрены следующие формы самостоятельной работы:

- подготовка к опросу;
- выполнение заданий самостоятельной работы,
- решение практико-ориентированных задач;
- выполнение контрольной работы (эссе, домашнего творческого задания, проектной работы);
- выполнение курсовой работы;
- подготовка к зачету и экзамену.

К выполнению заданий для самостоятельной работы предъявляются следующие требования: задания должны выполняться самостоятельно и представляться в установленный срок, а также должны соответствовать установленным требованиям по оформлению.

Студентам следует:

руководствоваться графиком самостоятельной работы, определенным РПД;

выполнять все плановые задания, выдаваемые преподавателем для самостоятельного выполнения, разбирать на занятиях и консультациях неясные вопросы;

прорабатывать соответствующие теоретические и практические разделы дисциплины, фиксируя неясные фрагменты для их обсуждения на консультации.

Методические рекомендации для обучающихся по выполнению контрольной работы

Контрольная работа является обязательной формой внеаудиторной самостоятельной работы студентов по дисциплине и может реализовываться как в письменном виде, так и с использованием информационных технологий и специализированных программных продуктов.

Цель выполнения контрольной работы, содержащей комплект заданий – овладение студентами навыками решения типовых расчетных задач, формирование учебно-исследовательских навыков, закрепление умений самостоятельно работать с различными источниками информации; проверка сформированности компетенций.

Целью выполнения контрольной работы является углубление и закрепление теоретических знаний и практических навыков студентов по дисциплине.

Контрольная работа по дисциплине выполняется по вариантам.

Содержание заданий контрольных работ охватывают основной материал соответствующих разделов (тем) дисциплин. Контрольные задания разрабатываются по многовариантной системе. Варианты контрольных работ равноценны по объему и сложности.

Контрольная работа выполняется студентом под руководством преподавателя кафедры «Математика и информатика», ведущим семинарские (практические) занятия.

Контрольная работа состоит из нескольких частей. Состав контрольной работы и очередность размещения отдельных частей:

- титульный лист;
- основная часть;
- список использованных источников;
- приложения (при наличии).

Титульный лист является первой страницей и заполняется по определенным правилам.

Основная часть выполняется согласно заданиям (вопросам) контрольных работ.

В список использованных источников включаются названия законодательных актов, нормативных документов, книг, статей, учебных пособий и т. п., которые, так или иначе, использовались студентом при выполнении работы.

В Приложения выносятся вспомогательные материалы, которые не содержат основную информацию, либо материалы, которые сложно разместить по тексту работы (большие схемы, таблицы, графические материалы, расчетные справочные данные, образцы первичных документов и т.п.). Непременным условием включения данных материалов в приложение является ссылка на них в тексте работы.

Требования к выполнению контрольной работы:

- четкость и последовательность изложения материала (решения) в соответствии с составленным планом;
- наличие обобщений и выводов, сделанных на основе изучения информационных источников по данной теме;
- предоставление в полном объеме решений имеющихся в задании практических задач;
- использование современных способов поиска, обработки и анализа информации;
- самостоятельность выполнения.

Требования к оформлению контрольной работы.

Контрольная работа выполняется на компьютере (гарнитура Times New Roman, шрифт 13 или 14) через 1-1,5 интервала с полями: верхнее, нижнее - 2; правое - 3; левое - 1,5. Отступ первой строки абзаца - 1,25. Нумерация страниц – внизу в центре.

Иллюстративный материал (схемы, диаграммы, рисунки, таблицы и др.) встраивается в текст работы или выносится в Приложения.

При написании допускаются только общепринятые сокращения (например, тыс. руб.).

В тексте обязательны ссылки на литературные источники, лучше всего постраничные.

Объем контрольной работы составляет не более 6 страниц, не включая таблиц, графиков и т.п. (при наличии).

Законченная контрольная работа, содержащая все требуемые элементы оформления, вставленная в папку (или файл) и скрепленная по левому краю, сдается на кафедру или непосредственно руководителю контрольной работы – преподавателю; ведущему семинарские (практические) занятия по дисциплине. Он осуществляет проверку контрольной работы, а также оказывает помощь при подготовке к ее защите.

Контрольная работа защищается в назначенные сроки. Защита работы проводится до начала сессии (в крайнем случае, до начала экзамена по соответствующему предмету). При защите студент кратко излагает основные положения работы, последовательность ее выполнения, свои предложения.

При защите работы студент должен свободно ориентироваться в изложенном материале работы; ответить на все замечания преподавателя; уметь отвечать на вопросы преподавателя по выполненной работе.

Оценка контрольных работ студентов проводится в процессе текущего контроля успеваемости студентов.

Критерии оценки контрольной работы

Оценка «отлично» (5-6 баллов) выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания вопросов контрольной работы /и/или умение уверенно применять их на практике при решении конкретных задач.

Оценка «хорошо» (3-4 балла) выставляется студенту, если он твердо знает материал контрольной работы, грамотно и по существу излагает его /и/или умеет применять полученные знания на практике при решении конкретных задач, но допускает некоторые неточности.

Оценка «удовлетворительно» (2 балла) выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, обнаружившему нарушения логической последовательности в изложении материала, но при этом владеющему основными вопросами, выносимыми на контрольную работу и необходимыми для дальнейшего обучения /и/или умеющему применять полученные знания по образцу в стандартной ситуации.

Оценки «неудовлетворительно» (0 баллов) заслуживает студент, который не знает большей части основного содержания выносимых на контрольную работу вопросов, тем дисциплины, допускает грубые ошибки в формулировках основных понятий /и/или не умеет использовать полученные знания при решении типовых практических задач.

При оценивании контрольной работы на «неудовлетворительно» она должна быть переделана (исправлена) в соответствии с полученными замечаниями, сдана на проверку заново и защищена не позднее срока окончания ее приёма и защиты.

Методические указания по выполнению курсовой работы

Студентам при выполнении курсовой работы следует руководствоваться Приказом Финансового университета № 1583/о от 02.06.2021г. «Об утверждении Положения о курсовом проектировании по образовательным программам высшего образования – программам бакалавриата и магистратуры в Финансовом университете».

Курсовая работа – важная форма изучения студентами дисциплины и одна из форм обязательной внеаудиторной самостоятельной учебной и научной работы студентов, осуществляемая под руководством преподавателя.

Целью курсовой работы является расширение, систематизация и закрепление теоретических и практических знаний, полученных студентами, а также выработка у студентов практических навыков самостоятельной работы.

Выполнение курсовой работы ориентировано на решение следующих задач:

- закрепление, углубление теоретических знаний и практическая реализация умений и навыков, полученных в ходе лекционных и практических (семинарских) занятий по дисциплине, по конкретной проблематике, практическая их реализация;
- получение опыта самостоятельного проведения исследования;
- формирование умения работать с источниками информации, Интернет-ресурсами, базами данных, информационно-аналитическими системами, навыков анализа и интерпретации показателей;
- развитие способности формулирования самостоятельных обоснованных выводов и рекомендаций, подготовки информационных и аналитических обзоров;
- развитие и повышение творческого потенциала студентов;
- получение опыта и формирование навыков подготовки, оформления и защиты научных работ для дальнейшего их использования при написании выпускной квалификационной работы.

Процесс подготовки и выполнения курсовой работы включает следующие этапы:

- 1) выбор из предложенного примерного списка темы курсовой работы и закрепление руководителя;
- 2) составление плана работы и согласование его с руководителем;
- 3) подбор литературы и сбор необходимой исходной информации по избранной теме; изучение выбранных источников литературы, а также обработка, систематизация, обобщение и анализ собранного фактического материала и полученной информации;
- 4) выполнение курсовой работы, формулирование выводов и рекомендаций; оформление курсовой работы в соответствии с требованиями;
- 5) подготовка к защите и защита курсовой работы.

Курсовая работа состоит из нескольких частей. Состав курсовой работы и очередность размещения отдельных частей в ней следующий:

- титульный лист (одна страница);
- содержание (одна страница);
- введение (2 страницы);
- основная часть, включающая две главы (20-25 страниц);
- заключение (2-3 страницы);
- список использованных источников (25-30 источников);
- приложения (при необходимости).

В процессе написания курсовой работы студент должен разобраться в теоретических вопросах избранной темы, самостоятельно проанализировать практический материал, разобрать и обосновать практические предложения.

Таким образом, качественно выполненная курсовая работа характеризуется двумя основными ее составляющими: во-первых, раскрытие экономической сущности изучаемой проблемы и изложение собственной позиции по дискуссионным вопросам; во-вторых, глубокий и всесторонний анализ действующей практики, исходя из конкретного фактического материала. Простое переписывание прочитанного материала; изложение дискуссионных вопросов без формирования собственной позиции; описание текущих инструкций, без аналитического осмысливания практического материала может послужить причиной низкой оценки выполненной курсовой работы. Т.к. такая работа не отражает умение автора самостоятельно и творчески использовать имеющийся материал и сочетать его с теоретическими знаниями, полученными при изучении данной дисциплины.

При подготовке основной части курсовой работы обязательными являются ссылки на использованные студентом источники (научную, методическую или учебную литературу, научные публикации и т. д.).

Курсовая работа, оформленная в соответствии с требованиями, подписывается студентом и представляется в электронном и бумажном носителях на кафедру не позднее, чем за три недели до установленного срока защиты курсовой работы.

Курсовая работа в обязательном порядке проверяется руководителем курсовой работы в системе «Антиплагиат». Руководитель проверяет курсовую работу и составляет о ней письменный отзыв, и при условии положительной оценки допускает курсовую работу к защите.

Завершающим этапом выполнения студентом курсовой работы является ее защита. Конкретную дату защиты устанавливает научный руководитель. В случае неявки студента в назначенную научным руководителем дату на защиту курсовой работы по уважительной причине, подтвержденной документально, научным руководителем устанавливается еще одна дата защиты курсовой работы в рамках установленного срока защиты курсовых работ.

На защите студент должен хорошо ориентироваться в представленной работе, уметь дать пояснения по использованным источникам информации, примененным методам исследования, полученным результатам, свободно отвечать на вопросы как теоретического, так и практического характера, относящиеся к теме работы.

Защита курсовой работы позволяет оценить полноту знания студентом исследованной темы, степень самостоятельности ее выполнения, уровень освоения им компетенций.

Оценка курсовой работы определяется преподавателем в процессе собеседования.

Основными критериями при оценивании курсовой работы являются:

1. Своевременное составление плана и библиографии по работе.
2. Актуальность темы исследования.
3. Правильное оформление работы.
4. Соответствие работы методическим рекомендациям.
5. Степень раскрытия темы.
6. Наличие анализа и обобщения по исследованию.
7. Аргументированность заключений по исследованию.
8. Процедура защиты – ответы на вопросы и замечания.

Конкретное распределение баллов отражается в Отзыве на курсовую работу.

Оценка результатов текущего контроля успеваемости и промежуточной аттестации обучающихся осуществляется в соответствии с Балльно-рейтинговой системой Финансового университета (Приказ Финансового университета № 2187/о от 01.10.2024 г. «Об утверждении Положения о проведении текущего контроля успеваемости и промежуточной аттестации студентов, обучающихся по образовательным программам высшего образования в Финансовом университете»).

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

11.1 Комплект лицензионного программного обеспечения:

- 1) Антивирусная защита Kaspersky Security для виртуальных и облачных сред;
- 2) Windows, Microsoft Office или Astra Linux, Libre Office.

11.2. Современные профессиональные базы данных и информационные справочные системы:

- Информационно-правовая система «Гарант»
- Информационно-правовая система «Консультант Плюс»
- Электронная энциклопедия: <http://ru.wikipedia.org/wiki/Wiki>
- Система комплексного раскрытия информации «СКРИН» -

<http://www.skrin.ru/>

11.3. Сертифицированные программные и аппаратные средства защиты информации:

Не используются.

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

Учебная аудитория для проведения учебных занятий, предусмотренных программой бакалавриата, оснащенная оборудованием и техническими средствами обучения

Аудитория № 36

Специализированная мебель:

Стол (учительский) – 1 шт.

Стол компьютерный – 1 шт.

Стол (студенческий) двухместный – 13 шт.

Стулья – 27 шт.

Доска меловая – 1 шт.

Технические средства обучения:

Компьютер в сборе – 1 шт.

Экран настенный – 1 шт.

Подключение к сети «Интернет» и обеспечение доступа в электронную информационно-образовательную среду Финансового университета

Учебная аудитория для проведения учебных занятий, предусмотренных программой бакалавриата, оснащенная оборудованием и техническими средствами обучения

Аудитория № 32

Специализированная мебель:

Стол компьютерный – 20 шт.

Стол (двухместный) – 7 шт.

Стул – 34 шт.

Шкаф – 1 шт.

Технические средства обучения:

Компьютер в сборе – 20 шт.

Мультимедиа-проектор – 1 шт.

Экран настенный – 1 шт.

Подключение к сети «Интернет» и обеспечение доступа в электронную информационно-образовательную среду Финансового университета

Помещение для самостоятельной работы обучающихся:

Кабинет № 55. Читальный зал:

Специализированная мебель:

Стол – 20 шт.

Стул – 40 шт.

Шкаф для книг – 4 шт.

Стеллаж книжный – 13 шт.

Стеллаж выставочный – 4 шт.

Технические средства обучения:

Компьютер в сборе – 6 шт.

Телевизор – 1 шт.

Помещение для самостоятельной работы обучающихся оснащено компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду Финансового университета