

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: j9KqU/i4sY+nEjm/qaXtXuxlZPmjSGYh  
Дата: 25.11.2025 г.

**И.И. Хасанов**

**Программная инженерия**

**Рабочая программа дисциплины**

для студентов, обучающихся по направлению подготовки:

09.03.03 - Прикладная информатика,

Образовательная программа «Прикладные информационные системы в экономике  
и финансах»

*Рекомендовано*

*Факультет информационных технологий и анализа больших данных  
(протокол № 03 от 16.12.2025 г.)*

*Одобрено*

*Кафедра информационных технологий  
(протокол № 12 от 03.12.2025 г.)*

## СОДЕРЖАНИЕ

1.	Наименование дисциплины	3
2.	Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине	3
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	6
5.1.	Содержание дисциплины	6
5.2.	Учебно-тематический план	8
5.3.	Содержание семинаров	9
6.	Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине	10
6.1.	Перечень вопросов, отводимых на самостоятельное освоение дисциплины, формы внеаудиторной самостоятельной работы	10
6.2.	Перечень вопросов, заданий, тем для подготовки к текущему контролю	11
7.	Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине	13
8.	Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины	19
9.	Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины	20
10.	Методические указания для обучающихся по освоению дисциплины	21
11.	Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень необходимого программного обеспечения и информационных справочных систем	22
12.	Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине	23

## 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
ПКН-3	Способность проектировать и реализовывать архитектуру и дизайн программной системы в соответствии с анализом задачи и требований к ней	Демонстрирует знание основных алгоритмов и структур данных, использует на практике простые структуры данных, оценивает сложность алгоритмов	<b>знать:</b> основные алгоритмы и структуры данных, принципы их применения и методы оценки сложности <b>уметь:</b> использовать простые структуры данных на практике, выбирать подходящие решения и анализировать эффективность алгоритмов
		Собирает, формулирует, систематизирует и анализирует функциональные и нефункциональные требования к информационной системе, выбирает архитектурные решения на их основе	<b>знать:</b> принципы сбора, формулирования, классификации и анализа функциональных и нефункциональных требований, а также основы выбора архитектурных решений <b>уметь:</b> выявлять и систематизировать требования, оценивать их влияние на архитектуру системы и обоснованно выбирать подходящие архитектурные решения
		Создает объектно-ориентированный код, инкапсулирующий условия задачи, производит декомпозицию задачи и проектирует систему в пределах одной платформы или технологии	<b>знать:</b> принципы объектно-ориентированного программирования, подходы к декомпозиции задач и проектированию программных систем в рамках выбранной платформы или технологии <b>уметь:</b> разрабатывать объектно-ориентированный код, корректно инкапсулировать логику задачи, выполнять декомпозицию и проектировать структуру системы в пределах используемой технологической среды

ПКН-5	Способность участвовать в документальном сопровождении разработки ИТ в рамках проектных групп, применять средства автоматизации управления проектами ИТ	Демонстрирует знание основ версионирования и управления изменениями при разработке ПО. Использует системы контроля версий для ведения совместной разработки	<b>знать:</b> основы версионирования, принципы управления изменениями и организацию совместной разработки <b>уметь:</b> использовать системы контроля версий для ведения командной разработки, отслеживания изменений и поддержания целостности проекта
		Демонстрирует знание основ тестирования программного обеспечения, умение создавать автоматизированные модульные и интеграционные тесты	<b>знать:</b> основы тестирования программного обеспечения, принципы модульного и интеграционного тестирования, а также подходы к автоматизации тестов <b>уметь:</b> разрабатывать автоматизированные модульные и интеграционные тесты, применять их для проверки корректности работы программных компонентов и анализа результатов тестирования
		Готовит документацию к программе, коммуницирует в пределах группы разработки и за ее границами о значимых аспектах информационной системы и информационной инфраструктуры в письменной и устной форме	<b>знать:</b> принципы подготовки технической и пользовательской документации, правила профессиональной коммуникации в команде разработки и с внешними участниками, а также основы описания информационных систем и инфраструктуры <b>уметь:</b> оформлять документацию к программным продуктам и ясно излагать значимые аспекты системы в устной и письменной форме внутри команды и за её пределами
		Демонстрирует знание жизненного цикла информационных систем, участвует в процессе разработки ПО на разных этапах	<b>знать:</b> основные модели и этапы жизненного цикла информационных систем, принципы организации процессов разработки ПО и роли участников этих процессов <b>уметь:</b> участвовать в разработке программного обеспечения на различных этапах жизненного цикла, ориентируясь в задачах, инструментах и требованиях каждого этапа

### 3. Место дисциплины в структуре образовательной программы

Дисциплина «Программная инженерия» относится к «Общепрофессиональному циклу».

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

Очная форма обучения

Вид учебной работы по дисциплине	Всего (в з/е и часах)	Семестр 3 (в часах)
<b>Общая трудоёмкость дисциплины</b>	3/108	108
<b>Контактная работа-Аудиторные занятия</b>	50	50
<b>Лекции</b>	16	16
<b>Семинары, практические занятия</b>	34	34
<b>Самостоятельная работа</b>	58	58
Вид текущего контроля	Контрольная работа	Контрольная работа
Вид промежуточной аттестации	Зачет	Зачет

## **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 и гибкие методологии**

Рассматриваются современные подходы к управлению процессом разработки программного обеспечения, включая традиционные и гибкие методологии. Обсуждаются роли, артефакты и процессы в жизненном цикле разработки, а также применение Scrum, Kanban и гибридных моделей. Анализируются принципы адаптивного планирования, итеративной поставки, управления рисками, командного взаимодействия и повышения эффективности процесса разработки в условиях быстро меняющихся требований.

#### **Тема 2. Формирование требований и бизнес-архитектура программных систем**

Изучаются методы выявления, анализа, формализации и согласования требований к программному обеспечению. Рассматривается связь между бизнес-архитектурой организации и архитектурой программного решения, переход от бизнес-моделей и процессов к архитектурно значимым требованиям и драйверам. Анализируются техники моделирования, формулирования функциональных и нефункциональных требований, а также их влияние на проектирование архитектуры.

#### **Тема 3. Масштабирование процессов разработки и современные подходы программной инженерии**

Рассматриваются методы масштабирования разработки ПО как на уровне процессов, так и на уровне организаций. Освещаются практики применения фреймворков SAFe, LeSS и других моделей масштабируемого Agile. Отдельное внимание уделено современным идеям программной инженерии, включающим научный подход, управление сложностью и эволюцией программных систем. Анализируются универсальные инженерные практики, применимые на этапах анализа, проектирования, разработки, поставки и эксплуатации.

#### **Тема 4. Архитектура распределённых приложений и атрибуты качества**

Рассматриваются современные архитектурные подходы к созданию распределённых систем, включая микросервисную архитектуру, облачно-нативные решения, контейнеризацию и оркестрацию. Анализируется влияние атрибутов качества —

масштабируемости, надёжности, производительности, модифицируемости, доступности и др. — на архитектурные решения. Изучаются сценарии атрибутов качества и методы их формализации для оценки архитектурных компромиссов и проектирования устойчивых систем.

### **Тема 5. Паттерны проектирования и рефакторинг**

Изучаются структурные, порождающие и поведенческие паттерны проектирования, применяемые для повышения гибкости, расширяемости и сопровождаемости программных систем. Рассматривается роль паттернов в структурировании архитектуры и устранении избыточной сложности. Отдельно анализируются техники рефакторинга, направленные на улучшение внутреннего качества кода без изменения внешнего поведения, включая устранение «запахов кода», оптимизацию зависимостей и восстановление архитектурной целостности.

### **Тема 6. Тестирование и обеспечение качества ПО**

Рассматриваются методы и процессы тестирования ПО, включая модульное, интеграционное, системное, нагрузочное, регрессионное и автоматизированное тестирование. Поднимаются вопросы построения стратегии обеспечения качества, организации тестовой документации и внедрения практик CI/CD. Изучаются современные инструменты автоматизации, принципы построения надежных тестов и методики оценки качества программных продуктов.

### **Тема 7. Сопровождение, эксплуатация и мониторинг программных систем**

Изучаются процессы сопровождения и эксплуатации программного обеспечения после его внедрения, включая управление обновлениями, устранение ошибок и технический долг. Рассматриваются современные подходы к мониторингу, наблюдаемости, логированию и анализу поведения системы в реальной среде. Анализируются практики DevOps и SRE, направленные на обеспечение непрерывной работоспособности, устойчивости и контроля качества работающих систем.

### **Тема 8. Метрики качества кода и процессы улучшения программных систем**

Рассматриваются количественные и качественные метрики кода и архитектуры, позволяющие оценивать сложность, сопровождаемость, надёжность и эффективность программных систем. Обсуждаются методы анализа метрик, использование инструментов статического анализа и способы принятия инженерных решений на основе объективных данных. Изучаются процессы улучшения качества ПО, включая оптимизацию архитектуры, устранение технического долга и внедрение инженерных практик для поддержания долгосрочной устойчивости системы.

## 5.2. Учебно-тематический план

№ п/п	Наименование тем(разделов) дисциплины	Трудоемкость в часах				
		Всего	Контактная работа - Аудиторная работа			Самостоя тельная работа
			Общая, в т.ч.:	Лекции	Семинары, практическ ие занятия	
1	Управление разработкой ПО и гибкие методологии	12	4	2	2	8
2	Формирование требований и бизнес-архитектура программных систем	12	6	2	4	6
3	Масштабирование процессов разработки и современные подходы программной инженерии	12	6	2	4	6
4	Архитектура распределённых приложений и атрибуты качества	12	6	2	4	6
5	Паттерны проектирования и рефакторинг	14	6	2	4	8
6	Тестирование и обеспечение качества ПО	14	8	2	6	6
7	Сопровождение, эксплуатация и мониторинг программных систем	14	6	2	4	8
8	Метрики качества кода и процессы улучшения программных систем	18	8	2	6	10
	Итого	108	50	16	34	58



### 5.3. Содержание семинаров

Наименование тем (разделов) дисциплины	Перечень вопросов для обсуждения на семинарах, практических занятиях	Формы проведения занятий
Управление разработкой ПО и гибкие методологии	Принципы гибких методологий разработки. Роли и артефакты в Scrum и Kanban. Планирование итераций и оценка задач в гибких командах.	Групповая работа.
Формирование требований и бизнес-архитектура программных систем	Методы выявления и формализации требований. Определение архитектурно значимых требований. Связь бизнес-архитектуры и функциональных требований.	Групповая работа.
Масштабирование процессов разработки и современные подходы программной инженерии	Масштабирование Agile в крупных командах. Основные идеи современной программной инженерии. Управление сложностью и эволюцией больших систем.	Дискуссия и разбор кейсов.
Архитектура распределённых приложений и атрибуты качества	Микросервисная архитектура и облако-нативные приложения. Атрибуты качества программных систем. Сценарии атрибутов качества и их влияние на архитектурные решения.	Командный анализ архитектурных решений. Проработка сценариев атрибутов качества.
Паттерны проектирования и рефакторинг	Применение паттернов проектирования в решении типовых задач. Выявление и устранение «запахов» кода. Техники рефакторинга и улучшения архитектуры.	Практическое моделирование. Анализ кода и применение паттернов и рефакторинга.
Тестирование и обеспечение качества ПО	Виды тестирования и их назначение. Автоматизация тестирования и роль CI/CD. Подходы к обеспечению качества и анализу результатов тестирования.	Практическая работа. Разбор и создание тестов, работа с инструментами автоматизации.
Сопровождение, эксплуатация и мониторинг программных систем	Особенности сопровождения ПО. Средства мониторинга и наблюдаемости. Управление техническим долгом и эксплуатационными рисками.	Групповая работа. Анализ кейсов сопровождения и построение схем мониторинга.
Метрики качества кода и процессы улучшения программных систем	Использование метрик для оценки качества кода. Статический анализ и интерпретация метрик. Управление качеством и совершенствование программных систем.	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 и гибкие методологии	История и эволюция гибких методологий. Отличия Scrum, Kanban и гибридных подходов. Основы управления рисками в проектах разработки ПО.	Изучение учебных материалов (статьи, стандарты). Изучение литературы.
Формирование требований и бизнес-архитектура программных систем	Техники моделирования процессов и бизнес-правил. Подходы к документированию требований. Методы анализа влияния требований на архитектуру.	Изучение документации.
Масштабирование процессов разработки и современные подходы программной инженерии	Структура и принципы SAFe и LeSS. Основные инженерные практики современного ПО. Методы управления сложностью и техническим долгом.	Изучение учебных материалов (статьи, стандарты). Изучение литературы.
Архитектура распределённых приложений и атрибуты качества	Архитектурные стили распределённых систем. Принципы проектирования облако-нативных решений. Формализация и анализ сценариев атрибутов качества.	Изучение учебных материалов (статьи, стандарты). Изучение литературы.
Паттерны проектирования и рефакторинг	Классификация паттернов проектирования. Инструменты автоматизации рефакторинга. Практики улучшения архитектурной структуры кода.	Изучение учебных материалов (статьи, стандарты). Изучение литературы.
Тестирование и обеспечение качества ПО	Стратегии тестирования в больших системах. Архитектура и принципы CI/CD. Инструменты автоматизации тестирования и метрики качества тестов.	Изучение учебных материалов (статьи, стандарты). Изучение литературы.
Сопровождение, эксплуатация и мониторинг программных систем	Подходы DevOps и SRE. Принципы наблюдаемости в программных системах. Методы анализа инцидентов и оптимизации эксплуатации.	Изучение учебных материалов (статьи, стандарты). Изучение литературы.
Метрики качества кода и процессы улучшения программных систем	Методы и инструменты статического анализа. Подходы к оценке сложности и сопровождаемости кода. Процессы принятия решений на основе метрик.	Изучение учебных материалов (статьи, стандарты). Изучение литературы.

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

### **Примерные задания 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**

**ПКН-3 Способность проектировать и реализовывать архитектуру и дизайн программной системы в соответствии с анализом задачи и требований к ней**

**1) Демонстрирует знание основных алгоритмов и структур данных, использует на практике простые структуры данных, оценивает сложность алгоритмов**

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

**Знать:** основные алгоритмы и структуры данных, принципы их применения и методы оценки сложности

**Уметь:** использовать простые структуры данных на практике, выбирать подходящие решения и анализировать эффективность алгоритмов

**Типовые контрольные задания**

Реализовать программный модуль, использующий выбранную структуру данных для решения поставленной задачи, обосновать выбор структуры данных, оценить временную и пространственную сложность реализованных алгоритмов

**2) Собирает, формулирует, систематизирует и анализирует функциональные и нефункциональные требования к информационной системе, выбирает архитектурные решения на их основе**

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

**Знать:** принципы сбора, формулирования, классификации и анализа функциональных и нефункциональных требований, а также основы выбора архитектурных решений

**Уметь:** выявлять и систематизировать требования, оценивать их влияние на архитектуру системы и обоснованно выбирать подходящие архитектурные решения

### **Типовые контрольные задания**

Разработать и обосновать архитектурное решение информационной системы на основе собранных и систематизированных функциональных и нефункциональных требований

**3) Создает объектно-ориентированный код, инкапсулирующий условия задачи, производит декомпозицию задачи и проектирует систему в пределах одной платформы или технологии**

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

**Знать:** принципы объектно-ориентированного программирования, подходы к декомпозиции задач и проектированию программных систем в рамках выбранной платформы или технологии

**Уметь:** разрабатывать объектно-ориентированный код, корректно инкапсулировать логику задачи, выполнять декомпозицию и проектировать структуру системы в пределах используемой технологической среды

### **Типовые контрольные задания**

Разработать объектно-ориентированный программный модуль, выполняющий декомпозицию поставленной задачи, корректно инкапсулирующий её логику и демонстрирующий проектирование системы в рамках выбранной платформы или технологии

**ПKN-5 Способность участвовать в документальном сопровождении разработки ИТ в рамках проектных групп, применять средства автоматизации управления проектами ИТ**

**1) Демонстрирует знание основ версионирования и управления изменениями при разработке ПО. Использует системы контроля версий для ведения совместной разработки**

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

**Знать:** основы версионирования, принципы управления изменениями и организацию совместной разработки

**Уметь:** использовать системы контроля версий для ведения командной разработки, отслеживания изменений и поддержания целостности проекта

### **Типовые контрольные задания**

Выполнить совместную работу над программным проектом с использованием системы контроля версий, оформляя изменения в соответствии с принципами версионирования и управления изменениями

**2) Демонстрирует знание основ тестирования программного обеспечения, умение создавать автоматизированные модульные и интеграционные тесты**

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

**Знать:** основы тестирования программного обеспечения, принципы модульного и интеграционного тестирования, а также подходы к автоматизации тестов

**Уметь:** разрабатывать автоматизированные модульные и интеграционные тесты, применять их для проверки корректности работы программных компонентов и анализа результатов тестирования

**Типовые контрольные задания**

Разработать и выполнить автоматизированные модульные и интеграционные тесты для программного компонента, обосновав выбор подходов к тестированию и проанализировав полученные результаты

**3) Готовит документацию к программе, коммуницирует в пределах группы разработки и за ее пределами о значимых аспектах информационной системы и информационной инфраструктуры в письменной и устной форме**

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

**Знать:** принципы подготовки технической и пользовательской документации, правила профессиональной коммуникации в команде разработки и с внешними участниками, а также основы описания информационных систем и инфраструктуры

**Уметь:** оформлять документацию к программным продуктам и ясно излагать значимые аспекты системы в устной и письменной форме внутри команды и за её пределами

**Типовые контрольные задания**

Подготовить комплект технической документации к программному продукту и представить ключевые особенности информационной системы и её инфраструктуры в устной и письменной форме для команды разработки и внешних участников

**4) Демонстрирует знание жизненного цикла информационных систем, участвует в процессе разработки ПО на разных этапах**

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

**Знать:** основные модели и этапы жизненного цикла информационных систем, принципы организации процессов разработки ПО и роли участников этих процессов

**Уметь:** участвовать в разработке программного обеспечения на различных этапах жизненного цикла, ориентируясь в задачах, инструментах и требованиях каждого этапа

### **Типовые контрольные задания**

Выполнить работу над программным проектом на нескольких этапах жизненного цикла информационной системы, продемонстрировав понимание моделей и последовательности стадий разработки ПО

### ***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***

1. Необходимо собрать требования к системе дистанционного обучения. Документация отсутствует, а пользователи неохотно идут на контакт, опасаясь изменений своих рабочих процессов. Какой способ сбора требований будет наиболее эффективным в данной ситуации?



2. Компания разрабатывает типовые корпоративные порталы для малого бизнеса. Какой способ сбора требований позволит наиболее быстро и точно определить потребности заказчика?
3. Команде предстоит разработать пилотную версию нового приложения для управления проектами, аналогов которого практически нет. Какие методы сбора и уточнения требований целесообразно использовать в этом случае?
4. Планируется модернизация существующей системы учёта рабочего времени с переходом на новую технологическую платформу. В работе участвуют опытные пользователи, хорошо знающие текущий функционал. Какой способ сбора требований будет оптимальным?
5. Руководителю проекта поручено внедрение системы мониторинга программных решений. Какую модель управления проектами следует выбрать для эффективного внедрения?
6. Необходимо организовать разработку модульной информационной системы для корпоративного заказчика. Какой метод управления разработкой наиболее подходит для такого типа проекта?
7. Поставлена задача развернуть новую виртуализированную инфраструктуру предприятия. Какую методологию управления стоит применить в данном случае?
8. Планируется внедрение набора процессов управления услугами— управление инцидентами, проблемами, изменениями, запросами и конфигурациями. Какая модель управления наиболее соответствует этим задачам?
9. Команде поручено создание распределённого решения на основе микросервисной архитектуры. Какую методологию управления разработкой следует использовать для достижения гибкости и качества?
10. При разработке корпоративного приложения требуется выстроить конвейер автоматизированной сборки и поставки программных решений. С каких основных инструментов следует начать настройку конвейера?
11. При создании конвейера поставки программных решений специалисты по информационной безопасности требуют интеграции инструментов контроля и защиты. Какие подходы можно предложить для обеспечения безопасности на всех этапах разработки?

12. Разработчики и тестировщики жалуются на большое количество ошибок из-за различий в настройках тестовых и рабочих сред. Какое решение позволит устранить эту проблему?
13. Перечислите основные практики интеграции разработки и эксплуатации программных систем и объясните, чем они отличаются друг от друга.
14. При внесении изменений в один из взаимосвязанных модулей корпоративной системы необходимо убедиться в стабильности всей системы. Какие виды тестирования следует обязательно провести перед вводом изменений в эксплуатацию?
15. Специалисты по информационной безопасности выявили, что в программном коде разработчики хранят пароли в открытом виде. Какой вид анализа или тестирования поможет устранить и предотвратить подобные ошибки в будущем?

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

### ***Нормативные 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»

### ***Основная литература:***

1. 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]

2. 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]

### ***Дополнительная литература:***

1. Горбенко, А.О. Программная инженерия : Учебник / А.О. Горбенко, А.А. Попов Электрон. дан. Москва : КноРус, 2025 307 с. Режим доступа: book.ru Internet access <https://book.ru/book/958136> ISBN 978-5-406-14751-1. [БИК ID: RU\bookru\bibl\958136]

2. Маран, М. М. Программная инженерия [ Электронный ресурс ] / Маран М. М. 3-е изд., стер. Санкт-Петербург : Лань, 2022 196 с. Книга из коллекции Лань - Информатика <https://e.lanbook.com/book/189470> ISBN 978-5-8114-9323-4. [БИК ID: RU-LAN-BOOK-189470]

## **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. Электронно-библиотечная система "Университетская библиотека ОНЛАЙН" <http://biblioclub.ru/>

4. Электронно-библиотечная система издательства "Лань" <https://e.lanbook.com/>

5. Образовательная платформа "ЮРАЙТ" <https://urait.ru/>

6. Научная электронная библиотека eLibrary.ru <http://elibrary.ru>

7. Информационно-образовательный портал Финуниверситета: <https://org.fa.ru>

## **10. Методические указания для обучающихся по освоению дисциплины**

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

### *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.

## **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. Комплект свободно распространяемого программного обеспечения
2. Пакет офисных программ
3. Kaspersky

### **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. Язык программирования Python 3. <https://pythonworld.ru/>
5. Размещение ЭУК: <https://www.campus.fa.ru/>
6. Создание тестов, система Moodle, осуществление прокторинга (асинхронный, синхронный): <https://edu.fa.ru/>
7. [www.iteam.ru/publications/strategy/](http://www.iteam.ru/publications/strategy/) - ITeam-Технологии корпоративного управления.
8. 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