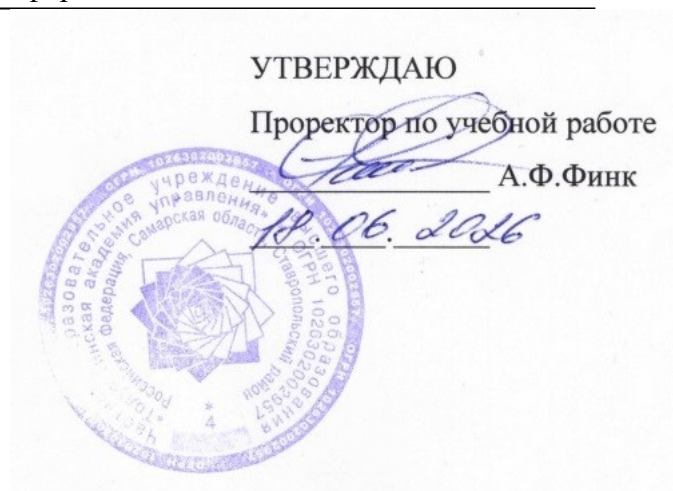


Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Финк Анастасия Федоровна
Должность: Проректор по учебной работе
Дата подписания: 01.07.2026 19:32:36
Уникальный программный ключ:
2431bd5130e74d20a9fc74baab365dd497e3afa3

НОУ ВО «Тольяттинская академия управления»

Кафедра

прикладной информатики и высшей математики



Б1.В.15

РАБОЧАЯ ПРОГРАММА

Учебная дисциплина	<i>Программная инженерия</i>
По направлению подготовки	<i>09.03.03 «Прикладная информатика»</i>
Профиль (программа бакалавриата)	<i>«Дизайн и разработка цифровых программных продуктов»</i>
Форма обучения	<i>Очная</i>

Программа дисциплины рассмотрена (актуализирована) и утверждена на заседании кафедры прикладной информатики и высшей математики

Протокол заседания № 10 от «25» мая 2026 г.

Заведующий кафедрой Стрекалова Наталья Борисовна

1. ОБЩАЯ ИНФОРМАЦИЯ

Рабочая программа дисциплины «Программная инженерия» составлена в соответствии с требованиями федерального государственного образовательного стандарта высшего образования – бакалавриат по направлению подготовки 09.03.03 «Прикладная информатика», утвержденного приказом Министерства образования и науки Российской Федерации 19.09.2017 №922 (с изменениями и дополнениями от 26.11.2020, 08.02.2021, 19.07.2022, 27.02.2023), и учебного плана направления подготовки 09.03.03 «Прикладная информатика», профиль «Дизайн и разработка цифровых программных продуктов» (программа бакалавриата).

Трудоемкость дисциплины: 5 ЗЕТ / 180 академических часа, в том числе 32 часа контактной работы и 112 часов самостоятельной работы обучающихся.

Распределение часов дисциплины по семестрам и видам занятий (по учебному плану)

Вид учебной работы		Количество часов								
		Всего по учебному плану	Семестры							
			1	2	3	4	5	6	7	8
Контактная работа (всего):		32							32	
в том числе:										
Лекции		12							12	
Практические занятия		16							16	
Контроль самостоятельной работы (КСР)		4							4	
Самостоятельная работа (всего):		112							112	
Виды промежуточной аттестации (экзамен)		экзамен 36							экзамен 36	
ОБЩАЯ трудоемкость дисциплины:	Часы:	180							180	
	Зач. ед.:	5							5	

2. ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ

Цель дисциплины – формирование целостного представления о ходе разработки программного продукта и принципах DevOps: предпроектный анализ, проектирование, технико-экономическое обоснование, кодирование, тестирование, внедрение и сопровождение; знакомство с типовыми моделями жизненного цикла программного обеспечения; изучение принципов и методов оценки качества программного продукта, обеспечение надежности его функционирования, управление процессом разработки.

Задачи дисциплины:

- сформировать у обучающихся компетенции в области создания надежного, качественного программного обеспечения

- научить обучающихся разрабатывать программные комплексы в соответствии с техническим заданием и с использованием средств автоматизации проектирования;
- научить обучающихся работать индивидуально и в качестве члена группы, состоящей из специалистов различных направлений и квалификаций;
- развивать навыки разработки программного обеспечения.

3. МЕСТО ДИСЦИПЛИНЫ В УЧЕБНОМ ПЛАНЕ

Учебная дисциплина «Программная инженерия» относится к дисциплинам части Блока 1. Дисциплины (модули), формируемой участниками образовательных отношений. Изучение данной дисциплины базируется на материале, изученном в дисциплинах «Управление ИТ-проектами», «Информационная безопасность», «Разработка гибридных приложений», «Контроль качества программных продуктов», «Архитектура web-приложений», «Теория систем и системный анализ». Знания, умения и навыки, приобретенные в результате изучения дисциплины, будут востребованы при изучении при написании выпускной квалификационной работы.

4. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ ДЛЯ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ ОБУЧАЮЩИХСЯ

Результаты освоения образовательной программы (компетенции обучающихся) устанавливаются в соответствии с федеральным государственным образовательным стандартом по направлению подготовки и профессиональными стандартами, соответствующими профессиональной деятельности выпускников, а также на основе анализа требований работодателей, предъявляемых к выпускникам. Планируемые результаты обучения по дисциплине (знания, умения, навыки) обеспечивают достижение результатов освоения образовательной программы.

Шифр и название компетенции	Индикаторы компетенции	Планируемые результаты обучения по дисциплине
ПК-1 Способен выявлять информационные потребности пользователей и составлять техническое задание на разработку цифрового программного продукта	ПК-1.2. Разрабатывает техническое задание, формализует требования заказчика к разработке цифрового программного продукта	Знать: - понятия, особенности и архитектуру программного обеспечения, профессиональные и этические требования профессионального сообщества программистов к деятельности по разработке информационных систем; Уметь: - составлять техническую документацию процесса разработки программного обеспечения; Владеть: - навыками разработки отдельных технических документов;

ПК-2 Способен проектировать цифровой программный продукт	ПК-2.1. Разрабатывает модель цифрового программного продукта с учетом внедрения передовых ИТ-решений прикладной сферы	Знать: - процесс и средства разработки информационных систем и программных комплексов; - факторы сложности разработки информационных систем, технологии, подходы и принципы к исследованию и созданию информационных систем, эргономические требования к ним Уметь: - определять факторы сложности разработки программных систем, выбирать адекватные технологии для их проектирования Владеть: - навыками критического анализа технической информации
ПК-4 Способен управлять процессом разработки цифрового программного продукта	ПК-4.1. Применяет гибкие технологии управления и цифровые средства контроля жизненного цикла разработки цифрового программного продукта	Знать: - суть процесса разработки программного обеспечения и виды деятельности в нем, модели жизненного цикла разработки программного обеспечения; - принципы конфигурационного управления и управления требованиями, суть методологии MSF - принципы и средства управления программным проектом и командой разработчиков; Уметь: - планировать этапы разработки информационной системы с использованием инструментальных средств; Владеть: - навыками управления программным проектом с помощью программно-технических средств, - навыками работы с сервисами контроля версий программного продукта

	ПК-4.2. Обеспечивает разработку проектной и технической документации цифрового программного продукта	Знать: - нормативные документы и стандарты в процессе создания информационных систем Уметь: - применять нормативные документы и стандарты в процессе создания цифровых программных продуктов Владеть: - навыками разработки цифровых программных продуктов с использованием нормативных документов и стандартов.
ПК-5 Способен обеспечивать качество работы цифрового программного продукта	ПК-5.3. Осуществляет работы по интеграционному тестированию цифрового программного продукта с внешним окружением	Знать: - международные и отечественные стандарты оценки качества разработки цифровых программных продуктов; - методы и способы тестирования программного обеспечения - основы эффективной организации разработки цифрового программного продукта и интеграцию всех технологических процессов (в том числе на основе облачных решений) для высокого качества программного продукта Уметь: - осуществлять эффективную организацию разработки цифрового программного продукта и интеграцию всех технологических процессов (в том числе на основе облачных решений) для высокого качества программного продукта - проводить тестирование цифровых программных продуктов Владеть: - навыками проверки работоспособности информационных систем и эффективность интеграции всех технологических процессов - навыками построения плана и протокола тестирования

5. СТРУКТУРА И ОБЪЕМ ДИСЦИПЛИНЫ

Семестр изучения: 7

Раздел, модуль	Подраздел, тема	Виды учебной работы					Промежуточная аттестация в часах	Форма текущего контроля	Формир уемые компете нции
		Контактная работа (в часах)			Самостоятельная работа				
		Лекц ии	Практи ческие занятия	КСР (пропо рциона льно темам)	в часах	формы организации самостоятельной работы			
	Тема 1. Введение в программную инженерию	1		-	8	Повторение лекционного материала	-	Устный опрос	ПК-1.2
	Тема 2. Разработка сложных программных систем	1		-	8	Повторение лекционного материала	-	Устный опрос	ПК-2.1
	Тема 3. Оценка качества процессов создания программного обеспечения	1		-	8	Повторение лекционного материала	-	Устный опрос	ПК-2.1 ПК-4.2
	Тема 4. Процесс разработки программного обеспечения	1	2	-	18	Повторение лекционного материала, подготовка доклада	-	Проверка выполненных работ, выступление с докладом	ПК-2.1, ПК-4.1, ПК-4.2

	Тема 5. DevOps и инфраструктурная инженерия	2	4	-	20	Повторение лекционного материала, подготовка к практическим занятиям	-	Устный опрос	ПК-2., ПК-4.1, ПК-5.3
	Тема 6. Архитектура программного обеспечения	2	2	-	18	Повторение лекционного материала, подготовка к практическим занятиям	-	Устный опрос	ПК-1.2, ПК-4.1
	Тема 7. Тестирование ПО	2	4	-	18	Повторение лекционного материала, подготовка к практическим занятиям	-	Проверка выполненных работ	ПК-5.3
	Тема 8. Управление программным проектом	2	4	-	14	Подготовка к практическим занятиям		Проверка выполненных работ	ПК-4.1
Форма промежуточной аттестации (экзамен)		-	-		-	Подготовка к промежуточной аттестации	36	-	-
Всего		12	16	4	112	-	36		
		180							

6. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Тема 1. Введение в программную инженерию

Понятие программной инженерии. Предпосылки и история возникновения программной инженерии. Основные определения: информатика, системотехника, бизнес-инжиниринг. Программное обеспечение: определение, свойства. Особенности программного обеспечения по Ф.Бруксу. Основные этапы развития технологий программирования. Профессиональные и этические требования профессионального сообщества программистов.

Тема 2. Разработка сложных программных систем

Факторы сложности разработки программных систем. Суть блочно-иерархического подхода к исследованию и созданию программных систем. Понятие декомпозиции. Принципы блочно-иерархического подхода. Использование CASE-технологий. Основные положения технологии RAD.

Тема 3. Оценка качества и технико-экономическое обоснование проектов программных средств

Международные стандарты оценки качества процессов разработки ПО: международные стандарты ISO 9000 (ISO 9000-9004), CMM, рабочая версия международного стандарта ISO/IEC 15504. Понятие технико-экономического обоснования программного средства. Прогнозирование использования ограниченных ресурсов при создании крупных комплексов программ. Базовые характеристики затрат на разработку программных средств.

Тема 4. Процесс разработки программного обеспечения

Понятие процесса разработки программного обеспечения. Содержание стандартного процесса. Деятельность по совершенствованию процесса. Pull/Push-стратегии в разработке ПО. Понятие фазы и вида деятельности в разработке ПО. Типовые модели жизненного цикла ПО: каскадная и спиральная модель. Другие модели жизненного цикла ПО: итерационная модель, V-образная модель, инкрементная (пошаговая) модель. Модели жизненного цикла в промышленных технологиях MSF, RUP, XP.

Тема 5. DevOps и инфраструктурная инженерия

Различие DevOps и DevSecOps. Цели методологии DevOps. Общее представление об инфраструктуре современной разработки. Обзор используемых инструментов. Циклы и этапы разработки ПО. Гибкие методологии разработки: Agile, Scrum, Kanban. Обзор систем управления версиями. Обзор архитектуры операционных систем на примере Linux. Основы сетевой архитектуры. Авторизация, аутентификация и безопасность пользователей. Основы работы с языками программирования в DevOps: Python, Bash,

YAML, JSON. Синтаксис разметки YAML. Синтаксис разметки JSON. Командная оболочка Bash: практические навыки. Использование Python для решения типовых задач в DevOps. Изучение задач, которые решает виртуализация. Технологии виртуализации. Системы управления виртуализацией

Контейнеризация на примере Docker и ContainerD. Практические навыки работы с Docker. Написание Dockerfile и использование Docker Compose, работа с Docker Swarm. Обзор систем управления конфигурациями. Концепция «инфраструктура как код». Процессы CI/CD: integration, delivery, deployment. Инфраструктура тестирования, сборки и доставки ПО. Практическое знакомство с Jenkins, TeamCity и Gitlab CI. Преимущества и недостатки микросервисной архитектуры. Отказоустойчивость систем. Nginx, HAProxy, Envoy. Кластеризация.

Тема 6. Архитектура программного обеспечения

Понятие архитектуры ПО. Множественность точек зрения при разработке ПО. Язык UML. Виды диаграмм. Управление требованиями. Виды и свойства требований. Варианты формализации требований. Ошибки при документировании требований. Цикл работы с требованиями. Конфигурационное управление. Единицы конфигурационного управления. Управление версиями. Управление сборками. Понятие baseline.

Тема 7. Тестирование ПО

Качество продукта и качество процесса. Стандарты качества. Методы обеспечения качества ПО. Процесс тестирования. Виды тестирования. Работа с ошибками. Эргономические требования к программному обеспечению. Человек в системе «человек-техника-среда». Основные термины и понятия эргономики.

Тема 8. Управление программным проектом

Проект и управление проектом. Треугольник ограничений проекта. Стандарт PMBOK. 34 компетенции IT-менеджера. Управление командой проекта. Ролевая модель команды. Модели организации команд (административная, модель хаоса, открытая архитектура). Планирование проекта и контроль. Метрики проекта. Средства управления проектом.

7. САМОСТОЯТЕЛЬНАЯ РАБОТА ОБУЧАЮЩИХСЯ

В рамках данной учебной дисциплины обучающиеся выполняют самостоятельную внеаудиторную работу в виде повторения пройденного материала по всем темам, выполнения практических заданий по темам 5-8 и подготовки доклада по теме 4.

8. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

При изучении данной дисциплины используются следующие образовательные технологии:

Тема занятия	Вид образовательной технологии	Форма проведения занятия
Тема 1. Введение в программную инженерию	Традиционная технология	Лекция
Тема 2. Разработка сложных программных систем	Традиционная технология	Лекция
Тема 3. Оценка качества и технико-экономическое обоснование проектов программных средств	Традиционная технология	Лекция
Тема 4. Процесс разработки программного обеспечения	Традиционная технология, интерактивная технология	Лекция
		Выступление с докладами и их обсуждение
Тема 5. DevOps и инфраструктурная инженерия	Традиционная технология	Лекция
		Практическое занятие
Тема 6. Архитектура программного обеспечения	Традиционная технология	Лекция
		Практическое занятие
Тема 7. Тестирование ПО	Традиционная технология	Лекция
		Практическое занятие
Тема 8. Управление программным проектом	Традиционная технология	Лекция
		Практическое занятие

9. ТЕКУЩИЙ КОНТРОЛЬ УСПЕВАЕМОСТИ И ПРОМЕЖУТОЧНАЯ АТТЕСТАЦИЯ ПО ДИСЦИПЛИНЕ

9.1. Формы контроля по дисциплине

Текущий контроль. В процессе изучения учебной дисциплины обучающиеся участвуют в устных теоретических опросах, выполняют практические задания, выступают с докладом. Результаты выполнения практических заданий являются основанием для выставления оценок текущего контроля по данной учебной дисциплине. Выполнение всех практических заданий является обязательным для всех обучающихся. Обучающиеся, не выполнившие в полном объеме все задания, не допускаются к сдаче экзамена по данной учебной дисциплине.

Промежуточная аттестация. Для контроля усвоения обучающимися данной дисциплины учебным планом предусмотрен экзамен, который проводится в форме устного ответа на вопрос и выполнения практического задания.

9.2. Оценочные материалы (оценочные средства) для текущего контроля успеваемости и промежуточной аттестации по дисциплине

Текущий контроль.

Список вопросов для устного опроса

Вопросы для проведения устного опроса

по теме «Введение в программную инженерию»

1. Что такое программная инженерия?
2. Назовите дату зарождения программной инженерии как отдельной науки.
3. В чем отличие программной инженерии от информатики?
4. В чем отличие программной инженерии от системотехники?
5. Приведите примеры дисциплин информатики и программной инженерии (дисциплины не путать с учебными предметами).
6. Что такое ПО?
7. С какими иными видами человеческой деятельности соотносится создание ПО в данном разделе?
8. Что такое процесс создания ПО?
9. Причины отсутствия универсального процесса разработки ПО.
10. Почему возможно и целесообразно стандартизировать процесс на уровне компании?
11. Какие методологии разработки ПО поддерживают понятие конкретного процесса и какими средствами?

Вопросы для проведения устного опроса

по теме «Разработка сложных программных систем»

1. Что такое процесс создания ПО?
2. Расскажите о причинах отсутствия универсального процесса разработки ПО.
3. Почему возможно и целесообразно стандартизировать процесс на уровне компании?
4. Что такое стандартный и конкретный процессы и как они соотносятся?
5. Чем отличаются между собой текущий и конкретный процессы? Какие методологии разработки ПО поддерживают понятие конкретного процесса и какими средствами?
6. Что такое процесс создания ПО?
7. Расскажите о причинах отсутствия универсального процесса разработки ПО.
8. Почему возможно и целесообразно стандартизировать процесс на уровне компании?

9. Что такое стандартный и конкретный процессы и как они соотносятся?
10. Чем отличаются между собой текущий и конкретный процессы? Какие методологии разработки ПО поддерживают понятие конкретного процесса и какими средствами?

Вопросы для проведения устного опроса

по теме «Оценка качества процессов создания программного обеспечения»

1. Какими характеристиками должен обладать качественный программный продукт?
2. Какие нефункциональные требования определяют качество программного продукта?
3. Какая роль тестирования в обеспечении качества программного продукта?
4. Какие типы тестов используют для проверки качества программного продукта?
5. Для чего применяется регрессионное тестирование?
6. Для чего применяется MicrosoftTestManager? Какие он имеет функциональные возможности?
7. Для чего используется LabManagement при тестировании?
8. Для чего применяют рефакторинг кода?
9. Приведите признаки некачественного кода.

Вопросы для проведения устного опроса

по теме «Процесс разработки программного обеспечения»

1. Что включает понятие «технология разработки программного обеспечения»?
2. Что определяет жизненный цикл программного обеспечения?
3. Поясните содержание каскадной модели разработки программного обеспечения.
4. Поясните содержание итерационной спиральной модели разработки программного обеспечения.
5. Поясните содержание итеративной модели разработки программного обеспечения.
6. Что должен обеспечивать эффективный подход к управлению процессом разработки ПО?
7. Что понимается под зрелостью процессов для компании, разрабатывающей ПО?
8. Приведите основные положения гибкого подхода к созданию ПО.
9. Приведите основное назначение методологии управления жизненным циклом приложений.
10. Какие инструментальные средства предлагает компания Microsoft для управления жизненным циклом приложений?

Вопросы для проведения устного опроса

по теме «Технико-экономическое обоснование проектов программных средств»

1. Краткая характеристика программных средств и стадии ее разработки
2. Определение трудоемкости разработки программных средств
3. Определение состава группы исполнителей
4. Расчет и построение сетевого плана-графика разработки и внедрения программных средств
5. Организация документирования программных средств
6. Требования к документации программных средств
7. Планирование документирования программных средств
8. Состав и содержание документов программных средств
9. Стандарты документирования программного обеспечения.

*Вопросы для проведения устного опроса
по теме «Архитектура программного обеспечения»*

1. Понятие архитектуры и задачи ее описания.
2. Основные классы архитектур программных средств.
3. Взаимодействие между подсистемами и архитектурные функции.
4. Контроль архитектуры программных средств.
5. Что такое архитектура программного обеспечения?
6. Важность архитектуры программного обеспечения
7. Каким способом программное обеспечение развертывается?
8. Кем обслуживаться программное обеспечение в процессе эксплуатации?
9. Требования к качеству программного обеспечения (безопасность, производительность, возможность параллельной обработки, интернационализация и конфигурация)?

*Вопросы для проведения устного опроса
по теме «Тестирование ПО»*

1. В чем отличие тестирования от обеспечения качества? Цель тестирования.
2. Какие виды деятельности выполняет тестировщик в процессе тестирования?
3. Какие существуют виды тестирования по глубине покрытия?
4. Какие уровни качества выставляются по результатам каждого вида тестирования по глубине покрытия?
5. Какие существуют тестовые активности?
6. На какой глубине покрытия выполняется каждая тестовая активность?
7. Какие существуют виды тестирования по знанию кода?
8. Какие существуют виды тестирования по степени автоматизации?
9. Какие существуют виды тестирования по изолированности компонентов?

10. Какие существуют виды тестирования в зависимости от объекта?
11. Какие существуют нефункциональные виды тестирования?
12. Комбинация тестов для первой поставки программного обеспечения на тестирование. Комбинация тестов для последующих поставок программного обеспечения на тестирование.

*Вопросы для проведения устного опроса
по теме «Управление программным проектом»*

1. Взаимосвязь управления проектами и функционального менеджмента.
2. Перспективы развития управления проектами. Переход к проектному управлению: задачи и этапы решения.
3. Классификация базовых понятий управления проектами. Классификация типов проектов. Цель и стратегия проектов. Результат проекта. Управление параметрами проекта. Проектный цикл.
4. Общая характеристика программных проектов.
5. Процессы управления проектом. Уровни зрелости процессов управления проектами.
6. Модели жизненного цикла ИТ-продукта. Соотношение жизненного цикла ИТ-решения и жизненного цикла проекта.
7. Теории управления программным проектом.
8. Классификация методов, моделей и стандартов разработки программного обеспечения. Методологии быстрой адаптивной разработки Agile(SCRUM).
9. Методологии разработки и внедрения ИТ-решений.

Промежуточная аттестация.

Список вопросов для подготовки к экзамену

1. Понятие программной инженерии. Предпосылки и история возникновения программной инженерии.
2. Основные определения: информатика, системотехника, бизнес-инжиниринг.
3. Программное обеспечение: определение, свойства.
4. Особенности программного обеспечения по Ф.Бруксу.
5. Профессиональные и этические требования профессионального сообщества программистов.
6. Основные этапы развития технологий программирования.
7. Факторы сложности разработки программных систем.
8. Суть блочно-иерархического подхода к исследованию и созданию программных систем.

9. Понятие декомпозиции. Принципы блочно-иерархического подхода.
10. Назначение и использование CASE-технологий в программной инженерии.
11. Основные положения технологии RAD.
12. Международные стандарты оценки качества процессов разработки ПО: международные стандарты ISO 9000 (ISO 9000-9004),
13. Международные стандарты оценки качества процессов разработки ПО: CMM, рабочая версия международного стандарта ISO/IEC 15504.
14. Понятие процесса разработки программного обеспечения. Содержание стандартного процесса.
15. Деятельность по совершенствованию процесса.
16. Pull/Push-стратегии в разработке ПО.
17. Понятие фазы и вида деятельности в разработке ПО. Каскадная модель деятельности: достоинства и недостатки.
18. Работа с рисками в каскадной модели деятельности.
19. Спиральная модель деятельности: достоинства и недостатки, ограничения модели.
20. Спиральная модель деятельности: условия завершения спирали, секторы витка спирали.
21. Другие модели жизненного цикла ПО: итерационная модель, V-образная модель, инкрементная (пошаговая) модель.
22. Модель жизненного цикла MSF (Microsoft Solution Framework)
23. Модель жизненного цикла RUP (Rational Unified Process)
24. Модель жизненного цикла XP (Extreme Programming).
25. Понятие архитектуры ПО. Множественность точек зрения при разработке ПО.
26. Язык UML. Виды диаграмм.
27. Управление требованиями. Виды и свойства требований. Варианты формализации требований.
28. Ошибки при документировании требований. Цикл работы с требованиями.
29. Конфигурационное управление. Единицы конфигурационного управления.
30. Управление версиями. Управление сборками. Понятие baseline.
31. Понятие технико-экономического обоснования программного средства.
32. Базовые характеристики затрат на разработку программных средств.
33. Качество продукта и качество процесса. Стандарты качества.
34. Методы обеспечения качества ПО.
35. Процесс тестирования. Виды тестирования. Работа с ошибками.

36. Эргономические требования к программному обеспечению. Человек в системе «человек-техника - среда».
37. Проект и управление проектом. Треугольник ограничений проекта.
38. Стандарт РМВОК. 34 компетенции IT-менеджера.
39. Управление командой проекта. Ролевая модель команды.
40. Модели организации команд (административная, модель хаоса, открытая архитектура).
41. Планирование проекта и контроль. Метрики проекта.

Средства управления проектом.

Порядок проведения текущего контроля успеваемости и промежуточной аттестации по дисциплине для инвалидов и лиц с ОВЗ предусмотрен Положением о текущем контроле успеваемости и промежуточной аттестации обучающихся.

10. РЕСУРСНАЯ СОСТАВЛЯЮЩАЯ

Для проведения занятий лекционного типа по данной дисциплине используются аудитории с медиа-оборудованием (проектор, экран, ноутбук) и учебной мебелью.

Для проведения занятий семинарского типа (практических занятий) по данной дисциплине используются компьютерные классы, оснащенные компьютерами с доступом в Интернет и необходимым программным обеспечением.

Для самостоятельной работы обучающихся используется помещение для самостоятельной работы обучающихся, оснащенное компьютерами с необходимым программным обеспечением и доступом в Интернет и электронную информационно-образовательную среду вуза.

Для проведения контроля самостоятельной работы по данной дисциплине используются компьютерные классы, оснащенные компьютерами с доступом в Интернет и необходимым программным обеспечением.

Для проведения текущего контроля успеваемости и промежуточной аттестации по данной дисциплине используются компьютерные классы, оснащенные компьютерами с доступом в Интернет и необходимым программным обеспечением.

Перечень лицензионного программного обеспечения

№ п/п	Наименование	Тип ресурса
1	Microsoft Office	Сублицензионный договор АО «СофтЛайн Трейд» № /131 от 10.07.2020. Период действия договора и лицензий бессрочный.
2	Microsoft Windows	

3	Visual Studio Enterprise (для учащихся, преподавателей и лабораторий)	Лицензионное соглашение Microsoft - Open Value Subscription для решений Education
4	КонсультантПлюс - справочно-правовая система «Комментарии законодательства» сетевая (смарт-комплект Оптимальный - отечественного производства)	Лицензионный договор ООО «Консультант Плюс Тольятти» договор № 251 от 01.01.2024 (лицензия бессрочная, договор ежегодно продлеваемый)
5	Антивирус Касперского 10 для Windows Workstations (Kaspersky Endpoint Security для бизнеса – стандартный Russian Edition, отечественного производства)	Сублицензионный договор АО «СофтЛайн Трейд» №Tr000947217 от 04.12.2025 срок действия 10.02.2026 - 11.02.2028. Тип лицензии - проприетарная (17E0-251210-090335-980-xxxx)

Перечень свободно распространяемого программного обеспечения

1. Браузеры Yandex, Google Chrome;
2. IDLE Python – бесплатная среда разработки устанавливается вместе с Python
3. Облачные сервисы Yandex, Google, Miro и др.;
4. Foxit Reader – Russian – бесплатное прикладное программное обеспечение для просмотра электронных документов в стандарте PDF;
5. СПС КонсультантПлюс - справочно-правовая система отечественного производства в свободном доступе в интернет.

В соответствии с Положением о создании специальных условий для инвалидов и лиц с ОВЗ информационно-технологическая база образовательного процесса предусматривает использование материально-технических средств с учетом различных нозологий инвалидов и лиц с ОВЗ.

11. ЛИТЕРАТУРА

11.1. Основная литература

№ п/п	Библиографическое описание	Тип издания	Количество в библиотеке
1.	Введение в программную инженерию : учебник / В. А. Антипов [и др.]. - Москва : Инфра-М, 2025. - 330 с. - ISBN 978-5-16-103172-8 (online). - ЭБС Знаниум. - URL: https://znanium.ru/catalog/document?id=455435	учебник	ЭБС
2.	Мартишин, С. А. Основы теории надежности информационных систем : учеб. пособие / С. А. Мартишин, В. Л. Симонов, В. М. Храпченко. - Москва : Инфра-М, 2026. - 254 с. - ISBN 978-5-16-106294-4 (online). - ЭБС Знаниум. - URL: https://znanium.ru/catalog/document?id=476639	учебное пособие	ЭБС

11.2 Дополнительная литература

1. Черников, Б. В. Управление качеством программного обеспечения : учебник / Б. В. Черников. - Москва : Инфра-М, 2019. - 240 с. - ISBN 978-5-16-102705-9 (online). - ЭБС Знаниум. - URL: <https://znanium.ru/catalog/document?id=339309>
2. Гагарина, Л. Г. Проектирование и архитектура программных систем : учеб. пособие / Л. Г. Гагарина, А. Р. Федоров, П. А. Федоров. - 2-е изд., испр. и доп. - Москва : Инфра-М, 2025. - 333 с. - ISBN 978-5-16-108394-9 (online). - ЭБС Знаниум. - URL: <https://znanium.ru/catalog/document?id=451039>
3. Трояновский, В. М. Программная инженерия информационно-управляющих систем в свете прикладной теории случайных процессов : учеб. пособие / В. М. Трояновский. - Москва : Инфра-М, 2024. - 325 с. - ISBN 978-5-16-106183-1 (online). - ЭБС Знаниум. - URL: <https://znanium.ru/catalog/document?id=430148>

Периодические издания:

1. Открытые системы. СУБД : журнал. – УБД ИВИС. – URL: <https://eivis.ru/browse/publication/64072>
2. Программные продукты и системы: международный научно-практический журнал. – ЭБС Знаниум. - URL : <https://znanium.ru/catalog/magazines/issues?ref=f9bfbfd0e-239e-11e4-99c7-90b11c31de4c>

11.3. Современные профессиональные базы данных, информационные справочные

системы, электронные библиотечные системы:

1. Znanium: электронно-библиотечная система. – URL: <http://znanium.ru>.
2. eLIBRARY.RU: научная электронная библиотека. – URL: <https://elibrary.ru/>.
3. Цифровая библиотека IPRsmart. – URL: <https://www.iprbookshop.ru/>.
4. Консультант плюс: справочно-правовая система. – URL: T:\consultantplus\cons.exe.
5. УБД ИВИС. – URL : <https://eivis.ru/basic/details>.
6. Электронная библиотека ТАУ. – URL: <http://83.234.207.58/MarcWeb2/Default.asp>.

12. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Успешное освоение дисциплины «Программная инженерия» предполагает посещение лекционных и практических занятий, самостоятельную работу над изучением дополнительного теоретического материала. Рекомендуется перед каждым занятием повторять материал предыдущей лекции, изучать документы, источники основной и дополнительной литературы по пройденной теме, структурировать изученный. Такая работа с учебным материалом позволит закрепить знания, более эффективно подготовиться к устным опросам и сдаче экзамена по дисциплине, а также выявить «проблемные» места в знаниях и подготовить вопросы к преподавателю.

При выполнении практических заданий по темам 7-8 обучающимся следует обратиться к опыту программирования на языке Python. Первоначальные практические навыки работы обучающиеся получают на практических занятиях в аудитории с преподавателем. Для самостоятельной отработки этих навыков рекомендуется использовать IDLE, либо воспользоваться аудиториями Академии, специально отведенными для самостоятельной работы. Обучающиеся могут повторять уже знакомые алгоритмы выполнения работ и проанализировать полученные результаты.

Все задания и теоретические материалы размещаются в локальной сети и электронной информационно-образовательной среде Академии.

При работе над докладами обучающемуся следует самостоятельно изучить необходимый материал, обобщить и структурировать его так, чтобы донести новую информацию до обучающихся наиболее понятно. Объем доклада должен быть не меньше 4-5 страниц печатного текста. Текст доклада должен быть оформлен в соответствии с требованиями Академии к оформлению текстовых учебных документов. Собранный материал необходимо визуализировать (таблицы, схемы, форматы, примеры) и представить на слайдах презентации. Все виды работ (выступление с докладом, участие в обсуждениях) являются элементами текущего контроля и оцениваются преподавателем.

Выполненные практические работы сдаются преподавателю непосредственно в аудитории либо одним из следующих способов: сохранение в электронной информационно-образовательной среде, отправка преподавателю на почтовый ящик. При отправке преподавателю выполненной работы, по почте обучающемуся следует обеспечить личную идентификацию. Как правило, в теме или тексте письма указывается курс, ФИО обучающегося, дисциплина, тема, по которой выполнена работы. Некоторые практические задания не могут быть сделаны только в рамках выделенного объема контактной работы (в аудитории) и «доделываются» в часы самостоятельной работы. Сдача таких работ на проверку осуществляется теми же самыми способами, что и по окончании практических занятий. Результаты проверки выполненных работ доводятся до сведения обучающегося во время аудиторных занятий и/или в часы КСР, а также размещаются в электронной информационно-образовательной среде.

Консультации по выполнению практических и самостоятельных работ, обсуждение отметок и допущенных ошибок осуществляется во время КСР на кафедре прикладной информатики или в аудитории по расписанию. Консультации могут осуществляться также посредством асинхронного (почта, ЭИОС) и синхронного (zoom) коммуникационного взаимодействия по предварительной договоренности с преподавателем.

Формой промежуточного контроля по дисциплине выступает экзамен, который проводится в форме устного ответа на теоретический вопрос и выполнения практического задания. Критерии допуска к экзамену и условия его сдачи озвучиваются преподавателем на первых занятиях по дисциплине.

К концу освоения дисциплины обучающийся должен предоставить полностью выполненные практические задания на проверку. По результатам текущего контроля и проверки индивидуального задания обучающийся допускается / не допускается к экзамену.

13. ЛИСТ СОГЛАСОВАНИЯ

Составил:

Н.Б. Стрекалова, д.п.н., доцент



(подпись)

Заведующий кафедрой

Н.Б. Стрекалова, д.п.н., доцент



(подпись)

Заведующий выпускающей кафедрой

Н.Б. Стрекалова, д.п.н., доцент



(подпись)

Директор БИК

О.В. Балакина



(подпись)

Начальник ООУП

С.В. Фирсова



(подпись)