

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Астраханский государственный университет имени В. Н. Татищева»
(Астраханский государственный университет им. В. Н. Татищева)

СОГЛАСОВАНО
Руководитель ОПОП

УТВЕРЖДАЮ
Заведующий кафедрой ИТ

_____ Ю.А. Головкин

_____ А.Н. Марьенков

«13» июня 2024 г.

«13» июня 2024 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

«ПРОГРАММНАЯ ИНЖЕНЕРИЯ»

Составитель(и)	Железняков Д.В., ассистент; Морозов Б.Б., к.х.н., доцент кафедры ИТ;
Направление подготовки / специальность	09.03.02 Информационные системы и технологии
Направленность (профиль) ОПОП	Технологии разработки и администрирования информационных систем
Квалификация (степень)	бакалавр
Форма обучения	очная
Год приёма	2022
Курс	3
Семестр(ы)	6

1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)

1.1. Целями освоения дисциплины (модуля) «Программная инженерия» являются познакомить студентов с технологиями создания и развития сложных, многоверсионных, тиражируемых программных средств и баз данных требуемого качества.

1.2. Задачи освоения дисциплины (модуля):

- познакомить с функциональными и технологическими стандартами разработки программных комплексов;
- познакомить с содержанием основных этапов разработки программных комплексов – стадий и процессов жизненного цикла программного продукта;
- познакомить с критериями качества программного продукта;
- познакомить с современными технологиями разработки программного обеспечения;
- сформировать умения формулировать требования к создаваемым программным продуктам и документировать их;
- познакомить с технологиями тестирования как одного из методов обеспечения качества программного продукта.

2. МЕСТО ДИСЦИПЛИНЫ (МОДУЛЯ) В СТРУКТУРЕ ОПОП

2.1. Учебная дисциплина (модуль) «Программная инженерия» относится к обязательной части и осваивается в 6 семестре.

2.2. Для изучения данной учебной дисциплины (модуля) необходимы следующие знания, умения, навыки, формируемые предшествующими учебными дисциплинами (модулями):

- *Информатика;*
- *Основы программирования;*
- *Базы данных.*

Знания: основные понятия информатики, принципы действия технических средств информатики – компьютеров, программное обеспечение компьютеров, базовые типы данных.

Умения: работа в интегрированных системах программирования, применение базовых алгоритмов в программировании.

Навыки: разработки алгоритмов, применение инструментария интегрированных сред программирования для решения различных прикладных задач.

2.3. Последующие учебные дисциплины (модули) и (или) практики, для которых необходимы знания, умения, навыки, формируемые данной учебной дисциплиной (модулем):

- *Инженерный практикум;*
- *IT-предпринимательство;*
- *Дипломный проект.*

3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

Процесс освоения дисциплины (модуля) направлен на формирование элементов следующей(их) компетенции(ий) в соответствии с ФГОС ВО и ОПОП ВО по данному направлению подготовки / специальности:

а) профессиональных (ПК): ПК-1, ПК-4

ПК-1. Способен проводить научные исследования при разработке, внедрении и сопровождении информационных технологий и систем на всех этапах жизненного цикла.

ПК-4. Способен проводить анализ требований к программному обеспечению,

выполнять работы по проектированию программного обеспечения

Таблица 1 – Декомпозиция результатов обучения

Код и наименование компетенции	Планируемые результаты обучения по дисциплине (модулю)		
	Знать (1)	Уметь (2)	Владеть (3)
ПК-1. Способен проводить научные исследования при разработке, внедрении и сопровождении информационных технологий и систем на всех этапах жизненного цикла	ИПК-1.1.1 методы проведения научных исследований на всех этапах жизненного цикла программных средств	ИПК-1.2.1 рационально планировать и выполнять научные исследования на всех этапах жизненного цикла программных средств	ИПК-1.3.1 навыками планирования и проведения научных исследований на всех этапах жизненного цикла программных средств
ПК-4. Способен проводить анализ требований к программному обеспечению, выполнять работы по проектированию программного обеспечения	ИПК-4.1.1 методы проведения анализа и разработки требований к программному обеспечению.	ИПК-4.2.1 выполнять работы по проектированию программного обеспечения	ИПК-4.3.1 методами проведения анализа требований к программному обеспечению, выполнять работы по проектированию программного обеспечения

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Объём дисциплины (модуля) составляет 5 зачётных(ые) единиц(ы), в том числе 51 часов(а), выделенных на контактную работу обучающихся с преподавателем (из них 17 часов(а) – лекции, 34 часов(а) – лабораторные работы), и 129 часов(а) – на самостоятельную работу обучающихся.

Таблица 2 – Структура и содержание дисциплины (модуля)

Раздел, тема дисциплины (модуля)	Семестр	Контактная работа (в часах)			Самост. работа		Форма текущего контроля успеваемости, форма промежуточной аттестации
		Л	ПЗ	ЛР	КР	СР	
Введение в программную инженерию	6	2	0	6	0	25	Отчет по лабораторной работе, вопросы к экзамену
Модели и профили жизненного цикла программных средств.		2	0	6	0	25	Отчет по лабораторной работе, вопросы к экзамену
Модели и процессы управления проектами программных средств.		4	0	6	0	25	Отчет по лабораторной

Раздел, тема дисциплины (модуля)	Семестр	Контактная работа (в часах)			Самост. работа		Форма текущего контроля успеваемости, форма промежуточной аттестации
		Л	ПЗ	ЛР	КР	СР	
							работе, вопросы к экзамену
Управление требованиями к программному обеспечению в областях техники		4	0	6	0	25	Отчет по лабораторной работе, вопросы к экзамену
Проектирование и разработка программного обеспечения.		5	0	10	0	29	Отчет по лабораторной работе, вопросы к экзамену
Итого		17	0	34	0	129	Экзамен

Примечание: Л – лекция; ПЗ – практическое занятие, семинар; ЛР – лабораторная работа; КР – курсовая работа; СР – самостоятельная работа.

Таблица 3 – Матрица соотношения разделов, тем учебной дисциплины (модуля) и формируемых компетенций

Раздел, тема дисциплины (модуля)	Кол-во часов	Код компетенции		Общее количество компетенций
		ПК-1	ПК-4	
Введение в программную инженерию	33		+	1
Модели и профили жизненного цикла программных средств.	33		+	1
Модели и процессы управления проектами программных средств.	35	+	+	2
Управление требованиями к программному обеспечению в областях техники	35	+	+	2
Проектирование и разработка программного обеспечения.	44		+	1
Итого	180			2

Краткое содержание каждой темы дисциплины (модуля)

Введение в программную инженерию

Общие сведения о программной инженерии, её история и значение. Понятия программного обеспечения (ПО) и его роль в современном мире. CASE-технологии: определение, преимущества и примеры использования. Методы инженерии ПО, включая традиционные и современные подходы. Характеристики качественного ПО: надежность, удобство использования, производительность и другие. Профессиональные и этические требования к специалистам по ПО, включая стандарты и кодексы поведения.

Модели и профили жизненного цикла программных средств:

Обзор различных моделей жизненного цикла ПО: каскадная, эволюционная, на основе ранее созданных компонентов, пошаговой разработки, спиральная. Преимущества и недостатки каждой модели жизненного цикла. Выбор модели жизненного цикла в зависимости от типа проекта и требований. Профили жизненного цикла: адаптация моделей под конкретные условия и задачи.

Модели и процессы управления проектами программных средств:

Основные процессы управления проектами ПО: планирование, мониторинг, контроль и закрытие. Методологии управления проектами: Agile, Scrum, Kanban и другие. Инструменты и техники управления проектами, включая диаграммы Ганта, PERT и другие. Управление рисками в проектах ПО: идентификация, оценка и минимизация рисков. Управление качеством ПО: стандарты, метрики и методы обеспечения качества.

Управление требованиями к программному обеспечению в областях техники:

Сбор и анализ требований к ПО: методы и инструменты. Документирование требований: спецификации, пользовательские истории и другие формы. Управление изменениями требований: процессы и инструменты. Валидация и верификация требований: методы и техники. Специфика управления требованиями в различных областях техники: промышленность, медицина, телекоммуникации и другие.

Проектирование и разработка программного обеспечения:

Основные этапы проектирования ПО: архитектурное проектирование, детальное проектирование. Языки моделирования ПО: UML и другие. Диаграммы классов, концепции связи и ассоциации, классы ассоциаций, квалификаторы, обобщение и наследование. Основные понятия объектно-ориентированного программирования (ООП): инкапсуляция, наследование, полиморфизм. Паттерны проектирования классов: определение, классификация и примеры использования. Программирование, отладка и тестирование: методы и инструменты. Unit-тестирование: принципы, методы и инструменты.

5. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ПРЕПОДАВАНИЮ И ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)

5.1. Указания для преподавателей по организации и проведению учебных занятий по дисциплине (модулю)

Учебная деятельность студента в процессе изучения строится из контактных форм работы с преподавателем (аудиторные занятия, экзамен) и самостоятельной работы.

Для успешного освоения дисциплины является обязательным посещение всех занятий, выполнение домашнего задания и лабораторно-практических работы, которые назначаются преподавателем.

Методическая поддержка дисциплины обеспечивается использованием дистанционных технологий. Студентам предлагается информационный ресурс, расположенный по адресу: <http://moodle.asu.edu.ru>, на сервере дистанционного обучения АГУ. Доступ студентов к учебным ресурсам осуществляется по учетной записи и паролю после регистрации на курс «Программная инженерия» на период обучения по данной дисциплине. На сервере размещен методический материал по данной дисциплине, в содержание которого входит: теоретический материал и указания по выполнению лабораторно-практических работ; вопросы к экзамену.

Аудиторные занятия проводятся на основе теоретического материала, опубликованного на образовательном портале, это позволяет студентам изучить пропущенный материал или самостоятельно разобраться с темой, не освоенной на занятии. Для исключения отрыва студентов от учебного процесса проводится учет посещаемости аудиторных занятий.

5.2. Указания для обучающихся по освоению дисциплины (модулю)

Для освоения дисциплины «Программная инженерия» студентам рекомендуется организация самостоятельной работы по следующим видам работ:

- самостоятельное изучение теоретического материала дисциплины с использованием Internet-ресурсов, информационных баз, методических разработок, специальной учебной и научной литературы;
- дополнительная подготовка к лабораторно-практическим работам или выполнение части лабораторной работы, которую не успели сделать в аудитории;
- подготовка к экзамену.

Теоретический материал, задания к лабораторно-практическим занятиям размещены на образовательном портале <http://moodle.asu.edu.ru>.

В процессе подготовки к аудиторным занятиям, необходимо обратить особое внимание на самостоятельное изучение рекомендованной литературы. Самостоятельная работа с 6 учебниками, учебными пособиями, научной, справочной литературой, материалами периодических изданий и Интернета является наиболее эффективным методом получения дополнительных знаний, позволяет значительно активизировать процесс овладения информацией, способствует более глубокому усвоению изучаемого материала.

Таблица 4 – Содержание самостоятельной работы обучающихся

Вопросы, выносимые на самостоятельное изучение	Кол-во часов	Форма работы
Введение в программную инженерию	25	отчет о выполнении ЛР, Устный опрос на экзамене
Модели и профили жизненного цикла программных средств.	25	отчет о выполнении ЛР, Устный опрос на экзамене
Модели и процессы управления проектами программных средств.	25	отчет о выполнении ЛР, Устный опрос на экзамене
Управление требованиями к программному обеспечению в областях техники	25	отчет о выполнении ЛР, Устный опрос на экзамене
Проектирование и разработка программного обеспечения.	29	отчет о выполнении ЛР, Устный опрос на экзамене

5.3. Виды и формы письменных работ, предусмотренных при освоении дисциплины (модуля), выполняемые обучающимися самостоятельно

Письменные работы, предусмотренные при освоении дисциплины, выполняемые студентами самостоятельно – это отчет по выполнению лабораторных практических работ. Тематика ЛПР представлена в таблице 4.

Отчеты по проектной работе оформляются в программе WORD согласно требованию ГОСТа 7.32- 2017 «Отчет о научно-исследовательской работе» о подготовке, содержанию и оформлению научно-исследовательских и курсовых работ.

Объем отчета не должен превышать 10 стр. Объем и состав демонстрационных материалов определяется требованиями индивидуального задания.

6. ОБРАЗОВАТЕЛЬНЫЕ И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

В рамках реализации компетентного подхода в соответствии с требованиями ФГОС ВО в учебном процессе предусмотрены активные и интерактивные формы проведения занятий.

Основой для выстраивания аудиторных занятий являются лабораторные работы. Это самостоятельная работа учащегося, выполненная с помощью консультаций преподавателя. Основное отличие такой деятельности — это то, что студент, прежде всего, получают практические навыки в области программирования.

6.1. Образовательные технологии

Для достижения поставленных целей преподавания дисциплины реализуются следующие образовательные технологии: - самостоятельное изучение теоретического материала дисциплины с использованием Internet-ресурсов, информационных баз, методических разработок, специальной учебной и научной литературы; - закрепление теоретического материала при проведении лабораторных работ, выполнения проблемно-ориентированных, поисковых, творческих заданий. Цели дисциплины достигаются путем сочетания комплекса методов обучения: проведения лабораторно-практических занятий на ЭВМ и организации самостоятельной работы студентов. Лабораторные работы ориентированы на формирование деятельностных компетентностей. Они заключаются в выполнении сквозного цикла лабораторных работ. В процессе выполнения лабораторных работ достигаются следующие цели: - изучаются инструментальные средства программных продуктов; - формируются практические навыки обработки информации различного вида и формы при решении конкретных практических задач; - формируется навык выявления ошибочных и нестандартных ситуаций и реагирования на них.

Таблица 5 – Образовательные технологии, используемые при реализации учебных занятий

Раздел, тема дисциплины (модуля)	Форма учебного занятия		
	Лекция	Практическое занятие, семинар	Лабораторная работа
Введение в программную инженерию	Лекция - презентация	Не предусмотрено	выполнение лабораторной работы
Модели и профили жизненного цикла программных средств.	Лекция - презентация	Не предусмотрено	выполнение лабораторной работы
Модели и процессы управления проектами программных средств.	Лекция - презентация	Не предусмотрено	выполнение лабораторной работы
Управление требованиями к программному обеспечению в областях техники	Лекция - презентация	Не предусмотрено	выполнение лабораторной работы
Проектирование и разработка программного обеспечения.	Лекция - презентация	Не предусмотрено	выполнение лабораторной работы

Учебные занятия по дисциплине могут проводиться с применением информационно-телекоммуникационных сетей при опосредованном (на расстоянии) интерактивном взаимодействии обучающихся и преподавателя в режимах on-line в формах: видеолекций, лекций-презентаций, видеоконференции, собеседования в режиме чат, форума, чата, выполнения виртуальных практических и/или лабораторных работ и др. Максимальный объем занятий обучающегося с применением электронных образовательных технологий не должен превышать 25%

6.2. Информационные технологии

- использование возможностей интернета в учебном процессе (использование сайта преподавателя (рассылка заданий, предоставление выполненных работ, ответы на вопросы, ознакомление обучающихся с оценками и т. д.));
- использование электронных учебников и различных сайтов (например, электронных библиотек, журналов и т. д.) как источников информации;

- использование возможностей электронной почты преподавателя;
- использование средств представления учебной информации (электронных учебных пособий и практикумов, применение новых технологий для проведения очных (традиционных) лекций и семинаров с использованием презентаций и т. д.);
- использование интегрированных образовательных сред, где главной составляющей являются не только применяемые технологии, но и содержательная часть, т. е. информационные ресурсы (доступ к мировым информационным ресурсам, на базе которых строится учебный процесс);
- использование виртуальной обучающей среды (LMS Moodle «Электронное образование»)

6.3. Программное обеспечение, современные профессиональные базы данных и информационные справочные системы

6.3.1. Программное обеспечение

– Лицензионное программное обеспечение

Adobe Reader – Программа для просмотра электронных документов

Платформа дистанционного обучения LMS Moodle – Виртуальная обучающая среда

Mozilla FireFox – Браузер

Microsoft Office 2013, Microsoft Office Project 2013, Microsoft Office Visio 2013 – Офисная программа

7-zip – Архиватор

Microsoft Windows 10 Professional – Операционная система

Kaspersky Endpoint Security – Средство антивирусной защиты

Eclipse – Среда разработки

6.3.2. Современные профессиональные базы данных и информационные справочные системы

1. Электронный каталог Научной библиотеки АГУ на базе MARKSQL НПО «Информ-систем». <https://library.asu.edu.ru>.
2. Электронный каталог «Научные журналы АГУ». <http://journal.asu.edu.ru>.
3. Универсальная справочно-информационная база данных периодических изданий ООО "ИВИС". <http://dlib.eastview.com>.
4. Корпоративный проект Ассоциации региональных библиотечных консорциумов (АРБИКОН) «Межрегиональная аналитическая роспись статей» (МАРС) - сводная база данных, содержащая полную аналитическую роспись 1800 названий журналов по разным отраслям знаний. Участники проекта предоставляют друг другу электронные копии отсканированных статей из книг, сборников, журналов, содержащихся в фондах их библиотек. <http://mars.arbicon.ru>.

7. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО КОНТРОЛЯ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

7.1. Паспорт фонда оценочных средств

При проведении текущего контроля и промежуточной аттестации по дисциплине (модулю) «Программная инженерия» проверяется сформированность у обучающихся компетенций, указанных в разделе 3 настоящей программы. Этапность формирования данных компетенций в процессе освоения образовательной программы определяется последовательным освоением дисциплин (модулей) и прохождением практик, а в процессе

освоения дисциплины (модуля) – последовательным достижением результатов освоения содержательно связанных между собой разделов, тем.

Таблица 6 – Соответствие разделов, тем дисциплины (модуля), результатов обучения по дисциплине (модулю) и оценочных средств

Контролируемый раздел, тема дисциплины (модуля)	Код контролируемой компетенции	Наименование оценочного средства
Введение в программную инженерию	ПК-4	Отчеты о ЛР, устный опрос на экзамене
Модели и профили жизненного цикла программных средств.	ПК-4	Отчеты о ЛР, устный опрос на экзамене
Модели и процессы управления проектами программных средств.	ПК-1,ПК-4	Отчеты о ЛР, устный опрос на экзамене
Управление требованиями к программному обеспечению в областях техники	ПК-1,ПК-4	Отчеты о ЛР, устный опрос на экзамене
Проектирование и разработка программного обеспечения.	ПК-4	Отчеты о ЛР, устный опрос на экзамене

Для оценивания результатов обучения в виде **знаний** используются следующие типы контроля:

- индивидуальное собеседование (устный опрос по ЛПР).
- письменные работы (отчеты о выполнении ЛПР).

Тестовые задания охватывают содержание всего пройденного материала. Индивидуальное собеседование проводится по разработанным вопросам к зачету/экзамену. Письменная работа (отчет о выполнении ЛПР) проводится по отдельному учебному элементу программы дисциплины.

Для оценивания результатов обучения в виде **умений и владений** используются практические контрольные задания (далее – ПКЗ), включающих одну или несколько задач (вопросов) в виде краткой формулировки действий (комплекса действий), которые следует выполнить, или описание результата, который нужно получить.

По сложности ПКЗ разделяются на простые и комплексные задания.

Простые ПКЗ предполагают решение в одно или два действия. К ним можно отнести: простые ситуационные задачи с коротким ответом или простым действием; несложные задания по выполнению конкретных действий. Простые задания применяются для оценки умений. Комплексные задания требуют многоходовых решений как в типичной, так и в нестандартной ситуациях. Это задания в открытой форме, требующие поэтапного решения и развернутого ответа, в т.ч. задания на индивидуальное или коллективное выполнение проектов, на выполнение практических действий или лабораторных работ. Комплексные практические задания применяются для оценки владений.

Типы практических контрольных заданий:

- задания на установление правильной последовательности, взаимосвязанности действий, выяснения влияния различных факторов на результаты выполнения задания;
- установление последовательности (описать алгоритм выполнения действия),
- нахождение ошибок в последовательности (определить правильный вариант последовательности действий);
- указать возможное влияние факторов на последствия реализации умения и т.д.
- задания на принятие решения в нестандартной ситуации (ситуации выбора, многоальтернативности решений, проблемной ситуации);
- задания на оценку последствий принятых решений;
- задания на оценку эффективности выполнения действия.

7.2. Описание показателей и критериев оценивания компетенций, описание шкал оценивания

Таблица 7 – Показатели оценивания результатов обучения в виде знаний

Шкала оценивания	Критерии оценивания
5 «отлично»	демонстрирует глубокое знание теоретического материала, умение обоснованно излагать свои мысли по обсуждаемым вопросам, способность полно, правильно и аргументированно отвечать на вопросы, приводить примеры
4 «хорошо»	демонстрирует знание теоретического материала, его последовательное изложение, способность приводить примеры, допускает единичные ошибки, исправляемые после замечания преподавателя
3 «удовлетворительно»	демонстрирует неполное, фрагментарное знание теоретического материала, требующее наводящих вопросов преподавателя, допускает существенные ошибки в его изложении, затрудняется в приведении примеров и формулировке выводов
2 «неудовлетворительно»	демонстрирует существенные пробелы в знании теоретического материала, не способен его изложить и ответить на наводящие вопросы преподавателя, не может привести примеры

Таблица 8 – Показатели оценивания результатов обучения в виде умений и владений

Шкала оценивания	Критерии оценивания
5 «отлично»	демонстрирует способность применять знание теоретического материала при выполнении заданий, последовательно и правильно выполняет задания, умеет обоснованно излагать свои мысли и делать необходимые выводы
4 «хорошо»	демонстрирует способность применять знание теоретического материала при выполнении заданий, последовательно и правильно выполняет задания, умеет обоснованно излагать свои мысли и делать необходимые выводы, допускает единичные ошибки, исправляемые после замечания преподавателя
3 «удовлетворительно»	демонстрирует отдельные, несистематизированные навыки, испытывает затруднения и допускает ошибки при выполнении заданий, выполняет задание по подсказке преподавателя, затрудняется в формулировке выводов
2 «неудовлетворительно»	не способен правильно выполнить задания

7.3. Контрольные задания и иные материалы, необходимые для оценки результатов обучения по дисциплине (модулю)

Темы практических и письменных работ

1. Системы управления версиями (VCS) ПО: история, виды VCS, современные VCS: SVN, GIT, Mercurial.
2. Разработка через тестирование (TDD). Разработка на основе поведения (BDD). Системы поддержки TDD и BDD (Cucumber, SpecFlow).
3. Концепция непрерывной интеграции ПО. Системы поддержки непрерывной интеграции. TeamCity, Jenkins (Hudson) - функциональные возможности, настройка и использование.
4. Методы и подходы к командной разработке ПО.

5. Системы и методологии поддержки командной разработки.
6. Модели лицензирования программного обеспечения.
7. Методы и стандарты документирования ПО.
8. Методы и стандарты качества разработки ПО.
9. Методы и стандарты проектирования ПО.
10. Методы и стандарты тестирования ПО.
11. Методы и стандарты сопровождения ПО.
12. Методы и стандарты управления конфигурацией ПО.
13. Методы и стандарты управления проектами по созданию ПО.
14. Методы и стандарты верификации и валидации ПО.
15. Методы и стандарты управления требованиями к ПО.
16. TickIT: сертификация систем качества для программного обеспечения.
17. Сравнительный анализ стандартов ISO/IEC 12207-95 и ISO/IEC 12207-2008.
18. Стандарт ISO/IEC 15504 (SPICE) оценки процессов разработки и поддержки ПО.
19. ISO 20000 – стандарт для управления и обслуживания ИТ-сервисов.
20. Сравнительный анализ моделей жизненного цикла ПО.
21. Системы управления версиями (VCS) ПО: история, виды VCS, современные VCS: SVN, GIT, Mercurial.
22. Разработка через тестирование (TDD). Разработка на основе поведения (BDD). Системы поддержки TDD и BDD (Cucumber, SpecFlow).
23. Концепция непрерывной интеграции ПО. Системы поддержки непрерывной интеграции. TeamCity, Jenkins (Hudson) - функциональные возможности, настройка и использование.
24. Методы и подходы к командной разработке ПО.
25. Системы и методологии поддержки командной разработки.
26. Модели лицензирования программного обеспечения.
27. Методы и стандарты документирования ПО.
28. Методы и стандарты качества разработки ПО.
29. Методы и стандарты проектирования ПО.
30. Методы и стандарты тестирования ПО.
31. Методы и стандарты сопровождения ПО.
32. Методы и стандарты управления конфигурацией ПО.
33. Методы и стандарты управления проектами по созданию ПО.
34. Методы и стандарты верификации и валидации ПО.
35. Методы и стандарты управления требованиями к ПО.
36. TickIT: сертификация систем качества для программного обеспечения.
37. Сравнительный анализ стандартов ISO/IEC 12207-95 и ISO/IEC 12207-2008.
38. Стандарт ISO/IEC 15504 (SPICE) оценки процессов разработки и поддержки ПО.
39. ISO 20000 – стандарт для управления и обслуживания ИТ-сервисов.
40. Сравнительный анализ моделей жизненного цикла ПО.

**Перечень вопросов и заданий,
выносимых на экзамен**

1. Что такое "Программная инженерия" и современные её проблемы?
2. Назовите важные признаки хорошего ПО.
3. Что такое SWEBOOK?
4. Что такое жизненный цикл ПО? Модели жизненного цикла ПО, достоинства и недостатки.
5. Работа с требованиями к ПО.
6. Проектирование и разработка (кодирование) ПО.
7. Тестирование и поддержка ПО.

8. Управление конфигурациями.
9. Процесс разработки ПО.
10. Модели и методы разработки ПО.
11. Качество ПО.
12. Компонентно-ориентированный подход, достоинства и недостатки.
13. Унифицированный процесс разработки программного обеспечения.
14. Аксиомы и итерации унифицированного процесса (UP).
15. Фазы, определение требований и структура UP.
16. Объектно-ориентированный анализ. Анализ прецедента, класс анализа (состав, признаки хорошего класса).
17. Отношения между объектами и между классами.
18. Стандарты жизненного цикла.
19. Структура и основные процессы РМВОК.
20. Структурное проектирование.
21. Объектно-ориентированное проектирование.
22. Метрики оценки качества программного продукта.
23. Модели лицензирования программного обеспечения.
24. Методы и стандарты документирования ПО.
25. Методы и стандарты качества разработки ПО.
26. Методы и стандарты проектирования ПО.
27. Методы и стандарты тестирования ПО.
28. Методы и стандарты сопровождения ПО.
29. Методы и стандарты управления конфигурацией ПО.
30. Методы и стандарты управления проектами по созданию ПО.
31. Методы и стандарты верификации и валидации ПО.
32. Методы и стандарты управления требованиями к ПО.
33. Гибкие методологии разработки ПО. Методология экстремального программирования. Scrum, Kanban.
34. Сравнительный анализ стандартов ISO/IEC 12207-95 и ISO/IEC 12207-2008.
35. Стандарт ISO/IEC 15504 (SPICE) оценки процессов разработки и поддержки ПО.
36. ISO 20000 – стандарт для управления и обслуживания ИТ-сервисов.
37. Методология RUP от IBM Rational Software.
38. Методология MSF от Microsoft.
39. Методология CDM от Oracle.
40. Характеристика стандартов ГОСТ 19, ГОСТ 24, ГОСТ 34.
41. CASE-средства разработки информационных систем.
42. ITIL/ITSM: методология управления и организации ИТ-услуг.
43. Стратегии и методы проектирования.
44. Структурное проектирование.
45. Объектно-ориентированное проектирование.
46. Нотации проектирования.
47. Повторное использование.
48. Интеграция классов, компонентов, подсистем.
49. Конструирование с возможностью тестирования.
50. Основные задачи верификации и валидации программ.
51. Методы тестирования программного обеспечения.
52. Формальные методы проверки правильности программ,
53. «конфигурационная единица».
54. Объекты тестирования и подходы к их тестированию.
55. Классификация ошибок в программном обеспечении
56. Необходимость сопровождения и природа сопровождения.
57. Категории сопровождения.

- 58. Процессы сопровождения.
- 59. Планирование конфигурационного управления.
- 60. Идентификация программных конфигураций.
- 61. Версионные библиотеки.
- 62. Реализация изменений.
- 63. Статусы конфигураций.
- 64. Аудит конфигураций.

Таблица 9 – Примеры оценочных средств с ключами правильных ответов

№ п/п	Тип задания	Формулировка задания	Правильный ответ	Время выполнения (в минутах)
ПК-2 Способен разрабатывать программное обеспечение, включая проектирование, отладку, проверку работоспособности и модификацию ПО.				
1.	Задание закрытого типа	<p>Что такое CASE-технологии?</p> <p>а) Технологии для создания аппаратного обеспечения</p> <p>б) Технологии для автоматизации процесса разработки ПО</p> <p>с) Технологии для тестирования ПО</p> <p>д) Технологии для управления проектами</p>	2	2
2.		<p>Какая модель процесса создания ПО называется каскадной?</p> <p>а) Модель, в которой все этапы выполняются последовательно</p> <p>б) Модель, в которой этапы выполняются параллельно</p> <p>с) Модель, в которой используются готовые компоненты</p> <p>д) Модель, в которой этапы повторяются циклично</p>	1	2
3.		<p>Что такое спецификация в процессе создания ПО?</p> <p>а) Процесс написания кода</p> <p>б) Процесс определения требований к ПО</p> <p>с) Процесс тестирования ПО</p> <p>д) Процесс сборки ПО</p>	2	2
4.		<p>Что такое диаграмма классов в моделировании ПО?</p> <p>а) Схема, показывающая последовательность выполнения программы</p>	2	2

№ п/п	Тип задания	Формулировка задания	Правильный ответ	Время выполнения (в минутах)
		б) Схема, показывающая структуру классов и их взаимосвязи с) Схема, показывающая алгоритмы работы программы д) Схема, показывающая интерфейсы программы		
5.		Что такое unit-тестирование? а) Процесс тестирования всей программы б) Процесс тестирования отдельных модулей программы с) Процесс тестирования интерфейсов программы д) Процесс тестирования алгоритмов программы	2	2
6.	Задание открытого типа	Опишите основные этапы процесса создания программного обеспечения.	Основные этапы процесса создания программного обеспечения включают спецификацию, проектирование, реализацию, сборку, аттестацию и эволюцию программных систем. На этапе спецификации определяются требования к ПО, на этапе проектирования разрабатывается архитектура и структура системы, на этапе реализации пишется код, на этапе сборки компоненты интегрируются в единую систему, на этапе аттестации проводится тестирование и проверка качества, а на этапе эволюции вносятся изменения и улучшения.	5
7.		Что такое CASE-технологии и какие преимущества они предоставляют?	CASE-технологии (Computer-Aided Software Engineering) представляют собой	5

№ п/п	Тип задания	Формулировка задания	Правильный ответ	Время выполнения (в минутах)
			<p>программные средства, автоматизирующие различные этапы разработки программного обеспечения. Они предоставляют преимущества, такие как ускорение процесса разработки, снижение количества ошибок, улучшение документации и повышение качества конечного продукта. CASE-технологии включают инструменты для моделирования, проектирования, кодирования, тестирования и управления проектами.</p>	
8.		<p>Объясните, что такое спиральная модель разработки ПО и в чем её отличие от каскадной модели.</p>	<p>Спиральная модель разработки ПО представляет собой итеративный подход, в котором процесс разработки повторяется циклично, включая этапы планирования, анализа рисков, инженерии и оценки. В отличие от каскадной модели, где этапы выполняются последовательно и однократно, спиральная модель позволяет вносить изменения и улучшения на каждом цикле, что делает её более гибкой и адаптивной к изменениям требований и условий.</p>	5
9.		<p>Какие основные понятия включает в себя объектно-ориентированное программирование (ООП)?</p>	<p>Объектно-ориентированное программирование (ООП) включает в себя</p>	5

№ п/п	Тип задания	Формулировка задания	Правильный ответ	Время выполнения (в минутах)
			<p>основные понятия, такие как инкапсуляция, наследование и полиморфизм. Инкапсуляция позволяет объединить данные и методы в единый объект, скрывая внутреннюю реализацию. Наследование позволяет создавать новые классы на основе существующих, перенимая их свойства и методы. Полиморфизм позволяет объектам разных классов обрабатывать вызовы методов по-разному, что обеспечивает гибкость и расширяемость кода.</p>	
10.		<p>Что такое паттерны проектирования и какую роль они играют в разработке ПО?</p>	<p>Паттерны проектирования представляют собой типовые решения для часто встречающихся задач проектирования программного обеспечения. Они помогают разработчикам использовать проверенные подходы и структуры, что улучшает качество кода, ускоряет процесс разработки и облегчает понимание и поддержку системы. Паттерны проектирования классифицируются на три основные категории: порождающие,</p>	5

№ п/п	Тип задания	Формулировка задания	Правильный ответ	Время выполнения (в минутах)
			структурные и поведенческие, каждая из которых решает определённые задачи в архитектуре программного обеспечения.	

Полный комплект оценочных материалов по дисциплине (модулю) (фонд оценочных средств) хранится в электронном виде на кафедре, утверждающей рабочую программу дисциплины (модуля), и в Центре мониторинга и аудита качества обучения.

7.4. Методические материалы, определяющие процедуры оценивания результатов обучения по дисциплине (модулю)

Дисциплина «Программная инженерия» изучается студентами 3 курса в течение 6 семестра. Форма аттестации по дисциплине в 6 семестре – «экзамен».

Итоговая оценка по промежуточной аттестации выставляется в соответствии с Положением АГУ о балльно-рейтинговой системе (БАРС). Итоговая оценка складывается из баллов, полученных студентами за текущую успеваемость в течении семестра и баллов, полученных студентом на зачетном экзамене. Для получения положительной оценки студенту необходимо набрать минимально 60 баллов.

Экзамен проходит в форме практического задания, примеры составленных заданий (п. 7.3). Один билет включает в себя одно задание. Выбор билета осуществляется в случайном порядке. На подготовку студенту отводится не менее 40 мин. Во время проведения экзамена студенту запрещено пользоваться сотовым телефоном и иными средствами связи, персональным компьютером, сетью Интернет, заготовленными заранее ответами и т.п. Студент, получивший замечание в использовании вышеперечисленного удаляется с экзамена с выставлением 0 баллов. Во время защиты к устному ответу студенты могут делать записи на чистом листе, а затем взять их для ответа. Во время устного ответа чтение текста, написанного при подготовке к устному ответу НЕ ДОПУСТИМО. Такой ответ будет оценен в 0 баллов.

Преподаватель, реализующий дисциплину (модуль), в зависимости от уровня подготовленности обучающихся может использовать иные формы, методы контроля и оценочные средства, исходя из конкретной ситуации.

Таблица 10 – Технологическая карта рейтинговых баллов по дисциплине (модулю)

№ п/п	Контролируемые мероприятия	Количество мероприятий / баллы	Максимальное количество баллов	Срок представления
Основной блок				
1.	<i>Ответ на занятия</i>	5/1	5	В соотв. с табл. 2
2.	<i>Выполнение лабораторной работы</i>	5/7	35	
Всего			40	-
Блок бонусов				
3.	<i>Посещение занятий без пропускам</i>		3	
4.	<i>Своевременное выполнение всех заданий</i>		3	

№ п/п	Контролируемые мероприятия	Количество мероприятий / баллы	Максимальное количество баллов	Срок представления
5.	<i>Активность студента на занятии</i>		4	
Всего			10	-
Дополнительный блок**				
6.	<i>Экзамен</i>		50	
Всего			50	-
ИТОГО			100	-

[Примечание: * – для дисциплины (модуля) с итоговой формой контроля «Зачёт» / «Дифференцированный зачёт», ** – для дисциплины (модуля) с итоговой формой контроля «Экзамен»]

Таблица 11 – Система штрафов (для одного занятия)

Показатель	Балл
<i>Опоздание на занятие</i>	-1
<i>Нарушение учебной дисциплины</i>	-1
<i>Неготовность к занятию</i>	-2
<i>Пропуск занятия без уважительной причины</i>	-2

Таблица 12 – Шкала перевода рейтинговых баллов в итоговую оценку за семестр по дисциплине (модулю)

Сумма баллов	Оценка по 4-балльной шкале	
90–100	5 (отлично)	Зачтено
85–89	4 (хорошо)	
75–84		
70–74		
65–69	3 (удовлетворительно)	Зачтено
60–64		
Ниже 60	2 (неудовлетворительно)	Не зачтено

При реализации дисциплины (модуля) в зависимости от уровня подготовленности обучающихся могут быть использованы иные формы, методы контроля и оценочные средства, исходя из конкретной ситуации.

8. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

8.1. Основная литература

1. Мейер, Б. Объектно-ориентированное программирование и программная инженерия / Б. Мейер. — 3-е изд. — М.: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Эр Медиа, 2019. — 285 с. — ISBN 978-5-4486-0513-0. — Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. — URL: <http://www.iprbookshop.ru/79706.html>.

2. Кознов Д.В. Введение в программную инженерию: учебное пособие/ Кознов Д.В. — Москва, Саратов: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Эр Медиа, 2020. — 305 с. — ISBN 978-5-4497-0311-8. — Текст: электронный // IPR SMART: [сайт]. — URL: <https://www.iprbookshop.ru/89428.html>— Режим доступа: для авторизир. пользователей

8.2. Дополнительная литература

1. Введение в программную инженерию: учебник / В. А. Антипов, А. А. Бубнов, А. Н. Пылькин, В. К. Столчев. — Москва: КУРС: ИНФРА-М, 2019. — 336 с. - ISBN 978-5-906923-22-6. - Текст: электронный. - URL: <https://znanium.com/catalog/product/1035160> – Режим доступа: по подписке.

2. Ехлаков, Ю. П. Введение в программную инженерию: учебное пособие / Ю. П. Ехлаков. — Томск: Томский государственный университет систем управления и радиоэлектроники, Эль Контент, 2011. — 148 с. — ISBN 978-5-4332-0018-0. — Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. — URL: <http://www.iprbookshop.ru/13923.html> — Режим доступа: для авторизир. пользователей

8.3. Интернет-ресурсы, необходимые для освоения дисциплины (модуля)

1. Электронная библиотечная система (ЭБС) ООО «Политехресурс» «Консультант студента»: www.studentlibrary.ru.

9. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Для проведения лекционных занятий необходима мультимедийная аудитория, оснащенная рабочими местами студентов. Для проведения лабораторных занятий необходима аудитория, оснащенная компьютерными рабочими местами студентов и доступом в Интернет.

Рабочая программа дисциплины (модуля) при необходимости может быть адаптирована для обучения (в том числе с применением дистанционных образовательных технологий) лиц с ограниченными возможностями здоровья, инвалидов. Для этого требуется заявление обучающихся, являющихся лицами с ограниченными возможностями здоровья, инвалидами, или их законных представителей и рекомендации психолого-медико-педагогической комиссии. Для инвалидов содержание рабочей программы дисциплины (модуля) может определяться также в соответствии с индивидуальной программой реабилитации инвалида (при наличии).