

Документ подписан простой электронной подписью

Информация о документе:

ФИО: Хоружий Людмила Ивановна

Должность: Директор Института экономики и управления АПК

Дата подписания: 15.07.2023 19:22:44

Уникальный программный ключ:

1e90b132d9b04dc667585166b015dddf2cb1e6a9



**МИНИСТЕРСТВО СЕЛЬСКОГО ХОЗЯЙСТВА РОССИЙСКОЙ ФЕДЕРАЦИИ**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ –**

**МСХА имени К.А. ТИМИРЯЗЕВА»**

**(ФГБОУ ВО РГАУ - МСХА имени К.А. Тимирязева)**

Институт экономики и управления АПК  
Кафедра прикладной информатики



Директор института

Экономики и управления АПК

Л.И. Хоружий

2022 г.

## РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

### Б1.В.05 Программная инженерия

для подготовки бакалавров

ФГОС ВО

Направление: 09.03.02 Информационные системы и технологии

Направленность: Компьютерные науки и интеллектуальный анализ данных  
(Computer Science and Data Mining)

Курс 3

Семестр 4, 5

Форма обучения заочная

Год начала подготовки 2022

Москва, 2022

Разработчики:

Лосев А. Н.,

ст. преподаватель кафедры прикладной информатики

Худякова Е.В.,

д.э.н., профессор, заведующий кафедры прикладной информатики

  
\_\_\_\_\_  
  
\_\_\_\_\_

« 20 » 08 2022 г.

Рецензент: Щедрина Е. В.,

доцент кафедры систем автоматизированного

проектирования и инженерных расчетов, к.п.н.

  
\_\_\_\_\_

« 20 » 08 2022 г.

Программа составлена в соответствии с требованиями ФГОС ВО по направлению подготовки 09.03.02 Информационные системы и технологии, профессиональных стандартов и учебного плана 2022 года начала подготовки.

Программа обсуждена на заседании кафедры прикладной информатики

протокол № 1 от « 20 » 08 2022 г.

Зав. кафедрой прикладной информатики

Худякова Е.В., д.э.н., профессор

  
\_\_\_\_\_

« 20 » 08 2022 г.

**Согласовано:**

Председатель учебно-методической

комиссии института экономики и управления АПК

Корольков А.Ф., к.э.н., доцент

№ 12

  
\_\_\_\_\_

« 20 » 08 2022 г.

Заведующий выпускающей кафедрой

статистики и кибернетики

Уколова А.В., к.э.н., доцент

  
\_\_\_\_\_

« 20 » 08 2022 г.

Заведующий отделом комплектования ЦНБ

  
\_\_\_\_\_

## Содержание

<b><u>АННОТАЦИЯ</u></b> .....	4
<b><u>1. ЦЕЛЬ ОСВОЕНИЯ ДИСЦИПЛИНЫ</u></b> .....	4
<b><u>2. МЕСТО ДИСЦИПЛИНЫ В УЧЕБНОМ ПРОЦЕССЕ</u></b> .....	4
<b><u>3. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ, СООТНЕСЕННЫХ С ПЛАНИРУЕМЫМИ РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ</u></b> .....	5
<b><u>4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ</u></b> .....	8
4.1 РАСПРЕДЕЛЕНИЕ ТРУДОЁМКОСТИ ДИСЦИПЛИНЫ ПО ВИДАМ РАБОТ ПО СЕМЕСТРАМ .....	8
4.2 СОДЕРЖАНИЕ ДИСЦИПЛИНЫ.....	8
4.3 ЛЕКЦИИ/ ПРАКТИЧЕСКИЕ ЗАНЯТИЯ.....	11
4.4 ПЕРЕЧЕНЬ ВОПРОСОВ ДЛЯ САМОСТОЯТЕЛЬНОГО ИЗУЧЕНИЯ ДИСЦИПЛИНЫ .....	13
<b><u>5. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ</u></b> .....	14
<b><u>6. ТЕКУЩИЙ КОНТРОЛЬ УСПЕВАЕМОСТИ И ПРОМЕЖУТОЧНАЯ АТТЕСТАЦИЯ ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ</u></b> .....	14
6.1. ТИПОВЫЕ КОНТРОЛЬНЫЕ ЗАДАНИЯ ИЛИ ИНЫЕ МАТЕРИАЛЫ, НЕОБХОДИМЫЕ ДЛЯ ОЦЕНКИ ЗНАНИЙ, УМЕНИЙ И НАВЫКОВ И (ИЛИ) ОПЫТА ДЕЯТЕЛЬНОСТИ .....	14
6.2. ОПИСАНИЕ ПОКАЗАТЕЛЕЙ И КРИТЕРИЕВ КОНТРОЛЯ УСПЕВАЕМОСТИ, ОПИСАНИЕ ШКАЛ ОЦЕНИВАНИЯ .....	16
<b><u>7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ</u></b> .....	17
7.1 ОСНОВНАЯ ЛИТЕРАТУРА .....	17
7.2 ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА.....	17
7.3 МЕТОДИЧЕСКИЕ УКАЗАНИЯ, РЕКОМЕНДАЦИИ И ДРУГИЕ МАТЕРИАЛЫ К ЗАНЯТИЯМ.....	18
<b><u>8. ПЕРЕЧЕНЬ РЕСУРСОВ ИНФОРМАЦИОННО-ТЕЛЕКОММУНИКАЦИОННОЙ СЕТИ «ИНТЕРНЕТ», НЕОБХОДИМЫХ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ</u></b> .....	18
<b><u>9. ПЕРЕЧЕНЬ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И ИНФОРМАЦИОННЫХ СПРАВОЧНЫХ СИСТЕМ (ПРИ НЕОБХОДИМОСТИ)</u></b> .....	18
<b><u>10. ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ (МОДУЛЮ)</u></b> .....	19
<b><u>11. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ СТУДЕНТАМ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ</u></b> .....	20
<b><u>12. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПРЕПОДАВАТЕЛЯМ ПО ОРГАНИЗАЦИИ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ</u></b> .....	21

## Аннотация

**рабочей программы учебной дисциплины Б1.В.05 «Программная инженерия» для подготовки бакалавров по направлению 09.03.02 "Информационные системы и технологии"**

**Направленности: Компьютерные науки и интеллектуальный анализ данных (Computer Science and Data Mining);**

**Цель освоения дисциплины:** Целью освоения дисциплины «Программная инженерия» является формирование у студентов теоретических и практических навыков по изучению и использованию современных технологий разработки программного обеспечения в соответствии с международными стандартами обучения программной инженерии. Основными задачами изучения дисциплины являются формирование у студентов умений и навыков по проблемам оценки требований, проектирования, разработки, качества, повышения надежности и документирования программного обеспечения, а также по вопросам управления коллективной разработкой программного обеспечения, с помощью таких цифровых технологий и инструментов, как Visual Studio, UML.

**Место дисциплины в учебном плане:** Дисциплина включена в часть, формируемую участниками образовательных отношений учебного плана по направлению подготовки 09.03.02 "Информационные системы и технологии".

**Требования к результатам освоения дисциплины:** в результате освоения дисциплины формируются следующие компетенции (индикаторы): ПКос-2(ПКос-2.1; ПКос-2.2; ПКос-2.3); ПКос-5(ПКос-5.1; ПКос-5.2; ПКос-5.3).

**Краткое содержание дисциплины:** Программное обеспечение (программный продукт). Software Engineering Body of Knowledge (Свод знаний по программной инженерии). Жизненный цикл ПО. Unified Process(UP), определение требований. Унифицированный процесс разработки программного обеспечения (Unified Software Development Process, USDP). История UP/ RUP. Аксиомы, итерации, фазы, структура UP. Поиск и выявление требований. Документация требований. UML – диаграммы. Анализ и проектирование ПО. Объектно-ориентированном проектировании посредством UML. Объектно-ориентированный анализ. Реализация прецедентов. Применение OPT и FLT. Применение LOOP. Модульность ПО. Cohesion (Внутренняя связанность). Coupling (Внешняя связанность). Кодирование. Метрики разработки ПО. Тестирование программного обеспечения. JUnit — библиотека для тестирования программного обеспечения на языке Java. Зрелость процесса разработки. Стандарт ISO 9001. Capability Maturity Model Integration (CMMI).

**Общая трудоемкость дисциплины составляет:** 3 зачетные единицы (108 часов, в том числе 2 часа практической подготовки)

**Промежуточный контроль:** экзамен.

### 1. Цель освоения дисциплины

Целью освоения дисциплины «Программная инженерия» является формирование у студентов теоретических и практических навыков по изучению и

использованию современных технологий разработки программного обеспечения в соответствии с международными стандартами обучения программной инженерии. Основными задачами изучения дисциплины являются формирование у студентов умений и навыков по проблемам оценки требований, проектирования, разработки, качества, повышения надежности и документирования программного обеспечения, а также по вопросам управления коллективной разработкой программного обеспечения. Software Engineering Body of Knowledge. Unified Process. UML. Стандарт ISO 9001. Capability Maturity Model Integration (CMMI).

## **2. Место дисциплины в учебном процессе**

Дисциплина «Программная инженерия» включена в часть. Формируемую участниками образовательных отношений учебно-образовательного плана. Дисциплина «Программная инженерия» реализуется в соответствии с требованиями ФГОС ВО, ОПОП ВО, профессиональных стандартов и Учебного плана по направлению 09.03.02 «Информационные системы и технологии». Предшествующими курсами, на которых базируется дисциплина, являются: «Системное программирование», «Интернет-программирование», «Высокоуровневое программирование», «Операционные системы», «Интернет-программирование». Последующие дисциплины: «Методы искусственного интеллекта», «Администрирование информационных систем», «ERP-системы в управлении бизнесом», «Методы и средства проектирования информационных систем и технологий».

Дисциплина «Программная инженерия» может быть использована при написании выпускной квалификационной работы.

Рабочая программа дисциплины «Программная инженерия» для инвалидов и лиц ограниченными возможностями здоровья разрабатывается индивидуально с учетом особенностей психологического развития, индивидуальных возможностей и состояния здоровья таких обучающихся.

## **3. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы**

Изучение данной учебной дисциплины направлено на формирование у обучающихся компетенций, представленных в таблице 1.

Таблица 1

## Требования к результатам освоения учебной дисциплины

№ п/п	Код компетенции	Содержание компетенции (или её части)	Индикаторы компетенций	В результате изучения учебной дисциплины обучающиеся должны:		
				знать	уметь	владеть
1.	ПКос-2	Способность выполнять интеграцию программных модулей и компонент	ПКос-2.1. Знать: модели процесса и принципы разработки информационных систем, основные подходы к интегрированию программных модулей в информационные системы	основные этапы, методологию, технологию и средства технического проектирования информационных систем; модели, методы, стандарты и инструменты интеграции при построении и сопровождении корпоративных информационных систем, в том числе с применением современных цифровых инструментов (Visual Studio, UML)		
2.			ПКос-2.2. Уметь: интегрировать модули в информационную систему; отлаживать программные модули		фиксировать ход технического проектирования, проводить сборку информационной системы из готовых компонентов, адаптировать приложения к изменяющимся условиям функционирования, в том числе с применением современных цифровых инструментов (Visual Studio, UML)	
3.			ПКос-2.3. Иметь навыки: интеграции модулей в информационную систему; отладки программных модулей			навыками выполнения технического проектирования с учетом принятого в отрасли инструментария, технологиями построения и сопровождения инфокоммуникационных систем и сетей
4.	ПКос-5	Способность создания технической документации на продукцию в сфере информационных технологий, управления технической информацией	ПКос-5.1. Знать: основные стандарты оформления технической документации в сфере информационных технологий; подходы и средства со-	основные методы и законы моделирования информационных процессов и систем, принципы моделирования, классификацию способов представления моделей систем, досто-		

№ п/п	Код компетенции	Содержание компетенции (или её части)	Индикаторы компетенций	В результате изучения учебной дисциплины обучающиеся должны:		
				знать	уметь	владеть
			ставления технической документации для сопровождения объектов автоматизации	инства и недостатки различных способов представления моделей систем, алгоритмы фиксации и обработки результатов моделирования систем, способы планирования машинных экспериментов с моделями, в том числе с применением современных цифровых инструментов (Visual Studio, UML)		
5.			ПКос-5.2. Уметь: применять стандарты оформления технической документации в сфере информационных технологий		использовать технологии моделирования, представлять модель в математическом и алгоритмическом виде, оценивать качество модели, показывать теоретические основания модели, проводить статистическое моделирование систем, моделировать процессы, протекающие в информационных системах и сетях, в том числе с применением современных цифровых инструментов (Visual Studio, UML)	
6.			ПКос-5.3. Иметь навыки: составления (разработки) технической документации в сфере информационных технологий			навыками построения имитационных моделей информационных процессов, методологией получения концептуальных моделей систем, навыками построения моделирующих алгоритмов, навыками обработки и интерпретации информации с помощью программных продуктов Visual Studio, Star UML и др.

## 4. Структура и содержание дисциплины

### 4.1 Распределение трудоёмкости дисциплины по видам работ по семестрам

Общая трудоёмкость дисциплины составляет 3 зач. ед. (108 часов), их распределение по видам работ семестрам представлено в таблице 2.

Таблица 2

#### Распределение трудоёмкости дисциплины по видам работ по семестрам

Вид учебной работы	Трудоёмкость		
	час.	в т.ч. по семестрам	
		№ 5	№ 6
<b>Общая трудоёмкость</b> дисциплины по учебному плану	<b>108</b>	<b>36</b>	<b>72</b>
<b>1. Контактная работа:</b>	<b>14,4</b>	<b>2</b>	<b>12,4</b>
<b>Аудиторная работа</b>	<b>14,4</b>	<b>2</b>	<b>12,4</b>
<i>в том числе:</i>			
<i>лекции (Л)</i>	6	2	4
<i>практические занятия (ПЗ)</i>	8/2		8/2
<i>контактная работа на промежуточном контроле (КРА)</i>	0,4		0,4
<b>2. Самостоятельная работа (СРС)</b>	<b>93,6</b>	<b>34</b>	<b>59,6</b>
<i>самостоятельное изучение разделов, самоподготовка (про- работка и повторение лекционного материала и материала учебников и учебных пособий, подготовка к практическим занятиям)</i>	85	34	51
<i>Подготовка к экзамену (контроль)</i>	8,6		8,6
Вид промежуточного контроля:	Экзамен		

\* в том числе практическая подготовка

### 4.2 Содержание дисциплины

Таблица 3

#### Тематический план учебной дисциплины

Наименование разделов и тем дисциплин	Всего	Аудиторная работа			Внеаудиторная работа СР
		Л	ПЗ /*	ПКР	
<b>Раздел 1. Основные принципы и понятия</b>					
Тема №1. Программная инженерия: назначение, основные принципы и понятия	18	1			17
Тема 2. Жизненный цикл ПО	18	1			17
<b>Всего за 5 семестр</b>	<b>36</b>	<b>2</b>			<b>34</b>
<b>Раздел 1. Основные принципы и понятия</b>					
Тема 3. Составление требований к ПО и варианты использования	20,5	2	1,5/ 1		17
<b>Раздел 2. Проектирование и тестирование</b>					
Тема 1. Объектно-ориентированное проектиро-	21	0,5	3,5/		17



Наименование разделов и тем дисциплин	Всего	Аудиторная работа			Внеаудиторная работа СР
		Л	ПЗ /*	ПКР	
вание ПО			1		
Тема 2. Кодирование и тестирование	21,5	1,5	3		17
Контактная работа на промежуточном контроле (КРА)	0,4			0,4	
Экзамен	8,6				8,6
<b>Всего за 6 семестр</b>	<b>72</b>	<b>4</b>	<b>8</b>	<b>0,4</b>	<b>59,6</b>
<b>Итого по дисциплине</b>	<b>108</b>	<b>6</b>	<b>8</b>	<b>0,4</b>	<b>93,6</b>

\* в том числе практическая подготовка

## Раздел 1. Основные принципы и понятия

### Тема №1. Программная инженерия: назначение, основные принципы и понятия

Программное обеспечение (программный продукт). Программа и программный продукт. Программная инженерия и компьютерные науки. Современные проблемы программной инженерии. Software Engineering Body of Knowledge (Свод знаний по программной инженерии). Работа с требованиями к ПО. Проектирование ПО. Разработка(кодирование) ПО. Тестирование ПО. Поддержка ПО. Управление конфигурациями. Процесс разработки ПО. Модели и методы разработки ПО. Качество ПО. Профессиональные практики. Экономика разработки ПО. Основы вычислений. Математические основы. Основы инженерии.

### Тема 2. Жизненный цикл ПО

Жизненный цикл ПО. Процесс разработки ПО. Жизненный цикл ПО. Развитие программного продукта. Модель процесса разработки ПО. Модели разработки ПО. Водопадная модель. Поэтапная(эволюционная) разработка. Компонентно-ориентированная разработка. Фазы водопадной модели. Достоинства и недостатки моделей разработки ПО. Инкрементальная разработка. Спиральная разработка. Этапы жизненного цикла ПО. Достоинства и недостатки компонентно-ориентированный подход. Формализм в процессе разработки ПО. Факторы, влияющие на оптимальный уровень формализма. Этапы разработки ПО. Относительная стоимость исправления ошибки. Постановка задачи. Процесс разработки ПО. Процесс валидации Тестирование. Разработка и поддержка ПО.

### Тема 3. Составление требований к ПО и варианты использования

Unified Process (UP), определение требований. Унифицированный процесс разработки программного обеспечения (Unified Software Development Process, USDP). История UP/ RUP. Аксиомы, итерации, фазы, структура UP. Основные рабочие потоки итерации. Определение требований. Требования и диаграмма прецедентов. Традиционное описание требований. Пример функциональных требований. Пример нефункциональных требований. Атрибуты требований.

Поиск и выявление требований. Документация требований. UML – диаграмма вариантов использования. Моделирование прецедентов. Идентификация актеров. Детализация прецедентов. Основной поток. Альтернативные потоки. Отображение требований. Применение модели прецедентов.

Анализ и проектирование ПО. Проектирование и архитектура. Влияние нефункциональных требований на архитектуру ПО. Паттерны архитектуры ПО. Каналы и фильтры. Многоуровневая (слоистая) архитектура. Архитектура, управляемая событиями. Микроядерная, микросервисная архитектура. Модульная архитектура. Информационная закрытость. Достоинства хорошей архитектуры. Подходы к модульной декомпозиции. Принципы объектно-ориентированного проектирования. SOLID - аббревиатура пяти основных принципов дизайна классов в объектно-ориентированном проектировании. Разделение интерфейса. Объектно-ориентированном проектировании посредством UML.

Объектно-ориентированный анализ. Анализ прецедента. Классы анализа. CRC – анализ. Стереотипы RUP. Отношение между объектами и классами. Диаграмма объектов. Диаграмма классов: ассоциация. Ассоциация и атрибуты. Ассоциация многие-ко-многим. Зависимости. Отношение: наследование. Обобщение

## **Раздел 2. Проектирование и тестирование**

### **Тема 1. Объектно-ориентированное проектирование ПО**

Реализация прецедентов. Анализ прецедента. Диаграммы последовательностей. Типы сообщений. Комбинированные фрагменты. Основные операторы. Применение OPT и FLT. Применение LOOP. Диаграммы деятельности. Разделы деятельности. Узел действия. Узлы управления. Узлы решения и слияния. Объектные узлы. Контакты. Узлы расширения. Отправка сигналов и прием событий. Обработка времени. Пример диаграммы деятельности.

Модульность ПО. Информационная закрытость. Cohesion (Внутренняя связность). Некорректные и корректные типы связности. Связность по совпадению. Логическая связность. Временная связанность. Процедурная связность. Коммуникативная связность. Последовательная связность. Функциональная связность. Типы связности. Coupling (Внешняя связанность). Связанность модулей. Типы связанности. Внутренняя связность и внешняя связанность. Принципы проектирования модульной структуры ПО.

### **Тема 2. Кодирование и тестирование**

Кодирование. Метрики разработки ПО. Метрики сложности программного кода. Размерно-ориентированные метрики. Физические и логические строки кода. Недостатки SLOC. Метрики сложности. Объектно-ориентированные метрики. Метрики Холстеда. Оценки на основе Метрики Холстеда. Цикломатическая сложность. Граф потока управления программы. Метрики Чепина. Метрика качества программного кода. Программная поддержка. Visual Studio 2013 Pro – метрики кода.

Тестирование программного обеспечения. История развития тестирования программного обеспечения. Понятие тестирования. Виды тестирования. Про-

цесс тестирования. Покрытие операторов. Покрытие решений. Покрытие путей. Тестирование по стратегии серого ящика. Тестовое покрытие. Покрытие требований. Покрытие кода. Документация тестирования. Автоматизированное тестирование компонентов. JUnit — библиотека для тестирования программного обеспечения на языке Java.

Зрелость процесса разработки. Система менеджмента качества разработки ПО. Качество процессов разработки ПО. Стандарт ISO 9001. Моель зрелости процессов разработки ПО. Capability Maturity Model Integration (CMMI). CMMI и эффективность. 5 уровней зрелости процесса разработки. Ключевые процессы на уровнях CMM.

### 4.3 Лекции/ практические занятия

Таблица 4

#### Содержание лекций/практических занятий и контрольные мероприятия

№ п/п	№ раздела	№ и название лекций/ практических занятий	Формируемые компетенции (индикаторы)	Вид контрольного мероприятия	Кол-во часов
<b>Раздел 1. Основные принципы и понятия</b>					<b>5,5</b>
1	Тема №1. Программная инженерия: назначение, основные принципы и понятия	Лекция 1. Основы программной инженерии. SWEBOOK - свод знаний по программной инженерии.	ПКос-5.1 ПКос-5.2 ПКос-5.3		1
2	Тема 2. Жизненный цикл ПО	Лекция 2. Жизненный цикл разработки ПО: водопадная и итерационная модель.	ПКос-5.1 ПКос-5.2 ПКос-5.3		0,5
3		Лекция 3. Жизненный цикл ПО: компонентно-ориентированная модель, этапы жизненного цикла.	ПКос-2.1 ПКос-2.2 ПКос-2.3		0,5
4	Тема 3. Составление требований к ПО и варианты использования	Лекция 4. UP (унифицированный процесс). Определение требований	ПКос-5.1 ПКос-5.2 ПКос-5.3		0,4
5		Практическая работа №1. Видение проекта в методике RUP	ПКос-2.1 ПКос-2.2 ПКос-2.3	защита практической работы	0,5
6		Практическая работа №2. Создание документа "Глоссарий". UML	ПКос-2.1 ПКос-2.2 ПКос-2.3	защита практической работы, дискуссия	0,5
		Лекция 5. Поиск и выявление требований. Варианты использования. Спецификация требований. UML	ПКос-2.1 ПКос-2.2 ПКос-2.3		0,4
		Лекция 6. Проектирование ПО. UML	ПКос-2.1 ПКос-2.2		0,4

№ п/п	№ раздела	№ и название лекций/ практических занятий	Формируемы е компетенции (индикаторы )	Вид контрольного мероприятия	Кол- во часов
			ПКос-2.3		
7		Практическая работа №3. Организация разработки. Каноническое проектирование. UML	ПКос-2.1 ПКос-2.2 ПКос-2.3	защита практической работы	0,5
8		Лекция 7. Выявление классов анализа. UML	ПКос-5.1 ПКос-5.2 ПКос-5.3		0,4
9		Лекция 8. Диаграммы классов анализа. UML	ПКос-5.1 ПКос-5.2 ПКос-5.3		0,4
10	<b>Раздел 2. Проектирование и тестирование</b>				<b>8,5</b>
11	Тема 1. Объектно-ориентированное проектирование ПО	Лекция 9. Реализация прецедентов. Диаграмма деятельности. Диаграмма последовательностей. UML	ПКос-2.1 ПКос-2.2 ПКос-2.3		0,25
		Практическая работа №4. Диаграмма вариантов использования. UML	ПКос-5.1 ПКос-5.2 ПКос-5.3	защита практической работы	0,5
12		Практическая работа №5. Диаграмма классов. UML	ПКос-2.1 ПКос-2.2 ПКос-2.3	защита практической работы	0,5
13		Практическая работа №6. Диаграммы коопераций и последовательностей. UML	ПКос-5.1 ПКос-5.2 ПКос-5.3	защита практической работы	0,5
14		Практическая работа №7. Диаграмма состояний. UML	ПКос-5.1 ПКос-5.2 ПКос-5.3	защита практической работы	0,5
15		Практическая работа №8. Диаграмма деятельности. UML	ПКос-2.1 ПКос-2.2 ПКос-2.3	защита практической работы	0,5
16		Практическая работа №9. Диаграмма компонентов. UML	ПКос-5.1 ПКос-5.2 ПКос-5.3	защита практической работы	0,5
17		Практическая работа №10. Диаграмма развертывания. UML	ПКос-2.1 ПКос-2.2 ПКос-2.3	защита практической работы	0,5
17		Лекция 10. Модульность. Внутренняя связность (cohesion) и внешняя связанность (coupling).	ПКос-5.1 ПКос-5.2 ПКос-5.3		0,25
19		Тема 2. Кодирование и тестирование	Лекция 11. Метрики ПО. Метрики исходного кода. (Visual Studio)	ПКос-5.1 ПКос-5.2 ПКос-5.3	
20	Лекция 12. Тестирование ПО. (Visual Studio)		ПКос-2.1 ПКос-2.2 ПКос-2.3		0,5

№ п/п	№ раздела	№ и название лекций/ практических занятий	Формируемые компетенции (индикаторы)	Вид контрольного мероприятия	Кол-во часов
21		Практическая работа №11. Организация ввода-вывода данных. Форматирование. (Visual Studio)	ПКос-2.1 ПКос-2.2 ПКос-2.3	защита практической работы, круглый стол	1
22		Практическая работа №12. Операции(Visual Studio)	ПКос-5.1 ПКос-5.2 ПКос-5.3	защита практической работы	0,5
23		Практическая работа №13. Операторы языка С#(Visual Studio)	ПКос-2.1 ПКос-2.2 ПКос-2.3	защита практической работы	1
24		Практическая работа №14. Методы: основные понятия(Visual Studio)	ПКос-2.1 ПКос-2.2 ПКос-2.3	защита практической работы, тестирование	0,5
25		Лекция 13. Зрелость управления процессами разработки ПО. (Visual Studio)	ПКос-2.1 ПКос-2.2 ПКос-2.3		0,5

#### 4.4 Перечень вопросов для самостоятельного изучения дисциплины

Таблица 5

##### Перечень вопросов для самостоятельного изучения дисциплины

№ п/п	№ раздела и темы	Перечень рассматриваемых вопросов для самостоятельного изучения
<b>Раздел 1. Основные принципы и понятия</b>		
1.	Тема №1. Программная инженерия: назначение, основные принципы и понятия	1. Жизненный цикл ПО. ПКос-2.1, ПКос-2.2, ПКос-2.3 2. Содержание основных этапов жизненного цикла ПО. ПКос-5.1, ПКос-5.2, ПКос-5.3 3. Тяжеловесные и легковесные методологии разработки ПО. ПКос-2.1, ПКос-2.2, ПКос-2.3
2.	Тема 2. Жизненный цикл ПО	4. Методы верификации и тестирования программ и систем. ПКос-2.1, ПКос-2.2, ПКос-2.3 5. Стандарты качества в области разработки ПО. ПКос-5.1, ПКос-5.2, ПКос-5.3 6. Основные принципы проектного управления при разработке ПО. ПКос-5.1, ПКос-5.2, ПКос-5.3 7. Стандарт SWEBOOK: области знаний (компетенций) программного инженера. ПКос-5.1, ПКос-5.2, ПКос-5.3 8. Понятие жизненного цикла программного обеспечения. ПКос-2.1, ПКос-2.2, ПКос-2.3
3.	Тема 3. Составление требований к ПО и варианты использования	9. Основные модели жизненного цикла программного обеспечения. ПКос-2.1, ПКос-2.2, ПКос-2.3

№ п/п	№ раздела и темы	Перечень рассматриваемых вопросов для самостоятельного изучения
		<p>10. Основные методологии жизненного цикла программного обеспечения. ПКос-2.1, ПКос-2.2, ПКос-2.3</p> <p>11. Диаграммы потоков данных. ПКос-5.1, ПКос-5.2, ПКос-5.3</p> <p>12. Основные характеристики набора требований к ПО. ПКос-5.1, ПКос-5.2, ПКос-5.3</p>
<b>Раздел 2. Проектирование и тестирование</b>		
4.	Тема 1. Объектно-ориентированное проектирование ПО	<p>13. Продуктивные свойства отдельного требования к ПО. ПКос-2.1, ПКос-2.2, ПКос-2.3</p> <p>14. Диаграмма вариантов использования (use case): основные характеристики. ПКос-5.1, ПКос-5.2, ПКос-5.3</p> <p>15. Понятие и определение архитектуры ПО. ПКос-2.1, ПКос-2.2, ПКос-2.3</p> <p>16. Основные этапы архитектурного проектирования ПО. ПКос-5.1, ПКос-5.2, ПКос-5.3</p>
5.	Тема 2. Кодирование и тестирование	<p>17. Понятие Usability ПО. ПКос-2.1, ПКос-2.2, ПКос-2.3</p> <p>18. Факторы, влияющие на принципы Usability ПО. ПКос-2.1, ПКос-2.2, ПКос-2.3</p> <p>19. Принципы Usability ПО: краткая характеристика. ПКос-2.1, ПКос-2.2, ПКос-2.3</p> <p>20. Описание процесса тестирования ПО. ПК-6</p> <p>21. Определение качества ПО. ПКос-2.1, ПКос-2.2, ПКос-2.3</p> <p>22. Характеристики и атрибуты качественного ПО. ПКос-2.1, ПКос-2.2, ПКос-2.3</p> <p>23. Стандарт ISO9000. ПКос-5.1, ПКос-5.2, ПКос-5.3</p>

## 5. Образовательные технологии

Таблица 6

### Применение активных и интерактивных образовательных технологий

№ п/п	Тема и форма занятия		Наименование используемых активных и интерактивных образовательных технологий
1	Раздел 1. Тема №1. Программная инженерия: назначение, основные принципы и понятия	Л	Неимитационный метод (проблемная лекция)
2	Раздел 1. Тема 3. Составление требований к ПО и варианты использования	ПЗ	Дискуссия
3	Раздел 1. Тема 3. Составление требований к ПО и варианты использования	Л	Неимитационный метод (проблемная лекция)
4	Раздел 2. Тема 2. Кодирование и тестирование	Л	Неимитационный метод (проблемная лекция)

№ п/п	Тема и форма занятия		Наименование используемых активных и интерактивных образовательных технологий
5	Раздел 2. Тема 2. Кодирование и тестирование	ПЗ	Круглый стол

## **6. Текущий контроль успеваемости и промежуточная аттестация по итогам освоения дисциплины**

### **6.1. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений и навыков и (или) опыта деятельности**

#### **1). Пример тестовых заданий**

1. Легкость применения программного обеспечения это:
  - ✓ характеристики ПО, позволяющие минимизировать усилия пользователя по подготовке исходных данных, применению ПО; \*
  - ✓ отношение уровня услуг, предоставляемых ПО пользователю при заданных условиях, к объему используемых ресурсов;
  - ✓ характеристики ПО, позволяющие минимизировать усилия по внесению изменений для устранения в нем ошибок и по его модификации.
2. Мобильность программного обеспечения это:
  - ✓ способность ПО выполнять набор функций, которые удовлетворяют потребности пользователей;
  - ✓ способность ПС безотказно выполнять определенные функции при заданных условиях в течение заданного периода времени;
  - ✓ способность ПО быть перенесенным из одной среды (аппаратного / программного) в другое. \*
3. UML — это:
  - ✓ язык программирования, имеющий синтаксис схож с C ++;
  - ✓ унифицированный язык визуального моделирования, использует нотацию диаграмм; \*
  - ✓ набор стандартов и спецификаций качества программного обеспечения.
4. При конструировании программного обеспечения на этапе разработки или выбора алгоритма решения реализуется следующее:
  - ✓ архитектурное обработки программы;
  - ✓ выбор языка программирования; \*
  - ✓ совершенствование программы.
5. Проектирование ПО в основном рассматривается как
  - ✓ архитектурное проектирование; \*
  - ✓ коммуникационные методы;
  - ✓ детальные методы.

#### **2). Примерный перечень дискуссионных тем для круглого стола (дискуссии)**

6. Перечислите области человеческой деятельности, где используются программные продукты.
7. Опишите структурный подход программирования.
8. Дайте определения программирования.
9. Дайте определение структуры данных.
10. Что такое поток управления и поток данных?
11. Перечислите основные типы операций.
12. Что такое качество программного обеспечения с Вашей точки зрения?
13. Что такое тестирование программных продуктов?

### **3). Примеры практических работ**

1. Видение проекта в методике RUP.
2. Создание документа "Глоссарий".
3. Организация разработки. Каноническое проектирование.
4. Диаграмма вариантов использования.
5. Диаграмма классов.
6. Диаграммы коопераций и последовательностей.
7. Диаграмма состояний.
8. Диаграмма деятельности.
9. Диаграмма компонентов.
10. Диаграмма развертывания.
11. Организация ввода-вывода данных. Форматирование.
12. Операции.
13. Операторы языка C#.
14. Методы: основные понятия

### **4) Вопросы к экзамену**

1. Жизненный цикл ПО. Содержание основных этапов жизненного цикла ПО.
2. Тяжеловесные и легковесные методологии разработки ПО.
3. Основные характеристики методологий разработки ПО.
4. Характеристики эффективных требований к ПО.
5. Основные методы выявления требований к ПО.
6. Виды архитектуры ПО.
7. Основные принципы юзабилити ПО.
8. Методы верификации и тестирования программ и систем.
9. Стандарты качества в области разработки ПО.
10. Основные принципы проектного управления при разработке ПО.
11. Определение программной инженерии.
12. Место программной инженерии среди других дисциплин, связанных с созданием программного обеспечения.
13. Ключевые отличия программной инженерии (software engineering) от информатики (computer science), от других инженерий.
14. Стандарт SWEBOOK: области знаний (компетенций) программного инженера.
15. Понятие жизненного цикла программного обеспечения.



16. Основные модели жизненного цикла программного обеспечения.
17. Основные методологии жизненного цикла программного обеспечения
18. Схема Захмана: основные характеристики
19. Диаграммы потоков данных. Нотации Йордана-Демарко и Гейна-Сарсона.
20. Основные характеристики набора требований к ПО
21. Продуктивные свойства отдельного требования к ПО
22. Ключевые атрибуты требований
23. Диаграмма вариантов использования (use case): основные характеристики
24. Понятие и определение архитектуры ПО
25. Основные этапы архитектурного проектирования ПО
26. Модели архитектуры на этапе структурирования ПО
27. Модели архитектуры на этапе определения принципов управления работой ПО
28. Модели архитектуры на этапе модульной композиции ПО
29. Виды диаграмм UML
30. Понятие Usability ПО
31. Факторы, влияющие на принципы Usability ПО
32. Принципы Usability ПО: краткая характеристика
33. Описание процесса тестирования ПО
34. Классификация видов тестирования ПО
35. Понятие «управления проектом»
36. Основные характеристики «проекта»
37. РМВОК: 9 областей управленческих знаний
38. Ролевая модель организации проектных команд
39. Модели управления командой. Критерии выбора оптимальной модели
40. Определение качества ПО
41. Характеристики и атрибуты качественного ПО
42. Стандарт ISO9000. TQM - фундаментальные требования к организациям-разработчикам ПО
43. Пять уровней зрелости модели СММ. Характеристика уровней.

## 6.2. Описание показателей и критериев контроля успеваемости, описание шкал оценивания

Для оценки знаний, умений, навыков и формирования компетенции по дисциплине применяется **балльно-рейтинговая** система контроля и оценки успеваемости студентов.

В основу балльно-рейтинговой системы (БРС) положены принципы, в соответствии с которыми формирование рейтинга студента осуществляется в ходе текущего, промежуточного контроля и промежуточной аттестации знаний.

Таблица 7

### Система рейтинговой оценки успеваемости

Баллы	Балльная оценка текущей успеваемости			
За круглый стол	2	3	4	5

Баллы	Балльная оценка текущей успеваемости			
За дискуссию	2	3	4	5
За тестирование	2	6	7	8
За практическую работу	0	1	2	3
За экзамен	2	3	4	5
Оценка	Неудовлетворительно. <b>Компетенции</b> , закреплённые за дисциплиной, <b>не сформированы</b>	Удовлетворительно. <b>Компетенции</b> , закреплённые за дисциплиной, <b>сформированы на уровне – достаточный.</b>	Хорошо. <b>Компетенции</b> , закреплённые за дисциплиной, <b>сформированы на уровне – хороший (средний).</b>	Отлично. <b>Компетенции</b> , закреплённые за дисциплиной, <b>сформированы на уровне – высокий.</b>

Таблица 8

### Итоговая сумма баллов

Виды контроля	Количество видов контроля	Максимальное возможное количество баллов за единицу	Количество баллов
Круглый стол	1	5	5
Дискуссия	1	5	5
Тестирование	1	8	8
Защита практической работы	14	3	42
Экзамен	1	40	40
Всего	-	-	100

Таблица 9

### Балльно-рейтинговая система контроля успеваемости

Шкала оценивания	Оценка
85–100	Отлично
70–84	Хорошо
61-69	Удовлетворительно
0-60	Неудовлетворительно

## 7. Учебно-методическое и информационное обеспечение дисциплины

### 7.1 Основная литература

1. Волк, В.К. Практическое введение в программную инженерию: учебное пособие / В.К. Волк. — Санкт-Петербург: Лань, 2019. — 100 с. — ISBN 978-5-

- 8114-3656-9. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/119634> Загл. с экрана.
2. Маран, М.М. Программная инженерия: учебное пособие / М.М. Маран. — Санкт-Петербург: Лань, 2018. — 196 с. — ISBN 978-5-8114-3032-1. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/106733> — Загл. с экрана.
  3. Пантелеев, Е.Р. Методы научных исследований в программной инженерии: учебное пособие / Е.Р. Пантелеев. — Санкт-Петербург: Лань, 2018. — 136 с. — ISBN 978-5-8114-3220-2. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/110936> — Загл. с экрана.
  4. Соловьев, Н.А. Исследование операций в задачах программной инженерии: учебное пособие / Н.А. Соловьев, Е.Н. Чернопрудова, Н.А. Тишина, А.Ф. Валева. — Санкт-Петербург: Лань, 2019. — 164 с. — ISBN 978-5-8114-3770-2. — Текст: электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/121486> - Загл. с экрана.

## 7.2 Дополнительная литература

1. Вейцман, В.М. Проектирование информационных систем: учебное пособие / В.М. Вейцман. — Санкт-Петербург: Лань, 2019. — 316 с. — ISBN 978-5-8114-3713-9. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/122172> — Загл. с экрана.
2. Шеина, Т. Ю. Основы программирования / Т. Ю. Шеина. — Санкт-Петербург : Лань, 2022. — 292 с. — ISBN 978-5-507-44046-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/247379> — Загл. с экрана.
3. Гвоздева, Т.В. Проектирование информационных систем. Стандартизация: учебное пособие / Т.В. Гвоздева, Б.А. Баллод. — Санкт-Петербург: Лань, 2019. — 252 с. — ISBN 978-5-8114-3517-3. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/115515> — Загл. с экрана.
4. Зубкова, Т.М. Технология разработки программного обеспечения: учебное пособие / Т.М. Зубкова. — Санкт-Петербург: Лань, 2019. — 324 с. — ISBN 978-5-8114-3842-6. — Текст: электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/122176> — Загл. с экрана.
5. Остроух, А.В. Проектирование информационных систем: монография / А.В. Остроух, Н.Е. Суркова. — Санкт-Петербург: Лань, 2019. — 164 с. — ISBN 978-5-8114-3404-6. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/118650> — Загл. с экрана.

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

1. <https://gostinform.ru/> Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления. (открытый доступ)
2. <http://sdo.timacad.ru> Система дистанционного обучения РГАУ МСХА им. К.А. Тимирязева (открытый доступ)
3. <https://www.google.com/chrome/> Браузер Google Chrome (открытый доступ)
4. <http://msdn.microsoft.com/> (открытый доступ)
5. <http://www.rsdn.ru/> Российский журнал для программистов (открытый доступ)
6. <https://www.embarcadero.com/products/cbuilder/> C++ Builder — программный продукт, инструмент быстрой разработки приложений (RAD), интегрированная среда программирования (IDE), система, используемая программистами для разработки программного обеспечения на языке программирования C++ (открытый доступ)
7. <https://visualstudio.microsoft.com/> Microsoft Visual Studio 2010 MSDN – (комплекс систем программирования, обеспечивающий разностороннюю работу с базами данных и сетью). (открытый доступ)

## 9. Перечень программного обеспечения и информационных справочных систем

Таблица 10

### Перечень программного обеспечения

№ п/п	Наименование раздела учебной дисциплины	Наименование программы	Тип программы	Автор	Год разработки
1	Раздел №1-3	Microsoft Office 2007	демонстрирующая	Microsoft	2007
2	Раздел №1-3	Windows Server 2003R2	демонстрирующая	Microsoft	2003
3	Раздел №1-3	WinRAR 3.8	демонстрирующая	Евгений Рошал, Александр Рошал	2008
4	Раздел №1-3	Notepad++	демонстрирующая	Notepad++ Contributors	2018

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

Таблица 11

### Сведения об обеспеченности специализированными аудиториями, кабинетами, лабораториями

Наименование специальных помещений и помещений для самостоятельной работы (№ учебного корпуса, № аудитории)	Оснащенность специальных помещений и помещений для самостоятельной работы
1	2

Наименование специальных помещений и помещений для самостоятельной работы (№ учебного корпуса, № аудитории)	Оснащенность специальных помещений и помещений для самостоятельной работы
1	2
1 учебный корпус, 110, 207 аудитория – учебная аудитория для проведения занятий лекционного типа групповых и индивидуальных консультаций	видеопроектор, экран настенный, ноутбук
Аудитории № 207, 214, уч. корпус №1 для проведения практических занятий, выполнения курсовых работ, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации	Экран настенный, видеопроектор, ноутбук, терминалы: ауд.207 – 18, ауд.214 – 16
Центральная научная библиотека имени Н.И. Железнова	Читальные залы библиотеки
Общежитие	Комната для самоподготовки

## 11. Методические рекомендации студентам по освоению дисциплины

В современных условиях творческая одаренность и нестандартная самостоятельная деятельность человека становятся основным ресурсом функционирования и развития общества. Процесс качественного обновления жизни в нашем обществе предполагает формирование устойчивого и долговременного спроса на творческую личность, яркую индивидуальность, на специалиста, свободно и критически мыслящего, самобытного и инициативного. Умение самостоятельно мыслить, свободно принимать решения, нести за них персональную ответственность необходимо молодежи еще и потому, что в современной жизни возросла автономия личности. И все же одним из важных требований социального заказа, предъявляемого выпускнику вуза в современных условиях, является умение самостоятельно пополнять свои знания, ориентироваться в стремительном потоке научной и культурной информации.

Промежуточным контролем по дисциплине является экзамен.

Организация самостоятельной работы обучающихся является одним из важнейших вопросов в условиях реализации компетентностной модели образования. Это связано не только с увеличением доли самостоятельной работы при освоении учебных дисциплин, но, прежде всего, с современным пониманием образования как жизненной стратегии личности. Мотивация к непрерывному образованию, общекультурные и профессиональные компетенции становятся необходимым ресурсом личности для успешного включения в трудовую деятельность и реализации своих жизненных планов. Основная задача высшего образования заключается в формировании творческой личности специалиста, способного к саморазвитию, самообразованию, инновационной деятельности.

Под самостоятельной работой обучающихся сегодня понимается вид учебно-познавательной деятельности по освоению основной образовательной программы высшего профессионального образования, осуществляемой в опре-

деленной системе, при партнерском участии преподавателя в ее планировании и оценке достижения конкретного результата.

Самостоятельная работа проводится с целью:

- систематизации и закрепления полученных теоретических знаний и практических умений обучающихся;
- углубления и расширения теоретических знаний;
- формирования умений использовать нормативную, правовую, справочную документацию и специальную литературу;
- развития познавательных способностей и активности обучающихся: творческой инициативы, самостоятельности, ответственности, организованности;
- формирование самостоятельности мышления, способностей к саморазвитию, совершенствованию и самоорганизации;
- формирования общих и профессиональных компетенций;
- развитию исследовательских умений.

При выполнении заданий, вынесенных на самостоятельное изучение, необходимо наряду с библиотечным фондом пользоваться различными базами знаний, размещенными в Интернет, к которым, в частности, относятся: Научная электронная библиотека, Российская государственная библиотека и многие другие.

В подготовке к занятиям по дисциплине студенты должны активно использовать дополнительную литературу, поскольку именно с ее помощью можно получить наиболее полное и верное представление о происходящих в стране и в мире процессах. Для этих же целей необходимо шире использовать имеющиеся информационные технологии. Изучение литературы очень трудоемкая и ответственная часть подготовки к лабораторному занятию, написанию доклада и т.п. Она, как правило, сопровождается записями в той или иной форме. Конспектом называется краткая схематическая запись основного содержания научной работы. Желательно использование логических схем, делающих наглядным ход мысли конспектируемого автора.

### **Виды и формы отработки пропущенных занятий**

Студент, пропустивший занятия обязан его отработать:

- лекцию отработывают путем устного ответа по пропущенной теме;
- практическое занятие путем выполнения практической работы, которая выполнялась на пропущенном практическом занятии, с разрешения преподавателя студент имеет право отработать пропущенное практическое задание самостоятельно и отчитаться по нему на ближайшем практическом занятии (если это не противоречит его плану) либо во время, назначенное преподавателем для индивидуальных консультаций.

## **12. Методические рекомендации преподавателям по организации обучения по дисциплине**

В процессе обучения по дисциплине «Программная инженерия» используются лекционно-практические занятия, деловые игры, разбор конкретных ситуаций, проводятся дискуссии по актуальным проблемам управления, орга-

низуются работа с методическими и справочными материалами, целесообразно применение современных технических средств обучения и информационных технологий. Освоение учебной дисциплины предполагает осмысление её разделов и тем на практических занятиях, в процессе которых бакалавр должен закрепить и углубить теоретические знания.

Своеобразие современной профессиональной деятельности преподавателя заключается в необходимости ведения, поддержки и сопровождения студентов, что позволит сформировать новое поколение специалистов, обладающих **современными компетенциями**.

Дисциплина «Программная инженерия» имеет прикладной характер, её теоретические положения и практические навыки могут быть использованы в будущей практической деятельности.

Преподавание учебного материала по курсу целесообразно вести исходя из научно-обоснованных рекомендаций, с учетом преобразований, происходящих в экономике страны. Это система гибкого управления, способного своевременно перестраиваться и реагировать на конъюнктуру рынка, условия конкурентной борьбы и социальные факторы развития.

В процессе изучения дисциплины «Программная инженерия» предусмотрены несколько форм контроля: текущий и промежуточный.

Текущий контроль предназначен для определения качества усвоения лекционного материала. В течение учебного семестра рекомендуется назначать контрольные точки для проверки качества усвоения изучаемого материала по определенным темам в форме опроса, тестирования и выполнения заданий практикума по дисциплине.

Рекомендуется определять сроки проведения контрольных мероприятий, максимальная оценка за каждое из них и правила перевода общего количества баллов, полученных при изучении дисциплины, в промежуточный результат (Экзамен).

Выполнение практических заданий является обязательным для всех обучающихся. Бакалавры, не выполнившие в полном объеме работы, предусмотренные учебным планом, не допускаются к сдаче экзамена.

Самостоятельная работа бакалавров по курсу должна обязательно сопровождаться проработкой конспекта, выполнением заданий и упражнений.

### **Программу разработали:**

Лосев А.Н., ст. преподаватель

Худякова Е.В., д.э.н., профессор



## РЕЦЕНЗИЯ

### на рабочую программу дисциплины Б1.В.05 «Программная инженерия»

ОПОП ВО по направлению 09.03.02 «Информационные системы и технологии», направленности: «Компьютерные науки и интеллектуальный анализ данных (Computer Science and Data Mining)» (квалификация выпускника – бакалавр)

Щедриной Еленой Владимировной, доцентом кафедры доцентом кафедры систем автоматизированного проектирования и инженерных расчетов, ФГБОУ ВО «Российский государственный аграрный университет – МСХА имени К.А. Тимирязева», кандидат педагогических наук (далее по тексту рецензент), проведено рецензирование рабочей программы дисциплины «Программная инженерия» ОПОП ВО по направлению 09.03.02 «Информационные системы и технологии», направленности «Компьютерные науки и интеллектуальный анализ данных (Computer Science and Data Mining)» (бакалавриат) разработанной в ФГБОУ ВО «Российский государственный аграрный университет – МСХА имени К.А. Тимирязева», на кафедре прикладной информатики (разработчики – Лосев Алексей Николаевич, старший преподаватель Худякова Елена Викторовна, д.э.н., профессор).

Рассмотрев представленные на рецензирование материалы, рецензент пришел к следующим выводам:

1. Предъявленная рабочая программа дисциплины «Программная инженерия» (далее по тексту Программа) соответствует требованиям ФГОС ВО по направлению 09.03.02 «Информационные системы и технологии». Программа содержит все основные разделы, соответствует требованиям к нормативно-методическим документам.

2. Представленная в Программе актуальность учебной дисциплины в рамках реализации ОПОП ВО не подлежит сомнению – дисциплина относится к части формируемой участниками образовательных отношений учебногo цикла – Б1.В.

3. Представленные в Программе цели дисциплины соответствуют требованиям ФГОС ВО направления 09.03.02 «Информационные системы и технологии».

4. В соответствии с Программой за дисциплиной «Программная инженерия» закреплено 2 компетенции (шесть индикаторов): ПКос-2(ПКос-2.1; ПКос-2.2; ПКос-2.3); ПКос-5(ПКос-5.1; ПКос-5.2; ПКос-5.3). Дисциплина «Программная инженерия» и представленная Программа способна реализовать их в объявленных требованиях.

5. Результаты обучения, представленные в Программе в категориях знать, уметь, владеть соответствуют специфике и содержанию дисциплины и демонстрируют возможность получения заявленных результатов.

6. Общая трудоёмкость дисциплины «Программная инженерия» составляет 3 зачётные единицы (108 часов, в том числе 2 часа практической подготовки).

7. Информация о взаимосвязи изучаемых дисциплин и вопросам исключения дублирования в содержании дисциплин соответствует действительности. Дисциплина «Программная инженерия» взаимосвязана с другими дисциплинами ОПОП ВО и Учебного плана по направлению 09.03.02 «Информационные системы и технологии» и возможность дублирования в содержании отсутствует.

8. Представленная Программа предполагает использование современных образовательных технологий, используемые при реализации различных видов учебной работы. Формы образовательных технологий соответствуют специфике дисциплины.

9. Программа дисциплины «Программная инженерия» предполагает проведение занятий в интерактивной форме.

10. Виды, содержание и трудоёмкость самостоятельной работы студентов, представленные в Программе, соответствуют требованиям к подготовке выпускников, содержащимся во ФГОС ВО направлению 09.03.02 «Информационные системы и технологии».

11. Представленные и описанные в Программе формы текущей оценки знаний (круглый стол как форма обсуждения отдельных вопросов, участие в дискуссиях, участие в тести-



ровании, защита практических работ) соответствуют специфике дисциплины и требованиям к выпускникам.

Форма промежуточного контроля знаний студентов, предусмотренная Программой, осуществляется в форме экзамена, что соответствует статусу дисциплины, как дисциплины части, формируемой участниками образовательных отношений учебного цикла – Б1.В. ФГОС ВО направления подготовки 09.03.02 «Информационные системы и технологии».

12. Формы оценки знаний, представленные в Программе, соответствуют специфике дисциплины и требованиям к выпускникам.

13. Учебно-методическое обеспечение дисциплины представлено: основной литературой – 4 источник (базовый учебник), дополнительной литературой – 5 наименований, периодическими изданиями – 7 источников со ссылкой на электронные ресурсы, Интернет-ресурсы – 3 источника и соответствует требованиям ФГОС ВО направлению 09.03.02 «Информационные системы и технологии».

14. Материально-техническое обеспечение дисциплины соответствует специфике дисциплины «Программная инженерия» и обеспечивает использование современных образовательных, в том числе интерактивных методов обучения.

15. Методические рекомендации студентам и методические рекомендации преподавателям по организации обучения по дисциплине дают представление о специфике обучения по дисциплине «Программная инженерия».

### **ОБЩИЕ ВЫВОДЫ**

На основании проведенного рецензирования можно сделать заключение, что характер, структура и содержание рабочей программы дисциплины «Программная инженерия» ОПОП ВО по направлению 09.03.02 «Информационные системы и технологии», направленности «Компьютерные науки и интеллектуальный анализ данных(Computer Science and Data Mining)» (квалификация выпускника – бакалавр), разработанная Лосевым А.Н., старшим преподавателем и Худяковой Е.В., д.э.н, профессором, заведующим кафедры прикладной информатики, соответствует требованиям ФГОС ВО, современным требованиям экономики, рынка труда и позволит при её реализации успешно обеспечить формирование заявленных компетенций.

Рецензент: Щедрина Е. В., доцент кафедры систем автоматизированного проектирования и инженерных расчетов, ФГБОУ ВО РГАУ – МСХА имени К.А. Тимирязева», кандидат педагогических наук



«29» августа 2022 г.