

Министерство науки и высшего образования Российской Федерации  
**Муромский институт (филиал)**  
федерального государственного бюджетного образовательного учреждения высшего образования  
**«Владимирский государственный университет  
имени Александра Григорьевича и Николая Григорьевича Столетовых»  
(МИ ВлГУ)**

Кафедра *ПИИ*

«УТВЕРЖДАЮ»  
Заместитель директора по УР  
\_\_\_\_\_ Д.Е. Андрианов  
\_\_\_\_\_ 23.05.2023

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**

*Методология программной инженерии*

**Направление подготовки**

*09.04.04 Программная инженерия*

**Профиль подготовки**

*Технологии разработки интеллектуальных систем*

Семестр	Трудоем- кость, час./зач. ед.	Лек- ции, час.	Практи- ческие занятия, час.	Лабора- торные работы, час.	Консуль- тация, час.	Конт- роль, час.	Всего (контак- тная работа), час.	СРС, час.	Форма промежу- точного контроля (экз., зач., зач. с оц.)
<b>1</b>	<b>144 / 4</b>	<b>16</b>	<b>28</b>		<b>3,6</b>	<b>0,35</b>	<b>57,95</b>	<b>69,4</b>	<b>Экз.(35,65)</b>
<b>Итого</b>	<b>144 / 4</b>	<b>16</b>	<b>28</b>		<b>3,6</b>	<b>0,35</b>	<b>57,95</b>	<b>69,4</b>	<b>35,65</b>

Муром, 2023 г.

## 1. Цель освоения дисциплины

Цель дисциплины: Изучение дисциплины обеспечивает прикладные научно-методические основы подготовки студента. Она способствует формированию у обучаемых представления о методах и процессах проектирования и реализации программных продуктов. Целью курса «Методология программной инженерии» является изучение современных инженерных принципов (методов) создания надежного, качественного программного обеспечения, удовлетворяющего предъявляемым к нему требованиям; формирование у студентов понимания необходимости применения данных принципов программной инженерии.

Задача изучения дисциплины состоит в том, чтобы обучающиеся овладели основами теоретических и практических знаний в области программной инженерии.

## 2. Место дисциплины в структуре ОПОП ВО

Дисциплина относится к базовой части учебного плана. Изучение дисциплины основано на умениях и компетенциях, полученных студентом при изучении дисциплин «Объектно-ориентированное программирование», «Организация баз данных», «Тестирование программного обеспечения», «Операционные системы». Дисциплина служит основой для дисциплин: "Параллельное и распределенное программирование", "Адаптивные web-системы", "Защищенные информационные системы".

## 3. Планируемые результаты обучения по дисциплине

Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения ОПОП (компетенциями и индикаторами достижения компетенций)

Формируемые компетенции (код, содержание компетенции)	Планируемые результаты обучения по дисциплине, в соответствии с индикатором достижения компетенции		Наименование оценочного средства
	Индикатор достижения компетенции	Результаты обучения по дисциплине	
УК-2 Способен управлять проектом на всех этапах его жизненного цикла	УК-2.2 Участвует в управлении проектом на всех этапах жизненного цикла	Имеет навыки участия в управлении проектом на всех этапах жизненного цикла (УК-2.2)	Задания на практическую работу, вопросы к устному опросу
УК-3 Способен организовывать и руководить работой команды, вырабатывая командную стратегию для достижения поставленной цели	УК-3.1 Демонстрирует понимание принципов командной работы	Умеет применять принципы командной работы (УК-3.1)	Задания на практическую работу, вопросы к устному опросу
	УК-3.2 Руководит членами команды для достижения поставленной цели	Владеет навыками руководства членами команды для достижения поставленной цели (УК-3.2)	
ОПК-8 Способен осуществлять эффективное управление разработкой программных средств и проектов	ОПК-8.1 Оценивает качество формализации поставленных задач в соответствии с требованиями технического задания	Знает методы оценки качества формализации поставленных задач в соответствии с требованиями технического задания (ОПК-8.1)	Задания на практическую работу, вопросы к устному опросу
	ОПК-8.2 Владеет методами принятия управленческих решений	Владеет методами принятия управленческих решений (ОПК-8.2)	

## 4. Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 4 зачетные единицы, 144 часа.

### 4.1. Форма обучения: очная

Уровень базового образования: высшее.

Срок обучения 2г.

#### 4.1.1. Структура дисциплины

№ п\п	Раздел (тема) дисциплины	Семестр	Контактная работа обучающихся с педагогическим работником							Самостоятельная работа	Форма текущего контроля успеваемости (по неделям семестра), форма промежуточной аттестации(по семестрам)
			Лекции	Практические занятия	Лабораторные работы	Контрольные работы	КП / КР	Консультация	Контроль		
1	Жизненный цикл проекта, этапы разработки программных систем	1	6	12						24	устный опрос, отчет по практической работе
2	Управление проектом, методологии разработки программных систем	1	6	2						16	устный опрос, отчет по практической работе
3	Методы тестирования и верификации программных продуктов	1	4	14						29,4	устный опрос, отчет по практической работе
Всего за семестр		144	16	28				3,6	0,35	69,4	Экз.(26,65)
Итого		144	16	28				3,6	0,35	69,4	26,65

#### 4.1.2. Содержание дисциплины

##### 4.1.2.1. Перечень лекций

##### Семестр 1

*Раздел 1. Жизненный цикл проекта, этапы разработки программных систем*

##### Лекция 1.

Стейкхолдеры. Извлечение требований (2 часа).

##### Лекция 2.

Анализ бизнес-драйверов. Функциональная декомпозиция (2 часа).

##### Лекция 3.

Задачи и особенности объектно-ориентированного проектирования программных средств. Основные понятия и модели объектно-ориентированного проектирования программных средств (2 часа).

*Раздел 2. Управление проектом, методологии разработки программных систем*

##### Лекция 4.

Различные методы организации разработки ПО. RUP, XP, MSF, SCRUM, FDD (2 часа).

##### Лекция 5.

Классификация рисков, определения. Стратегия управления рисками (2 часа).

##### Лекция 6.

Этапы и процедуры при управлении конфигурацией программных средств (2 часа).

### *Раздел 3. Методы тестирования и верификации программных продуктов*

#### **Лекция 7.**

Документирование программного обеспечения. Формирование требований к документации сложных программных средств (2 часа).

#### **Лекция 8.**

Технико-экономическое обоснование проектов программных средств (2 часа).

### **4.1.2.2. Перечень практических занятий**

#### **Семестр 1**

##### *Раздел 1. Жизненный цикл проекта, этапы разработки программных систем*

#### **Практическое занятие 1**

Стейкхолдеры. Формирование функциональных и нефункциональных требований (2 часа).

#### **Практическое занятие 2**

Формирование архитектурной схемы решения (2 часа).

#### **Практическое занятие 3**

Формирование архитектуры подсистем. Схемы взаимодействия подсистем (2 часа).

#### **Практическое занятие 4**

Контейнеризация решения (2 часа).

#### **Практическое занятие 5**

Создание демонстрационного решения. Анализ проблем и формирование планов по разрешению проблем (2 часа).

#### **Практическое занятие 6**

Методы рефакторинга (2 часа).

##### *Раздел 2. Управление проектом, методологии разработки программных систем*

#### **Практическое занятие 7**

Система контроля версий, правила работы команд над проектом (2 часа).

##### *Раздел 3. Методы тестирования и верификации программных продуктов*

#### **Практическое занятие 8**

Формулировка атрибутов качества (2 часа).

#### **Практическое занятие 9**

Управление изменениями (2 часа).

#### **Практическое занятие 10**

Юнит-тесты, статический анализ кода (2 часа).

#### **Практическое занятие 11**

Автоматический контроль кода. Автоматизированное тестирование (2 часа).

#### **Практическое занятие 12**

Реализация мониторинга в приложении (2 часа).

#### **Практическое занятие 13**

Автоматизация развертывания приложения (2 часа).

#### **Практическое занятие 14**

Ретроспектива разработки (2 часа).

### **4.1.2.3. Перечень лабораторных работ**

Не планируется.

### **4.1.2.4. Перечень тем и учебно-методическое обеспечение самостоятельной работы**

Перечень тем, вынесенных на самостоятельное изучение:

1. Структура и культура организаций. Мотивация работников. Управление конфликтами.
2. Основные факторы, определяющие качество сложных программных средств.
3. Разработка технического задания в соответствии с ГОСТ 34.602–89.
4. Системное проектирование программных средств.
5. Структурное проектирование сложных программных средств.

6. Применение модели СОСОМО II для оценки разработки программного обеспечения в Windows проектах.
7. Внешние и внутренние риски, сопровождающие программный проект.
8. Служба технической поддержки и ее задачи.
9. Принципы верификации и тестирования программ.
10. Технологические этапы и стратегии систематического тестирования.
11. Методы описания и систематизации требований.

Для самостоятельной работы используются методические указания по освоению дисциплины и издания из списка приведенной ниже основной и дополнительной литературы.

#### **4.1.2.5. Перечень тем контрольных работ, рефератов, ТР, РГР, РПР**

Не планируется.

#### **4.1.2.6. Примерный перечень тем курсовых работ (проектов)**

Не планируется.

### **5. Образовательные технологии**

В процессе изучения дисциплины применяется контактная технология преподавания (за исключением самостоятельно изучаемых студентами вопросов). При проведении практических работ применяется имитационный или симуляционный подход.

### **6. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины.**

Фонды оценочных материалов (средств) приведены в приложении.

### **7. Учебно-методическое и информационное обеспечение дисциплины.**

#### **7.1. Основная учебно-методическая литература по дисциплине**

1. Доррер, Г. А. Методология программной инженерии: учебное пособие. — Красноярск: Сибирский государственный университет науки и технологий имени академика М.Ф. Решетнева, 2021. — 190 с. — URL: <https://www.iprbookshop.ru/116641>
2. Мейер, Б. Объектно-ориентированное программирование и программная инженерия / Б. Мейер. — 3-е изд. — М.: ИНТУИТ, Ай Пи Эр Медиа, 2019. — 285 с. — URL: <https://www.iprbookshop.ru/79706>
3. Суханов, М. Б. Программная инженерия : учебное пособие. — СПб.: Санкт-Петербургский государственный университет промышленных технологий и дизайна, 2018. — 146 с. — URL: <https://www.iprbookshop.ru/102465>

#### **7.2. Дополнительная учебно-методическая литература по дисциплине**

1. Сухов, В. Д. Экономическое обоснование разработки программного продукта : учебно-методическое пособие для бакалавров / В. Д. Сухов, А. А. Киселев, А. И. Сазонов. — М.: Ай Пи Ар Медиа, 2022. — 108 с. — URL: <https://www.iprbookshop.ru/122612>
2. Сеницын, С. В. Верификация программного обеспечения: учебное пособие / С. В. Сеницын, Н. Ю. Налютин. — 3-е изд. — М.: ИНТУИТ, Ай Пи Ар Медиа, 2020. — 367 с. — URL: <https://www.iprbookshop.ru/97540>
3. Проскуряков, А. В. Качество и тестирование программного обеспечения. Метрология программного обеспечения : учебное пособие / А. В. Проскуряков. — Ростов-на-Дону, Таганрог : Издательство Южного федерального университета, 2022. — 197 с. — URL: <https://www.iprbookshop.ru/125702>
4. Кознов, Д. В. Введение в программную инженерию: учебное пособие. — М., Саратов: ИНТУИТ, Ай Пи Ар Медиа, 2020. — 305 с. — URL: <https://www.iprbookshop.ru/89428>
5. Бабич, А. В. Введение в UML: учебное пособие. — 4-е изд. — М.: ИНТУИТ, Ай Пи Ар Медиа, 2022. — 198 с. — URL: <https://www.iprbookshop.ru/120473>

### **7.3. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем**

В образовательном процессе используются информационные технологии, реализованные на основе информационно-образовательного портала института ([www.mivlgu.ru/iop](http://www.mivlgu.ru/iop)), и инфокоммуникационной сети института:

- предоставление учебно-методических материалов в электронном виде;
- взаимодействие участников образовательного процесса через локальную сеть института и Интернет;
- предоставление сведений о результатах учебной деятельности в электронном личном кабинете обучающегося.

Информационные справочные системы:

Электронная библиотечная система iprBooks.ru - <http://www.iprbooks.ru>

Электронная библиотека «ЭВРИКА» - <https://evrika.mivlgu.ru/>

Электронная библиотека ВлГУ - <https://dspace.www1.vlsu.ru/>

Программное обеспечение:

LibreOffice (Mozilla Public License v2.0)

Microsoft Visual Studio (Программа Microsoft Azure Dev Tools for Teaching (Order Number: IM126433))

Microsoft Visio (Программа Microsoft Azure Dev Tools for Teaching (Order Number: IM126433))

Microsoft Project (Программа Microsoft Azure Dev Tools for Teaching (Order Number: IM126433))

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

Информационно-образовательный портал МИ ВлГУ - <https://www.mivlgu.ru/iop/>

## **8. Материально-техническое обеспечение дисциплины**

Лаборатория системного и прикладного программирования

6 шт. компьютеров Intel Core i5, 3500 MHz/ O3Y 6Gb/ SSD-512Gb/ LG 22'; 6 шт. персональных компьютеров Digitech (комплект2) Intel Core i5 3000 MHz/ DDR-4 12Gb/ SSD-512Gb/ Philips 21eb; проектор NEC V300X 3D; экран проекционный настенный Lumien Master Picture; маршрутизатор Gigabit Switch TEG-S16S; макет системы мобильного мониторинга; лабораторный стенд для изучения микроконтроллера; роботизированная платформа IE-POP-BOT; аппаратно-программный комплекс «Изучение принципов построения и исследования инфокоммуникационных локальных сетей». Маркерная доска. Доступ к сети Интернет.

## **9. Методические указания по освоению дисциплины**

Для успешного освоения теоретического материала обучающийся: знакомится со списком рекомендуемой основной и дополнительной литературы; уточняет у преподавателя, каким дополнительным пособиям следует отдать предпочтение; ведет конспект лекций и прорабатывает лекционный материал, пользуясь как конспектом, так и учебными пособиями.:

На практических занятиях пройденный теоретический материал подкрепляется решением задач по основным темам дисциплины в соответствии с индивидуальными темами студентов. Каждой подгруппе обучающихся преподаватель выдает задачу, связанную с проектированием и реализацией программного продукта. В конце занятия обучающие демонстрируют полученные результаты преподавателю и при необходимости делают работу над ошибками.

Самостоятельная работа оказывает важное влияние на формирование личности будущего учащегося, она планируется обучающимся самостоятельно. Каждый обучающийся

самостоятельно определяет режим своей работы и меру труда, затрачиваемого на овладение учебным содержанием дисциплины. Он выполняет внеаудиторную работу и изучение разделов, выносимых на самостоятельную работу, по личному индивидуальному плану, в зависимости от его подготовки, времени и других условий.

Форма заключительного контроля при промежуточной аттестации – экзамен. Для проведения промежуточной аттестации по дисциплине разработаны фонд оценочных средств и балльно-рейтинговая система оценки учебной деятельности студентов. Оценка по дисциплине выставляется в информационной системе и носит интегрированный характер, учитывающий результаты оценивания участия студентов в аудиторных занятиях, качества и своевременности выполнения заданий в ходе изучения дисциплины и промежуточной аттестации.

Программа составлена в соответствии с требованиями ФГОС ВО по направлению  
*09.04.04 Программная инженерия* и профилю подготовки *Технологии разработки интеллектуальных систем*  
Рабочую программу составил *Холкина Н.Е.*\_\_\_\_\_

Программа рассмотрена и одобрена на заседании кафедры *ПИИ*

протокол № 13 от 05.05.2023 года.

Заведующий кафедрой *ПИИ* \_\_\_\_\_ *Жизняков А.Л.*

(Подпись)

Рабочая программа рассмотрена и одобрена на заседании учебно-методической комиссии факультета

протокол № 9 от 19.05.2023 года.

Председатель комиссии ФИТР \_\_\_\_\_ *Рыжкова М.Н.*

(Подпись)

(Ф.И.О.)



**Фонд оценочных материалов (средств) по дисциплине**  
Методология программной инженерии

**1. Оценочные материалы для проведения текущего контроля успеваемости по дисциплине**

Варианты заданий к практическим работам и перечень контрольных вопросов приведены в методических указаниях.

Темы для подготовке к текущему контролю в форме собеседования:

1. Понятие жизненного цикла ПО. Стандарты и проблемы жизненного цикла ПО.
2. Жизненный цикл в соответствии со стандартом ISO 12207. Структура жизненного цикла. Основные процессы.
3. Жизненный цикл в соответствии со стандартом ISO 15504. Классификация процессов. Категория процессов Потребитель-поставщик.
4. Жизненный цикл в соответствии со стандартом ISO 15504. Категории процессов «Инженерные процессы», «Вспомогательные процессы»
5. Жизненный цикл в соответствии со стандартом ISO 15504. Категории процессов «Управленческие процессы», «Организационные процессы».
6. Понятие Модели жизненного цикла программного продукта. Что является схемой модели ЖЦ ПО?
7. Модели жизненного цикла ПО. Принципы. Основные фазы. Преимущества и недостатки. Применимость.
8. Методология разработки ПО MSF. Базовые концепции и принципы. Преимущества и недостатки. Применимость.
9. Методология разработки ПО RUP. Структура RUP. Преимущества и недостатки. Применимость.
10. Методология разработки ПО Extreme Programming. Принципы. Схема. Основные фазы.
11. Методология SCRUM. Роли и их назначение.Arteфакты. Спринты.
12. Понятие надежности ПО.

**Общее распределение баллов текущего контроля по видам учебных работ для студентов**

Рейтинг-контроль 1	Устный опрос (2 вопроса)	До 6 баллов
Рейтинг-контроль 2	Устный опрос (2 вопроса)	До 6 баллов
Рейтинг-контроль 3	Устный опрос (2 вопроса)	До 6 баллов
Посещение занятий студентом	Отметка в журнале посещений	1 балл за каждое занятие
Дополнительные баллы (бонусы)		0
Выполнение семестрового плана самостоятельной работы	Защита практических работ	До 3 баллов за каждую практическую работу

**2. Промежуточная аттестация по дисциплине**

**Перечень вопросов к экзамену / зачету / зачету с оценкой.**

**Перечень практических задач / заданий к экзамену / зачету / зачету с оценкой (при наличии)**

Для проведения экзаменационного тестирования используются задания в тестовой форме. Примеры заданий приведены далее.

## Методические материалы, характеризующие процедуры оценивания

Промежуточная аттестация подводит итоги изучения дисциплины. Вопросы, выносимые на промежуточную аттестацию, доводятся до сведения учащихся за неделю до контрольной недели. Требования и задания соответствуют требуемому уровню усвоения дисциплины и отражают ее основное содержание.

Максимальная сумма баллов, набираемая студентом по дисциплине равна 100.

Оценка в баллах	Оценка по шкале	Обоснование	Уровень сформированности компетенций
Более 80	«Отлично»	Содержание курса освоено полностью, без пробелов, необходимые практические навыки работы с освоенным материалом сформированы, все предусмотренные программой обучения учебные задания выполнены, качество их выполнения оценено числом баллов, близким к максимальному	<b>Высокий уровень</b>
66-80	«Хорошо»	Содержание курса освоено полностью, без пробелов, некоторые практические навыки работы с освоенным материалом сформированы недостаточно, все предусмотренные программой обучения учебные задания выполнены, качество выполнения ни одного из них не оценено минимальным числом баллов, некоторые виды заданий выполнены с ошибками	<b>Продвинутый уровень</b>
50-65	«Удовлетворительно»	Содержание курса освоено частично, но пробелы не носят существенного характера, необходимые практические навыки работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий, возможно, содержат ошибки	<b>Пороговый уровень</b>
Менее 50	«Неудовлетворительно»	Содержание курса не освоено, необходимые практические навыки работы не сформированы, выполненные учебные задания содержат грубые ошибки	<b>Компетенции не сформированы</b>

### 3. Задания в тестовой форме по дисциплине

Примеры заданий:

Методология проектирования, соединяющая в себе процесс объектной декомпозиции и приёмы представления логической и физической, а также статической и динамической моделей проектируемой системы

- Объектно-ориентированное проектирование
- Объектно-ориентированный анализ
- Процедурно-ориентированное программирование
- Алгоритмическое программирование

Какой подход НЕ относится к основным подходам к организации процесса создания и использования программного средства

- Комплексное программирование
- Водопадный подход
- Прототипирование
- Исследовательское программирование

Абстракция, которая позволяет представить фактически неограниченный набор различных вычислений одной программой, которая есть абстракция всех этих наборов, называется абстракция через ... (параметризацию)

Аббревиатура методологии структурного анализа и проектирования, с представлением систем в виде иерархии функций с использованием специального графического языка описания моделей систем, интегрирующая процесс моделирования, управления конфигурацией проекта и руководство проектом: ... (SADT)

Под жизненным циклом программного средства (ПС) понимают

- период разработки и эксплуатации, начиная от момента возникновения замысла ПС и заканчивая прекращением всех видов его использования
- момент возникновения ПС и момент непосредственной его эксплуатации
- использование ПС для решения практических задач на компьютере путем выполнения ее программ
- хранение, внедрение и сопровождение ПС

На диаграмме потоков данных первичные информационные потоки порождаются

- внешними сущностями
- работами
- хранилищами данных
- подсистемами

Аббревиатура языка графического описания для объектного моделирования в области разработки программного обеспечения, для моделирования бизнес-процессов, системного проектирования и отображения организационных структур: ... (UML)

В функциональную спецификацию программной системы входит:

- описания внешней информационной среды, к которой должны применяться программное обеспечение
- описание нежелательных (исключительных) ситуаций, которые могут возникнуть при выполнении программ, и реакций на эти ситуации, которые должны обеспечить соответствующие программы.
- критерии качества программного средства

Упрощенная версия программного средства (ПС) для выявления действительных потребностей пользователей называется ... ПС (ответ в именительном падеже): (прототип)

В технологии проектирования SADT диаграммы DFD определяют для предметной области потоки ... (чего?) (данных)

В результате объектно-ориентированной разработки проекта с использованием UML для моделирования поведения объектов системы при переходе из одного состояния в другое формируются диаграмма ... (чего?): (состояния)

Критерий качества, определяющий способность программного средства безотказно выполнять определенные функции при заданных условиях в течение заданного периода времени с достаточно большой вероятностью, это – ... (ответ в именительном падеже) (надежность)

Полный перечень тестовых заданий с указанием правильных ответов, размещен в банке вопросов на информационно-образовательном портале института по ссылке <https://www.mivlgu.ru/iop/question/bank/managecategories/category.php?courseid=2036&cat=33134%2C60323>

Оценка рассчитывается как процент правильно выполненных тестовых заданий из их общего числа.