

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Волгоградский государственный социально-педагогический университет»
Факультет математики, информатики и физики
Кафедра информатики и методики преподавания информатики



«УТВЕРЖДАЮ»

Проректор по учебной работе

Ю. А. Жадаев

2019 г.

Программная инженерия

Программа учебной дисциплины

Направление 09.03.03 «Прикладная информатика»

Профиль «Прикладная информатика»

очная форма обучения

Волгоград
2019

Обсуждена на заседании кафедры информатики и методики преподавания информатики
«26» 02 2019 г., протокол № 7

Заведующий кафедрой _____ Сергеев А.Н. «26» 02 2019 г.
(подпись) (зав. кафедрой) (дата)

Рассмотрена и одобрена на заседании учёного совета факультета математики, информатики и
физики «02» 04 2019 г., протокол № 7

Председатель учёного совета Сергеев А.Н. «02» 04 2019 г.
(подпись) (дата)

Утверждена на заседании учёного совета ФГБОУ ВО «ВГСПУ»
«31» 05 2019 г., протокол № 10

Отметки о внесении изменений в программу:

Лист изменений № _____
(подпись) (руководитель ОПОП) (дата)

Лист изменений № _____
(подпись) (руководитель ОПОП) (дата)

Лист изменений № _____
(подпись) (руководитель ОПОП) (дата)

Разработчики:

Гермашев Илья Васильевич, доктор технических наук, профессор кафедры информатики и
методики преподавания информатики ФГБОУ ВО «ВГСПУ».

Программа дисциплины «Программная инженерия» соответствует требованиям ФГОС ВО
по направлению подготовки 09.03.03 «Прикладная информатика» (утверждён приказом
Министерства образования и науки Российской Федерации от 19 сентября 2017 г. № 922) и
базовому учебному плану по направлению подготовки 09.03.03 «Прикладная информатика»
(профиль «Прикладная информатика»), утверждённому Учёным советом ФГБОУ ВО
«ВГСПУ» (от 31 мая 2019 г., протокол № 10).

1. Цель освоения дисциплины

Сформировать систему компетенций бакалавра прикладной информатики в области разработки программных компонентов информационных систем, реализации проектных решений с использованием современных технологий программирования для решения задач проектной и производственно-технологической профессиональной деятельности.

2. Место дисциплины в структуре ОПОП

Дисциплина «Программная инженерия» относится к базовой части блока дисциплин.

Для освоения дисциплины «Программная инженерия» обучающиеся используют знания, умения, способы деятельности и установки, сформированные в ходе изучения дисциплин «Алгоритмизация и программирование», «Базы данных», «Вычислительные системы, сети и телекоммуникации», «Информационная безопасность», «Информационные системы и технологии», «Операционные системы», «Проектирование информационных систем».

Освоение данной дисциплины является необходимой основой для последующего изучения дисциплины «Проектный практикум», прохождения практик «Ознакомительная практика», «Технологическая (проектно-технологическая) практика».

3. Планируемые результаты обучения

В результате освоения дисциплины выпускник должен обладать следующими компетенциями:

- способен использовать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности (ОПК-2);
- способен участвовать в разработке стандартов, норм и правил, а также технической документации, связанной с профессиональной деятельностью (ОПК-4);
- способен устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем (ОПК-5);
- способен разрабатывать алгоритмы и программы, пригодные для практического применения (ОПК-7);
- способен принимать участие в управлении проектами создания информационных систем на стадиях жизненного цикла (ОПК-8).

В результате изучения дисциплины обучающийся должен:

знать

- функциональные и технологические стандарты разработки программных комплексов, модели и процессы жизненного цикла программных средств ИС;
- содержание основных этапов процесса разработки и проектирования программных комплексов, принципы организации проектирования программных комплексов, способы выравнивания загрузки ресурсов, средства автоматической генерации тестов;
- экономико-правовые основы разработки программных продуктов, международные и отечественные стандарты;

– классификация принципов и методов обеспечения надежности программных средств, способы использования системы отслеживания задач TFS;

уметь

– формировать архитектуру программных комплексов для информатизации предприятий, разрабатывать программные приложения, отслеживать проекты, формулировать требования к создаваемым программным комплексам;
– формулировать требования к создаваемым программным комплексам, планировать работу на основе методологии Scrum;
– использовать международные и отечественные стандарты, провести ретроспективный анализ выполненного Scrum sprint, выявить потенциальные способы оптимизации, а затем и применить их, используя средства настройки процесса разработки TFS;

владеть

– навыками создания и отслеживания проектов, проведения анализ хода работ в проекте и осуществлять прогнозирование результатов;
– навыками использования функциональных и технологических стандартов разработки программных комплексов;
– навыками разработки модульных тестов средствами Visual Studio Team Developer.

4. Объём дисциплины и виды учебной работы

Вид учебной работы	Всего часов	Семестры
		5
Аудиторные занятия (всего)	58	58
В том числе:		
Лекции (Л)	20	20
Практические занятия (ПЗ)	–	–
Лабораторные работы (ЛР)	38	38
Самостоятельная работа	50	50
Контроль	36	36
Вид промежуточной аттестации		ЭК
Общая трудоемкость	часы	144
	зачётные единицы	4

5. Содержание дисциплины

5.1. Содержание разделов дисциплины

№ п/п	Наименование раздела дисциплины	Содержание раздела дисциплины
1	Введение в программную инженерию	Понятие программной инженерии. Основные определения: информатика, Системотехника, Бизнес-реинжиниринг. Программное обеспечение: определение, свойства. Понятие процесса разработки ПО. Универсальный процесс. Текущий процесс. Конкретный процесс. Стандартный процесс. Совершенствование процесса. Pull/Push стратегии. Классические модели жизненного цикла программных средств: водопадная модель, спиральная модель. Фазы и виды деятельности.
2	Проектирование	Рабочий продукт. Дисциплина обязательств. Проект.

	программных средств	Управление проектами. Типовые приемы конструирования пакетов программ сложной структуры. Методы реализации работ. Выбор и обоснование методов и средств реализации проекта. Детализация проектных решений. Определение назначений. Анализ и оптимизация плана проекта. Процесс управления разработкой программных средств. Прогнозирование и оценка затрат основных ресурсов, необходимых для реализации процессов жизненного цикла программных средств. Создание инфраструктуры. Организация проектирования программного обеспечения; этапы проектирования. Организация коллектива разработчиков. Управление разработкой программных средств на базе стандартов и применения автоматизированных средств и математических методов. Понятие архитектуры ПО. Точка зрения и характеристики точек зрения. Множественность точек зрения при разработке ПО. Виды требований: функциональные требования, нефункциональные требования. Свойства требований: ясность и недвусмысленность, полнота и непротиворечивость, необходимый уровень детализации, прослеживаемость, тестируемость и проверяемость, модифицируемость. Формализация требований. Цикл работы с требованиями.
3	Требования к программным средствам	Понятие конфигурационного управления. Управление версиями. Понятие "ветки" проекта. Управление сборками. Средства версионного контроля. Единицы конфигурационного управления. Понятие baseline. Стандартизация качества. Методы обеспечения качества ПО. Понятие тестирования. Тестирование черного ящика. Тестирование белого ящика. Инструменты тестирования. Критерии тестирования. Виды тестирования. Работа с ошибками. Средства контроля ошибок (bug tracking systems). Диаграммные техники в работе со знаниями. Случаи использования. Работа с требованиями. Случаи использования в управлении разработкой. Итеративный цикл автор/рецензент. Карты памяти. IT решение. Основные принципы MSF. Модель команды: основные принципы, ролевые кластеры. Масштабирование команды MSF. Модель процесса. Управление компромиссами. Понятие CMMI. Уровни зрелости процессов по CMMI. Области усовершенствования. Гибкие методологии разработки. Методология экстремального программирования. Scrum. Общее описание "гибких" методов разработки ПО. Extreme Programming: общее описание, основные принципы организации процесса. Scrum: общее описание, роли, практики. Обследование объекта и определение функциональных характеристик и технико-экономических показателей программного

		изделия. Спецификация программного средства. Анализ осуществимости разработки. Оформление технико-экономического обоснования, технического задания на разработку программного изделия.
4	Качество программных средств, зрелость управления процессами разработки ПО	Технология Microsoft Visual Studio Team System (VSTS). Состав. Архитектура. Клиентская часть . Серверная часть . Элементы работы при планировании. конфигурационное управление. Система контроля версий. Создание сборочного агента. . Capture & Playback при тестировании Web-интерфейсов. VSTS: поддержка различных моделей процесса. Обзор существующих шаблонов. MSF for Agile Software Development. Scrum. Классификация принципов и методов обеспечения надежности программных средств. Модели надежности программных средств. Оценка качественных и количественных характеристик программного обеспечения. Уровни зрелости процесса управления требованиями. Шкала зрелости. Сертификация как средство обеспечения адекватности, надежности и безопасности использования программных средств. Сертификация программного обеспечения. Обобщенные и частные показатели эффективности и качества. Стандарты качества программной продукции.

5.2. Количество часов и виды учебных занятий по разделам дисциплины

№ п/п	Наименование раздела дисциплины	Лекц.	Практ. зан.	Лаб. зан.	СРС	Всего
1	Введение в программную инженерию	2	—	—	12	14
2	Проектирование программных средств	4	—	16	11	31
3	Требования к программным средствам	7	—	8	13	28
4	Качество программных средств, зрелость управления процессами разработки ПО	7	—	14	14	35

6. Перечень основной и дополнительной учебной литературы

6.1. Основная литература

1. Липаев В.В. Программная инженерия сложных заказных программных продуктов [Электронный ресурс]: учебное пособие/ Липаев В.В.— Электрон. текстовые данные.— М.: МАКС Пресс, 2014.— 309 с.— Режим доступа: <http://www.iprbookshop.ru/27297>.— ЭБС «IPRbooks».

2. Кознов Д.В. Введение в программную инженерию [Электронный ресурс]/ Кознов Д.В.— Электрон. текстовые данные.— М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016.— 306 с.— Режим доступа: <http://www.iprbookshop.ru/16697>.— ЭБС «IPRbooks», по паролю.

6.2. Дополнительная литература

1. Кознов Д.В. Основы визуального моделирования [Электронный ресурс]: учебное пособие/ Кознов Д.В.— Электрон. текстовые данные.— М.: БИНОМ. Лаборатория знаний, Интернет-Университет Информационных Технологий (ИНТУИТ), 2008.— 246 с.— Режим доступа: <http://www.iprbookshop.ru/22423>.— ЭБС «IPRbooks».

2. Мейер Б. Объектно-ориентированное программирование и программная инженерия [Электронный ресурс]/ Мейер Б.— Электрон. текстовые данные.— М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016.— 285 с.— Режим доступа: <http://www.iprbookshop.ru/39552>.— ЭБС «IPRbooks».

3. Липаев В.В. Сертификация программных средств [Электронный ресурс]: учебник/ Липаев В.В.— Электрон. текстовые данные.— М.: СИНТЕГ, 2010.— 338 с.— Режим доступа: <http://www.iprbookshop.ru/27299>.— ЭБС «IPRbooks».

4. Липаев В.В. Документирование сложных программных комплексов [Электронный ресурс]: электронное дополнение к учебному пособию «Программная инженерия сложных заказных программных продуктов» (для бакалавров)/ Липаев В.В.— Электрон. текстовые данные.— Саратов: Вузовское образование, 2015.— 115 с.— Режим доступа: <http://www.iprbookshop.ru/27294>.— ЭБС «IPRbooks».

7. Ресурсы Интернета

Перечень ресурсов Интернета, необходимых для освоения дисциплины:

1. Электронная библиотечная система IPRbooks. URL: <http://iprbookshop.ru>.

8. Информационные технологии и программное обеспечение

Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем (при необходимости):

1. Пакет офисных приложений (редактор текстовых документов, презентаций, электронных таблиц).
2. CASE-средства программной инженерии.

9. Материально-техническая база

Для проведения учебных занятий по дисциплине «Программная инженерия» необходимо следующее материально-техническое обеспечение:

1. Компьютерный класс для самостоятельной работы студентов, оборудованный необходимым количеством персональных компьютеров, подключённых к единой локальной сети с возможностью централизованного хранения данных и выхода в Интернет, использования офисных приложений и CASE-средств.
2. Аудитория для проведения учебных занятий, оснащённая аудиторной доской, стационарным или переносным комплексом мультимедийного презентационного оборудования, имеющего доступ к Интернету и локальной сети.

10. Методические указания для обучающихся по освоению дисциплины

Дисциплина «Программная инженерия» относится к базовой части блока дисциплин. Программой дисциплины предусмотрено чтение лекций и проведение лабораторных работ. Промежуточная аттестация проводится в форме экзамена.

Лекционные занятия направлены на формирование глубоких, систематизированных

знаний по разделам дисциплины. В ходе лекций преподаватель раскрывает основные, наиболее сложные понятия дисциплины, а также связанные с ними теоретические и практические проблемы, даёт рекомендации по практическому освоению изучаемого материала. В целях качественного освоения лекционного материала обучающимся рекомендуется составлять конспекты лекций, использовать эти конспекты при подготовке к практическим занятиям, промежуточной и итоговой аттестации.

Лабораторная работа представляет собой особый вид индивидуальных практических занятий обучающихся, в ходе которых используются теоретические знания на практике, применяются специальные технические средства, различные инструменты и оборудование. Такие работы призваны углубить профессиональные знания обучающихся, сформировать умения и навыки практической работы в соответствующей отрасли наук. В процессе лабораторной работы обучающийся изучает практическую реализацию тех или иных процессов, сопоставляет полученные результаты с положениями теории, осуществляет интерпретацию результатов работы, оценивает возможность применения полученных знаний на практике.

При подготовке к лабораторным работам следует внимательно ознакомиться с теоретическим материалом по изучаемым темам. Необходимым условием допуска к лабораторным работам, предполагающим использованием специального оборудования и материалов, является освоение правил безопасного поведения при проведении соответствующих работ. В ходе самой работы необходимо строго придерживаться плана работы, предложенного преподавателем, фиксировать промежуточные результаты работы для отчета по лабораторной работе.

Контроль за качеством обучения и ходом освоения дисциплины осуществляется на основе рейтинговой системы текущего контроля успеваемости и промежуточной аттестации студентов. Рейтинговая система предполагает 100-балльную оценку успеваемости студента по учебной дисциплине в течение семестра, 60 из которых отводится на текущий контроль, а 40 – на промежуточную аттестацию по дисциплине. Критериальная база рейтинговой оценки, типовые контрольные задания, а также методические материалы по их применению описаны в фонде оценочных средств по дисциплине, являющемся приложением к данной программе.

11. Учебно-методическое обеспечение самостоятельной работы

Самостоятельная работа обучающихся является неотъемлемой частью процесса обучения в вузе. Правильная организация самостоятельной работы позволяет обучающимся развивать умения и навыки в усвоении и систематизации приобретаемых знаний, обеспечивает высокий уровень успеваемости в период обучения, способствует формированию навыков совершенствования профессионального мастерства.

Самостоятельная работа обучающихся во внеаудиторное время включает в себя подготовку к аудиторным занятиям, а также изучение отдельных тем, расширяющих и углубляющих представления обучающихся по разделам изучаемой дисциплины. Такая работа может предполагать проработку теоретического материала, работу с научной литературой, выполнение практических заданий, подготовку ко всем видам контрольных испытаний, выполнение творческих работ.

Учебно-методическое обеспечение для самостоятельной работы обучающихся по дисциплине представлено в рабочей программе и включает в себя:

- рекомендуемую основную и дополнительную литературу;
- информационно-справочные и образовательные ресурсы Интернета;
- оценочные средства для проведения текущего контроля и промежуточной аттестации по дисциплине.

Конкретные рекомендации по планированию и проведению самостоятельной работы

по дисциплине «Программная инженерия» представлены в методических указаниях для обучающихся, а также в методических материалах фондов оценочных средств.

12. Фонд оценочных средств

Фонд оценочных средств, включающий перечень компетенций с указанием этапов их формирования, описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания, типовые контрольные задания и методические материалы является приложением к программе учебной дисциплины.