ПРОГРАММНАЯ ИНЖЕНЕРИЯ

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 инсталлировать 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, повести 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. Общая трудоёмкость дисциплины и её распределение

количество зачётных единиц – 4,
общая трудоёмкость дисциплины в часах – 144 ч. (в т. ч. аудиторных часов – 58 ч., СРС – 50 ч.),
распределение по семестрам – 5,
форма и место отчётности – экзамен (5 семестр).

5. Краткое содержание дисциплины

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.

Проектирование программных средств.
Рабочий продукт. Дисциплина обязательств. Проект. 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.

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.

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.

6. Разработчик

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