**Санкт-Петербургский государственный университет**

**Р А Б О Ч А Я П Р О Г Р А М М А**

**УЧЕБНОЙ ДИСЦИПЛИНЫ**

Разработка программного обеспечения

Software Development

**Язык(и) обучения**

русский

Трудоемкость в зачетных единицах: 3

Регистрационный номер рабочей программы: 003617

**Раздел 1. Характеристики учебных занятий**

**1.1. Цели и задачи учебных занятий**

Дисциплина «Разработка программного обеспечения» предназначена для систематизации знаний и практических навыков, а также для поднятия профессионального технического уровня обучающихся бакалавриата «Программная инженерия».

**1.2. Требования подготовленности обучающегося к освоению содержания учебных занятий (пререквизиты)**

Программа курса предназначена для обучающихся второго курса бакалавриата «Программная инженерия». Курс рассчитан на обучающихся, которые освоили программу первого курса направления 09.03.04 «Программная инженерия» или близких направлений.

**1.3. Перечень результатов обучения (learning outcomes)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Наименование категории (группы) компетенций | Код и наименование компетенции | Планируемые результаты обучения, обеспечивающие формирование компетенции | Код индикатора и индикатор достижения универсальной компетенции |
|  | 1 | 2 | 3 | 4 |
| 1 | Общепрофессиональные компетенции | ОПК-1 – способен применять естественнонаучные и общеинженерные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности | Систематизация знания в области современных методологий, методов и средств разработки ПО | ОПК-1.2 Определение качественных характеристик каждого компонента |
| 2 | Общепрофессиональные компетенции | ОПК-2 Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности | Систематизация знания в области современных методологий, методов и средств разработки ПО | ОПК-2.1 Уметь писать программный код с использованием языков программирования, определения и манипулирования данными |
| 3 | Общепрофессиональные компетенции | ОПК-6 – способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов; | Систематизация знания в области современных методологий, методов и средств разработки ПО | ОПК-6.2 Уметь определять перечень возможных шаблонов (стилей) проектирования для каждого слоя или компонента |
| 4 | Общепрофессиональные компетенции | ОПК-7 – способен применять в практической деятельности основные концепции, принципы, теории и факты, связанные с информатикой; | Систематизация знания в области современных методологий, методов и средств разработки ПО | ОПК-7.2 Оценка осуществимости функционирования и сопровождения программного средства |
| 5 | Профессиональные компетенции | ПКП-1 – способен проектировать программные системы; | Систематизация знания об основных видах деятельности при разработке ПО – управление требований, анализ и проектирование, разработка, тестирование, управление проектами, конфигурационное управление и др. | ПКП-1.1 Уметь разрабатывать технические спецификации на программные компоненты и их взаимодействие |
| 6 | Профессиональные компетенции | ПКП-2 – способен использовать основные модели информационных технологий и способы их применения для решения задач в предметных областях | Систематизация знания об основных видах деятельности при разработке ПО – управление требований, анализ и проектирование, разработка, тестирование, управление проектами, конфигурационное управление и др. | ПКП-2.1 Уметь описывать алгоритмы компонентов, включая методы и схемы |
| 7 | Профессиональные компетенции | ПКП-3 – способен разрабатывать моделирующие алгоритмы и реализации их на базе языков и пакетов прикладных программ моделирования | Систематизация знания об основных видах деятельности при разработке ПО – управление требований, анализ и проектирование, разработка, тестирование, управление проектами, конфигурационное управление и др. | ПКП-3.1 Разработка модели бизнес-процессов заказчика |
| 8 | Профессиональные компетенции | ПКП-4 – способен выбирать архитектуру и комплексирование современных компьютеров, систем, комплексов и сетей системного администрирования | Систематизация знания об основных видах деятельности при разработке ПО – управление требований, анализ и проектирование, разработка, тестирование, управление проектами, конфигурационное управление и др. | ПКП-4.1 Оценка и выбор варианта архитектуры программного средства |
| 9 | Профессиональные компетенции | ПКП-5 – способен использовать современные системные программные средства: операционные системы, операционные и сетевые оболочки, сервисные программы; | Систематизация знания об основных видах деятельности при разработке ПО – управление требований, анализ и проектирование, разработка, тестирование, управление проектами, конфигурационное управление и др. | ПКП-5.1 Уметь выбирать технологии и средства разработки программного обеспечения |
| 10 | Профессиональные компетенции | ПКП-6 – способен формировать суждения о проблемах современной информатики, ее категорий и связей с другими научными дисциплинами; | Систематизация знания об основных видах деятельности при разработке ПО – управление требований, анализ и проектирование, разработка, тестирование, управление проектами, конфигурационное управление и др. | ПКП-6.1 Уметь разрабатывать процедуры проверки работоспособности и измерения характеристик программного обеспечения |
| 11 | Профессиональные компетенции | ПКП-7 – способен систематизировать и применять знания о содержании основных этапов и тенденций развития программирования, математического обеспечения и информационных технологий; | Систематизация знания об основных видах деятельности при разработке ПО – управление требований, анализ и проектирование, разработка, тестирование, управление проектами, конфигурационное управление и др. | ПКП-7.2 Оценка и выбор архитектуры развертывания каждого компонента |
| 12 | Универсальные компетенции | УК-1 – способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач; | Знание базовых понятий программной инженерии – процесс, модель процесса, улучшение процесса, рабочий продукт, дисциплина обязательств | УК 1.1. Анализирует задачу, выделяя ее базовые составляющие; |
| 13 | Универсальные компетенции | УК-2 – способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений; | Знание базовых понятий программной инженерии – процесс, модель процесса, улучшение процесса, рабочий продукт, дисциплина обязательств | УК-2.3. Оценивает соответствие способов решения цели проекта; |
| 14 | Универсальные компетенции | УКБ-1 – способен участвовать в разработке и реализации проектов, в т.ч. предпринимательских; | Знание базовых понятий программной инженерии – процесс, модель процесса, улучшение процесса, рабочий продукт, дисциплина обязательств | УКБ-1.6. Представляет результаты проекта; |
| 15 | Универсальные компетенции | УКБ-3 – способен понимать сущность и значение информации в развитии общества, использовать основные методы получения и работы с информацией с учетом современных технологий цифровой экономики и информационной безопасности; | Знание базовых понятий программной инженерии – процесс, модель процесса, улучшение процесса, рабочий продукт, дисциплина обязательств | УКБ-3.1. Находит и использует различные источники информации. |

**1.4. Перечень и объём активных и интерактивных форм учебных занятий**

Практические занятия - 10 ак. часов.

**Раздел 2. Организация, структура и содержание учебных занятий**

**2.1. Организация учебных занятий**

**2.1.1 Основной курс**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Трудоёмкость, объёмы учебной работы и наполняемость групп обучающихся | | | | | | | | | | | | | | | | | | |
| Код модуля в составе дисциплины,  практики и т.п. | Контактная работа обучающихся с преподавателем | | | | | | | | | | | | Самостоятельная работа | | | | Объём активных и интерактивных  форм учебных занятий | Трудоёмкость |
| лекции | семинары | консультации | практические  занятия | лабораторные работы | контрольные работы | коллоквиумы | текущий контроль | промежуточная  аттестация | итоговая аттестация | под руководством преподавателя | в присутствии  преподавателя | сам. раб. с использованием  методических материалов | текущий контроль (сам.раб.) | промежуточная аттестация (сам.раб.) | итоговая аттестация  (сам.раб.) |
| ОСНОВНАЯ ТРАЕКТОРИЯ | | | | | | | | | | | | | | | | | | |
| Форма обучения: очная | | | | | | | | | | | | | | | | | | |
| Семестр 4 | 15 |  |  | 15 |  |  |  |  | 2 |  |  |  | 66 |  | 10 |  | 10 | 3 |
|  | 2-42 |  |  | 2-25 |  |  |  |  | 2-25 |  |  |  | 1-1 |  | 1-1 |  |  |  |
| ИТОГО | 15 |  |  | 15 |  |  |  |  | 2 |  |  |  | 66 |  | 10 |  |  | 3 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Виды, формы и сроки текущего контроля успеваемости и промежуточной аттестации | | | | | | |
| Код модуля в составе дисциплины, практики и т.п. | Формы текущего контроля успеваемости | | Виды промежуточной аттестации | | Виды итоговой аттестации  (только для программ итоговой аттестации и дополнительных образовательных программ) | |
| Формы | Сроки | Виды | Сроки | Виды | Сроки |
| ОСНОВНАЯ ТРАЕКТОРИЯ | | | | | | |
| Форма обучения: очная | | | | | | |
| Семестр 4 |  |  | зачёт, устно, традиционная форма | по графику промежуточной аттестации |  |  |

**2.2. Структура и содержание учебных занятий**

Период обучения (модуль): семестр 4.

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Наименование темы (раздела, части) | Вид учебных занятий | Количество часов |
| 1. | Введение в специальность | лекции | 2 |
| практические занятия | 2 |
| по методическим материалам | 14 |
| 2. | Процесс разработки ПО | лекции | 2 |
| практические занятия | 2 |
| по методическим материалам | 14 |
| 3. | Конфигурационное управление | лекции | 2 |
| практические занятия | 2 |
| по методическим материалам | 10 |
| 4. | Тестирование | лекции | 2 |
| практические занятия | 2 |
| по методическим материалам | 8 |
| 5. | Управление проектами | лекции | 2 |
| практические занятия | 2 |
| по методическим материалам | 6 |
| 6. | Управление требованиями | лекции | 2 |
| практические занятия | 2 |
| по методическим материалам | 6 |
| 7. | Работа с картами памяти (Mindmaps) | лекции | 1 |
| практические занятия | 1 |
| по методическим материалам | 4 |
| 8. | Разработка документации | лекции | 2 |
| практические занятия | 2 |
| по методическим материалам | 4 |
| 9. | Промежуточная аттестация | зачёт | 2 |
| самостоятельная работа | 10 |

1. **Введение в специальность.**
   1. История программной инженерии как науки. Основные достижения. Цикл: передовой опыт/модели, методы, программные средства/распространение и использование в проектах. Мировое software engineering-сообщество.
   2. Виды компаний и проектов по разработке и сопровождению ПО.
   3. Варианты карьеры (технический рост, переход в административную плоскость, переход в смежную предметную область).
   4. Необходимые навыки программиста, в том числе и при карьерном росте.
   5. Пример карьеры: рост менеджера из тестировщика.
   6. Текущие тенденции рынка производства ПО в России и Санкт-Петербурге.
   7. Startups: собственная идея, собственный бизнес и собственная компания. Обзор успешных случаев, рассказ о способах первичной раскрутки.
2. **Процесс разработки ПО.**
   1. Определения процесса разработки ПО. О необходимости разработки и установления процесса до начала проекта. Фазы и виды деятельности процесса. Понятие модели процесса, виды модели процесса: водопадная, спиральная, итеративно-инкрементальная.
   2. Артефакты разработки (код, спецификации модели, тесты, документация, процедуры процесса, средства разработки). Понятие рабочего продукта (workproduct).Подробный разбор группы рабочих продуктов под названием «Диплом». Сравнение процесса с проектными планами, а также другими артефактами разработки.
   3. Роли в проекте. Дисциплина обязательств как основа рабочей этики. Хирургическая бригада Брукса. Модель команды в MSF. Модель команды в Scrum. Команда и другие stakeholders проекта.
   4. Scrum: история основные идеи, структура команды, порядок проведения итерации, отдельные практики (planningpoker и др.), ограничения и предпосылки успешного применения.
   5. CMM: история, 5-ти уровневая структура стандарта, понятие ключевой области процесса, краткий обзор каждого из уровней.
   6. Обзор open-source модели разработки.
3. **Конфигурационное управление.**
   1. Примеры проблем отсутствия порядка в артефактах проекта. Аналогия со складским учетом. Свойства файлов как единиц учета. Определение конфигурационного управления.
   2. Практики версионного контроля исходных кодов проекта. Варианты организации структуры версионного хранилища (различные «ветки»). Процедуры доступа к веткам, перемещение исходников из одной ветки в другую. Введение в svn.
   3. Управление сборками (build management). Сценарии работы сборщика. Регулярная ночная сборка, сборка по запросу, continuous integration. Сборщики и регрессионное тестирование. Примеры простейших сборщиков на основе bat-файлов. Обзор современных средств разработки сборщиков (MSBuild, ant, TFS).
   4. Управление версиями продукта. Особенности управления версиями многоверсионного продукта, поставляемого разным заказчикам. Release management.
4. **Тестирование.**
   1. Определение тестирования. Парадигмы тестирования – белый и черный ящик. Виды тестирования: модульное, системное, интеграционное, стрессовое, нагрузочное, регрессионное. Другие средства обеспечения качества ПО: верификация, улучшение процесса и др.
   2. Варианты организации команды тестировщиков и ее взаимодействия с разработчиками.
   3. Bugtracking системы. Варианты реализации жизненного цикла ошибки.
   4. Модульное тестирование. Дисциплина и правила написания модульных тестов. Обзор продуктов NUnit/JUnit. Самостоятельная разработка пакета модульных тестов.
5. **Управление проектами.**
   1. Понятие проекта, основные особенности: нетипичная задача, необходимость спец усилий по фокусировке, команда. Отличие проекта от функциональной модели организации производства. Небольшие «живые» проекты, большие проекты, в которых участвует много компаний. Stakeholders проекта.
   2. Рамки проекта. Требования и план проекта. Управление деньгами проекта: маленькие/ большие деньги, свои/чужие. Внешнее оправдание (довольный заказчик, менеджмент, удачный бизнес) и внутреннее совершенство (хорошая и стабильная архитектура, налаженный процесс). Треугольник компромиссов MSF. Понятие рисков. Понятие безнадежных проектов по Йордану.
   3. Карьера менеджера softwareпроекта. Основные навыки: умение программировать (в прошлом программист или тестировщик), умение ладить с командой, с начальством и заказчиком, организационные навыки (организация и удерживание разномасштабных начинаний), умение работать с деньгами, чувствование хода проекта
6. **Управление требованиями.**
   1. Трудности управления требованиями к программным продуктам. Понятие требований как иерархической системы дискретных описаний свойств ПО. Функциональные и нефункциональные требования. Трудности выявления нефункциональных требований. Проблемы недовыявленных требований.
   2. Техническое задание проекта: назначение, варианты структуры, примеры. Язык написания технических заданий. Типичные ошибки. Самостоятельная работа по написанию технических заданий.
   3. Приемы интервьюирования пользователей/экспертов предметной области для выявления требований: иметь предварительную подготовку, задавать много вопросов, не давать читать толстых документов, пользоваться диаграммами (в частности, UML-диаграммами). Активность и ответственность разработчика, «беззаботность» источника требований. Примеры UML-диаграмм для выявления требований. Использование диаграмм случаев использования. Самостоятельная работа по выявлению требований для разных вариантов проектов.
7. **Работа с картами памяти (Mindmaps)**
   1. Основные идеи и элементы. История карт памяти. Обзор областей применения. Способы разработки – «от руки», цветными карандашами, в компьютерных средствах. Разработка собственных примеров. Разработка карт памяти для лекций данного курса, и разбор ошибок.
   2. Программные средства групповой разработки карт памяти. Обзор продукта Comapping. Самостоятельные упражнения в построение карт памяти с помощью Comapping.
8. **Разработка документации.**
   1. Основные виды документации программного проекта: документы с требованиями, проектные спецификации, руководства пользователя. Самостоятельная разработка этих видов документов для проектов, в которых участвовали обучающиеся. Разбор типовых ошибок.
   2. Обучение навыкам использования продукта Microsoft Word: использование заголовков и стилей, нумерация страниц и генерация оглавления, различные ссылки, параметры страницы и параметры печати, форматы хранения. Обзор системы LaTex, разработка собственных примеров, dvi-утилита, генерация pdf-файлов.

**Раздел 3. Обеспечение учебных занятий**

**3.1. Методическое обеспечение**

**3.1.1 Методические указания по освоению дисциплины**

Для освоения дисциплины обучающиеся должны посещать лекции и практические занятия, выполнять задания преподавателей.

**3.1.2 Методическое обеспечение самостоятельной работы**

При самостоятельном изучении теоретического материала и выполнении практических заданий целесообразно использовать рекомендованную основную и дополнительную литературу.

**3.1.3 Методика проведения текущего контроля успеваемости и промежуточной аттестации и критерии оценивания**

Зачёт проводится в устной форме. Билет состоит из двух вопросов. Время подготовки ответа на вопросы билета составляет 60 минут.

При подготовке к ответу обучающийся может использовать литературу, личный конспект и интернет-источники.

После ответа на вопросы билета преподаватель вправе задать дополнительные вопросы по любой теме из списка вопросов, вынесенных на теоретический зачёт, на основании оценки ответов на которые итоговая оценка по предмету может быть повышена или понижена. В качестве дополнительных используются вопросы, не требующие длительного вывода, в том числе основные определения, примеры и логические связи, введенные в дисциплине.

Критерии аттестации:

* 1. Оценка «зачтено» выставляется, если выполняются оба условия:
     1. обучающимся в целом дан ответ по всем вопросам билета (возможно с помощью наводящих подсказок преподавателя), либо дан полный ответ на один из вопросов билета, по второму вопросу представлены основные определения и формулировки;
     2. обучающийся отвечает более чем на 60% дополнительных вопросов.
  2. Оценка «не зачтено» выставляется, если не выполняются условия для получения оценки «зачтено».

Критерии выставления оценок в системе ECTS:

1. Оценка «A» — полный ответ по всем вопросам билета, ответ на 90% или более дополнительных вопросов.
2. Оценка «B» — полный ответ по всем вопросам билета, ответ на 60–90% дополнительных вопросов.
3. Оценка «C» — полный ответ по всем вопросам билета после наводящих подсказок преподавателя, ответ более, чем на 60% дополнительных вопросов.
4. Оценка «D» — по одному из вопросов билета представлены лишь определения и формулировки, но дан ответ на 90% или более дополнительных вопросов.
5. Оценка «E» — по одному из вопросов билета представлены лишь определения и формулировки, на 60–90% или более дополнительных вопросов.
6. Оценка «F» — в остальных случаях.

**3.1.4 Методические материалы для проведения текущего контроля успеваемости и промежуточной аттестации (контрольно-измерительные материалы, оценочные средства)**

Примерный список вопросов к зачёту:

1. История программной инженерии как науки. Основные достижения.
2. Виды компаний и проектов по разработке и сопровождению ПО.
3. Варианты карьеры (технический рост, переход в административную плоскость, переход в смежную предметную область). Необходимые навыки программиста, в том числе и при карьерном росте.
4. Текущие тенденции рынка производства ПО в России и Санкт-Петербурге.
5. Определения процесса разработки ПО. Фазы и виды деятельности процесса. Понятие модели процесса, виды модели процесса: водопадная, спиральная, итеративно-инкрементальная.
6. Артефакты разработки (код, спецификации модели, тесты, документация, процедуры процесса, средства разработки). Понятие рабочего продукта (work product).
7. Сравнение процесса с проектными планами, а также другими артефактами разработки.
8. Роли в проекте. Дисциплина обязательств как основа рабочей этики.
9. Хирургическая бригада Брукса. Модель команды в MSF.
10. Модель команды в Scrum. Команда и другие stakeholders проекта.
11. Scrum: история основные идеи, структура команды, порядок проведения итерации, практики, ограничения и предпосылки успешного применения.
12. CMM: история, 5-ти уровневая структура стандарта, понятие ключевой области процесса, краткий обзор каждого из уровней.
13. Open-source модель разработки.
14. Свойства файлов как единиц учета. Определение конфигурационного управления.
15. Практики версионного контроля исходных кодов проекта. Варианты организации структуры версионного хранилища. Процедуры доступа к веткам, перемещение исходников из одной ветки в другую. Svn.
16. Управление сборками (build management). Сценарии работы сборщика. Регулярная ночная сборка, сборка по запросу, continuous integration.
17. Сборщики и регрессионное тестирование. Примеры простейших сборщиков на основе bat-файлов. Обзор современных средств разработки сборщиков (MSBuild, ant, TFS).
18. Управление версиями продукта. Особенности управления версиями многоверсионного продукта, поставляемого разным заказчикам. Release management.
19. Определение тестирования. Парадигмы тестирования – белый и черный ящик. Виды тестирования: модульное, системное, интеграционное, стрессовое, нагрузочное, регрессионное. Другие средства обеспечения качества ПО: верификация, улучшение процесса и др.
20. Варианты организации команды тестировщиков и ее взаимодействия с разработчиками.
21. Bugtracking системы. Варианты реализации жизненного цикла ошибки.
22. Модульное тестирование. Дисциплина и правила написания модульных тестов. NUnit/JUnit.
23. Понятие проекта, основные особенности: нетипичная задача, необходимость спец усилий по фокусировке, команда. Отличие проекта от функциональной модели организации производства. Stakeholders проекта.
24. Рамки проекта. Требования и план проекта. Управление деньгами проекта. Внешнее оправдание и внутреннее совершенство. Треугольник компромиссов MSF.
25. Понятие рисков. Понятие безнадежных проектов по Йордану.
26. Карьера менеджера software-проекта.
27. Трудности управления требованиями к программным продуктам. Понятие требований как иерархической системы дискретных описаний свойств ПО.
28. Функциональные и нефункциональные требования. Трудности выявления нефункциональных требований. Проблемы недовыявленных требований.
29. Техническое задание проекта. Типичные ошибки.
30. Приемы интервьюирования пользователей/экспертов предметной области для выявления требований. Примеры UML-диаграмм для выявления требований. Использование диаграмм случаев использования.
31. Карты памяти: основные идеи и элементы. История карт памяти. Обзор областей применения.
32. Основные виды документации программного проекта: документы с требованиями, проектные спецификации, руководства пользователя.
33. Система LaTeX.

|  |  |  |
| --- | --- | --- |
| № | Код индикатора и индикатор достижения универсальной компетенции | Контрольно-измерительные материалы (КИМ) (тестовые вопросы, контрольные задания, кейсы и пр.) |
|  | 1 | 2 |
| 1 | ОПК-1.2 Определение качественных характеристик каждого компонента | активность обучающегося на практических занятиях модулей 1, 2, 4, 7, 8 независимо оценивается по шкале от 0 (не сделано) до 10 (очень хорошо), далее оценка усредняется. Результат переводится в диапазон от 0 до 100 |
| 2 | ОПК-2.1 Уметь писать программный код с использованием языков программирования, определения и манипулирования данными | активность обучающегося на практических занятиях модулей 1, 2, 4, 7, 8 независимо оценивается по шкале от 0 (не сделано) до 10 (очень хорошо), далее оценка усредняется. Результат переводится в диапазон от 0 до 100 |
| 3 | ОПК-6.2 Уметь определять перечень возможных шаблонов (стилей) проектирования для каждого слоя или компонента | активность обучающегося на практических занятиях модуля 5 оценивается по шкале от 0 (не сделано) до 100 (очень хорошо) |
| 4 | ОПК-7.2 Оценка осуществимости функционирования и сопровождения программного средства | активность обучающегося на практических занятиях модулей 1, 2, 4, 7, 8 независимо оценивается по шкале от 0 (не сделано) до 10 (очень хорошо), далее оценка усредняется. Результат переводится в диапазон от 0 до 100 |
| 5 | ПКП-1.1 Уметь разрабатывать технические спецификации на программные компоненты и их взаимодействие | активность обучающегося на практических занятиях модуля 6 оценивается по шкале от 0 (не сделано) до 100 (очень хорошо) |
| 6 | ПКП-2.1 Уметь описывать алгоритмы компонентов, включая методы и схемы | активность обучающегося на практических занятиях модуля 3 оценивается по шкале от 0 (не сделано) до 100 (очень хорошо) |
| 7 | ПКП-3.1 Разработка модели бизнес-процессов заказчика | активность обучающегося на практических занятиях модуля 5 оценивается по шкале от 0 (не сделано) до 100 (очень хорошо) |
| 8 | ПКП-4.1 Оценка и выбор варианта архитектуры программного средства | активность обучающегося на практических занятиях модуля 6 оценивается по шкале от 0 (не сделано) до 100 (очень хорошо) |
| 9 | ПКП-5.1 Уметь выбирать технологии и средства разработки программного обеспечения | активность обучающегося на практических занятиях модуля 5 оценивается по шкале от 0 (не сделано) до 100 (очень хорошо) |
| 10 | ПКП-6.1 Уметь разрабатывать процедуры проверки работоспособности и измерения характеристик программного обеспечения | ответы на оба вопроса на зачёте и ответы на дополнительные вопросы независимо оцениваются по шкале от 0 (нет ответа) до 10 (очень хорошо), далее оценка усредняется. Результат переводится в диапазон от 0 до 100 |
| 11 | ПКП-7.2 Оценка и выбор архитектуры развертывания каждого компонента | активность обучающегося на практических занятиях модулей 1, 2, 4, 7, 8 независимо оценивается по шкале от 0 (не сделано) до 10 (очень хорошо), далее оценка усредняется. Результат переводится в диапазон от 0 до 100 |
| 12 | УК 1.1. Анализирует задачу, выделяя ее базовые составляющие; | активность обучающегося на практических занятиях модуля 5 оценивается по шкале от 0 (не сделано) до 100 (очень хорошо) |
| 13 | УК-2.3. Оценивает соответствие способов решения цели проекта; | активность обучающегося на практических занятиях каждого модуля независимо оценивается по шкале от 0 (не сделано) до 10 (очень хорошо), далее оценка усредняется. Результат переводится в диапазон от 0 до 100 |
| 14 | УКБ-1.6. Представляет результаты проекта; | активность обучающегося на практических занятиях модуля 6 оценивается по шкале от 0 (не сделано) до 100 (очень хорошо) |
| 15 | УКБ-3.1. Находит и использует различные источники информации. | ответы на оба вопроса на зачёте и ответы на дополнительные вопросы независимо оцениваются по шкале от 0 (нет ответа) до 10 (очень хорошо), далее оценка усредняется. Результат переводится в диапазон от 0 до 100 |

**3.1.5 Методические материалы для оценки обучающимися содержания и качества учебного процесса**

Для оценки обучающимися содержания и качества учебного процесса применяется анкетирование в соответствии с методикой и графиком, утвержденными в установленном порядке.

**3.2. Кадровое обеспечение**

**3.2.1 Образование и (или) квалификация штатных преподавателей и иных лиц, допущенных к проведению учебных занятий**

К преподаванию дисциплины могут быть допущены преподаватели, имеющие диплом о высшем образовании по соответствующему направлению.

**3.2.2 Обеспечение учебно-вспомогательным и (или) иным персоналом**

Специальных требований нет.

**3.3. Материально-техническое обеспечение**

**3.3.1 Характеристики аудиторий (помещений, мест) для проведения занятий**

Учебные аудитории для проведения учебных занятий, оснащенные стандартным оборудованием, используемым для обучения в СПбГУ в соответствии с требованиями материально-технического обеспечения.

**3.3.2 Характеристики аудиторного оборудования, в том числе неспециализированного компьютерного оборудования и программного обеспечения общего пользования**

Стандартное оборудование, используемое для обучения в СПбГУ. MS Windows, MS Office, Mozilla FireFox, Google Chrome, Acrobat Reader DC, WinZip, Антивирус Касперского.

**3.3.3 Характеристики специализированного оборудования**

Не требуется.

**3.3.4 Характеристики специализированного программного обеспечения**

Не требуется.

**3.3.5 Перечень и объёмы требуемых расходных материалов**

Для аудиторий с маркерными досками необходимы стирающиеся маркеры в объёме, достаточном для проведения курса. Для аудиторий с меловыми досками необходим мел в объёме, достаточном для проведения курса. Канцелярские принадлежности в объёме, достаточном для проведения курса.

**3.4. Информационное обеспечение**

1. Д.В.Кознов. Программная инженерия. ЧастьI. - СПб: Изд-во СПбГУ, 2005. Мм – 75 экз.

2. Басс, Лен . Архитектура программного обеспечения на практике: учебное пособие / Л. Басс, П. Клементс, Р. Кацман. -2-е изд. - СПб: Питер, 2006. - 574 с. Мм – 13 экз.

3. Брукс Ф. Мифический человеко-месяц или как создаются программные системы. - СПб, 2007. – 298 с. Мм – 10 экз.

4. Липаев, Владимир Васильевич . Качество программного обеспечения: производственно-практическое издание / В. В. Липаев. - М. : Финансы и статистика, 1983. - 263 с. Мм – 11 экз.

5. Сайт Научной библиотеки им. М. Горького СПбГУ: <http://www.library.spbu.ru/>

6. Электронный каталог Научной библиотеки им. М. Горького СПбГУ: <http://www.library.spbu.ru/cgi-bin/irbis64r/cgiirbis_64.exe?C21COM=F&I21DBN=IBIS&P21DBN=IBIS>

7. Перечень электронных ресурсов, находящихся в доступе СПбГУ: <http://cufts.library.spbu.ru/CRDB/SPBGU/>

8. Перечень ЭБС, на платформах которых представлены российские учебники, находящиеся в доступе СПбГУ: <http://cufts.library.spbu.ru/CRDB/SPBGU/browse?name=rures&resource%20type=8>

**Раздел 4. Разработчики программы**

Кознов Дмитрий Владимирович. д.тех.н. доцент профессор кафедры системного программирования [d.koznov@spbu.ru](mailto:d.koznov@spbu.ru)