**Санкт-Петербургский государственный университет**

**Р А Б О Ч А Я П Р О Г Р А М М А**

**УЧЕБНОЙ ДИСЦИПЛИНЫ**

Введение в программную инженерию

Introduction to Software Engineering

**Язык(и) обучения**

русский

Трудоемкость в зачетных единицах: 3

Регистрационный номер рабочей программы: 003590

**Раздел 1. Характеристики учебных занятий**

**1.1. Цели и задачи учебных занятий**

Дисциплина «Введение в программную инженерию» является одной из важных дисциплин основной образовательной программы бакалавриата СВ.5080.\* «Программная инжененрия», предназначенной для введения обучающихся в специальность программиста, а также для сообщения знаний основ процесса разработки ПО. Она представляет ориентацию в мире различных ИТ-профессий, а также знания методологий, моделей и различных видов деятельности при разработке ПО. Отдельные параметры курса могут варьироваться по степени сложности в зависимости от уровня подготовки обучающихся.

Цель изучения дисциплины: сориентировать обучающихся в их дальнейшей профессиональной деятельности и обеспечить их базовыми знаниями в области процесса разработки ПО.

Задачи курса: изучение различных моделей процесса разработки (водопадной, спиральной, итеративно-инкрементальной, V-образной), изучение современных методологий и стандартов разработки (CMM, Scrum, EP), а также основ тестирования, управления требованиями, конфигурационного управления и управления проектами.

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

Программа курса предназначена для обучающихся 3 курса, изучавших программирование в объеме первых двух курсов и владеющих базовыми навыками работы с компьютером.  
Максимальная эффективность программы будет обеспечена при условии, что обучающийся:  
• освоил базовый курс по программированию (языки Java, C#, Phyton);  
• владеет основами работы с известными IDEs (Visual Studio, IDEA).

**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 | Определение и свойства ПО и процесса разработки | ОПК-1.1 Уметь идентифицировать возможные проблемы и пути их решения |
| 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 | Модели разработки ПО (водопадная, спиральная, итеративно-инкрементальная, V-образная) | 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 | Знание современных методологий и стандартов разработки (CMM, Scrum, EP). Agile-манифест | ОПК-6.2 Уметь определять перечень возможных шаблонов (стилей) проектирования для каждого слоя или компонента |
| 4 | Общепрофессиональные к6омпетенции | ОПК-7 Способен применять в практической деятельности основные концепции, принципы, теории и факты, связанные с информатикой | Знание современных методологий и стандартов разработки (CMM, Scrum, EP). Agile-манифест | ОПК-7.2 Оценка осуществимости функционирования и сопровождения программного средства |
| 5 | Профессиональные компетенции | ПКП-1 Способен проектировать программные системы | Краткие основы UML | ПКП-1.1 Уметь разрабатывать технические спецификации на программные компоненты и их взаимодействие |
| 6 | Профессиональные компетенции | ПКП-5 Способен использовать современные системные программные средства: операционные системы, операционные и сетевые оболочки, сервисные программы | Основы тестирования, управления требованиями, конфигурационного управления и управления проектами | ПКП-5.1 Уметь выбирать технологии и средства разработки программного обеспечения |
| 7 | Профессиональные компетенции | ПКП-6 Способен формировать суждения о проблемах современной информатики, ее категорий и связей с другими научными дисциплинами | Понимание ИТ-проекта как единого целого, а также его отдельных составных частей | ПКП-6.1 Уметь разрабатывать процедуры проверки работоспособности и измерения характеристик программного обеспечения |
| 8 | Профессиональные компетенции | ПКП-7 Способен систематизировать и применять знания о содержании основных этапов и тенденций развития программирования, математического обеспечения и информационных технологий | Начальные навыки планирования проектами | ПКП-7.1 Умеет проводить технические и управленческие ревизии создаваемого программного средства |
| 9 | Универсальные компетенции | УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач | Структурирование информации с помощью mind maps | УК 1.4. Оценивает достоинства, недостатки и последствия вариантов решения поставленных задач; |
| 10 | 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 | Понимание ИТ-индустрии, вариантов карьеры и карьерного роста | УКБ-3.2. Точно определяет тип и форму необходимой информации. |

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

Аудиторная учебная работа:   
• Лекционные занятия в объеме 2 часов в неделю.  
  
Самостоятельная работа:  
• Выполнение домашних заданий по курсу.

**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 3 | 30 |  | 2 |  |  |  |  |  | 2 |  |  |  | 37 |  | 37 |  | 10 | 3 |
|  | 2-42 |  | 2-25 |  |  |  |  |  | 2-25 |  |  |  | 1-1 |  | 1-1 |  |  |  |
| ИТОГО | 30 |  | 2 |  |  |  |  |  | 2 |  |  |  | 37 |  | 37 |  |  | 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 3 |  |  | экзамен, устно, традиционная форма | по графику промежуточной аттестации |  |  |

**2.2. Структура и содержание учебных занятий**

Период обучения (модуль): Семестр 3

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Наименование темы (раздела, части) | Вид учебных занятий | Количество часов |
| I. | Варианты карьеры в ИТ-индустрии. | Лекции | 2 |
| Практические занятия | 0 |
| II. | Программное обеспечение: определение, свойства, виды | Лекции | 2 |
| Практические занятия | 0 |
| III. | Процесс разработки ПО | Лекции | 2 |
| Практические занятия | 0 |
| IV. | Модели процесса разработки ПО | Лекции | 4 |
| Практические занятия | 0 |
| V. | Разработка требований | Лекции | 2 |
| Практические занятия | 0 |
| VI. | Анализ и проектирование | Лекции | 2 |
| Практические занятия | 0 |
| VII. | Конфигурационное управление и управление изменениями | Лекции | 2 |
| Практические занятия | 0 |
| VIII. | Основы тестирования | Лекции | 2 |
| Практические занятия | 0 |
| IX. | Управление проектами: планирование | Лекции | 2 |
| Практические занятия | 0 |
| X. | Управление проектами: работа с рисками | Лекции | 2 |
| Практические занятия | 0 |
| XI. | Команда проекта | Лекции | 2 |
| Практические занятия | 0 |
| XII. | Capability Maturity Model | Лекции | 2 |
| Практические занятия | 0 |
| XIII. | Agile-манифест, XP, Scrum | Лекции | 3 |
| Практические занятия | 0 |
| XIV. | Mind Maps | Лекции | 1 |
| Практические занятия | 0 |
|  | Промежуточная аттестация | Консультация | 2 |
| Промежуточная аттестация (экзамен) | 10 |

**Содержание модулей**

1. Варианты карьеры в ИТ-индустрии.

Рассмотрены следующие варианты карьеры: Технический специалист; менеджмент, бизнес; переход в смежную предметную область; «зацепиться» за технологию технологию (например, Oracle, C++, Java); карьера в большой компании; фрилансер, эксперт; академическая карьера. Обсуждается динамическая модель карьеры.

2. Программное обеспечение: определение, свойства, виды

Вводится определение ПО по Х.Милсу. Рассматриваются свойства ПО по Ф.Бруксу. Обсуждаются виды ПО по И.Сомервилу.

3. Процесс разработки ПО

Даётся определение процесса разработки по И.Сомервилу. Определяются фазы и виды деятельности процесса. Рассматривается деятельность по совершенствованию процесса (software process improvement). Рассматриваются две стратегии улучшения: technology push и organization pull.

4. Модели процесса разработки

Определение модели процесса разработки. Водопадная, спиральная, итеративно-инкрементальная и V-образные модели. Достоинства и недостатки этих моделей. От моделей процесса к методологиям.

5. Разработка требований

Определение требования. Атрибуты требования Фазы работы с требованиями. Виды спецификаций требований: техническое задание, формальная модель, граф, неформальный подход. Примеры технических заданий. Трудности работы с техническими заданиями. Изменчивость требования и задача requirement traceability. Современные программные средства работы с требованиями. О языке описания требований.

6. Анализ и проектирование

Анализ – построение функциональной модели ПО. Понятие архитектуры ПО. Легковесный и тяжеловесные подходы к проектированию. Что проектируем (разбиение ПО на компоненты и модули, интерфейсы, производительность и пр.). Обзор UML. Мотивация и слабые стороны практического использования UML. Прагматичный подход к проектированию

7. Конфигурационное управление и управление изменениями

Версионный контроль кода проекта. Элементы конфигурационного управления. «Ветки» проекта. Сборка проекта (build management). Современные средства создания сборщиков. Интеграция версионного контроля, сборки и регрессионного тестирования. Современные средства версионного контроля. GitHub и его место в современной разработке. Понятие baseline и процедура внесения изменений в проект.

8. Основы тестирования

Определение теста. Общая схема тестирования. Виды тестирования (модульное, интеграционное, стрессовое, системное, нагрузочное и пр.). Обзор современных средств тестирования. Тестировани е, основанное на моделях. Тестирование Web-приложений. Тестирование систем реального времени. О карьере тестировщика.

9. Управление проектами: планирование

О необходимости планов. Параметры планирования: сроки, люди, деньги, фичи. Отличие планов от требований и от архитектуры. Средства планирования – Microsoft Excel и Microsoft Project. Отслеживание планов. Рабочий продукт. Дисциплина обязательств.

10. Управление проектами: управление рисками

Понятие риска. Атрибуты рисков. Техники предупреждения рисков. Что делать, когда риски произошли (реализовались). Треугольник компромиссов.

11. Команда проекта

Общепринятые роли в проекте. Модель бригады главного хирурга. Модель команды MSF. Совместимые и несовместимые роли. Иерархические и «плоские» команды: сильные и слабые стороны. О масштабируемости команды.

12. Capability Maturity Model (CMM)

Тяжеловесные методологии разработки ПО. История CMM. Понятие ключевой области процесса (key process area). Понятие уровни зрелости процесса. Обзор уровней зрелости процесса. Процедура сертификации компании. ISO 9000. Сертификация в авионике.

13. Agile-манифест, XP, Scrum

Agile-движение и agile-манифест. Основные практики XP. История появления XP. Scrum: модель команды, модель процесса разработки, основные артефакты.

14. Mind Maps

Предназначение и история создания. Минималистическая и детальная стратегии создания mind maps. Программные средства работы с mind maps. Основные ошибки.

Лектору следует использовать mind maps для предложения различных заданий по структурированию материала лекций и организации связи между различными темами. Примеры заданий:

* 1. Создать схему вакансий в ИТ-индустрии (на примере российского рынка).
  2. Начертить схему различных карьер, включая необходимые компетенции.
  3. Идентифицировать и детализировать (включая связи) рабочие продукта ВКР бакалавра.

**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.

На лекционных занятиях преподаватель рассказывает материал курса согласно следующему содержанию в разбивке по разделам и модулям. Помимо предложенных информационных источников, могут цитироваться и демонстрироваться фрагмменты из следующих источников:

1. Карл И. Вигерс. Разработка требований к программному обеспечению. Русская редакция, 2004.
2. Хаммер М., Чампи Дж. Реинжениринг корпорации: Манифест революции в бизнесе. СПб.: Издательство С.-Петербургского университета, 1997.
3. Эдвард Йордан. Путь камикадзе. Как разработчику программного обеспечения выжить в безнадёжном проекте. BOOKS-SHOP.COM. 2002.

Также целесообразно предложить обучающимся задание по планированию – «планирование ближайшей сессии».

Следует наладить рабочий процесс по высылке и проверке домашних заданий, в частности, следует назначить точное время высылки сделанных заданий, чтобы у преподавателя была возможность их проверить перед занятием, а также согласовать форматы для оформления заданий.

**3.1.2 Методическое обеспечение самостоятельной работы**

При самостоятельном изучении теоретического материала, выполнении практических заданий и во время подготовки доклада целесообразно использовать рекомендованную основную и дополнительную литературу. По согласованию с преподавателем обучающийся может осваивать теоретическую часть курса по одному из онлайн-курсов, представленных на образовательных платформах в п. 3.4.3. При этом преподаватель обязан сообщить обучающимся, какие разделы выбранного им онлайн-курса недостаточно полно раскрывают ту или иную тему курса и порекомендовать дополнительные источники по данной теме. Преподаватель вправе отказать обучающемуся в самостоятельном освоении теоретической части дисциплины по онлайн-курсу в случае нахождения в нем существенных расхождений с содержанием курса в разделе 2 и п. 3.1.1.

Практическая часть дисциплины может быть полностью выполнена самостоятельно по методическим материалам.

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

Текущий контроль успеваемости состоит в выполнении всех практических заданий по работе с базой данных, предлагаемых обучающимся в течение семестра. По мере выполнения этого требования обучающемуся выдается индивидуальное проектное задание (определенная предметная область и набор требований для реализации). Проект должен быть сдан преподавателю не позднее установленной даты. Выбор даты рекомендуется сделать не позднее первых четырех недель семестра. Рекомендуемый диапазон для даты сдачи проекта – 2 недели, предшествующие зачетной.

Промежуточная аттестация представляет собой экзамен в устной форме. Допуском к экзамену является выполнение всех практических заданий. Экзаменационный билет состоит из двух вопросов. Время подготовки ответа на вопросы билета составляет не менее 1 академического часа. Использование конспектов и учебников, а также электронных устройств хранения, обработки или передачи информации при подготовке и ответе на вопросы экзамена запрещено.

После ответа на основные вопросы билета, преподаватель вправе задать уточнящие вопросы по услышанному. Затем преподаватель задает дополнительные вопросы по любой теме из списка вопросов, вынесенных на экзамен. В качестве дополнительных используются вопросы, не требующие длительного ответа, в том числе основные определения и понятия. Рекомендуется задавать 3-4 дополнительных вопроса на билет и не более 2-3 уточняющих вопросов. Преподаватель вправе увеличить количество вопросов в случае, если у него не возникает понимание, освоил обучающийся материал учебного курса или нет.

**Критерии оценивания по пятибалльной шкале**

Оценка «отлично» ставится в случае, если обучающийся ответил на все вопросы из билета, уточняющие и дополнительные. Допускаются небольшие неточности и недоговорки.

Оценка «хорошо» ставится в случае, если обучающийся ответил на все вопросы из билета и уточняющие и на половину дополнительных (с округлением вверх) или ответил на все основные, уточняющие и доролнительные, но допустил несколько негрубых ошибок.

Оценка «удовлетворительно» ставится, если обучающийся ответил на все вопросы из билета, уточняющие и один дополнительный, или ответил на все основные, уточняющие и дополнительные вопросы с существенными ошибками.

Оценка «неудовлетворительно» ставится во всех случаях, не подходящих под вышеуказанные критерии.

**Критерии оценивания в системе ECTS.**

Оценка «A» ставится в тех же случаях, что и оценка «отлично».

Оценка «B» ставится, если выполнены требования для оценки «хорошо» и при этом в ответе допущено не более двух неточностей.

Оценка «C» ставится, если выполнены требования для оценки «хорошо» и при этом в ответе допущено более двух неточностей.

Оценка «D» ставится, если выполнены требования для оценки «удовлетворительно» и при этом в ответе допущено не более одной грубой ошибки.

Оценка «E» ставится, если выполнены требования для оценки «удовлетворительно» и при этом в ответе допущено более одной грубой ошибки.

Оценка «F» ставится в тех же случаях, что и оценка «неудовлетворительно».

В любой момент обучающийся имеет право отказаться от ответа с выставлением оценки «неудовлетворительно» (F).

По желанию преподавателя на экзамен допустимо приглашать других преподавателей с квалификацией не ниже изложенной в п. 3.2.1 как для независимого оценивания ответов обучающихся, так и для коллегиального. В последнем случае оценка за экзамен ставится на основании голосования простого большинства. В спорных ситуациях преподаватель, ведущий дисциплину, имеет право принятия окончательного решения.

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

Контроль выполнения практических заданий осуществляется путем проверки на соответствие демонстрируемого обучающимся результата требованиям задания, перечисленным в п. 3.1.1.

Пример списка вопросов, выносимых на экзамен.

1. Определение программного обеспечения.
2. Свойства (признаки) свойства ПО. Виды ПО
3. Процесс разработки ПО
4. Водопадная модель
5. Спиральная модель
6. Итеративно-инкрементальная модель
7. V-образная модель
8. Определение требования. Атрибуты требования.
9. Жизненный цикл разработки требований
10. Способы формализации требований
11. Инструменты разработки требований
12. Анализ ПО: функциональная модель ПО.
13. Определение архитектуры ПО
14. Аспекты ПО, которые следует проектировать.
15. Элементы конфигурационного управления
16. Сборка проекта
17. Baseline и управление изменениями
18. Средств версионного контроля
19. Средства разработки сборщиков
20. GitHub и его роль в современной разработке
21. Определение тестирования
22. Виды тестирования
23. Тестирование Web-приложений
24. Тестирование систем реального времени
25. Современные инструменты тестирования
26. Карьера тестировщика
27. Управление проектами: планирование
28. Управление проектами: работа с рисками
29. Команда проекта
30. Capability Maturity Model: назначение и история
31. Ключевые области процесса
32. Уровни зрелости процесса
33. Обзор уровней зрелости
34. Agile-манифест
35. Методология XP
36. Методология Scrum

|  |  |  |
| --- | --- | --- |
| № | Код индикатора и индикатор достижения универсальной компетенции | Контрольно-измерительные материалы (КИМ) (тестовые вопросы, контрольные задания, кейсы и пр.) |
|  | 1 | 2 |
| 1 | ОПК-1.1 Уметь идентифицировать возможные проблемы и пути их решения | Ответы на оба вопроса на экзамене и ответы на дополнительные теоретические вопросы независимо оцениваются по шкале от 0 (нет ответа/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 | Ответы на оба вопроса на экзамене и ответы на дополнительные теоретические вопросы независимо оцениваются по шкале от 0 (нет ответа/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 | Ответы на оба вопроса на экзамене и ответы на дополнительные теоретические вопросы независимо оцениваются по шкале от 0 (нет ответа/не сделано) до 10 (очень хорошо), далее оценка усредняется. Результат переводится в диапазон от 0 до 100 |
| 4 | ОПК-7.2 Оценка осуществимости функционирования и сопровождения программного средства | Ответы на оба вопроса на экзамене и ответы на дополнительные теоретические вопросы независимо оцениваются по шкале от 0 (нет ответа/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 | Ответы на оба вопроса на экзамене и ответы на дополнительные теоретические вопросы независимо оцениваются по шкале от 0 (нет ответа/не сделано) до 10 (очень хорошо), далее оценка усредняется. Результат переводится в диапазон от 0 до 100 |
| 6 | ПКП-5.1 Уметь выбирать технологии и средства разработки программного обеспечения | Ответы на оба вопроса на экзамене и ответы на дополнительные теоретические вопросы независимо оцениваются по шкале от 0 (нет ответа/не сделано) до 10 (очень хорошо), далее оценка усредняется. Результат переводится в диапазон от 0 до 100 |
| 7 | ПКП-6.1 Уметь разрабатывать процедуры проверки работоспособности и измерения характеристик программного обеспечения | Ответы на оба вопроса на экзамене и ответы на дополнительные теоретические вопросы независимо оцениваются по шкале от 0 (нет ответа/не сделано) до 10 (очень хорошо), далее оценка усредняется. Результат переводится в диапазон от 0 до 100 |
| 8 | ПКП-7.1 Умеет проводить технические и управленческие ревизии создаваемого программного средства | Ответы на оба вопроса на экзамене и ответы на дополнительные теоретические вопросы независимо оцениваются по шкале от 0 (нет ответа/не сделано) до 10 (очень хорошо), далее оценка усредняется. Результат переводится в диапазон от 0 до 100 |
| 9 | УК 1.4. Оценивает достоинства, недостатки и последствия вариантов решения поставленных задач; | Ответы на оба вопроса на экзамене и ответы на дополнительные теоретические вопросы независимо оцениваются по шкале от 0 (нет ответа/не сделано) до 10 (очень хорошо), далее оценка усредняется. Результат переводится в диапазон от 0 до 100 |
| 10 | УКБ-3.2. Точно определяет тип и форму необходимой информации. | Ответы на оба вопроса на экзамене и ответы на дополнительные теоретические вопросы независимо оцениваются по шкале от 0 (нет ответа/не сделано) 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 техническом образовании по направлениям «Программная инженерия», «Математическое обеспечение и администрирование информационных систем» и другим смежным направлениям, связанным с информационными технологиями.

**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**

Специализированное оборудование не требуется.

**3.3.4 Характеристики специализированного программного обеспечения**

Специализированное программного обеспечения не требуется.

**3.3.5 Перечень и объёмы требуемых расходных материалов**

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

**3.4. Информационное обеспечение**

**3.4.1 Список литературы**

1. Сомервилл И. Инженерия программного обеспечения. — М.: Вильямс, 2002  
2. Кулямин В.В. Технология программирования. Компонентный подход. М.: Интернет-Университет, 2010.  
3. Брукс Ф. Мифический человеко-месяц или как создаются программные системы. Символ-Плюс. Любой год издания.

4. Guide to the Software Engineering Body of Knowledge (SWEBOK), русский перевод <http://studopedia.su/11_49856_perevod-SWEBOK-na-russkiy-yazik.html>

5. J. von Neumann. First Draft of a Report on the EDVAC. University of Pennsylvania. 1945. <https://fab.cba.mit.edu/classes/862.16/notes/computation/vonNeumann-1945.pdf>

6. Кознов Д.В. Введение в программную инженерию. Часть 1. СПбГУ, 2005.   
7. Кознов Д.В. Бугайченко Д. Введение в программную инженерию. 2008. <http://window.edu.ru/resource/409/61409/files/koznov-lectures.pdf>

**3.4.2 Перечень иных информационных источников**

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

• 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>

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

• 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 Доктор технических наук доцент Профессор кафедры системного программирования d.koznov@spbu.ru