**Санкт-Петербургский государственный университет**

**Р А Б О Ч А Я П Р О Г Р А М М А**

**УЧЕБНОЙ ДИСЦИПЛИНЫ**

Программная инженерия

Software Engineering

**Язык(и) обучения**

русский

Трудоемкость в зачетных единицах: 2

Регистрационный номер рабочей программы: 003735

**Раздел 1. Характеристики учебных занятий**

**1.1. Цели и задачи учебных занятий**

Дисциплина «Программная инженерия» является обобщающим курсом, суммирующим знания обучающихся ООП бакалавриата 09.03.04 «Программная инженерия» в данной области и освещающем современные тенденции мира промышленной разработки программного обеспечения. Некоторая эклектичность курса позволяет обратиться к различным, новым, обновляемым и активно развивающимся аспектам, таким как лицензирование, современные IDEs, современное понимание архитектуры ПО, документирование и др. Приведение в более стройную систему этой информации требует времени, эклектичность же позволяет остановиться на отдельных, очень современных вопросах.

Цель изучения дисциплины: повторить и углубить знания о методах и процессе разработке ПО, ознакомиться с современными реалиями промышленной разработки ПО.

Задачи курса: изучение способов наладки процесса разработки ПО, вопросы управления командой и требованиями, лицензирование, стандартизация в промышленной разработке, свойства и тенденции развития современных средств разработки (IDEs, открытых фреймворков), углубленное изучение вопросов разработки документации (комментарии к коду, дизайн-спецификации, API-документация, пользовательская документация к фреймворкам разработки ПО, а также спецификации требований), включая знакомство с современными средствами разработки документации, такими как JavaDoc.

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

Программа курса предназначена для обучающихся 4 курса, имеющих навыки программирования и использования сред разработки (MS Visual Studio, Intellij IDEA Eclipse и др.), а также имеющих знания в области управления проектами и требованиями, проектировании и тестировании.

**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 | Знания современных тенденции разработки ПО: процесс разработки, управления командой и требованиями, нове понимание архитектуры ПО | 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 | Знания современных тенденции разработки ПО: процесс разработки, управления командой и требованиями, нове понимание архитектуры ПО | ОПК-2.2 Уметь проводить формализацию и алгоритмизацию поставленных задач |
| 3 | Общепрофессиональные компетенции | ОПК-5 – способен инсталировать программное и аппаратное обеспечение для информационных и автоматизированных систем; | Знание структуры и функций современных IDEs; открытых фреймворков разработки и их места в современной индустрии | ОПК-5.1 Установка и настройка системного и прикладного ПО, необходимого для функционирования ИС в соответствии с трудовым заданием |
| 4 | Общепрофессиональные компетенции | ОПК-8 – способен осуществить поиск, хранение, обработку и анализ информации из различных источников и баз данных, представлять ее в требуемом формате с использованием информационных, компьютерных и сетевых технологий; | Знание структуры и функций современных IDEs; открытых фреймворков разработки и их места в современной индустрии | ОПК-8.1 Оценка и выбор технологии доступа к данным |
| 5 | Профессиональные компетенции | ПКП-1 – способен проектировать программные системы; | Знание структуры и функций современных IDEs; открытых фреймворков разработки и их места в современной индустрии | 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; | Навыки использования технологии JavaDoc | 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; | Навыки использования технологии JavaDoc | 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; | Знание структуры и функций современных IDEs; открытых фреймворков разработки и их места в современной индустрии | ПКП-4.1 Оценка и выбор варианта архитектуры программного средства |
| 9 | Профессиональные компетенции | ПКП-6 – способен формировать суждения о проблемах современной информатики, ее категорий и связей с другими научными дисциплинами; | Знание особенностей задачи сопровождения документации, включая внесение точечных изменений, поддержание актуальности и корректности документации, её связи с другими артефактами разработки (прежде всего, с программным кодом). | ПКП-6.1 Уметь разрабатывать процедуры проверки работоспособности и измерения характеристик программного обеспечения |
| 10 | Профессиональные компетенции | ПКП-7 – способен систематизировать и применять знания о содержании основных этапов и тенденций развития программирования, математического обеспечения и информационных технологий; | Знание современных тенденций развития программного обеспечения широкого диапазона типов вычислительных систем, в том числе суперкомпьютерных комплексов | ПКП-7.1 Умеет проводить технические и управленческие ревизии создаваемого программного средства |
| 11 | Универсальные компетенции | УК-1 – способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач; | Понимание современных реалий мира промышленной разработки ПО | УК 1.5. Грамотно, логично, аргументированно формирует собственные суждения, решения и оценки. |
| 12 | 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; | Понимание современных реалий мира промышленной разработки ПО | УК-2.4. Планирует реализацию задач в зоне своей ответственности с учетом имеющихся ресурсов и ограничений, действующих правовых норм; |
| 13 | Универсальные компетенции | УКБ-1 – способен участвовать в разработке и реализации проектов, в т.ч. предпринимательских; | Понимание основных задач и инструментов организации процесса разработки | УКБ-1.5. Выполняет задачи в зоне своей ответственности в соответствии с запланированными результатами и точками контроля; |
| 14 | 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, искусственного интеллекта и науки о данных, а также информационной безопасности | Понимание необходимости документации ПО, трудоёмкости разработки и сопровождения документации. | УКБ-3.4. Проверяет достоверность собранной информации. |

1) Знания:  
 современных тенденции разработки ПО: процесс разработки, управления командой и требованиями, нове понимание архитектуры ПО;  
 о стандартизации в разработке ПО;  
 о лицензиях в мире разработки ПО (GPL, GDPR);  
 структура и функции современных IDEs; открытые фреймворки разработки и их место в современной индустрии;   
 особенностей разработки документации ПО – комментариев к коду, дизайн-спецификаций, API-документации, пользовательской документации к фреймворкам разработки ПО, а также спецификации требований;  
 особенностей задачи сопровождения документации, включая внесение точечных изменений, поддержание актуальности и корректности документации, её связи с другими артефактами разработки (прежде всего, с программным кодом).   
   
2) Умения:  
 навыки использования технологии JavaDoc.   
  
3) Понимание:  
 современных реалий мира промышленной разработки ПО;  
 основных задач и инструментов организации процесса разработки;  
 необходимости документации ПО, трудоёмкости разработки и сопровождения документации.

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

Лекции – 10 ак. ч.

**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 7 | 60 |  |  |  |  |  |  |  | 2 |  |  |  |  |  | 10 |  | 10 | 2 |
|  | 10-25 |  |  |  |  |  |  |  | 10-25 |  |  |  |  |  | 1-1 |  |  |  |
| ИТОГО | 60 |  |  |  |  |  |  |  | 2 |  |  |  |  |  | 10 |  |  | 2 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 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 7 |  |  | зачёт, устно, традиционная форма | по графику промежуточной аттестации |  |  |

**2.2. Структура и содержание учебных занятий**

Период обучения (модуль): Семестр 7

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Наименование темы (раздела, части) | Вид учебных занятий | Количество часов |
| I. | Исторический и современный взгляды на процесс разработки ПО | Лекции | 4 |
| II. | Пример «тяжеловесной» методологии разработки: RUP/USDP | Лекции | 4 |
| III. | Agile-манифест. Методологии XP и Scrum | Лекции | 4 |
| IV. | Современное понятие архитектуры ПО | Лекции | 6 |
| V. | Обзор современных IDEs и фреймворков разработки | Лекции | 4 |
| VI. | Команда проекта | Лекции | 4 |
| VII. | Управление проектами | Лекции | 4 |
| VIII. | Разработка требований | Лекции | 4 |
| IX. | Стандартизация в индустрии | Лекции | 4 |
| X. | Лицензия по защите персональных данных GDPR | Лекции | 4 |
| XI. | Документация ПО | Лекции | 4 |
| XII. | Особенности разработки документации ПО | Лекции | 4 |
| XIII. | Комментарии к коду | Лекции | 4 |
| XIV. | Документация ПО и верификация | Лекции | 6 |
|  | Аттестация | Сам. работа | 10 |
| Промежуточная аттестация (зачет) | 2 |

На лекционных занятиях преподаватель рассказывает материал курса согласно следующему содержанию в разбивке по разделам:

1. Исторический и современный взгляды на процесс разработки ПО.

Интуитивное определение процесса: есть процесс – нет процесса. Определение процесса разработки по И.Сомервилу. Фазы и виды деятельности процесса. Естественные (человеческие) аспекты процесса. Совершенствованию процесса (software process improvement). Стратегии улучшения: technology push и organization pull. Комплексные средства автоматизации процесса (единый workflow). Bug trakers/issue trackers – бездна возможностей. «Тяжеловесные» и «легковесные» процессы. Модель Open Source разработки. Проблемы Open Source, примеры известных проектов. Лицензия GPL (GNU General Public License).

1. Пример «тяжеловесной» методологии разработки: RUP/USDP.

Итеративно-инкрементальная модель процесса. Виды деятельности. Итерации. Инструментальная поддержка процесса.

1. Agile-манифест. Методологии XP и Scrum.

Agile-движение и agile-манифест. История и основные практики XP. Scrum: модель команды, модель процесса разработки, основные артефакты. Когда Scrum работает, а когда нет.

1. Современное понятие архитектуры ПО.

Современные тенденции разработки ПО как вызовы для разработки архитектуры. Исторические аспекты архитектуры. Современное определение архитектуры. Атрибуты качества архитектуры. Гибкие методы (agile) и архитектура. Фреймворки, библиотеки, контейнеры. Deployability Tactics. Technical debts.

1. Обзор современных IDEs и фреймворков разработки.

IDE – не только компилятор: обзор функциональности. Фичинг и современные средства обработки данных (машинное обучение, обработка текстов на естественных языках, рекомендательные системы). Языки программирования и фитчинг (на примере языка С++). Академические методы и фитчинг (на примере алгоритмов clone detection). Обзор среды разработки Intellij Idea. Свободно распространяемые фреймворки и их место в современных средствах разработки. Примеры.

1. Команда проекта

Иерархические и «плоские» команды. О совмещении функций управления проектом и разработкой ПО. Отношения команды и компании, рабство и демократия. И снова «брать на себя». Команда и заказчик. Команда и пользователи. Команда и начальство. О масштабировании команды. О рекрутинге и кадровых проблемах в индустрии разработки ПО.

1. Управление проектами.

Понятие проекта. Масштаб проекта и основные действующие лица (stakeholders). Управление людьми, деньгами, информацией. Что значит «брать на себя». О коммуникациях в проекте: электронные и человеческие. Шаблоны взаимодействия с членами коллектива. Понятие риска и атрибуты рисков. Виды рисков. Техники предупреждения рисков. Управление рисками.

1. Разработка требований.

Техническое задание, виды технических заданий. Жизненный цикл работы с требованиями. Примеры технических заданий. Трудности работы с техническими заданиями. Изменчивость требования и задача requirement traceability. О языке описания требований в документации. Юридические аспекты требований. Различные способы использования требований в разработке.

1. Стандартизация в индустрии.

Комитет ISO. Стандарт ISO 9000. О сертификации в авионике. Стандарт CMM-I: назначение, история, структура. Сертификация команды/компании по CMM-I.

1. Лицензия по защите персональных данных GDPR.

История вопроса. Цель данной лицензии. Кто и как может под неё подпасть, и чем это грозит. Особенности создания новых средств разработки в свете GDPR. Пример со средством обработки crash-отчётов.

1. Документация ПО.

История вопроса. Виды документации. Выгоды, цена и трудозатраты и проблемы документации. Двухуровневая модель разработки документации (примеры: LaTex и DocBook). Классификация документации по Д.Парнасу: reference и narrative документация.

1. Особенности разработки документации ПО.

Практические виды документации: комментарии к коду, дизайн-спецификации, API-документация, пользовательская документация к фреймворкам разработки ПО, спецификации требований. Особенности жизненного цикла и разработки этих видов документации. Проблемы и задачи в разработке документации. Целостность и актуальность документации. Связь документации с другими артефактами разработки (прежде всего, с кодом). О задаче автоматической генерации документации по различным артефактам (прежде всего, по коду).

1. Комментарии к коду.

Понятие комментария. Введение в JavaDoc: виды сущностей, которые комментируются; обзор языка разметки; средства автоматической обработки. Обзор практики комментирования кода в Java-мире. Виды комментариев к коду для различных языков – Java, Python, C++. Краткий обзор языков и технологий разработки комментариев для различных языков программирования.

1. Документация ПО и верификация.

О верификационной «вилке» или где брать нужную информацию. Верификационные движки, схемы верификации. «Слои» документации и спецификаций в проекте. Проблема documentation inconsistency. Формальные и неформальные документы. Формальные, полуформальные и неформальные визуальные модели, средства анализа таких моделей. Использование онтологий. Что получаем в итоге? О трудозатратах и рисках верифкационного процесса. Когда это действительно работает.

**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 занятиях могут демонстрироваться и цитироваться фрагменты следующих источников:

* Карл И. Вигерс. Разработка требований к программному обеспечению. Русская редакция, 2015-2017.

**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 дате проведения зачета.

Промежуточная аттестация представляет собой зачет в устной форме. Допуском к зачету является выполнение всех практических заданий. Обучающийся получает два вопроса. Время подготовки ответа на вопросы 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 запрещено.

После ответа на основные вопросы, преподаватель вправе задать уточнящие вопросы по услышанному. Затем преподаватель задает дополнительные вопросы по любой теме из списка вопросов, вынесенных на зачет. В качестве дополнительных используются вопросы, не требующие длительного ответа, в том числе основные определения и понятия. Рекомендуется задавать 3-4 дополнительных вопроса 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.

Оценивание производится по следующим критериям.

| **Качество ответов обучающегося во время промежуточной аттестации** | **Оценка СПбГУ** | **Оценка ECTS** |
| --- | --- | --- |
| Обучающийся ответил на все вопросы из билета, уточняющие и дополнительные. | Зачтено | A |
| Обучающийся ответил на все вопросы из билета, уточняющие и дополнительные. Допускаются небольшие неточности и недоговорки. | B |
| Обучающийся ответил на все вопросы из билета и уточняющие и на 2/3 дополнительных (с округлением вверх) или ответил на все основные, уточняющие и дополнительные, но допустил не более двух негрубых ошибок. | C |
| Обучающийся ответил на все вопросы из билета и уточняющие и 1/3 дополнительных (с округлением вверх) или ответил на все основные, уточняющие и дополнительные, но допустил несколько негрубых ошибок. | D |
| Обучающийся ответил на все вопросы из билета, уточняющие и один дополнительный, или ответил на все основные, уточняющие и дополнительные вопросы с существенными ошибками. | E |
| Остальные случаи, включая отказ от ответа в любой момент аттестации. | Не зачтено | F |

По желанию преподавателя на зачет допустимо приглашать других преподавателей с квалификацией не ниже изложенной в п. 3.2.1 как для независимого оценивания ответов обучающегося, так и для коллегиального. В последнем случае оценка за зачет 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 зачет.

1. Интуитивное определение процесса: есть процесс/нет процесса.

2. Определение процесса разработки по И.Сомервилу.

3. Фазы и виды деятельности процесса.

4. Совершенствованию процесса (software process improvement). Стратегии совершенствования процесса: technology push и organization pull.

5. Комплексные средства автоматизации процесса (единый workflow).

6. Возможности bug trakers/issue trackers по организации процесса.

7. Модель Open Source разработки. Проблемы Open Source, примеры известных проектов.

8. Лицензия GPL (GNU General Public License).

9. Итеративно-инкрементальная модель процесса RUP/USDP. Виды деятельности. Итерации.

10. Инструментальная поддержка процесса.

11. Agile-движение и agile-манифест.

12. История и основные практики XP.

13. Scrum: модель команды, модель процесса разработки, основные артефакты.

14. Когда Scrum работает, а когда нет.

15. Современные тенденции разработки ПО как вызовы для разработки архитектуры.

16. Исторические аспекты архитектуры. Современное определение архитектуры.

17. Атрибуты качества архитектуры. Гибкие методы (agile) и архитектура.

18. Фреймворки, библиотеки, контейнеры.

19. Deployability Tactics.

20.Technical debts.

21. IDE – не только компилятор: обзор функциональности.

22. Фичинг и современные средства обработки данных (машинное обучение, обработка текстов на естественных языках, рекомендательные системы).

23. Языки программирования и фитчинг (на примере языка С++).

24. Академические методы и фитчинг (на примере алгоритмов clone detection).

25. Обзор среды разработки Intellij Idea.

26. Свободно распространяемые фреймворки и их место в современных средствах разработки. Примеры.

27. Иерархические и «плоские» команды.

28. О совмещении функций управления проектом и разработкой ПО.

29. Отношения команды и компании, рабство и демократия. И снова «брать на себя».

30. Команда и заказчик. Команда и пользователи. Команда и начальство. О масштабировании команды.

31. О рекрутинге и кадровых проблемах в индустрии разработки ПО.

32. Понятие проекта. Масштаб проекта и основные действующие лица (stakeholders).

33. Управление людьми, деньгами, информацией. Что значит «брать на себя».

34. О коммуникациях в проекте: электронные и человеческие. Шаблоны взаимодействия с членами коллектива.

35. Понятие риска и атрибуты рисков. Виды рисков. Техники предупреждения рисков. Управление рисками.

36. Техническое задание, виды технических заданий.

37. Жизненный цикл работы с требованиями. Примеры технических заданий.

38. Трудности работы с техническими заданиями. Изменчивость требования и задача requirement traceability.

39. О языке описания требований в документации.

40. Юридические аспекты требований.

41. Способы использования требований в разработке.

42. Комитет ISO. Стандарт ISO 9000.

44. О сертификации в авионике.

45. Стандарт CMM-I: назначение, история, структура. Сертификация команды/компании по CMM-I.

46. Лицензия по защите персональных данных GDPR.

47. Документация ПО: история вопроса.

48. Виды документации.

49. Выгоды, цена и трудозатраты и проблемы документации.

50. Двухуровневая модель разработки документации (примеры: LaTex и DocBook).

51. Классификация документации по Д.Парнасу: reference и narrative документация.

52. Практические виды документации: комментарии к коду, дизайн-спецификации, API-документация, пользовательская документация к фреймворкам разработки ПО, спецификации требований.

53 О задаче автоматической генерации документации по различным артефактам (прежде всего, по коду).

54. Комментарии к коду. Введение в JavaDoc: виды сущностей, которые комментируются; обзор языка разметки; средства автоматической обработки.

55. Обзор практики комментирования кода в Java-мире.

56. Виды комментариев к коду для различных языков – Java, Python, C++. Краткий обзор языков и технологий разработки комментариев для различных языков программирования.

57. О верификационной «вилке» или где брать нужную информацию. Верификационные движки, схемы верификации.

58. «Слои» документации и спецификаций в проекте. Проблема documentation inconsistency. Формальные и неформальные документы. Формальные, полуформальные и неформальные визуальные модели, средства анализа таких моделей.

59. О трудозатратах и рисках верифкационного процесса. Когда это действительно работает.

|  |  |  |
| --- | --- | --- |
| № | 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 зачёте и ответы на дополнительные теоретические вопросы независимо оцениваются по шкале от 0 (нет ответа) до 10 (очень хорошо), далее оценка усредняется. Результат переводится в диапазон от 0 до 100 |
| 2 | ОПК-2.2 Уметь проводить формализацию и алгоритмизацию поставленных задач | обучающемуся предлагается продемонстрировать, как реализуются та или иная функциональность, рассмотренная в модуле 5, в установленной среде или платформе. Результат оцениваются по шкале от 0 (не сделано) до 100 (очень хорошо). |
| 3 | ОПК-5.1 Установка и настройка системного и прикладного ПО, необходимого для функционирования ИС в соответствии с трудовым заданием | обучающемуся предлагается установить одну из сред или платформ, рассмотренных в модуле 5. Результат оцениваются по шкале от 0 (не сделано) до 100 (очень хорошо). |
| 4 | ОПК-8.1 Оценка и выбор технологии доступа к данным | обучающемуся предлагается продемонстрировать, как реализуются та или иная функциональность, рассмотренная в модуле 5, в установленной среде или платформе. Результат оцениваются по шкале от 0 (не сделано) до 100 (очень хорошо). |
| 5 | ПКП-1.1 Уметь разрабатывать технические спецификации на программные компоненты и их взаимодействие | обучающемуся предлагается рассказать, как установленная среда или платформа влияет на архитектуру проекта (или наоборот) и на разработку конкретных модулей. Результат оцениваются по шкале от 0 (не сделано) до 100 (очень хорошо). |
| 6 | ПКП-2.1 Уметь описывать алгоритмы компонентов, включая методы и схемы | обучающемуся предлагается продемонстрировать, как реализуются та или иная функциональность, рассмотренная в модуле 5, в установленной среде или платформе. Результат оцениваются по шкале от 0 (не сделано) до 100 (очень хорошо). |
| 7 | ПКП-3.1 Разработка модели бизнес-процессов заказчика | обучающемуся предлагается рассказать, как установленная среда или платформа влияет на архитектуру проекта (или наоборот) и на разработку конкретных модулей. Результат оцениваются по шкале от 0 (не сделано) до 100 (очень хорошо). |
| 8 | ПКП-4.1 Оценка и выбор варианта архитектуры программного средства | обучающемуся предлагается рассказать, как установленная среда или платформа влияет на архитектуру проекта (или наоборот) и на разработку конкретных модулей. Результат оцениваются по шкале от 0 (не сделано) до 100 (очень хорошо). |
| 9 | ПКП-6.1 Уметь разрабатывать процедуры проверки работоспособности и измерения характеристик программного обеспечения | ответы на оба вопроса на зачёте и ответы на дополнительные теоретические вопросы независимо оцениваются по шкале от 0 (нет ответа) до 10 (очень хорошо), далее оценка усредняется. Результат переводится в диапазон от 0 до 100 |
| 10 | ПКП-7.1 Умеет проводить технические и управленческие ревизии создаваемого программного средства | ответы на оба вопроса на зачёте и ответы на дополнительные теоретические 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 | УК 1.5. Грамотно, логично, аргументированно формирует собственные суждения, решения и оценки. | обучающемуся предлагается продемонстрировать, как реализуются та или иная функциональность, рассмотренная в модуле 5, в установленной среде или платформе. Результат оцениваются по шкале от 0 (не сделано) до 100 (очень хорошо). |
| 12 | УК-2.4. Планирует реализацию задач в зоне своей ответственности с учетом имеющихся ресурсов и ограничений, действующих правовых норм; | ответы на оба вопроса на зачёте и ответы на дополнительные теоретические вопросы независимо оцениваются по шкале от 0 (нет ответа) до 10 (очень хорошо), далее оценка усредняется. Результат переводится в диапазон от 0 до 100 |
| 13 | УКБ-1.5. Выполняет задачи в зоне своей ответственности в соответствии с запланированными результатами и точками контроля; | обучающемуся предлагается рассказать, как установленная среда или платформа влияет на архитектуру проекта (или наоборот) и на разработку конкретных модулей. Результат оцениваются по шкале от 0 (не сделано) до 100 (очень хорошо). |
| 14 | УКБ-3.4. Проверяет достоверность собранной информации. | ответы на оба вопроса на зачёте и ответы на дополнительные теоретические 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 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**

Не требуется.

**3.3.4 Характеристики специализированного программного обеспечения**

Не требуется.

**3.3.5 Перечень и объёмы требуемых расходных материалов**

Не требуется.

**3.4. Информационное обеспечение**

1. Кознов Д.В. Программная инженерия. Часть 1. СПбГУ, 2005.

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

3. Кознов Д.В. Бугайченко Д. Введение в программную инженерию. 2008. <http://window.edu.ru/resource/409/61409/files/koznov-lectures.pdf> - ЭР открытого доступа в сети Интернет

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

5.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>

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

7. 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