**Правительство Российской Федерации**

**Санкт-Петербургский государственный университет**

**Р А Б О Ч А Я П Р О Г Р А М М А**

**УЧЕБНОЙ ДИСЦИПЛИНЫ**

Технология разработки программного обеспечения

Software Technology

**Язык(и) обучения**

русский

Трудоемкость в зачетных единицах: 2

Регистрационный номер рабочей программы: 002190

Санкт-Петербург

2024

**Раздел 1. Характеристики учебных занятий**

**1.1. Цели и задачи учебных занятий**

Целью учебных занятий по дисциплине является формирование у обучающихся знаний о процессах, практиках и стандартах, принятых в индустрии при работе над проектами по разработке программного обеспечения (ПО).

Задачами учебных занятий по дисциплине являются:

* сформировать у обучающихся систему знаний о жизненном цикле ПО и основных его этапах, об актуальных методологиях разработки ПО и применяемых в них практиках, о работе по выявлению и управлению требованиями, о выделении рисков проекта и построении стратегий противодействия им, о критериях качества ПО и методах повышения качества кода в проектах, о групповой разработке и правилах построения команд, о практиках эффективного управления проектами, о методах и средствах проектирования пользовательских интерфейсов и архитектуры ПО;
* сформировать умения эффективно управлять системами конфигурационного управления, инструментами сборки проекта, системами непрерывной интеграции кода;
* сформировать навыки декомпозиции проектов, оценки задач, построения графика проекта, написания читаемого и сопровождаемого кода, создания модульных и других тестов.

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

Учебные занятия по дисциплине «Технология разработки программного обеспечения» проводятся в 6-м семестре для обучающихся бакалавриата по направлению подготовки 02.03.03 «Математическое обеспечение и администрирование информационных систем».

Для успешного освоения дисциплины обучающийся должен иметь предварительную подготовку по дисциплинам информатики и программирования, изучаемых на I-II курсах математико-механического факультета СПбГУ или аналогичную подготовку.

**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 | Теоретические и практические основы профессиональной деятельности | ОПК-1 — способен применять фундаментальные знания, полученные в области математических и (или) естественных наук, и использовать их в профессиональной деятельности | * способен применять основы теории графов при планировании проекта | ОПК-1.002190.1. Строит план проекта с учётом последовательности работ, выраженных в виде сетевого графика. |
| 2 | Информационно-коммун икационные технологии для профессиональной деятельности | ОПК-3 — способен понимать и применять современные информационные технологии, в том числе отечественные, при создании программных продуктов и программных комплексов различного назначения | * понимает значение информационных технологий в планировании и контроле исполнения проекта | ОПК-3.002190.1. Применяет инструменты планирования для создания графика работ, оценки загруженности ресурсов и бюджета проекта. |
| 3 | Информационно-коммун икационные технологии для профессиональной деятельности | ОПК-4 — способен участвовать в разработке технической документации программных продуктов и программных комплексов | * владеет принципами составления и документирования требований к программному обеспечению | ОПК-4.002190.1. Перечисляет и поясняет основные виды документов, относящихся к требованиям. |
| 4 | Информационно-коммун икационные технологии для профессиональной деятельности | ОПК-5 — способен инсталлировать и сопровождать программное обеспечение для информационных систем и баз данных, в том числе отечественного производства | * владеет принципами сопровождения программного обеспечения | ОПК-5.002190.1. Объясняет принципы и факторы стоимости сопровождения программного обеспечения, понятие реинжиниринга. |
| 5 | Профессиональные компетенции (академические) | ПКА-1 — способен демонстрировать базовые знания математических и естественных наук, программирования и информационных технологий | * демонстрирует знания в сфере управления проектами | ПКА-1.002190.1. Перечисляет и поясняет фазы жизненного цикла программного обеспечения, основные роли в команде, принципы оценки, планирования и контроля проектов, управления требованиями и дефектами. |
| 6 | Профессиональные компетенции (практические) | ПКП-1 — способность проводить под научным руководством исследование на основе существующих методов в конкретной области профессиональной деятельности | * обладает навыками планирования программной части научных исследований | ПКП-1.002190.1. Строит план своей научной работы по правилам и практикам планирования проекта. |
| 7 | Профессиональные компетенции (практические) | ПКП-2 — способен решать задачи в области развития науки, техники и технологии с учетом нормативного правового регулирования в сфере интеллектуальной собственности | * обладает представлением об общей структуре хода проекта по разработке ПО, владеет основами связанной с этим правовой базы | ПКП-2.002190.1. Делает осознанный выбор модели проекта исходя из требований интеллектуальной собственности. |
| 8 | Профессиональные компетенции (практические) | ПКП-3 — способен преподавать математику и информатику в средней школе, специальных учебных заведениях на основе полученного фундаментального образования и научного мировоззрения | * умеет внятно рассказать и объяснить материал курса | ПКП-3.002190.1. Строит корректный и связный рассказ по предложенной теме из покрываемых курсом. |
| 9 | Профессиональные компетенции (практические) | ПКП-5 — способен использовать основные методы и средства автоматизации проектирования, реализации, испытаний и оценки качества при создании конкурентоспособного программного продукта и программных комплексов, а также способен использовать методы и средства автоматизации, связанные с сопровождением, администрированием и модернизацией программных продуктов и программных комплексов | * знает о средствах автоматизации планирования, декомпозиции задач, отслеживания дефектов, отслеживания задач, системах модульного тестирования и непрерывной интеграции | ПКП-5.002190.1. Перечисляет и указывает достоинства и недостатки популярных систем автоматизации планирования, декомпозиции задач, отслеживания дефектов, отслеживания задач, систем модульного тестирования и непрерывной интеграции. |
| 10 | Профессиональные компетенции (практические) | ПКП-6 — способен использовать знания направлений развития компьютеров с традиционной (нетрадиционной) архитектурой; современных системных программных средств: операционных систем, операционных и сетевых оболочек, сервисных программ; тенденции развития функций и архитектур проблемно-ориентированных программных систем и комплексов в условиях цифровой экономики | * применяет знания о современных процессорных архитектурах, операционных системах, оболочках, сервисных программах, тенденции развития проблемно-ориентированных программных систем при планировании проекта | ПКП-6.002190.1. Планирует проект по заданным требованиям с учётом знаний о направлениях развития процессорных архитектур, операционных систем и оболочек, сервисных программ, проблемно-ориентированных программных систем. |
| 11 | Профессиональные компетенции (практические) | ПКП-7 — способен использовать основные концептуальные положения функционального, логического, объектно-ориентированного и визуального направлений программирования, методы, способы и средства разработки программ в рамках этих направлений | * владеет знаниями о рефакторингах в объектно-ориентированных программах | ПКП-7.002190.1. Объясняет основные виды рефакторингов и ситуации их применимости. |
| 12 | Универсальные компетенции бакалавра | 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 | * понимает сущность и значение информации в развитии общества; * владеет методами получения и работы с информацией. | УКБ-3.1. Находит и использует различные источники информации.  УКБ-3.2. Точно определяет тип и форму необходимой информации.  УКБ-3.3. Получает информацию и сохраняет ее в удобном для работы формате.  УКБ-3.4. Проверяет достоверность собранной информации. |

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

В качестве основных интерактивных форм (общее количество 3 часа) предполагается:

* практическая работа в аудитории, где обучающиеся в командах создают план некоего проекта (2 часа);
* активные лекции, предполагающие обсуждение с преподавателем (1 час).

**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 6 | 30 |  |  |  |  |  |  |  | 2 |  |  |  | 14 |  | 26 |  | 3 | 2 |
|  | 2-100 |  |  |  |  |  |  |  | 1-30 |  |  |  | 1-1 |  | 1-1 |  |  |  |
| ИТОГО | 30 |  |  |  |  |  |  |  | 2 |  |  |  | 14 |  | 26 |  |  | 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 6 |  |  | зачёт, устно, традиционная форма | по графику промежуточной аттестации |  |  |

**2.2. Структура и содержание учебных занятий**

Период обучения (модуль): семестр 6.

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Наименование темы (раздела, части) | Вид учебных занятий | Количество часов |
| I. | Введение | лекции | 2 |
| по методическим материалам | 1 |
| II. | Жизненный цикл ПО, методологии и процессные фреймворки | лекции | 4 |
| по методическим материалам | 2 |
| III. | Требования и проектирование GUI | лекции | 4 |
| по методическим материалам | 2 |
| IV. | Роль и функции менеджера проекта | лекции | 6 |
| по методическим материалам | 3 |
| V. | Групповая разработка | лекции | 2 |
| по методическим материалам | 1 |
| VI. | Качество программного обеспечения | лекции | 4 |
| по методическим материалам | 2 |
| VII. | Сопровождение, рефакторинг и реинжиниринг | лекции | 4 |
| по методическим материалам | 1 |
| VIII. | Развёртывание и непрерывная интеграция | лекции | 2 |
| по методическим материалам | 1 |
| IX. | Экономические аспекты разработки ПО | лекции | 2 |
| по методическим материалам | 1 |
| X. | Промежуточная аттестация | самостоятельная работа | 26 |
| зачёт | 2 |
| **Итого** | | | **108** |

Раздел 1: Введение

1. Программа и программный продукт, понятие программной инженерии, состав команды, виды деятельности разработчика ПО.

Раздел 2: Жизненный цикл ПО, методологии и процессные фреймворки

1. Жизненный цикл ПО, модели ЖЦ. Rational Unified Process и eXtreme Programming как примеры методологий разработки.
2. Процессный фреймворк Scrum: роли, артефакты, основные практики и мероприятия.

Раздел 3: Требования и проектирование GUI

1. Виды и характеристики требований. Инженерия требований: выявление, анализ, провека, спецификация, управление требованиями.
2. Проектирование GUI. Понятие User Experience, User-centered design, Activity-centred design. Персонажи и сценарии. Подходы к прототипированию пользовательских интерфейсов: storytelling, бумажные прототипы, bodystorming, макеты, дизайн-макеты, интерактивные прототипы. Подходы к исследованию удобства использования продукта, юзабилити-исследования.

Раздел 4: Роль и функции менеджера проекта

1. Функции менеджера проекта: определение проекта, планирование, управление и контроль. Матрица ответственностей, план коммуникаций. Управление рисками. Декомпозиция проекта.
2. Построение графика работ: матрица зависимостей, сетевой график, диаграмма Гантта. Оценка задач и графика в целом. Типичные ошибки при оценке проектов. Треугольник равновесия проекта. Балансирование равновесия проекта: основные подходы и практики. Отслеживание прогресса хода проекта и реагирование на отклонения от плана. Организация команды: типы команд, роли участников.
3. Практическая работа: составление диаграммы Гантта для проекта.

Раздел 5: Групповая разработка

1. Понятие команды. Составляющие эффективной команды. Практики формирования позитивной экосистемы и формирования навыков совместного решения задач. Особенности формирования команды. Командная разработка ПО. Системы контроля версий. Разработка с использованием веток в git. Удачные модели коллективной разработки ПО в распределённых системах контроля версий.

Раздел 6: Качество программного обеспечения

1. Понятие тестирования, виды тестирования. Тестирование и жизненный цикл программного обеспечения. Тестовые сценарии. Инструменты тестирования: библиотеки модульного тестирования, мок-объектьов, тестирвоания пользовательских интерфейсов, системы управления тестированием. Понятие ошибки, атрибуты отчётов об ошибках, жизненный цикл ошибки, системы отслеживания ошибок.
2. Понятие качества программного обеспечения, характеристики качества. Модели и метрики качества программного обеспечения. Качество процессов, модель CMMI, стандарты качества.

Раздел 7: Сопровождение, рефакторинг и реинжиниринг

1. Динамика развития программных проектов, законы Лемана. Понятие унаследованной системы. Сопровождение, организация процесса сопровождения. Понятие реинжиниринга, применимость и оценка стоимости реинжиниринга.
2. Понятие рефакторинга. Описание популярных рефакторингов.

Раздел 8: Развёртывание и непрерывная интеграция

1. Автоматизация развёртывания. Антипаттерны управления релизами. Основные принципы непрерывного развёртывания. Конфигурационное управление, непрерывная интеграция. Структура и шаги построения конвейера развёртывания. Модель зрелости процесса управления релизами.

Раздел 9: Экономические аспекты разработки ПО

1. Финансовые риски проекта, оценка рисков, типовые риски. Модели финансовых взаимоотношений с заказчиком. Основы российского налогового законодательства касательно сферы информационных технологий.

**Раздел 3. Обеспечение учебных занятий**

**3.1. Методическое обеспечение**

**3.1.1 Методические указания по освоению дисциплины**

Основными видами занятий при изучении дисциплины «Технология разработки программного обеспечения» являются лекции и самостоятельная работа обучающихся.  
В рамках лекционного материала основное внимание уделяется изложению теоретических основ курса.

**3.1.2 Методическое обеспечение самостоятельной работы**

Основная и дополнительная литература.

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

***3.1.3.1. Методика проведения текущего контроля успеваемости и промежуточной аттестации***

В формировании итоговой оценки по курсу участвует оценка за практическую работу и результаты устного зачёта.

Практическая работа выполняется в виде индивидуального задания, где прямо на занятии обучающимся предлагается выполнить планирование проекта, указанного в задании, и сдать его на проверку с помощью системы поддержки обучения.Использование любых источников разрешено. На выполнение задания выделяется два академических часа.

Устный зачёт проводится в соответствии с заранее опубликованным списком вопросов, покрывающих рассмотренные на лекционных занятиях темы. Обучающийся должен в ходе зачёта ответить на два вопроса, без подготовки и без использования литературы и других материалов. Преподаватель вправе задавать уточняющие вопросы по ходу ответа, а также после ответа на основные два вопроса задавать дополнительные вопросы по всему материалу курса.

***3.1.3.2. Критерии оценивания итогового процента освоения дисциплины***

Решения практической работы оцениваются по шкале от 0 (нет решения) до 5 (очень хорошее решение, полностью выполняющее все требования условия и отражающее знания, полученные в ходе освоения дисциплины).

На зачёте каждый ответ оценивается по шкале от 0 (нет ответа) до 20 (очень хороший ответ). Результирующий процент освоения дисциплины получается следующим образом:

1. Оценки за ответы на вопросы усредняются.
2. К полученной оценке прибавляется оценка за проверочную работу.
3. Результат сложения приводится линейным преобразованием из диапазона от 0 до 25 в диапазон от 0 до 100.

Далее применяется единая для СПбГУ шкала оценивания ECTS:

|  |  |  |
| --- | --- | --- |
| Итоговый процент  выполнения, % | Оценка СПбГУ при  проведении зачёта | Оценка  ECTS |
| 90-100 | зачтено | A |
| 80-89 | зачтено | B |
| 70-79 | зачтено | C |
| 61-69 | зачтено | D |
| 50-60 | зачтено | E |
| менее 50 | не зачтено | F |

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

***3.1.4.1. Формируемые дисциплиной компетенции***

Дисциплина развивает следующие компетенции:

* ОПК-1 – способен применять фундаментальные знания, полученные в области математических и (или) естественных наук, и использовать их в профессиональной деятельности;
* ОПК-3 – способен применять современные информационные технологии, в том числе отечественные, при создании программных продуктов и программных комплексов различного назначения;
* ОПК-4 – способен участвовать в разработке технической документации программных продуктов и программных комплексов;
* ОПК-5 – способен инсталлировать и сопровождать программное обеспечение для информационных систем и баз данных, в том числе отечественного производства;
* ПКА-1 – способен демонстрировать базовые знания математических и естественных наук, программирования и информационных технологий;
* ПКП-1 – способность проводить под научным руководством исследование на основе существующих методов в конкретной области профессиональной деятельности;
* ПКП-2 – способен решать задачи в области развития науки, техники и технологии с учетом нормативного правового регулирования в сфере интеллектуальной собственности;
* ПКП-3 – способен преподавать математику и информатику в средней школе, специальных учебных заведениях на основе полученного фундаментального образования и научного мировоззрения;
* ПКП-5 – способен использовать основные методы и средства автоматизации проектирования, реализации, испытаний и оценки качества при создании конкурентоспособного программного продукта и программных комплексов, а также способен использовать методы и средства автоматизации, связанные с сопровождением, администрированием и модернизацией программных продуктов и программных комплексов;
* ПКП-6 – способен использовать знания направлений развития компьютеров с традиционной (нетрадиционной) архитектурой; современных системных программных средств: операционных систем, операционных и сетевых оболочек, сервисных программ; тенденции развития функций и архитектур проблемно-ориентированных программных систем и комплексов в условиях цифровой экономики;
* ПКП-7 – способен использовать основные концептуальные положения функционального, логического, объектно-ориентированного и визуального направлений программирования, методы, способы и средства разработки программ в рамках этих направлений;
* 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.

Шкала оценивания для каждой компетенции линейная, определяется долей успешно выполненных заданий, проверяющих данную компетенцию.

***3.1.4.2. Контрольно-измерительные материалы (примеры)***

***Пример списка вопросов для устного зачёта:***

1. Программная инженерия как область знания
2. Отличия разработки программного обеспечения от других инженерных областей
3. Компетенции и профстандарты в области программной инженерии
4. Понятие жизненного цикла программного обеспечения
5. Водопадная модель жизненного цикла
6. Итеративная, итеративно-инкрементальная и спиралевидная модели жизненного цикла
7. Rational Unified Process
8. Agile-подход к разработке
9. eXtreme Programming: общий подход, достоинства и недостатки
10. XP: практики «Короткий цикл разработки»
11. XP: практики «Непрерывность процесса»
12. XP: практики «Понимание, разделяемое всеми»
13. Scrum, роли в команде
14. Scrum, Backlog и спринты
15. ScrumAnd, ScrumBut, достоинства и недостатки методологии
16. Понятие и виды требований
17. Требования к требованиям
18. Работа с требованиями: выявление, анализ, проверка
19. Навыки и трудовые функции аналитика
20. Документы, связанные с требованиями
21. Моделирование требований
22. Спецификация требований к программному обеспечению (SRS)
23. Управление требованиями
24. Понятие User experience
25. Стадии проектирования пользовательского интерфейса
26. User-centered design, персонажи и сценарии
27. Activity-centred design
28. Data-driven design
29. Сторителлинг и раскадровки как инструмент проектирования UI
30. Макеты и дизайн-макеты как инструмент проектирования UI
31. Варианты исследования UX продукта
32. Usability-тестирование
33. Функции менеджера проекта
34. Матрица ответственности и план коммуникаций
35. Основные действия по управлению рисками
36. Декомпозиция проекта
37. Построение графика работ: матрица зависимостей, сетевой график
38. Построение графика работ: диаграмма Гантта
39. Типичные ошибки при оценке проектов
40. Треугольник равновесия проекта
41. Приёмы балансирования равновесия на уровне проекта
42. Приёмы балансирования равновесия на уровне бизнес-целей
43. Отслеживание прогресса проекта
44. Позитивная экосистема команды: базовые правила и сплочённость команды
45. Позитивная экосистема команды: умение слушать, умение проводить совещания
46. Совместное решение задач: анализ задач и варианты принятия решений
47. Совместное решение задач: разрешение конфликтов и непрерывное обучение
48. Особенности формирования команды
49. Особенности командной разработки программного обеспечения, ревью кода
50. Системы контроля версий: модели ветвления «Разработка в главной ветке» и «Gitflow»
51. Системы контроля версий: модели ветвления «GitHub flow», «GitLab flow», «OneFlow»
52. Понятие качества программного обеспечения
53. Характеристики качества программного обеспечения по ISO 25010:2011
54. Метрики качества программного обеспечения, классификация
55. Метрики Холстеда и цикломатическая сложность
56. Метрики Чидамбера и Кемерера
57. Метрики Лоренца и Кидда
58. Метрики MOOD
59. Модель зрелости компаний CMMI
60. Понятие и виды тестирования программного обеспечения, пирамида тестирования
61. Тестирование требований, тестирование архитектуры
62. Тестовые сценарии
63. Инструменты тестирования
64. Отслеживание ошибок, жизненный цикл ошибки
65. Законы Лемана эволюции программного обеспечения
66. Понятие и особенности унаследованных систем
67. Сопровождение программного обеспечения: факторы стоимости, прогнозирование, процесс
68. Организация технической поддержки программного обеспечения
69. Особенности работы с унаследованным кодом
70. Реинжиниринг программного обеспечения: варианты, факторы стоимости, основные этапы
71. Понятие рефакторинга
72. «Code smells»
73. Рефакторинги «Выделение метода» и «Перемещение метода»
74. Рефакторинги «Выделение класса» и «Выделение подкласса»
75. Рефакторинги «Сокрытие делегирования» и «Введение внешнего метода»
76. Рефакторинги «Самоинкапсуляция поля>> и «Введение Null-объекта»
77. Рефакторинги «Замена кода типа подклассами» и «Замена условного оператора полиморфизмом»
78. Рефакторинги «Замена конструктора фабричным методом» и «Замена наследования делегированием»
79. Проблемы при проведении рефакторинга. Причины не проводить рефакторинг
80. Понятие Continuous Delivery
81. Антипаттерны управления релизами
82. Преимущества частых автоматизированных релизов
83. Принципы непрерывного развёртывания, конфигурационное управление
84. Continuous Integration

*Проверяемые компетенции***:** ОПК-1, ОПК-3, ОПК-4, ОПК-5, ПКА-1, ПКП-1, ПКП-2, ПКП-3, ПКП-5, ПКП-6, ПКП-7, УКБ-3.

*Критерии оценивания***:** обучающемуся задаётся два вопроса и, возможно, задаётся несколько дополнительных вопросов по курсу. Ответ на каждый вопрос оценивается по шкале от 0 (нет ответа) до 20 (очень хороший ответ), далее оценка усредняется. Для вычисления уровня сформированности компетенций полученная оценка приводится из диапазона от 0 до 20 в диапазон от 0 до 100 линейным преобразованием.

***Пример задания для практической работы:***

Нарисовать диаграмму Гантта для проекта своей учебной практики в предположении, что над проектом будут работать 2-3 человека.

*Проверяемые компетенции***:** ОПК-1, ОПК-3, ОПК-4, ОПК-5, ПКА-1, ПКП-1, ПКП-2, ПКП-3, ПКП-5, ПКП-6, ПКП-7, УКБ-3.

*Критерии оценивания*: Задание оценивается по шкале от 0 (нет решения) до 5 (очень хорошее решение, полностью выполняющее все требования условия и отражающее знания, полученные в ходе освоения дисциплины).

***3.1.4.3. Соответствие индикаторов достижения компетенций и контрольно-измерительных материалов***

|  |  |  |
| --- | --- | --- |
| № | Код индикатора и индикатор достижения универсальной компетенции | Контрольно-измерительные материалы (КИМ) (тестовые вопросы, контрольные задания, кейсы и пр.) |
|  | 1 | 2 |
| 1 | ОПК-1.002190.1. Строит план проекта с учётом последовательности работ, выраженных в виде сетевого графика. | Пример задания для практической работы. |
| 2 | ОПК-3.002190.1. Применяет инструменты планирования для создания графика работ, оценки загруженности ресурсов и бюджета проекта. | Пример задания для практической работы. |
| 3 | ОПК-4.002190.1. Перечисляет и поясняет основные виды документов, относящихся к требованиям. | Пример вопросов для устного зачёта. |
| 4 | ОПК-5.002190.1. Объясняет принципы и факторы стоимости сопровождения программного обеспечения, понятие реинжиниринга. | Пример вопросов для устного зачёта. |
| 5 | ПКА-1.002190.1. Перечисляет и поясняет фазы жизненного цикла программного обеспечения, основные роли в команде, принципы оценки, планирования и контроля проектов, управления требованиями и дефектами. | Пример вопросов для устного зачёта. |
| 6 | ПКП-1.002190.1. Строит план своей научной работы по правилам и практикам планирования проекта. | Пример задания для практической работы. |
| 7 | ПКП-2.002190.1. Делает осознанный выбор модели проекта исходя из требований интеллектуальной собственности. | Пример вопросов для устного зачёта. |
| 8 | ПКП-3.002190.1. Строит корректный и связный рассказ по предложенной теме из покрываемых курсом. | Пример вопросов для устного зачёта. |
| 9 | ПКП-5.002190.1. Перечисляет и указывает достоинства и недостатки популярных систем автоматизации планирования, декомпозиции задач, отслеживания дефектов, отслеживания задач, систем модульного тестирования и непрерывной интеграции. | Пример вопросов для устного зачёта. |
| 10 | ПКП-6.002190.1. Планирует проект по заданным требованиям с учётом знаний о направлениях развития процессорных архитектур, операционных систем и оболочек, сервисных программ, проблемно-ориентированных программных систем. | Пример вопросов для устного зачёта, пример задания для практической работы. |
| 11 | ПКП-7.002190.1. Объясняет основные виды рефакторингов и ситуации их применимости. | Пример вопросов для устного зачёта. |
| 12 | УКБ-3.1. Находит и использует различные источники информации. | Пример вопросов для устного зачёта. |
| 13 | УКБ-3.2. Точно определяет тип и форму необходимой информации. | Пример вопросов для устного зачёта. |
| 14 | УКБ-3.3. Получает информацию и сохраняет ее в удобном для работы формате. | Пример вопросов для устного зачёта. |
| 15 | УКБ-3.4. Проверяет достоверность собранной информации. | Пример вопросов для устного зачёта. |

**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 и либо прошедшие обучение в аспирантуре не менее года, либо имеющие ученую степень.

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

Персонал для обслуживания проектора.

**3.3. Материально-техническое обеспечение**

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

В аудитории, где проводятся занятия, необходимо наличие досок и средств письма на них, а также технических средств, указанных в пункте 3.3.2.

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

В аудитории, где проводятся занятия, должен иметься проектор и экран.

**3.3.3 Характеристики специализированного оборудования**

Специальных требований нет.

**3.3.4 Характеристики специализированного программного обеспечения**

Специальных требований нет.

**3.3.5 Перечень и объёмы требуемых расходных материалов**

Специальных требований нет.

**3.4. Информационное обеспечение**

**3.4.1 Список обязательной литературы**

1. Kerzner, Harold. Project management a systems approach to planning, scheduling, and controlling. Hoboken, N.J.: John Wiley & Sons, Inc., 2013. URL: https://find.library.spbu.ru/vufind/Record/EBC1113482 (дата обращения: 31.03.2017)

2. Charrel, Pierre-Jean, Galarreta, Daniel. Project Management and Risk Management in Complex Projects. Dordrecht: Springer Netherlands, 2007. URL: https://find.library.spbu.ru/vufind/Record/978-1-4020-5837-0 (дата обращения: 31.03.2017)

3. Mohapatra, Pratap K. J. Software engineering (a lifecycle approach). New Delhi: New Age International, 2010. URL: https://find.library.spbu.ru/vufind/Record/EBC3017407 (дата обращения: 31.03.2017)

4. Tomayko, J. E. Human aspects of software engineering. Hingham, MA: Charles River Media, 2004. URL: https://find.library.spbu.ru/vufind/Record/EBC3135679 (дата обращения: 31.03.2017)

5. Mistrík, Ivan, Grundy, John, Hoek, André, Whitehead, Jim. Collaborative Software Engineering. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. URL: https://find.library.spbu.ru/vufind/Record/978-3-642-10294-3 (дата обращения: 31.03.2017)

6. Hazzan, Orit, Dubinsky, Yael. Agile Software Engineering. London: Springer London, 2008. URL: https://find.library.spbu.ru/vufind/Record/978-1-84800-199-2 (дата обращения: 31.03.2017)

7. Davis, Barbara. Mastering software project requirements : a framework for successful planning, development & alignment. Plantation, Florida: J. Ross Publishing, 2013. URL: https://find.library.spbu.ru/vufind/Record/EBC3319548 (дата обращения: 31.03.2017)

8. Chemuturi, Murali. Mastering software quality assurance best practices, tools and techniques for software developers. Fort Lauderdale, Fla.: J. Ross Pub., 2011. URL: https://find.library.spbu.ru/vufind/Record/EBC3319498 (дата обращения: 31.03.2017)

**3.4.2 Список дополнительной литературы**

1. Lewis, James P. Fundamentals of project management. New York: American Management Association, 2007. URL: https://find.library.spbu.ru/vufind/Record/EBC3001873 (дата обращения: 31.03.2017)

2. Qin, Zheng. Zheng, Xiang,.Xing, Jiankuan. Software Architecture, Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, 337pp., URL: https://find.library.spbu.ru/vufind/Record/978-3-540-74343-9 (дата обращения: 31.03.2017)

3. Aurum, Aybüke, Wohlin, Claes. Engineering and Managing Software Requirements. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005. URL: https://find.library.spbu.ru/vufind/Record/978-3-540-28244-0 (дата обращения: 31.03.2017)

4. Evans, Isabel. Achieving software quality through teamwork. Boston: Artech House, 2004. URL: https://find.library.spbu.ru/vufind/Record/EBC227672 (дата обращения: 31.03.2017)

**3.4.3 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_type=8>

**Раздел 4. Разработчики программы**

Литвинов Юрий Викторович, к.т.н., старший преподаватель кафедры системного программирования, y.litvinov@spbu.ru.

По материалам Брыксина Тимофея Александровича, к.т.н.