**Санкт-Петербургский государственный университет**

**Р А Б О Ч А Я П Р О Г Р А М М А**

**УЧЕБНОЙ ДИСЦИПЛИНЫ**

Прикладные технологии программной инженерии

Software Engineering Applied Technology

**Язык(и) обучения**

русский

Трудоемкость в зачетных единицах: 4

Регистрационный номер рабочей программы: 042915

**Раздел 1. Характеристики учебных занятий**

**1.1. Цели и задачи учебных занятий**

Ввести обучающихся в предмет программной инженерии как в будущую профессию разработчиков ПО, показать различные варианты проектов, познакомить с разными видами деятельности при разработке ПО. Дать необходимые знания для организации индивидуального процесса разработки, управления проектом в небольшой команде разработчиков, быстрой интеграции в существующие крупные проекты. В курсе рассматриваются базовые понятия и практики, общие для широких групп проектов – процесс, модели процесса, «гибкие» методологии, анализ требований, проектирование, тестирование, управление конфигурациями, управление проектом, работа с требованиями, разработка документации, переиспользование кода, лицензирование. Весь материал снабжается задачами, которые обучающиеся в качестве заданий выполняют в течение курса.

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

Программа курса предназначена для обучающихся 3 курса. Курс рассчитан на студентов, имеющих базовые навыки программирования на языках Python/Java/С#/С++ и реализовавших в своей жизни самостоятельно хотя бы одно законченное программное приложение. Предполагается, что данные навыки обучающиеся получили на практикуме по программированию на I курсе.

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

По окончании обучения обучающийся должен:  
1. Первичная ориентация в мире промышленного программирования.  
2. Понимание необходимости использования методов программной инженерии.  
3. Понимание отличий программной инженерии от программирования.  
4. Базовые навыки выявления и анализа требований к системе.  
5. Базовые навыки проектирования системы.  
6. Навыки организации процесса тестирования и определения достаточности тестового покрытия системы в зависимости от требований к качеству системы.  
7. Навыки разработки документации и определения необходимого количества документации исходя и требований.  
8. Умение пользоваться базовыми инструментами разработки (IDE, bug tracker, builder, version control, wiki, continuous integration).  
9. Навыки по работе с картами памяти (Mindmaps) для быстрого и компактного изложение имеющихся знаний и неясных вопросов в рассматриваемой области.

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

Аудиторная учебная работа: теоретические занятия в объеме 4 часов в неделю.  
Самостоятельная работа: без участия преподавателя: индивидуальная работа с доступными математическими текстами, а также удовлетворение личных познавательных потребностей.

**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 | 16 | 16 | 2 |  |  |  |  |  | 2 |  |  |  | 72 | |  | 36 |  | 20 | | 4 | |
|  | 2-100 | 10-25 | 2-100 |  |  |  |  |  | 2-100 |  |  |  | 1-1 | |  | 1-1 |  |  | |  | |
| ИТОГО | 16 | 16 | 2 |  |  |  |  |  | 2 |  |  |  | 72 | |  | 36 |  |  | | 4 | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 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** | **Кол-во часов** | |
| 1 | Введение в специальность.   1. Цели и задачи, решаемые методами программной инженерии 2. История программной инженерии как науки. Основные достижения. Цикл: передовой опыт/модели, методы, программные средства/распространение и использование в проектах. Мировое software engineering сообщество. 3. Виды компаний и проектов по разработке и сопровождению ПО. 4. Варианты карьеры (технический рост, переход в административную плоскость, переход в смежную предметную область). 5. Необходимые навыки программиста, в том числе и при карьерном росте. 6. Startups: собственная идея, собственный бизнес и собственная компания. Обзор успешных случаев, рассказ о способах первичной раскрутки. 7. Некоммерческие проекты: научные, гуманитарные, хобби проекты. | лекции | 2 | |
| семинары | 2 | |
| самостоятельная работа по методическим материалам | 9 | |
| 2 | Процесс разработки ПО.   1. Определения процесса разработки ПО. О необходимости разработки и установления процесса до начала проекта. Фазы и виды деятельности процесса. Понятие модели процесса, виды модели процесса: водопадная, спиральная, итеративно-инкрементальная. 2. Артефакты разработки (код, спецификации модели, тесты, документация, процедуры процесса, средства разработки).Понятие рабочего продукта (workproduct).Подробный разбор группы рабочих продуктов под названием «Диплом». Сравнение процесса с проектными планами, а также другими артефактами разработки. 3. Роли в проекте. Дисциплина обязательств как основа рабочей этики. Хирургическая бригада Брукса. Модель команды в MSF.Модель команды в Scrum. Команда и другие stakeholders проекта. 4. Scrum: история основные идеи, структура команды, порядок проведения итерации, отдельные практики (planningpokerи др.), ограничения и предпосылки успешного применения. 5. CMM: история, 5-ти уровневая структура стандарта, понятие ключевой области процесса, краткий обзор каждого из уровней. 6. Обзор open-source модели разработки. | лекции | 2 | |
| семинары | 2 | |
| самостоятельная работа по методическим материалам | 9 | |
| 3 | Конфигурационное управление.   1. Примеры проблем отсутствия порядка в артефактах проекта. Аналогия со складским учетом. Свойства файлов как единиц учета. Определение конфигурационного управления. 2. Практики версионного контроля исходных кодов проекта. Варианты организации структуры версионного хранилища (различные «ветки»). Процедуры доступа к веткам, перемещение исходников из одной ветки в другую. 3. Управление сборками (build management). Сценарии работы сборщика. Регулярная ночная сборка, сборка по запросу, continuous integration. Сборщики и регрессионное тестирование. Примеры простейших сборщиков на основе bat-файлов.Обзор современных средств разработки сборщиков (ant, Maven, CMake, ). 4. Управление версиями продукта. Особенности управления версиями многоверсионного продукта, поставляемого разным заказчикам. Release management. | лекции | 2 | |
| семинары | 2 | |
| самостоятельная работа по методическим материалам | 9 | |
| 4 | Тестирование.   1. Определение тестирования. Парадигмы тестирования – белый и черный ящик. Виды тестирования: модульное, системное, интеграционное, стрессовое, нагрузочное, регрессионное. Другие средства обеспечения качества ПО: верификация, улучшение процесса и др. 2. Варианты организации команды тестировщиков и ее взаимодействия с разработчиками. 3. Bugtracking системы. Варианты реализации жизненного цикла ошибки. 4. Модульное тестирование. Дисциплина и правила написания модульных тестов. Обзор продуктов Nunit/Junit. Самостоятельная разработка пакета модульных тестов. | лекции | 2 | |
| семинары | 2 | |
| самостоятельная работа по методическим материалам | 9 | |
| 5 | Управление проектами.   1. Понятие проекта, основные особенности: нетипичная задача, необходимость спец усилий по фокусировке, команда. Отличие проекта от функциональной модели организации производства. Небольшие «живые» проекты, большие проекты, в которых участвует много компаний. Stakeholders проекта. 2. Рамки проекта. Требования и план проекта. Управление деньгами проекта: маленькие/ большие деньги, свои/чужие. Внешнее оправдание (довольный заказчик, менеджмент, удачный бизнес) и внутреннее совершенство (хорошая и стабильная архитектура, налаженный процесс). Треугольник компромиссов MSF. Понятие рисков. Понятие безнадежных проектов по Йордану. 3. Карьера менеджера softwareпроекта. Основные навыки: умение программировать (в прошлом программист или тестеровщик), умение ладить с командой, с начальством и заказчиком, организационные навыки (организация и удерживание разномасштабных начинаний), умение работать с деньгами, чувствование хода проекта. | лекции | 2 | |
| семинары | 2 | |
| самостоятельная работа по методическим материалам | 9 | |
| 6 | Управление требованиями.   1. Трудности управления требованиями к программным продуктам. Понятие требований как иерархической системы дискретных описаний свойств ПО. Функциональные и нефункциональные требования. Трудности выявления нефункциональных требований. Проблемы недовыявленных требований. 2. Техническое задание проекта: назначение, варианты структуры, примеры. Язык написания технических заданий. Типичные ошибки. Самостоятельная работа по написанию технических заданий. 3. Приемы интервьюирования пользователей/экспертов предметной области для выявления требований: иметь предварительную подготовку, задавать много вопросов, не давать читать толстых документов, пользоваться диаграммами (в частности, UML-диаграммами). Активность и ответственность разработчика, «беззаботность» источника требований. Примеры UML-диаграмм для выявления требований. Использование диаграмм случаев использования. Самостоятельная работа по выявлению требований для разных вариантов проектов. | лекции | 2 | |
| семинары | 2 | |
| самостоятельная работа по методическим материалам | 9 | |
| 7 | Переиспользование кода, защита авторских и смежных прав   1. Авторское право, патенты, торговые марки, лицензирование ПО 2. Переиспользование библиотек с открытым и закрытым кодом. 3. Предоставление кода в открытый доступ. 4. Другие варианты лицензирования. | лекции | 2 | |
| семинары | 2 | |
| самостоятельная работа по методическим материалам | 9 | |
| 8 | Разработка документации.   1. Основные виды документации программного проекта: документы с требованиями, проектные спецификации, руководства пользователя. Самостоятельная разработка этих видов документов для проектов, в которых участвовали студенты. Разбор типовых ошибок. 2. Самодокументирующийся код. 3. Передача знаний в команде. | лекции | 2 | |
| семинары | 2 | |
| самостоятельная работа по методическим материалам | 9 | |
| 9 | Промежуточная аттестация | самостоятельная работа | 36 | |
| консультации | 2 | |
| экзамен | 2 | |
| **Итого** | | | | **144** |

**Раздел 3. Обеспечение учебных занятий**

**3.1. Методическое обеспечение**

**3.1.1 Методические указания по освоению дисциплины**

Освоение дисциплины состоит в разборе основных понятий, принципов и типичных задач по темам курса и решения практических заданий с использованием компьютера. Методические материалы включают в себя следующие типы материалов — учебники, учебные пособия, методические указания для студентов, Интернет-ресурсы, электронные учебные пособия, с опорой на которые проводится аудиторная работа.

**3.1.2 Методическое обеспечение самостоятельной работы**

Самостоятельная работа обучающегося, как вид деятельности, стимулирующий активность, самостоятельность, познавательный интерес с целью поиска необходимой информации, приобретения знаний, использования этих знаний для решения учебных, научных и профессиональных задач, представляет собой важную составляющую учебного процесса, которой отводится не менее половины учебного времени при очной форме обучения. Время, отводимое на самостоятельную работу, должно использоваться студентами для наиболее полного освоения учебной дисциплины. Следовательно, организация эффективной внеаудиторной самостоятельной работы в процессе обучения требует, с одной стороны, создание условий, призванных обеспечить рациональное и планомерное управление учебной деятельностью, протекающей в отсутствие преподавателя, и тщательной подготовки целого ряда учебных пособий, снабженных методическими указаниями, с другой стороны.

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

Аппарат контроля усвоения материала включает в себя задания для самостоятельной работы по вопросам курса, а также итоговый экзамен по изложенному материалу.

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

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

Промежуточная аттестация проводится в виде экзамена в традиционной устной форме.

Экзаменационный билет содержит один вопрос

1. Предварительно экзаменующийся должен ответить на вопросы по основным курса. Предлагается три вопроса, каждый из которых оценивается в 11 баллов. Баллы снижаются за неточности в ответе. При наборе менее 15 баллов ответ считается неудовлетворительным (оценка F). Максимальное количество баллов за ответ --- 33.

2. После этого следует ответ на вопрос с определения основных понятий и теорем рассматриваемого билета. За верные определения и формулировки основных терем рассматриваемой темы начисляется максимально 35 баллов. При наборе менее 15 баллов выставляется оценка «неудовлетворительно» (F).

3. Доказательство основных теорем билета --- 17 баллов.

4. Дополнительные баллы – до 6 баллов.

Критерии оценок

«Отлично» – A --- выше 85 баллов.

«Хорошо» – B --- 81 – 84 балла;

– C – 71 – 80 баллов.

«Удовлетворительно» – D – 61 – 70 баллов;

– E – 50 – 60 баллов.

Обучающийся, набравший строго менее 50 баллов, получает оценку F --- «неудовлетворительно».

Замечание. Оценка «неудовлетворительно» (F) может быть выставлена по итогам ответа на билет на этапе 1 или 2.

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

Оценка обучающимися содержания и качества учебного процесса по дисциплине осуществляется в установленном в СПбГУ порядке.

**3.2. Кадровое обеспечение**

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

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

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

Специальных требований нет.

**3.3. Материально-техническое обеспечение**

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

В аудиториях, где проводятся занятия, необходимо наличие досок для письма мелом. Помещения должны быть светлыми, с достаточным объёмом и притоком свежего воздуха.

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

Наличие компьютера с широкополостным выходом в интернет, проектора, экрана.

**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**

**3.4.1 Список обязательной литературы**

1. Мартин, Роберт. Чистый код: создание, анализ и рефакторинг. Библиотека программиста. Издательский дом «Питер», 2013.

**3.4.2 Список дополнительной литературы**

1. Beck, Kent, and Erich Gamma. Extreme programming explained: embrace change. addison-wesley professional, 2000.  
2. Brooks Jr, Frederick P. "The mythical man-month (anniversary ed.)." (1995).  
3. Brown, William H., et al. AntiPatterns: refactoring software, architectures, and projects in crisis. John Wiley & Sons, Inc., 1998.  
4. Coad, Peter, Edward Yourdon, and Peter Coad. Object-oriented analysis. Vol. 2. Englewood Cliffs, NJ: Yourdon press, 1991.  
5. DeMarco, Tom, and Tim Lister. Peopleware: productive projects and teams. Addison-Wesley, 2013.  
6. Fowler, Martin. Refactoring: improving the design of existing code. Addison-Wesley Professional, 2018.  
7. Lessig, Lawrence. Code: And other laws of cyberspace. ReadHowYouWant. com, 2009.  
8. Martin, Robert C. Clean code: a handbook of agile software craftsmanship. Pearson Education, 2009.  
9. McConnell, Steve. Code complete. Pearson Education, 2004.  
10. Myers, Glenford J., Corey Sandler, and Tom Badgett. The art of software testing. John Wiley & Sons, 2011.  
11. Pree, Wolfgang, and Erich Gamma. Design patterns for object-oriented software development. Vol. 183. Reading, MA: Addison-wesley, 1995.  
12. Schwaber, Ken, and Mike Beedle. Agile software development with Scrum. Vol. 1. Upper Saddle River: Prentice Hall, 2002.  
13. Shlaer, Sally, and Stephen J. Mellor. Object lifecycles: modeling the world in states. Yourdon Press, 1992.  
14. Westby, Emma Jane Hogbin. Git for teams: a user-centered approach to creating efficient workflows in Git. " O'Reilly Media, Inc.", 2015.  
15. Yourdon, Edward. "Modern structured analysis." Edward Yourdon Englewood: Prentice-Hall International 4 (1989).  
16. Yourdon, Edward. Death march. Pearson Education, 2003.

**3.4.3 Перечень иных информационных источников**

Любые библиотечные книги по теме курса, журнальная литература, интернет.

**Раздел 4. Разработчики программы**

Ловягин Юрий Никитич, ст. преп. каф. информатики, к. ф.-м. н., доцент. y.lovyagin@spbu.ru, +7 905 255 02 94.