**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: 3

Регистрационный номер рабочей программы: 002190

Санкт-Петербург

2020

**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 пользоваться 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)**

В результате освоения дисциплины обучающиеся должны:

* обладать представлением об общей структуре хода проекта по разработке ПО;
* уметь идентифицировать основные роли в команде и их области ответственности;
* обладать навыками декомпозиции и оценки задач, построения графика работ и оценки бюджета проекта;
* понимать важность работы с требованиями, знать основные виды деятельности при сборе и управлении требованиями;
* уметь обоснованно выбрать методологию разработки проекта и парадигму программирования в зависимости от его параметров и требований;
* иметь представление о подходах и инструментах проектирования макетов пользовательского интерфейса приложения;
* осознавать важность архитектуры проекта, понимать критерии оценки качества архитектуры и методов её проектирования;
* понимать особенности работы в команде, уметь использовать средства контроля версий для коллективной разработки ПО;
* уметь обнаруживать, классифицировать и отслеживать дефекты в ПО;  
  понимать основные характеристики качества ПО и методы по его достижению.

Курс способствует формированию следующих компетенций:

* 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 профессиональной деятельности;
* 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.

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

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

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

**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 | 29 |  |  |  |  | 1 |  |  | 2 |  |  |  | 50 |  | 26 |  | 3 | 3 |
|  | 2-100 |  |  |  |  | 1-30 |  |  | 1-30 |  |  |  | 1-1 |  | 1-1 |  |  |  |
| ИТОГО | 29 |  |  |  |  | 1 |  |  | 2 |  |  |  | 50 |  | 26 |  |  | 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 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 | 4 |
| II. | Жизненный цикл ПО, методологии и процессные фреймворки | лекции | 4 |
| по методическим материалам | 7 |
| III. | Требования и проектирование GUI | лекции | 4 |
| по методическим материалам | 7 |
| IV. | Роль и функции менеджера проекта | лекции | 7 |
| контрольные работы | 1 |
| по методическим материалам | 13 |
| V. | Групповая разработка | лекции | 8 |
| по методическим материалам | 14 |
| VI. | Качество программного обеспечения | лекции | 4 |
| по методическим материалам | 5 |
| VII. | 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.
3. Балансирование равновесия проекта: основные подходы и практики. Отслеживание прогресса хода проекта и реагирование на отклонения от плана. Организация команды: типы команд, роли участников.
4. Расчёт бюджета проекта, варианты финансового взаимодействия с заказчиком. Инструментальная поддержка для управления проектами.

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.
2. Групповая разработка. Управление версиями: централизованные и распределённые системы. Удачные модели коллективной разработки ПО в распределённых системах контроля версий.
3. Динамика развития программных проектов, "законы Лемана". Legacy-системы. Сопровождение и поддержка, эволюция систем, реинжиниринг.
4. Рефакторинг: что, когда и зачем. Понятие и основные code smells. Примеры рефакторингов.
5. Антипаттерны управления релизами. Основные принципы Continuous Delivery. Конфигурационное управление, Continuous Integration. Полезные практики при внедрении Continuous Delivery. Структура и шаги построения конвейера развёртывания. Модель зрелости процесса управления релизами.

Раздел 6: Качество программного обеспечения

1. Понятие дефекта. Методы и системы управления дефектами в программных проектах. Характеристики дефектов.
2. Качество ПО: характеристики качества, стандарты. Оценка качества ПО: основные метрики. Методы обеспечения качества ПО.

**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 контрольную работу и результаты устного зачёта.

Контрольная работа выполняется в виде командного задания, где прямо на занятии обучающимся предлагается разделиться на команды по 2-3 человека, выполнить задание и сдать его на проверку с помощью системы поддержки обучения. Преподаватель может переформировать команды по своему усмотрению. Общение внутри команды и использование любых источников разрешено. На выполнение задания выделяется один академический час. В случае пропуска занятия с текущим контролем задание предлагается выполнить индивидуально на зачёте, на что выделяется дополнительно два академических часа в процессе экзамена.

Поскольку дисциплина имеет практическую направленность и ориентирована, в том числе, на успешное применение полученных обучающимися навыков и знаний на практике при дальнейшей работе по специальности, подготовка ответов на зачёте проводится без ограничения доступа к материалам дисциплины, литературе и иным информационным источникам.

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, на подготовку ответа на которые даётся не менее 40 минут, а также на несколько дополнительных вопросов, без подготовки. Количество и содержание дополнительных вопросов – на усмотрение преподавателя, принимающего зачёт.

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

Решения контрольной 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 10 (очень хороший ответ). Результирующий процент освоения дисциплины получается следующим образом:

1. Оценки за ответы на два основных вопроса усредняются.
2. Оценки за ответы на дополнительные вопросы усредняются.
3. Полученные оценки складываются, к ним прибавляется оценка за контрольную работу.
4. 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 профессиональной деятельности;
* 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. Программирование и разработка коммерческих продуктов
2. Особенности разработки ПО от других инженерных областей
3. Компетенции профессионального программиста
4. Типы требований к программным системам
5. Основные характеристики требований
6. Основные действия, выполняемые при разработке требований
7. Основные действия, выполняемые при управлении требованиями
8. Основные документы, создаваемые при работе с требованиями
9. Понятие жизненного цикла ПО
10. Водопадная модель жизненного цикла
11. Итеративная и спиральная модель жизненного цикла
12. Rational Unified Process
13. Agile подход к разработке
14. eXtreme Programming: общий подход, достоинства и недостатки
15. eXtreme Programming: практики “Короткий цикл разработки”
16. eXtreme Programming: практики “Непрерывность процесса”
17. eXtreme Programming: практики “Понимание, разделяемое всеми”
18. Scrum: назначение, особенности, общий подход
19. Scrum: роли в команде
20. Scrum: артефакты, используемые в планировании и разработке
21. Scrum: основные процессные мероприятия
22. Понятие User experience
23. User-centered design
24. Создание персонажей при проектировании интерфейсов
25. Создание сценариев при проектировании интерфейсов
26. Activity-centred design
27. Data-driven design
28. Сторителлинг и раскадровки как инструмент проектирования UI
29. Макеты и дизайн-макеты как инструмент проектирования UI
30. Варианты исследования UX продукта
31. Usability-тестирование
32. Функции менеджера проекта
33. Матрица ответственности и план коммуникаций
34. Основные действия по управлению рисками
35. Декомпозиция проекта
36. Построение графика работ: матрица зависимостей, сетевой график
37. Построение графика работ: сетевой график, диаграмма Гантта
38. Типичные ошибки при оценке проектов
39. Треугольник равновесия проекта
40. Приёмы балансирования равновесия на уровне проекта
41. Приёмы балансирования равновесия на уровне бизнес-целей
42. Отслеживание прогресса проекта
43. Организация коммуникаций внутри команды
44. Управление изменениями
45. Основные действия по завершению проекта
46. Позитивная экосистема команды: базовые правила и сплочённость команды
47. Позитивная экосистема команды: умение слушать
48. Позитивная экосистема команды: умение проводить совещания
49. Совместное решение задач: анализ задач и варианты принятия решений
50. Совместное решение задач: разрешение конфликтов
51. Совместное решение задач: непрерывное обучение
52. Особенности формирования команды
53. Особенности командной разработки ПО
54. Системы контроля версий: назначение и виды
55. Модель командной разработки на основе веток
56. Модель командной разработки на основе единой ветки
57. Понятие качества ПО
58. Характеристики качества ПО: функциональность и надёжность
59. Характеристики качества ПО: удобство использования и эффективность
60. Характеристики качества ПО: сопровождаемость и переносимость
61. Метрики качества ПО: метрики Холстеда и цикломатическая сложность
62. Метрики качества ПО: метрики Чидамбера и Кемерера
63. Метрики объектно-ориентированного тестирования
64. Модель зрелости компаний CMMI
65. Понятие и виды тестирования ПО
66. Понятие ошибки, сбоя и дефекта ПО. Характеристики дефектов
67. Отладка ПО: воспроизведение дефектов
68. Отладка ПО: локализация ошибки
69. Отладка ПО: исправление ошибки
70. Понятие защитного программирования
71. Общие принципы защитного программирования
72. Стратегии реакции на ошибку в программе
73. Использование барьеров для изоляции ошибок в ПО
74. Понятие и мотивация для осуществления рефакторинга
75. Основные “дурные запахи” в коде
76. Рефакторинги “Выделение метода” и “Перемещение метода”
77. Рефакторинги “Выделение класса” и “Выделение подкласса”
78. Рефакторинги “Сокрытие делегирование” и “Введение внешнего метода”
79. Рефакторинги “Самоинкапсуляция поля” и “Введение Null-объекта”
80. Рефакторинги “Замена кода типа подклассами” и “Замена условного оператора полиморфизмом”
81. Рефакторинги “Замена конструктора фабричным методом” и “Замена наследования делегированием”
82. Проблемы при проведении рефакторинга. Причины не проводить рефакторинг
83. Понятие Continuous Delivery
84. Антипаттерны управления релизами
85. Преимущества частых автоматизированных релизов
86. Принципы непрерывного развёртывания ПО
87. Понятие Continuous Integration
88. Модель зрелости процесса управления релизами

*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 даётся два билета и задаётся несколько дополнительных вопросов по курсу. Ответ на каждый вопрос билета и на дополнительные вопросы оценивается по шкале от 0 (нет ответа) до 10 (очень хороший ответ), далее оценка усредняется. Результирующий процент освоения дисциплины получается следующим образом:

1. Оценки за ответы на два основных вопроса усредняются.
2. Оценки за ответы на дополнительные вопросы усредняются.

Полученные оценки складываются. 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 контрольной работы:***

Поделиться на команды по 2-3 человека. Нарисовать диаграмму Гантта для проекта учебной практики одного из сокомандников.

*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).

**3.1.5 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**

Не предусмотрены.

**Раздел 4. Разработчики программы**

Брыксин Тимофей Александрович, к.ф.-м.н., доцент, t.bryksin@spbu.ru