*Годы обучения по образовательной программе 2022-2024*

Петрозаводский государственный университет

Институт математики и информационных технологий

Кафедра информатики и математического обеспечения

УТВЕРЖДАЮ

Проректор по учебной работе

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ К.Г. Тарасов

«\_\_\_\_» \_\_\_\_\_\_\_\_ 2022 г.

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**

**РУКОВОДСТВО ПРОЦЕССОМ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

Направление подготовки магистратуры

01.04.02 Прикладная математика и информатика

Магистерская программа

«Интеллектуальные Интернет-технологии»

Форма обучения очная

Петрозаводск

2022

Рабочая программа дисциплины разработана в соответствии с ФГОС ВО, утвержденным приказом Министерства науки и высшего образования Российской Федерации от 10.01.2018 г. № 13 (с изменениями от 08.02.2021 № 82, от 26.11.2020 № 1456), и учебным планом по направлению подготовки магистратуры 01.04.02 Прикладная математика и информатика (профиль «Интеллектуальные Интернет-технологии»).

Разработчик:

Богоявленский Юрий Анатольевич, заведующий кафедрой, кафедра информатики и математического обеспечения; заведующий лабораторией, научно-исследовательская лаборатория "Информационно-телекоммуникационные системы", кандидат технических наук, доцент

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(подпись)*

Эксперт:

Печников Андрей Анатольевич, руководитель лаборатории телекоммуни­каци­онных систем, Институт прикладных математических исследований КарНЦ РАН, к.ф.-м.н., д.т.н, доцент

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(подпись)*

Рабочая программа дисциплины рассмотрена и одобрена на заседании кафедры Информатики и математического обеспечения

Протокол № 6 от « 14 » июня 2022 г.

Заведующий кафедрой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ю. А. Богоявленский, к.т.н., доцент

 *(подпись)*

СОГЛАСОВАНО:

Рабочая программа дисциплины рассмотрена и утверждена на заседании учебно-методической комиссии института математики и информационных технологий

Протокол № \_\_\_\_\_\_\_\_\_ от «\_\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2022 г.

Директор института \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Н. Ю. Светова, к.ф.-м.н., доцент

 *(подпись)*

Начальник методического отдела

учебно-методического управления ПетрГУ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ И.В. Маханькова

**1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения основной профессиональной образовательной программы (ОПОП) магистратуры**

Компетенции обучающегося, формируемые в результате освоения дисциплины:

|  |  |  |
| --- | --- | --- |
| **Код компетенции. Этап формирования компетенции** | **Формулировка компетенции** | **Планируемые результаты обучения** (индикаторы достижения компетенции) |
| ПК-1основной | Способность профессионально организовать и выполнять разработку, отладку, модификацию, поддержку и исследование программного обеспечения, эффективно применяя математические методы и информационно-коммуникационные технологии | ПК-1.1. Знает современные методики разработки программного обеспечения для интернета вещей и киберфизических систем, включая гибкую методологию разработки и математические модели оценки сложности разработки и планирования разработки в ИТ-проектах.ПК-1.2. Умеет разрабатывать и исследовать различные типы программного обеспечения, включая компьютерные сети, распределенные системы, вычислительные платформы и комплексы, библиотеки и пакеты программ, веб-сервисы, мобильные приложения, встроенное ПО, интеллектуальные информационные системы, а также другие продукты системного и прикладного программного обеспечения в ИТ-проектах; ПК-1.3.Владеет навыками обоснованного использования различных языков программирования и систем программирования при разработке интернет-технологий и интернет-приложений в ИТ-проектах, включая процедурные, объектно-ориентированные, функциональные, логические парадигмы программирования и средства генерации программ. |
| ПК-3начальныйосновнойитоговый | Способность управления разработкой программного обеспечения в области искусственного интеллекта, интернета вещей и киберфизических систем | ПК-3.1. Знает модели разработки и нормативно-методическую документацию по существующим стандартам и рекомендациям программной инженерии, принципы управления ИТ-проектами, методы автоматизации управления разработкой.ПК-3.2. Умеет создавать условия для разработки программного обеспечения, оценивать трудоемкость работ с учетом возможностей коллектива ИТ-проекта, планировать графики работ, проводить аналитическую работу, принимать решения по использованию готового программного обеспечения и доводить эти решения до исполнителей.ПК-3.3. Владеет навыками управления командой, деления поставленной задачи на подзадачи, проработка решений с руководителем и архитектором проекта в области искусственного интеллекта, интернета вещей и киберфизических систем. |

В процессе изучения дисциплины обучающийся будет систематизировать компетенции разработчика ПО, необходимые для соответствия квалификациям системного инженера, системного архитектора, технического директора, руководителя проекта.

В результате освоения дисциплины обучающийся должен:

**Знать:**

Модели процессов жизненного цикла программного продукта и их теоретические основы. Методы планирования программного проекта: структуры плана управления и графика работ. Управление риском. Управление персоналом. Управление документацией и конфигурацией, Оценки программного проекта. Процессы обеспечения качества программного продукта. Программные инструменты управления программным проектом.

Модели и методы анализа и процессы работы при определении требований к программной системе, методы синтеза при проектировании и кодировании программной системы, модели композиции приложений и оценок программного продукта.

Системные и прикладные программные инструменты построения моделей, инструменты кодирования и сборки проектов, системы контроля версий, системы управления проектами.

**Уметь:**

Выбирать модель процесса программного проекта. Оценить параметры программного проекта. Строить план управления проектом и график работ. Подбирать персонал команды проекта. Организовать работу с требованиями к программному продукту. Выбрать программный инструмент управления проектом.

Применять методы анализа при определении требований к программной системе и методы синтеза при проектировании и кодировании программной системы.

Путем анализа выбирать и, при необходимости, комбинировать модели для процессов жизненного цикла программного продукта, оценок программного продукта, анализа требований, композиции приложений.

Свободно применять существующие и быстро осваивать новые системные и прикладные программные инструменты построения моделей, инструменты кодирования и сборки проектов, системы контроля версий, системы управления проектами.

**Владеть:**

Понятием жизненного цикла программного продукта. Комплексом знаний по управлению программным проектом. Методами планирования и управления программным проектом.

Методами управления требованиями пользователя и детальными требованиями. Методами проектирования архитектуры и ее декомпозиции на модули.

Навыками применения моделей для процессов жизненного цикла программного продукта, оценок программного продукта, анализа требований, композиции приложений.

Системными и прикладными программными инструментами построения моделей, инструментами кодирования и сборки проектов, системами контроля версий, системами управления проектами.

**2. Место дисциплины в структуре ОПОП магистратуры и язык преподавания**

Дисциплина «Руководство процессом разработки программного обеспечения» входит в часть, формируемую участниками образовательных отношений учебного плана основной образовательной программы магистратуры по данному направлению подготовки и является обязательной для изучения дисциплиной.

Согласно учебному плану дисциплина проводится в 3 семестре.

Изучение дисциплины опирается на знания, умения и навыки, приобретенные при освоении образовательной программы предыдущего уровня, а также при изучении дисциплин: «Информатика», «Введение в архитектуру ЭВМ», «Операционные оболочки», «Компьютерные сети», «Технология производства программного обеспечения», «Web технологии», «Базы данных». «Формальные языки и компиляторы», «Системное программирование», «Операционные системы», «Сетевые инструменты распределенной разработки ПО», «Архитектура современных ЭВМ».

Язык преподавания – русский

**3. Виды учебной работы и тематическое содержание**

Общая трудоемкость дисциплины составляет 3 зачетных единиц или 108 академических часов.

**3.1 Виды учебной работы**

|  |  |
| --- | --- |
| Виды учебной работы | Объем в академических часах |
| **Общая трудоемкость дисциплины по учебному плану** | 108 |
| В том числе: |  |
| **Контактная работа (работа во взаимодействии с преподавателем). Всего** | 30 |
| В том числе: |  |
| Лекции (Л) | 15 |
| Практические занятия (Пр) | 15 |
| Лабораторные занятия (Лаб) | - |
| Вид промежуточной аттестации | зачет |
| **Самостоятельная работа обучающихся (СР) (всего)** | 78 |
| В том числе: |
| Самостоятельное изучение разделов дисциплины, подготовка к занятиям — 60 |
| Подготовка к промежуточной аттестации — 18 |

**3.2. Краткое содержание дисциплины по разделам и видам учебной работы**

|  |  |  |  |
| --- | --- | --- | --- |
| № |  |  |  |
| Всего | Лекции | Практическиезанятия | Лабораторные занятия | Самостоятельная работа обучающихся |
| Семестр № 3 |
| 1 | Модели процессов разработки программного обеспечения. | 18 | 2 | 4 | 0 | 12 | Доклад с презентацией |
| 2 | Руководство программным проектом. | 18 | 4 | 2 | 0 | 12 | Доклад с презентацией |
| 3 | Требования к продукту. | 20 | 4 | 4 | 0 | 12 | Доклад с презентацией |
| 4 | Проектирование программных систем. | 17 | 3 | 3 | 0 | 12 | Доклад с презентацией |
| 5 | Обеспечение качества. | 16 | 2 | 2 | 0 | 12 | Доклад с презентацией |
| Вид промежуточной аттестации в семестре |
|  | Зачет | 18 |  |  |  | 18 |  |
| **Итого:** | 108 | 15 | 15 | 0 | 78 |  |

**3.3. Содержание аудиторных занятий**

**Содержание лекционных занятий**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № раздела | № лекции | Основное содержание | Количество часов | В т.ч. с использованием ДОТ (\*) |
| Семестр № 3 |
| 1 | 1.1 | Организационно-вводная часть. Методы, средства и процессы программной инженерии. Модели, классификация процессов - основные, вспомогательные и организационные процессы. Структура модели процесса: деятельность, задача, действие. Пять видов основной деятельности. Восемь видов защитной деятельности. Различия в организации процессов.Модель «классический жизненный цикл»: этапы, достоинства, недостатки. Макетирование: цель, шаги, достоинства, недостатки. Стратегии разработки. Инкрементная, спиральная и компонентная модели. Тяжеловесные и облегченные процессы. Обзор agile моделей. Процессы Scrum. Модель качества процессов разработки. Модель CMM. | 2 |  |
| 2 | 2.1 | Основные понятия руководства проектом. Начало, измерения, процесс оценки, анализ риска, планирование, трассировка и контроль. Планирование программного проекта: структуры плана управления и графика работ программного проекта. Управление риском: идентификация, анализ риска, ранжирование, планирование управления, разрешение и наблюдение риска. Управление персоналом: подбор, взаимодействия в команде, ее состав. | 2 |  |
| 2 | 2.2 | Метрики сложности проекта LOC и FP. Оценка на их основе при планировании. Конструктивная модель стоимости. Модель композиции приложения. Модели раннего этапа проектирования и этапа постархитектуры. Предварительная оценка проекта. Анализ чувствительности. Сценарий понижения зарплаты стр. | 2 |  |
| 3 | 3.1 | Виды требований. Формирование, анализ, аттестация требований. Характеристики детального требования. Спецификация и управление требованиями. | 2 |  |
| 3 | 3.2 | Визуальные модели требований. Цели визуального моделирования. Диаграммы: потоков данных и рабочих потоков (swimlane). Карты диалоговых окон. Таблицы и деревья решений. Таблицы событий и реакций. Диаграмма сущность-связь. | 2 |  |
| 4 | 4.1 | Процесс синтеза программной системы. Проектирование архитектуры. Структурирование. Шаблоны архитектуры. Шаблон MVC. Шаблоны архитектуры: с хранилищем данных, клиент-сервер, многоуровневая, канала и фильтра. Шаблоны управления: вызов-возврат, менеджера, широковещательное, по прерываниям. Проектирование интерфейсов | 2 |  |
| 4 | 4.2 | Декомпозиция подсистем на модули. Модульность: информационная закрытость, связность. Виды связности: функциональная, информационная, коммуникативная, процедурная, временная, логическая, связность, по совпадению. Нахождение связности. Структурное проектирование. Типы информационных потоков. Проектирование для потоков данных типа «преобразование» и «запрос». | 1 |  |
| 5 | 5.1 | Методы обеспечения качества программного продукта. Стандарты и модели качества. Характеристики качества: результативность;производительность; совместимость; удобство использования (юзабилити); надёжность; защищённость; сопровождаемость; переносимость (мобильность). Тестирование программного обеспечения. | 2 |  |
|  |  | **Итого:** | **15** |  |

**Содержание практических (или семинарских) занятий**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № раздела | № занятия | Основное содержание | Количество часов | В т.ч. с использованием ДОТ (\*) |
| Семестр № 3 |
| 1 | 1.1 | Распределение тем для докладов с презентациями. | 1 |  |
| 1 | 1.2-1.4 | Доклады с презентациями по распределенным темам. | 3 |  |
| 2 | 2.1-2.2 | Доклады с презентациями по распределенным темам. | 2 |  |
| 3 | 3.1-3.4 | Доклады с презентациями по распределенным темам. | 4 |  |
| 4 | 4.1-4.3 | Доклады с презентациями по распределенным темам. | 3 |  |
| 5 | 5.1-5.2 | Доклады с презентациями по распределенным темам. | 2 |  |
|  |  | **Итого:** | **15** |  |

**3.4. Организация самостоятельной работы обучающегося**

|  |  |  |  |
| --- | --- | --- | --- |
| № раздела | Задания для самостоятельной работы | Количество часов | В т.ч. с использованием ДОТ (\*) |
| Семестр № 3 |
| 1 | Изучение литературы. Подготовка докладов и презентаций по разделам 1-5 | 60 |  |
|  | Подготовка к зачету | 18 |  |
| **Итого** | **78** |  |

**4. Образовательные технологии по дисциплине**

Образовательные технологии основываются на смешанной модели обучения, содержащей пассивную, и интерактивную формы организации познавательной деятельности.

Пассивная форма обучения представлена в виде проведения лекционных занятий. Интерактивная форма реализуется на практических занятиях, когда докладчик представляет презентацию доклада по заданной теме, а обучающиеся и лектор оценивают качество и полноту доклада, задают вопросы, дают комментарии, делятся собственным опытом. Важной составляющей при этом является подготовка обучающихся в процессе самостоятельной работы.

**5. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации обучающихся по дисциплине**

5.1. Текущий контроль осуществляется преподавателем дисциплины при проведении занятий в форме оценки докладов с презентациями по заданным темам.

**Оценочное средство. Доклад с презентацией**

Доклад с презентацией, являющийся продуктом самостоятельной работы обучающегося, представляющий собой публичное выступление с изложением результатов полученных при изучении материалов дисциплины.

Примерные темы докладов

1. Модель «классический жизненный цикл»: этапы, достоинства, недостатки.

 Макетирование: цель, шаги, достоинства, недостатки.

2. Стратегии разработки. Инкрементная, спиральная и компонентная модели

3. Тяжеловесные и облеrченные процессы. Обзор agile моделей.

4. Модель качества процессов разработки. Модель CMM.

5. Процессы Scrum.

6. Основные понятия руководства проектом. Начало, измерения, процесс оценки,

 анализ риска, планирование, трассировка и контроль.

7. Планирование программного проекта: структуры плана управления и

 графика работ программного проекта.

8. Управление риском: идентификация, анализ риска, ранжирование,
 планирование управления, разрешение и наблюдение риска.

 9. Управление персоналом: подбор, взаимодействия в команде, ее состав.

10. Управление документацией: стандарты, полнота, согласованность.

Управление конфигурацией: идентификация объектов, контроль верcий и
 изменений.

11. Планирование программного проекта. LOC и FP метрики..

12. Оценка на основе LOC и FP метрик при планировании. Конструктивная
 модель стоимости. Модель композиции приложения.

13. Модели paннero этапа проектирования и этапа постархитектуры.

14. Предварительная оценка проекта. Анализ чувствительности. Сценарий

 понижения зарплаты.

15. Виды требований.

16. Формирование, анализ, аттестация требований.

17. Рекомендуемые приемы формулирования требований.

18. Каркас процесса создания требований.

19. Определение бизнес требованияй.

20. Документ о концепции и границах.

21. Способы представления границ проекта.

22. Характеристики детального требования.

23. Спецификация и управление требованиями.

24. Обзор и цели методов визуального моделирования трнбований.

25. Выбор правильного визуального представления.

26. Диаграммы: потоков данных, рабочих потоков (swimlane).

27. Диаграмма переходов состояний и таблица состояний.

28. Карты диалоговых окон.

29 Таблицы и деревья решений. Таблицы событий и реакций.

30. Моделирование ланных. Диаграмма сущность-связь.

31. Моделирование данных. Словарь данных.

32. Процесс синтеза программной системы. Проектирование архитектуры.

структурирование. Шаблоны архитектуры. Шаблон MVC.

33. Шаблоны архитектуры: с хранилищем данных, клиент-сервер.

34. Шаблоны многоуровневая архитектуры и канала и фильтра. \

35. Шаблоны управления: вызов-возврат, менеджера, широковещательное,

по прерываниям

36. Проектирование интерфейсов.

37. Декомпозиция подсистем на модули (без разделения понятий). Модульность.

 Информационная закрытость, связность.

38. Виды связности: функциональная, информационная, коммуникативная,
 процедурная, временная, логическая связность, по совпадению. Определение
 связности.

39. Структурное проектирование. Типы информационных потоков. Проек­тирова­ ние для потоков данных типа «преобразование» и «запрос».

40. Обеспечение качества. Определение, цели, факторы качества.

41. Деятельность по обеспечению качества: технические проверки и аудиты.

42. Деятельность по обеспечению качества: инспектирование, верификация,
 валидация.

43. Атрибуты качества. План обеспечения качества.

44. Разработка требований к ответственным системам..

45. Анализ инструментов управления требованиями.

46. Основы тестирования.

47. Процесс тестирования.

48. Тестирование в течение жизненного цикла разработки ПО.

49. Уровни тестирования.

50. Типы тестирования.

51. Инструменты управления проектом: Microsoft project.

52. Инструменты управления проектом: Jira.

53. Инструменты управления проектом: Redmine.

54. Инструменты управления проектом: Jenkins.

55. Наиболее важные свойства инструмента для работы с требованиями Requality.

 Критерии оценивания доклада с презентацией.

 Оценка **«зачтено»** выставляется обучающемуся, если он представил на слайдах все основные разделы рассматриваемой темы, свободно излагает ее материал, ориентируется во взаимосвязях его разделов, дает полноценные ответы на вопросы. Структура презентации должна соответствовать структуре докладываемого материала. Презентация должна удовлетворять следующим техническим требованиям: наличие слайдов титульного листа и содержания, отсутствие на слайдах длинных фрагментов текста, использование всего поля слайда, отсутствие шрифтов мелкого размера.

 Оценка **«не зачтено»** выставляется обучающемуся, если он не освоил тему доклада, не структурировал презентацию, не может пояснить понятия, представленные на слайдах, не способен ответить на вопросы, грубо нарушил технические требования к презентации.

5.2. Промежуточная аттестация проводится в виде: зачета:

Вопросы к зачету

1. Определение программной инженерии.
2. Определение жизненного цикла ПО.
3. Официальная классификации процессов программной инженерии.
4. Виды основной деятельности для базиса процессов.
5. Специфика видов защитной деятельности.
6. Классический жизненный цикл, этапы, достоинства, недостатки.
7. Макетирование (прототипирование).
8. Стратегии разработки ПО.
9. Инкрементная модели.
10. Спиральная модель.
11. Компонентно-ориентированная модель.
12. Тяжеловесные и облегченные процессы.
13. Модель Scrum.
14. Модель СММ.
15. Модель процесса разработки в знакомой вам фирме.
16. Меры и метрики при оценке проекта, примеры.
17. Процесс оценки программного проекта.
18. Трассировка и контроль.
19. Последовательность действий при планировании проекта.
20. Разделы плана программного проекта.
21. Графика работ программного проекта.
22. Источники риска, действия по управлению риском.
23. Риск: анализ, ранжирование, планирование управления, разрешение, наблюдение.
24. Подбор членов команды для программ ого проекта.
25. Ответственность лидера команды.
26. Цели имеет управление документацией.
27. Обеспечение полноты и согласованности документации.
28. Определение и элементы конфигурация.
29. Задачи и план управления конфигурацией.
30. Объекты конфигурации, возможные отношения между ними.
31. Архитектура системы контроля версий.
32. Выполнение оценки программного проекта, метрики.
33. Функциональные указатели, их информационные характеристики и вычисление.
34. Шаги оценки проекта на основе LOC- и FP-метрик.
35. Конструктивная модель стоимости и ее применение.
36. Назначение модели композиции, ее базовые оценки.
37. 16.Модели раннего этапа проектирования, Основное уравнеие.
38. Модели этапа пост-архигектуры, их отличие от моделей раннего этапа проектирования.
39. Анализ чувствительности программного проекта.
40. Четыре вида требований к ПО.
41. Сравнение процессов формирования требований и анализа требований.
42. Различия требований заказчика и разработчика.
43. Различия детальных требований и требований разработчика.
44. Желаемые характеристики детальных требований.
45. Главные идеи организации детальных требований.
46. Задачи, решаемые в ходе аттестации требований и методы их рещения.
47. Цели визуального моделирования требований. Диаграммы потоков данных, их иерархия.
48. Диаграммы рабочих потоков (swimlane).
49. Диаграммы вариантов использования.
50. Диаграммы переходов состояний.
51. Таблицы состояний, карты диалоговых окон.
52. Таблицы и деревья решений.
53. Таблицы событий и реакций.
54. Диаграмма сущность-связь.
55. Какие задачи решают методы анализа, ориентированные на структуры данных.
56. Цель и этапы синтеза программной системы
57. Модели данных, архитектуры подсистем.
58. Шаблоны (паттерны) архитектуры. Шаблоны MVC и с хранилищем данных.
59. Шаблоны клиент-сервер, мноrоуровневой архитектура, канала и фильтра.
60. Шаблоны управления: вызов-возврат, менеджера, широковещательное, по прерываниям.
61. Цель декомпозиции подсистем на модули.
62. Информационная закрытость и связность модулей.
63. Функциональная, информационная, коммуникативная связности.
64. Связности процедурная, временная, логическая, по совпадению.
65. Метод структурного проектирования.
66. Типы информационных потоков.
67. Проектирование для потоков данных типа «преобразование».
68. Проектирование для потоков данных типа «запрос»
69. Метод Джексона. Пример.
70. Определение качества ПО.
71. Факторы и метрики качества ПО
72. Внешние и внутренние метрики качества ПО.
73. Функциональности, надежность, практичность ПО.
74. Эффективность, сопровождаемость переносимость ПО.
75. Содержание деятельности но обеспечению качества.
76. Инспектирование и техническая проверка.
77. Процесс верификации.
78. Процесс валидации.
79. План обеспечения качества.
80. Программные системы управления проектами.

Критерии оценивания

**«Зачтено»** выставляется обучающемуся, если он при собеседовании по вопросам достаточно полно ответил на оба вопроса в билете, т. е. показал достаточно прочные знания основных положений учебной дисциплины, умение самостоятельно решать конкретные лабораторные задания, предусмотренные рабочей программой, ориентироваться в рекомендованной справочной литературе, умеет правильно оценить полученные результаты.

**«Не зачтено»** выставляется обучающемуся, если при ответе выявились существенные пробелы в знаниях основных положений учебной дисциплины.

Подробно средства оценивания для проведения промежуточной аттестации обучающихся приведены в Фонде оценочных средств по данной дисциплине.

**6. Методические рекомендации обучающимся по дисциплине, в том числе для самостоятельной работы**

Обучающимся рекомендуется запланировать в распорядке дня регулярную работу по подготовке докладов с презентациями по распределенным им темам. Презентации рекомендуется оформлять с помощью класса Beamer макропакета LaTeX. При этом следует не размещать на слайдах длинные фрагменты текста, полностью использовать все поле слайда, не использовать плохо видимые элементы — мелкие шрифты, не контрастные цвета и т. п. Рекомендуется размещать на слайде от 10 до 17 строк, используя максимально возможный при этом размер шрифта.

При изучении литературы следует уделить особое внимание взаимосвязи материалов разделов.

**7. Методические рекомендации преподавателям по дисциплине**

Особое внимание при подготовке необходимо уделить первому практическому занятию, где обучающиеся знакомятся с планом дисциплины, получают темы для подготовки докладов и знакомятся с требованиями по ее освоению. На первом занятии также необходимо ознакомить обучающихся со списком литературы и пояснить, где искать материал для докладов. Также следует прокомментировать студентам список вопросов к экзамену и ответить на их вопросы.

**8. Учебно-методическое и информационное обеспечение дисциплины**

**8.1. Основная литература:**

* 1. Ехлаков, Ю.П. Управление программными проектами : учебник / Ю.П. Ехлаков ; Министерство образования и науки Российской Федерации, Томский Государственный Университет Систем Управления и Радиоэлектроники (ТУСУР). - Томск : Томский государственный университет систем управления и радиоэлектроники, 2015. - 217 с. : схем., табл. - Библиогр. в кн. - ISBN 978-5-86889-723-8 ; То же [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book&id=480634 (23.03.2022.
	2. Зубкова, Т.М. Технология разработки программного обеспечения / Т.М. Зубкова ; Министерство образования и науки Российской Федерации, Федеральное государственное бюджетное образовательное учреждение высшего образования «Оренбургский государственный университет», Кафедра программного обеспечения вычислительной техники и автоматизированных систем. – Оренбург : ОГУ, 2017. – 469 с. : ил. – Режим доступа: по подписке. – URL: http://biblioclub.ru/index.php?page=book&id=485553 (дата обращения: 26.03.2022. – Библиогр.: с. 454-459. – ISBN 978-5-7410-1785-2. – Текст : электронный.
	3. Кознов Д. В. Введение в программную инженерию. [Электронный ресурс]. - URL:<https://www.intuit.ru/studies/courses/497/353/info> (23.03.2022)
	4. Маглинец, Ю.А. Анализ требований к автоматизированным информационным системам : учебное пособие / Ю.А. Маглинец. - Москва : Интернет-Университет Информационных Технологий, 2008. - 200 с. : ил., табл., схем. - (Основы информационных технологий). - ISBN 978-5-94774-865-9 ; То же [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book&id=233195 (24.03.2022).

**8.2. Дополнительная литература:**

1. Васючкова Т. С., Держо М. А., Иванчева, Н. А. Пухначева Т. П. Управление проектами с использованием Microsoft Project / - 2-е изд., испр. - Москва : Национальный Открытый Университет «ИНТУИТ», 2016. - 148 с. : ил. - Библиогр. в кн. ; То же [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book&id=429881 (22.03.2022).

2. ГОСТ Р ИСО/МЭК 12207-2010 Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств. – М.: Стандартинформ, 2011. – 106 с. [Электронный ресурс]: — URL: http://protect.gost.ru/v.aspx?control=8&baseC=-1&page=0&month=-1&year=-1&search=&RegNum=1&DocOnPageCount=15&id=16909 (23.03.2022.

3. Методология Agile. - [Электронный ресурс] — URL: https://worksection.com/blog/agile.html (15.03.2022)

4. Никитин И. А., Цулая М. Т. Аналитические шаблоны проектирования приложений. - [Электронный ресурс]. - URL: https://www.intuit.ru/studies/courses/3611/853/info (19.03.2022)

**8.3. Программное обеспечение и Интернет-ресурсы:**

Используются стандартные средства показа pdf файлов.

1. Пример шаблона технического задания (ТЗ) на сайт [Электронный ресурс]. - URL: http://www.rugost.com/index.php?option=com\_content&view=article&id=182:43&catid=33&Itemid=85 (28.03.2022)

2. Atlassian Лучший продукт для agile-команд [Электронный ресурс]. - URL: https://ru.atlassian.com/software/jira (26.03.2022)

3. Jenkins User Documentation [Электронный ресурс]. - URL: https://jenkins.io/doc/ (28.03.2022)

4. Redmine guide [Электронный ресурс]. - URL: https://www.redmine.org/projects/redmine/wiki/Guide (27.03.2022)

5. Scrum Guides.- [Электронный ресурс] — URL:https://www.scrumguides.org/ (22.03.2022)

**9. Материально-техническое обеспечение дисциплины**

Материально-техническая база ПетрГУ обеспечивает проведение всех видов дисциплинарной и междисциплинарной подготовки, практической и научно-исследовательской работы обучающихся, предусмотренных учебным планом и соответствует действующим санитарным и противопожарным правилам и нормам.

Минимально-необходимый перечень для информационно-технического и материально-технического обеспечения дисциплины:

* аудитория для проведения лекционных и практических занятий, оснащенная рабочими местами для обучающихся и преподавателя, доской, мультимедийным оборудованием;
* библиотека с читальным залом и залом для самостоятельной работы обучающегося, оснащенная компьютером с выходом в Интернет, книжный фонд которой составляет специализированная научная, учебная и методическая литература, журналы (в печатном или электронном виде).

Дата « 08 » апреля 2022 г.