Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«Национальный исследовательский ядерный университет «МИФИ»

ИНСТИТУТ ИНТЕЛЛЕКТУАЛЬНЫХ КИБЕРНЕТИЧЕСКИХ СИСТЕМ

КАФЕДРА КИБЕРНЕТИКИ

ОДОБРЕНО  
  
протокол № 20 / 03  
  
от « 31 » мая 2020 г.

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ**

по дисциплине

УПРАВЛЕНИЕ ЖИЗНЕННЫМ ЦИКЛОМ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ (SOFTWARE LIFECYCLE MANAGEMENT)

|  |  |
| --- | --- |
| Направление подготовки (специальность) | 09.04.04 Программная инженерия |
| Квалификация (степень) выпускника | Магистр |
| Форма обучения | очная |

**ПАСПОРТ**

**фонда оценочных средств**

**по дисциплине «Управление жизненным циклом программного обеспечения (Software Lifecycle Management)»**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Семестр** | **Интерактив** | **Трудоемкость, кред.** | **Общий объем курса, час.** | **Лекции, час.** | **Практич. занятия, час.** | **Лаборат. работы, час.** | **СРС, час.** | **КСР, час.** | **Форма(ы) контроля, экз./зач./КР/КП** |
| 2 |  | 2 | 72 | 0 | 30 | 0 | 42 | 0 | З |
| ИТОГО | 0 | 2 | 72 | 0 | 30 | 0 | 42 | 0 |  |

**1.1. Область применения**

Фонд оценочных средств (ФОС) *–* является неотъемлемой частью учебно-методического комплекса учебной дисциплины «Управление жизненным циклом программного обеспечения (Software Lifecycle Management)» и предназначен для контроля и оценки образовательных достижений обучающихся, освоивших программу данной дисциплины.

**1.2. Цели и задачи фонда оценочных средств**

Целью Фонда оценочных средств является установление соответствия уровня подготовки обучающихся требованиям ОС НИЯУ МИФИ.

Для достижения поставленной цели Фондом оценочных средств по дисциплине «Управление жизненным циклом программного обеспечения (Software Lifecycle Management)» решаются следующие задачи:

– контроль и управление процессом приобретения обучающимися знаний, умений и навыков предусмотренных в рамках данного курса;

– контроль и оценка степени освоения общекультурных, общепрофессиональных и профессиональных компетенций предусмотренных в рамках данного курса;

– обеспечение соответствия результатов обучения задачам будущей профессиональной деятельности через совершенствование традиционных и внедрение инновационных методов обучения в образовательный процесс в рамках данного курса.

**1.3. Модели контролируемых компетенций**

В результате освоения дисциплины у выпускника формируются следующие компетенции:

УК-2 – Способен управлять проектом на всех этапах его жизненного цикла

УК-3 – Способен организовывать и руководить работой команды, вырабатывая командную стратегию для достижения поставленной цели

ОПК-8 – Способен осуществлять эффективное управление разработкой программных средств и проектов

ПК-6 – Способен к пониманию существующих подходов к верификации моделей программного обеспечения

ПК-13 – Владеет навыками программной реализации распределенных информационных систем

ПК-20 – Владеет навыками организации промышленного тестирования создаваемого программного обеспечения

ПК-2.1 – Владеет знаниями основ технологии управления проектами

ПК-2.2 – Способен определять последовательность выполнения работ

ПК-2.3 – Способен оценивать качество проводимых работ

4. Структура и содержание учебной дисциплины

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **№ п.п** | **Наименование раздела учебной дисциплины** | **Недели** | **Лекции, час.** | **Практ. занятия/ семинары, час.** | **Лабораторные работы, час.** | **Обязат. текущий контроль (форма\*, неделя)** | **Аттестация раздела (форма\*, неделя)** | **Максимальный балл за раздел\*\*** | **Компетенции по разделам, проверяемые при текущем и рубежном контроле** | **Компетенции, проверяемые на зач. /экз.** |
|  | 2 семестр |  |  |  |  |  |  |  |  |  |
| 1 | Модели, методологии и средства управления разработкой программного обеспечения | 1-6 | 0 | 12 | 0 | БДЗ-5 | КИ-6 | 20 | УК-2, УК-3, ОПК-8, ПК-6, ПК-13, ПК-20, ПК-2.1, ПК-2.2, ПК-2.3 | УК-2, УК-3, ОПК-8, ПК-6, ПК-13, ПК-20, ПК-2.1, ПК-2.2, ПК-2.3 |
| 2 | Управление анализом требований и проектированием программного обеспечения | 7-13 | 0 | 14 | 0 | БДЗ-10 | КИ-6 | 20 | УК-2, УК-3, ОПК-8, ПК-6, ПК-13, ПК-20, ПК-2.1, ПК-2.2, ПК-2.3 | УК-2, УК-3, ОПК-8, ПК-6, ПК-13, ПК-20, ПК-2.1, ПК-2.2, ПК-2.3 |
| 3 | Управление реализацией и сопровождением программного обеспечения | 14-15 | 0 | 4 | 0 | БДЗ-16,к.р-16 | КИ-6 | 20 | УК-2, УК-3, ОПК-8, ПК-6, ПК-13, ПК-20, ПК-2.1, ПК-2.2, ПК-2.3 | УК-2, УК-3, ОПК-8, ПК-6, ПК-13, ПК-20, ПК-2.1, ПК-2.2, ПК-2.3 |
|  | *Итого за 2 семестр* |  | 16 | 16 | 0 |  |  | 60 |  |  |
|  | **Контрольные мероприятия после 2 семестра** |  |  |  |  |  | З | 40 | УК-2, УК-3, ОПК-8, ПК-6, ПК-13, ПК-20, ПК-2.1, ПК-2.2, ПК-2.3 | УК-2, УК-3, ОПК-8, ПК-6, ПК-13, ПК-20, ПК-2.1, ПК-2.2, ПК-2.3 |

\* – сокращенное наименование формы контроля

\*\* – сумма максимальных баллов должна быть равна 100 за семестр, включая зачет и (или) экзамен

Сокращение наименований форм текущего контроля и аттестации разделов:

КИ Контроль по итогам

КАЛЕНДАРНЫЙ ПЛАН

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Недели** | **Темы занятий / Содержание** | **Лек., час.** | **Пр./сем., час.** | **Лаб., час.** |
|  | *2 Семестр* | 0 | 30 | 0 |
| **1-6** | **Модели, методологии и средства управления разработкой программного обеспечения** | 0 | 12 | 0 |
| 1 - 2 | **Вводная лекция. Жизненный цикл разработки.** Понятие программной системы. Процессы проектирования и конструирования программного обеспечения. Архитектуры программных систем. История развития. Цели и задачи учебной дисциплины. Применяемые методы и средства. Проблемы учебной дисциплины. Основные разделы курса. Структура учебной программы дисциплины (контрольные мероприятия в семестре). | Всего аудиторных часов | | |
|  | 4 |  |
| Онлайн | | |
|  |  |  |
| 3 - 6 | **Модели жизненного цикла. Методологии разработки программных систем.** Понятие методологии. Методология Microsoft Solution Framework. Подход Rational Unified Process. Процессы, роли, артефакты. Активности, потоки работ. Гибкие методологии (Agile, SCRUM, eXtreme Programming). Сходства и различия между методологией разработки и моделью жизненного цикла программной системы. Преимущества и недостатки методологий. | Всего аудиторных часов | | |
|  | 8 |  |
| Онлайн | | |
|  |  |  |
| **7-13** | **Управление анализом требований и проектированием программного обеспечения** | 0 | 14 | 0 |
| 7 - 13 | **Управление анализом требований и проектированием программного обеспечения** Управление анализом требований и проектированием программного обеспечения | Всего аудиторных часов | | |
|  | 14 |  |
| Онлайн | | |
|  |  |  |
| **14-15** | **Управление реализацией и сопровождением программного обеспечения** | 0 | 4 | 0 |
| 14 - 15 | **Управление реализацией и сопровождением программного обеспечения** Управление реализацией и сопровождением программного обеспечения | Всего аудиторных часов | | |
|  | 4 |  |
| Онлайн | | |
|  |  |  |

В качестве оценочного средства используется 100 бальная семестровая система, учитывающая посещаемость занятий, активность (выполнение домашних занятий), выполнение тематических домашних заданий по каждому разделу, контрольно-тестовая работа по каждому разделу. Каждый раздел проходит аттестацию.

ПЗ – посещаемость семинарских занятий (еженедельно) не менее 80% +2 балла

не менее 50% +1 балл

менее 50% 0 баллов

ТДЗ – выполнения тематического ДЗ (по каждому разделу)

Выполнено не менее 90% +10 баллов

Выполнено от 80-до 89% +8 балла

Выполнено от 70-до 79% +6 балла

Выполнено от 60-до 69% +4 балла

Выполнено от 40-до 59% +2 балл

Менее 39% 0 баллов

КТР - контрольно-тестовая работа (продолжительность – 1 а/час

(проводится в аудитории) Выполнено не менее 90% +8 баллов

Выполнено от 70-до 89% +6 баллов

Выполнено от 40-до 69% +4 балла

Менее 39% 0 баллов

АМ – аттестация раздела Раздел аттестуется, если набрано не менее 60%

Зачет (40 баллов). На зачет выносятся вопросы, относящиеся ко всем разделам. Зачет проводится в письменном виде по индивидуальному билету. Каждый билет содержит 10 заданий. Два из них - теоретические, которые выбираются из списка вопросов к зачету. Остальные задания связаны с проверкой теоретических и практических знаний по всем разделам дисциплины. Письменные ответы студента регистрируются на специальных бланках. Студент обязательно отмечает на этих бланках символом "+" те вопросы и задачи из билета, на которые даны полные о развернутые ответы. Отмечает символом "?", если ответ не полон или решение задачи не выполнено до конца. Символом "-" отмечаются те вопросы (задачи), которые не нашли своего отражения в ответах. Каждый вопрос оценивается по следующей схеме:

• 4 балла (полный развернутый ответ на теоретический вопрос или полное и обоснованное решение практической задачи);

• 2 балла (ответ на теоретический вопрос не полон, имеются отдельные неточности в определениях и свойствах, получены частичные результаты решения практической задачи);

• 0 баллов (ответы на теоретический вопрос отсутствуют, обоснование оперирует ложными понятиями либо полностью отсутствует и т.п.).

**Вопросы к зачету**

1. Понятие программной системы.
2. Архитектуры программных систем.
3. Понятие жизненного цикла.
4. Этапы жизненного цикла.
5. Анализ и спецификация требований.
6. Предварительное и детальное проектирование.
7. Реализация и тестирование.
8. Интеграция.
9. Сопровождение.
10. Экономика жизненного цикла.
11. Модель Build-and-Fix.
12. Водопадная модель.
13. Быстрое прототипирование.
14. Инкрементная модель.
15. Модель синхронизации и стабилизации.
16. Спиральная модель.
17. Объектно-ориентированная модель.
18. Преимущества и недостатки моделей.
19. Методология Microsoft Solution Framework.
20. Подход Rational Unified Process.
21. Гибкие методологии (Agile, SCRUM, eXtreme Programming).
22. Сходства и различия между методологией разработки и моделью жизненного цикла программной системы.
23. Преимущества и недостатки методологий.

Автор:

|  |  |  |
| --- | --- | --- |
|  | Зыков Сергей Викторович к.т.н. доцент | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) |