

Рабочая программа составлена на основании федерального государственного образовательного стандарта высшего образования по направлению (специальности): 09.04.04 Программная инженерия

ФГОС введен в действие приказом №1406 от 30.10.2014 г. , дата утверждения: 28.11.2014 г.

Место дисциплины в структуре учебного плана: Блок 1, базовая

Рабочая программа разработана на основе компетентностной модели выпускника по направлению (специальности): 09.04.04 Программная инженерия

Рабочая программа обсуждена на заседании кафедры

ВТ, протокол заседания кафедры №6 от 20.06.2017

Утверждена на совете факультета автоматики и вычислительной техники, протокол № 6 от 21.06.2017

Программу разработал:

доцент, к.т.н. Романов Е. Л.

Заведующий кафедрой:

доцент, к.т.н. Якименко А. А.

Ответственный за образовательную программу:

доцент Якименко А. А.

1. Внешние требования

Таблица 1.1

Компетенция ФГОС: ОК.5 использованием на практике умений и навыков в организации исследовательских и проектных работ, в управлении коллективом; <i>в части следующих результатов обучения:</i>
31. знать особенности организации и управления программным проектом в различных методологиях
Компетенция ФГОС: ОК.6 способность проявлять инициативу, в том числе в ситуациях риска, брать на себя всю полноту ответственности; <i>в части следующих результатов обучения:</i>
31. знать принципы оценки и управления рисками программного проекта, методы управления и организации коллектива разработчиков
Компетенция ФГОС: ОК.8 способность к профессиональной эксплуатации современного оборудования и приборов (в соответствии с целями магистерской программы; <i>в части следующих результатов обучения:</i>
31. основных принципов и процессов сопровождения программного обеспечения
Компетенция ФГОС: ПК.6 пониманием существующих подходов к верификации моделей программного обеспечения; <i>в части следующих результатов обучения:</i>
31. методы верификации моделей проектирования и программного обеспечения
Компетенция НГТУ: ПК.22.В способность управлять средой функционирования объектов профессиональной деятельности; <i>в части следующих результатов обучения:</i>
32. инструменты и методы управления требованиями
38. методы и приемы формализации задач
Компетенция НГТУ: ПК.23.В способность к управлению процессами жизненного цикла программного обеспечения; <i>в части следующих результатов обучения:</i>
32. этапы, технологические процессы, артефакты унифицированного процесса разработки ПО, содержание свода знаний о программной инженерии SWEBOK, стандартов по процессам жизненного цикла ПО
35. методологии разработки программного обеспечения

2. Требования НГТУ к результатам освоения дисциплины

Таблица 2.1

Результаты изучения дисциплины по уровням освоения (иметь представление, знать, уметь, владеть)	Формы организации занятий
<i>Методология программной инженерии</i>	
ОК.5.31 знать особенности организации и управления программным проектом в различных методологиях	
1. знать особенности организации и управления программным проектом в различных методологиях	Лекции; Самостоятельная работа
ОК.6.31 знать принципы оценки и управления рисками программного проекта, методы управления и организации коллектива разработчиков	
2. знать принципы оценки и управления рисками программного проекта, методы управления и организации коллектива разработчиков	Лекции; Самостоятельная работа
ПК.6.31 методы верификации моделей проектирования и программного обеспечения	
3. методы верификации программного обеспечения	Лекции; Самостоятельная работа
ПК.22.В.32 инструменты и методы управления требованиями	
4. инструменты и методы управления требованиями	Лекции; Самостоятельная работа
ПК.22.В.38 методы и приемы формализации задач	
5. методы и приемы формализации задач	Лекции; Самостоятельная работа

ПК.23.В.32 этапы, технологические процессы, артефакты унифицированного процесса разработки ПО, содержание свода знаний о программной инженерии SWEBOOK, стандартов по процессам жизненного цикла ПО	
6.этапы, технологические процессы, артефакты унифицированного процесса разработки ПО, содержание свода знаний о программной инженерии SWEBOOK, стандартов по процессам жизненного цикла ПО	Лекции; Самостоятельная работа
ПК.23.В.35 методологии разработки программного обеспечения	
7.методологии разработки программного обеспечения	Лекции; Самостоятельная работа
8.умеет строить объектные модели информационных систем	Лекции; Самостоятельная работа
9.умеет строить модели потоков данных информационных систем	Лекции; Самостоятельная работа
11.знает структурный и объектно-ориентированный подходы к проектированию информационных систем	Лекции; Самостоятельная работа
ОК.8.31 основных принципов и процессов сопровождения программного обеспечения	
12.основных принципов и процессов сопровождения программного обеспечения	Лекции; Самостоятельная работа
ОК.5.31 знать особенности организации и управления программным проектом в различных методологиях	
13.знать особенности организации и управления программным проектом в различных методологиях	Лекции; Самостоятельная работа

3. Содержание и структура учебной дисциплины

Таблица 3.1

Темы лекций	Активные формы, час.	Часы	Ссылки на результаты обучения	Учебная деятельность
Семестр: 1				
Дидактическая единица: UML как средство поддержки процесса проектирования				

1. Структура UML, статическая и динамическая составляющие модели	0	2	11, 5	Структура UML, статическая и динамическая составляющие модели. Составляющие элементы, механизмы, архитектура. Составные элементы: сущности, отношения, диаграммы. Виды сущностей: структурные сущности - класс, интерфейс, кооперация, прецедент, активный класс, компонент, узел; поведенческие сущности - взаимодействия, деятельности, автоматы; группирующая сущность - пакет, аннотационная сущность - примечание. Виды отношений: зависимость, ассоциация, агрегация, композиция, включение, обобщение, реализация. Отношения. Связи - отношения между объектами. Направленность связи, Сообщения. Ассоциации - отношения между классами. Свойства ассоциации: имя, кратность, навигация, атрибуты. Рефлексивные ассоциации, деревья и сети. Классы атрибутов ассоциаций (классы-ассоциации). Зависимости. Зависимости использования "use", "call", "parameter", "send" и "instantiate". Зависимости абстракции. Зависимости доступа. Виды диаграмм. Спецификации. Дополнения. Принятые деления: классификатор-экземпляр, интерфейс-реализация. Стереотипы
--	---	---	-------	---

2. Диаграммы UML. Структуры и поведение	0	2	11, 5, 8	Диаграммы взаимодействия. Линия жизни, активности, сообщения, их виды. Комбинированные фрагменты и операторы. Включения, параметры, шлюзы. Продолжения. Их аналоги в программировании. Коммуникационные диаграммы. Диаграммы деятельности. Технология сетей Петри. Параллелизм. Поток управления, узел действия, ребро, узел управления, объектный узел, буферизация и в объектном узле. Объектные узлы - параметры, состояния объектных узлов. Контакты. Прерывающие ребра. Контакты исключений. Потоки объектов. Их аналоги в программировании (фаза реализации). Диаграммы состояний. Конечные автоматы. Состояния. Переходы. События. Сигналы. Составные состояния. Подавтоматы. Предыстория.
Дидактическая единица: Предмет программной инженерии, жизненный цикл проекта, методологии проектирования и модели процессов ЖЦ				
3. Жизненный цикл ПО	1	1	12, 5, 6, 7	Свод знаний по программной инженерии SWEBOOK Жизненный цикл (ЖЦ) программного продукта и проекта. "Легкие" и "тяжелые" модели процессов разработки ПО. Обзор: "как получится", ГОСТы, RUP, Agile, XP. Модели жизненного цикла. Каскадная, итерационная и спиральная модели. Организационная и технологическая составляющие жизненного цикла. Роль UML в формализации технологической составляющей. Моделирование, виды моделей. UML как средство описания различных аспектов "мысленной" модели проектируемой системы.

4. Унифицированный процесс UP	1	1	12, 13, 6, 7	Унифицированный процесс UP. Фазы жизненного цикла: начало, уточнение, построение, внедрение. Содержание и результаты фаз. Итерация и ее рабочие потоки: требования, анализ, проектирование, реализация, тестирование, их содержание. Фаза начала. Видение проекта, глоссарий. Рабочий поток анализа предметной области. Диаграммы классов предметной области.
5. Бизнес-аналитика и системная аналитика	2	2	11, 3, 6, 7, 8, 9	Моделирование предметной области и бизнес-процессов. Бизнес-процессы, диаграммы потоков данных. Язык описания бизнес-процессов BPMN. Различие между бизнес-аналитикой и системной аналитикой. Модель предметной области
6. Управление требованиями	2	2	4, 6, 7	Требования. Место рабочего потока определения требований. Функциональные и не-функциональные, перечень. Атрибуты требований: приоритеты, состояние, трудоемкость, стабильность, риск, полезность. Способы извлечения и фильтрации требований. Актеры и роли. Прецеденты. Выявление актеров и прецедентов: бизнес-модель предметной области, модель требований, глоссарий проекта. Диаграммы прецедентов, детализация прецедентов, спецификации прецедентов. Отображение требований на прецеденты.
7. Архитектура и проектирование ПС	2	3	11, 8	Архитектура как система представлений: представление классов, процессов, реализации, развертывания и прецедентов (требований). Классы анализа. Виды классов: граница, управление, сущность. Выявление классов анализа. Диаграммы устойчивости. Архитектурные аспекты технологического процесса проектирования (по SWEBOOK)

8. Проектирование графического интерфейса	0	1		Проектирование графического интерфейса (GUI). Основные аспекты. Архитектурное проектирование, основанное на GUI. Факторы, характеризующие GUI: производительность, человеческие ошибки, обучение, субъективное восприятие, запоминание, поиск, визуализация, навигация
9. Примеры проектирования ПС	0	2	11, 5	Примеры проектирования. Многослойная структура клиент-серверных приложений. Слои, интерфейсы, протоколы. Тонкие и толстые клиенты. Примеры проектирования и основного набора моделей и документов: система выдачи ключей аудиторий. . система подсчета рейтинга успеваемости. Распределение компонентов. Клиент-серверные приложения: тонкий и толстый клиент. Обработка ошибок и исключительных ситуаций, управление отказоустойчивостью и надежностью. Разделение представления и сущности: концепция модель-вид-контроллер.
10. Экстремальное и гибкое программирование/проектирование	0	2	1, 13, 2	Экстремальное и гибкое программирование. Манифест экстремального программирования (XP). Гибкие (agile) технологии. SCRUM. Agile UP, ICONIX. SCRUM как технологический фреймворк.. Терминология. Спринт. Митинг. Собственник проекта. Команда. SCRUM-мастер. Беклог проекта и спринта. Планирование спринта. Диаграмма сгорания. Оценка трудоемкости. Покер-планирование

4. Самостоятельная работа обучающегося

№	Виды самостоятельной работы	Ссылки на результаты обучения	Часы на выполнение	Часы на консультации
Семестр: 1				

1	Курсовая работа	1, 11, 12, 13, 2, 3, 4, 5, 6, 7, 8, 9	46	6
В курсовой работе необходимо провести анализ выпускной квалификационной работы бакалавра или другого программного проекта, в котором студент принимал участие, на предмет наличия в нем элементов, определенных сводом знаний о ПИ (SWEBOK). В пояснительной записке обязательны следующие разделы: 1. Содержательное описание анализируемой разработки, процесса ее проектирования и результатов. 2. Описание частей процесса выполнения проекта, соответствующих содержанию раздела SWEBOK (например, разработка программных требований). 3. Анализ п.2. с точки зрения влияния каждого раздела на успех (неудачу) проекта, соответствия содержанию раздела SWEBOK. 4. Выводы об уровне исполнения проекта с точки зрения методологии ПИ, анализ сильных и слабых сторон. 5. Рекомендации по совершенствованию учебного процесса и преподаванию дисциплин, связанных с ПИ, на основе анализа проекта. : Романов Е. Л. Программная инженерия [Электронный ресурс] : электронный учебно-методический комплекс / Е. Л. Романов ; Новосиб. гос. техн. ун-т. - Новосибирск, [2015]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000220170. - Загл. с экрана.				
2	Подготовка к занятиям	1, 11, 12, 13, 2, 3, 4, 5, 6, 7	23	0
Подготовка к лекции в форме дискуссии по предлагаемому и найденному материалу: Романов Е. Л. Программная инженерия [Электронный ресурс] : электронный учебно-методический комплекс / Е. Л. Романов ; Новосиб. гос. техн. ун-т. - Новосибирск, [2015]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000220170. - Загл. с экрана.				
3	Подготовка к аттестации	1, 11, 12, 13, 2, 3, 4, 5, 6, 7	10	3
: Романов Е. Л. Программная инженерия [Электронный ресурс] : электронный учебно-методический комплекс / Е. Л. Романов ; Новосиб. гос. техн. ун-т. - Новосибирск, [2015]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000220170. - Загл. с экрана. Пустовалова Н. В. Программная инженерия (метрическая теория программ) [Электронный ресурс] : электронный учебно-методический комплекс [для студентов направления 080800 Прикладная информатика] / Н. В. Пустовалова, Г. И. Кайгородцев ; Новосиб. гос. техн. ун-т. - Новосибирск, [2014]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000208496. - Загл. с экрана.				

5. Технология обучения

Для организации и контроля самостоятельной работы обучающихся, а также проведения консультаций применяются информационно-коммуникационные технологии (табл. 5.1).

Таблица 5.1

Деятельность	Информационно-коммуникационные технологии
Информирование	Социальные сети: YandexDisk vk.com/cprog_cs Skype ИДО - дисц. "Программная инженерия"; Чат: Skype
Консультирование	e-mail: romanov@corp.nstu.ru; Чат: Skype
Размещение учебных материалов	Социальные сети: YandexDisk vk.com/cprog_cs Skype ИДО - дисц. "Программная инженерия"

6. Правила аттестации обучающихся по учебной дисциплине

Для аттестации обучающихся по дисциплине используется балльно-рейтинговая система (БРС), позволяющая выставять оценки по традиционной шкале и 15-уровневой ECTS. Краткая информация о БРС приведена в табл. 6.1.

Таблица 6.1

Оцениваемые виды деятельности обучающихся	Мин. балл	Максимальный балл
Семестр: 1		
<i>Лабораторная:</i>	24	48
<i>РГЗ:</i>	6	12
<i>Экзамен:</i>	0	40
Контролирующие материалы приводятся в "Романов Е. Л. Программная инженерия [Электронный ресурс] : электронный учебно-методический комплекс / Е. Л. Романов ; Новосиб. гос. техн. ун-т. - Новосибирск, [2015]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vils000220170 . - Загл. с экрана."		

В таблице 6.2 представлено соответствие форм контроля заявляемым требованиям к результатам освоения дисциплины.

Таблица 6.2

Коды компетенций ФГОС	Результаты обучения	Формы контроля	
		Защита КП/КР	Экзамен
ОК.5	з1. знать особенности организации и управления программным проектом в различных методологиях	+	+
ОК.6	з1. знать принципы оценки и управления рисками программного проекта, методы управления и организации коллектива разработчиков		+
ОК.8	з1. основных принципов и процессов сопровождения программного обеспечения		+
ПК.6	з1. методы верификации моделей проектирования и программного обеспечения		+
	ПК.22.В з2. инструменты и методы управления требованиями		+
	ПК.22.В з8. методы и приемы формализации задач		+
	ПК.23.В з2. этапы, технологические процессы, артефакты унифицированного процесса разработки ПО, содержание свода знаний о программной инженерии SWEBOK, стандартов по процессам жизненного цикла ПО		+
	ПК.23.В з5. методологии разработки программного обеспечения	+	+

Фонд оценочных средств по дисциплине представлен в приложении № 1 к рабочей программе.

7. Литература

Основная литература

1. Липаев В. В. Программная инженерия. Методологические основы : [учебник для вузов по направлению "Бизнес-информатика" (080700)] / В. В. Липаев ; Гос. ун-т - высш. шк. экономики. - М., 2006. - 605, [1] с.
2. Беркун С. Искусство управления IT- проектами / Скотт Беркун ; [пер. с англ. Н. Вильчинский]. - М. [и др.], 2010. - 431 с. : ил.
3. Буч Г. Язык UML : руководство пользователя / Г. Буч, Д. Рамбо, И. Якобсон ; [пер. с англ. Н. Мухина]. - М., 2007. - 493 с. : ил.

Дополнительная литература

1. Рейнвотер Д. Х. Как пасти котов : наставление для программистов, руководящих другими программистами / Дж. Ханк Рейнвотер ; [пер. с англ. Ю. Гороховский]. - СПб., 2008. - 255 с. : ил.

2. Якобсон А. Унифицированный процесс разработки программного обеспечения / А. Якобсон, Г. Буч, Дж. Рамбо ; [пер. с англ. В. Горбункова]. - СПб. [и др.], 2002. - 492 с. : ил.
3. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на C++ / Гради Буч ; пер. с англ. под ред. И. Романовского и Ф. Андреева. - М., 1998. - 558 с. : ил. - Тит. л. парал. рус., англ..
4. Орлов С. А. Технологии разработки программного обеспечения. Разработка сложных программных систем : Учебное пособие для вузов / С. А. Орлов. - СПб., 2003. - 473 с. : ил.
5. Вязовик Н. А. Программирование на Java. Курс лекций / Н. А. Вязовик ; Интернет ун-т информ. технологий. - М., 2003. - 586 с. : ил., схемы

Интернет-ресурсы

1. Архипенков С. Лекции по управлению программными проектами [Электронный ресурс] / С. Архипенков. – Москва, 2009. — 128 с. – Режим доступа : http://www.arkhipenkov.ru/resources/sw_project_management.pdf. – Загл. с экрана.
2. ЭБС НГТУ : <http://elibrary.nstu.ru/>
3. ЭБС «Издательство Лань» : <https://e.lanbook.com/>
4. ЭБС IPRbooks : <http://www.iprbookshop.ru/>
5. ЭБС "Znaniyum.com" : <http://znaniyum.com/>
6. :

8. Методическое и программное обеспечение

8.1 Методическое обеспечение

1. Пустовалова Н. В. Программная инженерия (метрическая теория программ) [Электронный ресурс] : электронный учебно-методический комплекс [для студентов направления 080800 Прикладная информатика] / Н. В. Пустовалова, Г. И. Кайгородцев ; Новосиб. гос. техн. ун-т. - Новосибирск, [2014]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000208496. - Загл. с экрана.
2. Романов Е. Л. Программная инженерия [Электронный ресурс] : электронный учебно-методический комплекс / Е. Л. Романов ; Новосиб. гос. техн. ун-т. - Новосибирск, [2015]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000220170. - Загл. с экрана.

8.2 Специализированное программное обеспечение

1 StarUML

2 COSOMO® П.2000.4

9. Материально-техническое обеспечение

Презентационное оборудование

№	Наименование	Назначение
1	Презентационное оборудование (мультимедиа-проектор, экран, компьютер для управления)	Проведение лекций

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Новосибирский государственный технический университет»

Кафедра вычислительной техники

“УТВЕРЖДАЮ”
ДЕКАН АВТФ
к.т.н., доцент И.Л. Рева
“ ” _____ Г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

УЧЕБНОЙ ДИСЦИПЛИНЫ

Методология программной инженерии

Образовательная программа: 09.04.04 Программная инженерия, магистерская программа:
Разработка программного обеспечения информационных систем

1. Обобщенная структура фонда оценочных средств учебной дисциплины

Обобщенная структура фонда оценочных средств по дисциплине **Методология программной инженерии** приведена в Таблице.

Таблица

Формируемые компетенции	Показатели сформированности компетенций (знания, умения, навыки)	Темы	Этапы оценки компетенций	
			Мероприятия текущего контроля (курсовой проект, РГЗ(Р) и др.)	Промежуточная аттестация (экзамен, зачет)
ОК.5 использованием на практике умений и навыков в организации исследовательских и проектных работ, в управлении коллективом	з1. знать особенности организации и управления программным проектом в различных методологиях	Унифицированный процесс UP Экстремальное и гибкое программирование/проектирование	Курсовая работа	Экзамен, вопросы 4,13,14
ОК.6 способность проявлять инициативу, в том числе в ситуациях риска, брать на себя всю полноту ответственности	з1. знать принципы оценки и управления рисками программного проекта, методы управления и организации коллектива разработчиков	Экстремальное и гибкое программирование/проектирование		Экзамен, вопросы 13,14
ОК.8 способность к профессиональной эксплуатации современного оборудования и приборов (в соответствии с целями магистерской программы	з1. основных принципов и процессов сопровождения программного обеспечения	Жизненный цикл ПО Унифицированный процесс UP		Экзамен, вопросы 3,4
ПК.22.В способность управлять средой функционирования объектов профессиональной деятельности	з2. инструменты и методы управления требованиями	Управление требованиями		Экзамен, вопросы 6
ПК.22.В	з8. методы и приемы формализации задач	Диаграммы UML. Структуры и поведение Примеры проектирования ПС Структура UML, статическая и динамическая составляющие модели		Экзамен, вопросы 9-12
ПК.23.В способность к управлению процессами жизненного цикла программного обеспечения	з2. этапы, технологические процессы, артефакты унифицированного процесса разработки ПО, содержание свода знаний о программной инженерии SWEBOOK,	Бизнес-аналитика и системная аналитика Управление требованиями		Экзамен, вопросы 5,6

	стандартов по процессам жизненного цикла ПО			
ПК.23.В	35. методологии разработки программного обеспечения	Архитектура и проектирование ПС Бизнес-аналитика и системная аналитика Диаграммы UML. Структуры и поведение Жизненный цикл ПО Структура UML, статическая и динамическая составляющие модели Унифицированный процесс UP Управление требованиями	Курсовая работа	Экзамен, вопросы 5-12
ПК.6/НИ пониманием существующих подходов к верификации моделей программного обеспечения	31. методы верификации моделей проектирования и программного обеспечения	Бизнес-аналитика и системная аналитика		Экзамен, вопросы 5,6

2. Методика оценки этапов формирования компетенций в рамках дисциплины.

Промежуточная аттестация по дисциплине проводится в 1 семестре - в форме экзамена, который направлен на оценку сформированности компетенций ОК.5, ОК.6, ОК.8, ПК.22.В, ПК.23.В, ПК.6/НИ.

Кроме того, сформированность компетенций проверяется при проведении мероприятий текущего контроля, указанных в таблице раздела 1.

В 1 семестре обязательным этапом текущей аттестации является курсовая работа. Требования к выполнению курсовой работы, состав и правила оценки сформулированы в паспорте курсовой работы.

Общие правила выставления оценки по дисциплине определяются балльно-рейтинговой системой, приведенной в рабочей программе учебной дисциплины.

На основании приведенных далее критериев можно сделать общий вывод о сформированности компетенций ОК.5, ОК.6, ОК.8, ПК.22.В, ПК.23.В, ПК.6/НИ, за которые отвечает дисциплина, на разных уровнях.

Общая характеристика уровней освоения компетенций.

Ниже порогового. Уровень выполнения работ не отвечает большинству основных требований, теоретическое содержание курса освоено частично, пробелы могут носить существенный характер, необходимые практические навыки работы с освоенным материалом сформированы не достаточно, большинство предусмотренных программой обучения учебных заданий не выполнены или выполнены с существенными ошибками.

Пороговый. Уровень выполнения работ отвечает большинству основных требований, теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые практические навыки работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые виды заданий выполнены с ошибками.

Базовый. Уровень выполнения работ отвечает всем основным требованиям, теоретическое содержание курса освоено полностью, без пробелов, некоторые практические навыки работы с освоенным материалом сформированы недостаточно, все предусмотренные программой обучения

учебные задания выполнены, качество выполнения ни одного из них не оценено минимальным числом баллов, некоторые из выполненных заданий, возможно, содержат ошибки.

Продвинутый. Уровень выполнения работ отвечает всем требованиям, теоретическое содержание курса освоено полностью, без пробелов, необходимые практические навыки работы с освоенным материалом сформированы, все предусмотренные программой обучения учебные задания выполнены, качество их выполнения оценено числом баллов, близким к максимальному.

Паспорт экзамена

по дисциплине «Методология программной инженерии», 1 семестр

1. Методика оценки

Экзамен проводится в устной форме по билетам. Билет формируется по следующему правилу: оба вопроса выбирается из общего диапазона. В ходе экзамена преподаватель вправе задавать студенту дополнительные вопросы по существу содержания вопроса.

Форма экзаменационного билета

НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
Факультет АВТФ

Билет № _____

к экзамену по дисциплине «Методология программной инженерии»

Вопрос 1. Фаза исследования. Основные дисциплины и артефакты. Дисциплина «анализ предметной области», бизнес-анализ. Диаграммы потоков данных, деятельности. Моделирование предметной области.

Вопрос 2. Оценка программного кода. Метрики кода. Метрики количественные, сложности потока управления и потока данных, метрики ООП, прагматические метрики. Средства оценки качества программного кода.

Утверждаю: зав. кафедрой _____ должность, ФИО
(подпись)
(дата)

2. Критерии оценки

- Ответ на экзаменационный билет считается **неудовлетворительным**, если студент не ответил на один из вопросов, оценка составляет *<30% базовой*.
- Ответ на экзаменационный билет (тест) засчитывается на **пороговом** уровне, если студент при ответе на теоретический вопрос дает определение основных понятий, может показать причинно-следственные связи явлений, допускает не принципиальные ошибки, оценка составляет *30-60% от базовой*.
- Ответ на экзаменационный билет засчитывается на **базовом** уровне, если студент при дает развернутый ответ на вопросы с несущественными ошибками, оценка составляет *60-80% от базовой*.
- Ответ на экзаменационный билет (тест) билет засчитывается на **продвинутом** уровне,

если студент дает развернутый ответ на вопросы билета и дополнительные вопросы к ним, оценка составляет *80-100% от базовой*.

3. Шкала оценки

В общей оценке по дисциплине экзаменационные баллы учитываются в соответствии с правилами балльно-рейтинговой системы, приведенными в рабочей программе дисциплины, т.е. 40 баллов за экзамен.

4. Вопросы к экзамену по дисциплине «Методология программной инженерии»

1. Системная и программная инженерия. Их общность и различия. Понятие системы. Архитектура в системной инженерии.
2. Сущность программной инженерии (ПИ). Связь с computer science. Особенности в сравнении и другими инженерными дисциплинами. Свод знаний и ПИ SWEBOK
3. Жизненный цикл (ЖЦ) программного продукта и проекта. «Легкие» и «тяжелые» модели процессов разработки ПО. Этапы и технологические процессы (дисциплины) ЖЦ. Результаты этапов и основные документы. Каскадная, итеративная и спиральная модели.
4. Унифицированный процесс UP. Фазы жизненного цикла: исследование, анализ, реализация, внедрение. Содержание и результаты фаз. Итерация и ее рабочие потоки: требования, анализ, проектирование, реализация, тестирование, их содержание.
5. Фаза исследования. Основные дисциплины и артефакты. Дисциплина «анализ предметной области», бизнес-анализ. Диаграммы потоков данных, деятельности. Моделирование предметной области.
6. Фаза анализа и проектирования. Дисциплина «анализ требований». Способы извлечения и фильтрации требований. Бизнес-требования, бизнес-требования, системные требования, функциональные требования. Разработка и управление требованиями. Документ «спецификация требований к ПО ». Диаграммы прецедентов.
7. Фаза анализа и проектирования. Понятие архитектуры, ее многомерность. Основные методы проектирования и их особенности: структурное, функциональное, объектно-ориентированное, компонентное, проектирование на основе структур данных. Классы анализа. Виды классов: граница, управление, сущность. Диаграммы устойчивости. Архитектурные аспекты технологического процесса проектирования (по SWEBOK)
8. Фаза анализа и проектирования. Дисциплина проектирование (design). Ключевые моменты проектирования по SWEBOK: параллелизм, контроль и обработка событий, распределение компонентов, обработка ошибок и исключительных ситуаций и обеспечение отказоустойчивости, взаимодействие и представление (MVC), сохраняемость данных (доступность «долгоживущих» данных).
9. Виды моделей. Сущность UML как средства моделирования. Структура UML, статическая и динамическая составляющие модели. Составные элементы: сущности, отношения, диаграммы. Виды сущностей: структурные сущности – класс, интерфейс, кооперация, прецедент, активный класс, компонент, узел; поведенческие сущности –

взаимодействия, деятельности, автоматы; группирующая сущность – пакет, аннотационная сущность – примечание.

10. UML. Виды отношений: зависимость, ассоциация, агрегация, композиция, включение, обобщение, реализация. Отношения. Связи – отношения между объектами. Направленность связи, Сообщения. Диаграммы объектов. Ассоциации – отношения между классами. Свойства ассоциации: имя, кратность, навигация, атрибуты. Рефлексивные ассоциации, деревья и сети. Классы атрибутов ассоциаций (классы-ассоциации). Зависимости. Зависимости использования «use», «call», «parameter», «send» и «instantiate». Зависимости абстракции. Зависимости доступа.
11. UML. Принятые деления: классификатор-экземпляр, интерфейс-реализация. Расширения: ограничения, стереотипы. Классификация диаграмм. Диаграммы классов (объектов). Диаграммы взаимодействий, коммуникационные диаграммы.
12. UML. Диаграммы деятельности. Технология сетей Петри. Параллелизм. Поток управления, узел действия, ребро, узел управления, объектный узел, буферизация и в объектном узле. Объектные узлы – параметры, состояния объектных узлов. Контакты. Прерывающие ребра. Контакты исключений. Потоки объектов. Их аналоги в программировании. Диаграммы состояний. Конечные автоматы.
13. Экстремальное и гибкое программирование. Манифест экстремального программирования (XP). Гибкие (agile) технологии. SCRUM. Agile UP, ICONIX.
14. SCRUM как технологический фреймворк. Терминология. Спринт. Митинг. Собственник проекта. Команда. SCRUM-мастер. Беклог проекта и спринта. Планирование спринта. Диаграмма сгорания. Оценка трудоемкости. Покер-планирование.
15. Оценка программного кода. Метрики кода. Метрики количественные, сложности потока управления и потока данных, метрики ООП, прагматические метрики. Средства оценки качества программного кода.

Паспорт курсовой работы

по дисциплине «Методология программной инженерии», 1 семестр

1. Методика оценки

В курсовой работе необходимо провести анализ выпускной квалификационной работы бакалавра или другого программного проекта, в котором студент принимал участие, на предмет наличия в нем элементов, определенных сводом знаний о ПИ (SWEBOOK).

В пояснительной записке обязательны следующие разделы:

1. Содержательное описание анализируемой разработки, процесса ее проектирования и результатов.
2. Описание частей процесса выполнения проекта, соответствующих содержанию раздела SWEBOOK (например, разработка программных требований).
3. Анализ п.2. с точки зрения влияния каждого раздела на успех (неудачу) проекта, соответствия содержанию раздела SWEBOOK.
4. Выводы об уровне исполнения проекта с точки зрения методологии ПИ, анализ сильных и слабых сторон.
5. Рекомендации по совершенствованию учебного процесса и преподаванию дисциплин, связанных с ПИ, на основе анализа проекта.

2. Критерии оценки

- КР считается **невыполненным**, если отсутствуют необходимые разделы описания программы, либо программа не работает, либо студент не в состоянии объяснить принципов ее работы (алгоритма) – оценка составляет <50% баллов;
- КР засчитывается на **пороговом** уровне, разработка не полностью соответствует заданию, имеются методологические ошибки, пояснительная записка оформлена со значительными структурными, стилистическими и грамматическими ошибками - оценка составляет не более 50% базовой
- КР засчитывается на **базовом** уровне, если содержание разработки соответствует заданию, пояснительная записка оформлена в целом грамотно - оценка составляет 50-80% базовой
- КР засчитывается на **продвинутом** уровне, если тематика раскрыта полностью и разносторонне, в пояснительной записке отражены все аспекты проблемы и имеется аналитическая составляющая - оценка составляет 80-100% базовой

3. Шкала оценки

За курсовую работу ставится отдельная оценка по 100-балльной шкале

4. Примерный перечень тем курсового проекта (работы).

Темы КР формулируются по названиям выпускных работ или программных проектов с

добавлением «Анализ проекта:».

5. Перечень вопросов к защите курсового проекта (работы).

Вопросы задаются по содержанию перечисленных разделов.