

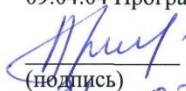


МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«Дальневосточный федеральный университет»
(ДВФУ)

ШКОЛА ЕСТЕСТВЕННЫХ НАУК


«СОГЛАСОВАНО»

Руководитель ОП Разработка программно-информационных систем по направлению 09.04.04 Программная инженерия


(подпись) Артемяева И.Л.
(Ф.И.О. рук. ОП)
« 21 » 07 2018 г.



«УТВЕРЖДАЮ»
Заведующая кафедрой прикладной математики,
механики, управления и программного обеспечения


(подпись) Артемяева И.Л.
(Ф.И.О. зав. каф.)
« 21 » 07 2018 г.

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

Методы коллективной разработки и верификации программного обеспечения

Направление подготовки – 09.04.04 Программная инженерия

Магистерская программа «Разработка программно-информационных систем»

Форма подготовки (очная)

курс 2 семестр 3

лекции 18 час.

практические занятия 0 час.

лабораторные работы 18 час.

в том числе с использованием МАО лек. 0 / пр. 0/ лаб. 18 час.

в том числе в электронной форме лек. ____/пр. ____/лаб. ____ час.

всего часов аудиторной нагрузки – 36 час.

в том числе с использованием МАО – 18 час.

в том числе контролируемая самостоятельная работа 0 час.

в том числе в электронной форме ____ час.

самостоятельная работа 72 час.

в том числе на подготовку к экзамену _45_ час.

курсовая работа / курсовой проект не предусмотрено

зачет не предусмотрен

экзамен 3 семестр

Рабочая программа составлена в соответствии с требованиями образовательного стандарта, самостоятельно устанавливаемого ДВФУ, утвержденного приказом ректора от 07.07.2015 № 12-13-1282.

Рабочая программа обсуждена на заседании кафедры прикладной математики, механики, управления и программного обеспечения, протокол № 7.2 от 21.07.2018 г.

Заведующая кафедрой прикладной математики, механики, управления и программного обеспечения Артемяева И.Л., д.т.н., профессор

Составитель: профессор каф. прикладной математики, механики, управления и программного обеспечения ДВФУ Гриняк В. М., д.т.н., доцент

Оборотная сторона титульного листа РПУД

I. Рабочая программа пересмотрена на заседании кафедры:

Протокол от «_____» _____ 20__ г. № _____

Заведующий кафедрой _____
(подпись) (И.О. Фамилия)

II. Рабочая программа пересмотрена на заседании кафедры:

Протокол от «_____» _____ 20__ г. № _____

Заведующий кафедрой _____
(подпись) (И.О. Фамилия)

ABSTRACT

Master's degree in 09.04.04 – Software engineering

Master's Program “Development of software and information systems”

Course title: Methods of the collective design and verification of software

Variable part of Block, 3 credits

Instructor: Grinyak V.

At the beginning of the course a student should be able to: analyse the problems and the development trends of the technology of programming; apply main methods and tools of the automation of the design, the production and the evaluation tests of software in a professional activity; know and apply the main conceptual regulations of functional, logic, object-oriented and visual kinds of programming and also the methods and the tools of the software design in these kinds; know and apply methods of the design and the production of software; know about the principles of the structure and the operation of the tools used for the creation of software; organize the working process in a group of software developers; know about the trends of the methods and the tools of program designing

Learning outcomes: readiness to show leadership skills and organize a teamwork, to possess effective methods of solving professional problems; an ability to work in project interdisciplinary teams including as the head of the team; practical using of abilities and skills in the organization of research and project works, in team management; an ability to understand existing approaches to software verification; possession of skills in the organization of industrial software testing

Course description: methods of the design of complex software for its application in professional activity in various subject domains including poor formalized ones

Main course literature:

1. Kotlyarov V.P. Osnovy testirovaniya programmnoho obespecheniya [Bases of software testing]. Moscow, BINOM, 2012. 285 p.

<http://lib.dvfu.ru:8080/lib/item?id=chamo:668103&theme=FEFU>

2. Sinitsyn S.V. Verifikatsiya programmnoho obespecheniya [Software verification]. Moscow, BINOM, 2008. – 367 p.

<https://lib.dvfu.ru:8443/lib/item?id=chamo:274428&theme=FEFU>

Form of final knowledge control: Examination

**Аннотация рабочей программы учебной дисциплины
«Методы коллективной разработки
и верификации программного обеспечения»**

Рабочая программа дисциплины «Методы коллективной разработки и верификации программного обеспечения» разработана для студентов 2 курса магистратуры, обучающихся по направлению 09.04.04 Программная инженерия, магистерская программа «Разработка программно-информационных систем». Дисциплина является дисциплиной выбора вариативной части учебного плана Б1.В.ДВ.01.01.

Трудоемкость дисциплины 3 зачетных единицы (108 часов). Дисциплина реализуется в 3 семестре Учебным планом предусмотрено: 18 часов лекций, 18 часов лабораторных работ (все в интерактивной форме), 0 часов контролируемой самостоятельной работы студента, самостоятельная работа студента – 72 часа, в том числе 45 на подготовку к экзамену.

Дисциплина «Методы коллективной разработки и верификации программного обеспечения» базируется на дисциплинах, связанных с технологией программирования, как правило, изучаемых на профильных направлениях бакалавриата. Знания, полученные при ее изучении, будут использованы в дисциплинах «Научно-исследовательская работа», «Производственная практика по получению профессиональных умений и опыта профессиональной деятельности», «Преддипломная практика» учебного плана, а также в практической деятельности выпускника.

Цель дисциплины – обучение студентов методам коллективной разработки сложного программного обеспечения для использования в профессиональной деятельности в различных предметных областях, в том числе и в слабо формализованных. Рассматриваются процессы: выявления участков профессиональной деятельности, которые подлежат автоматизации с использованием компьютера; моделирование соответствующих предметных областей; разработка постановок задач на построенных моделях; формулирования требований к создаваемой программой системе, ее разработки и сопровождения.

Задачи дисциплины:

1.Формирование готовности проявлять качества лидера и организовать работу коллектива разработчиков при разработке проектов информационных систем для автоматизации профессиональной деятельности, владений эффективными технологиями решения профессиональных проблем

2.Овладение умением работать в проектных междисциплинарных командах, в том числе в качестве руководителя

3.Овладение навыками организации промышленного тестирования создаваемого программного обеспечения

Для успешного изучения дисциплины «Методы коллективной разработки и верификации программного обеспечения» у обучающихся должны быть сформированы следующие предварительные компетенции:

готовность анализировать проблемы и направления развития технологий программирования;

способность применять в профессиональной деятельности основные методы и средства автоматизации проектирования, производства, испытаний и оценки качества программного обеспечения;

способность использовать знания основных концептуальных положений функционального, логического, объектно-ориентированного и визуального направлений программирования, методов, способов и средств разработки программ в рамках этих направлений;

способность использовать знания методов проектирования и производства программного продукта, принципов построения, структуры и приемов работы с инструментальными средствами, поддерживающими создание программного обеспечения;

способность использовать знания методов организации работы в коллективах разработчиков ПО, направления развития методов и программных средств коллективной разработки.

Планируемые результаты обучения по данной дисциплине (знания, умения, владения), соотнесенные с планируемыми результатами освоения образовательной программы, характеризуют этапы формирования следующих компетенций (общекультурные/ общепрофессиональные/ профессиональные компетенции (элементы компетенций)):

Код и формулировка компетенции	Этапы формирования компетенции	
ОК-2 готовностью проявлять качества лидера и организовать работу коллектива, владеть эффективными технологиями решения профессиональных проблем	Знает	Специфику работы коллектива разработчиков программного обеспечения
	Умеет	Ставить задачу членам коллектива разработчиков программного обеспечения согласно их роли
	Владеет	Инструментами поддержки командной работы при разработке программного обеспечения
ОК-3 умением работать в проектных междисциплинарных командах, в том числе в качестве руководителя	Знает	Подходы к формулировке требований к программному обеспечению
	Умеет	Организовать коллективную работу по формулировке требований к программному обеспечению

	Владеет	Инструментами поддержки командной работы при разработке программного обеспечения
ПК-6 пониманием существующих подходов к верификации моделей программного обеспечения	Знает	Методы верификации моделей программного обеспечения
	Умеет	Проводить процедуру верификации моделей программного обеспечения
	Владеет	Методикой обоснования правильности моделей программного обеспечения
ПК-11 способностью проектировать сетевые службы	Знает	Особенности архитектуры сетевых служб
	Умеет	Проводить анализ данных, необходимых для функционирования сетевых служб
	Владеет	Методами сбора и анализа данных, обрабатываемых в процессе функционирования сетевых служб
ПК-14 способностью руководить коллективом разработчиков при разработке проектов информационных систем для автоматизации профессиональной деятельности	Знает	Подходы к организации работы коллектива разработчиков
	Умеет	Разработать систему измерений компании, проанализировать данные системы измерений
	Владеет	Навыками коллективной работы при разработке проектов информационных систем для автоматизации профессиональной деятельности
ПК-15 способностью проектировать программное обеспечение, имеющее встроенные средства адаптации к изменяемым условиям эксплуатации	Знает	Основные приёмы, связанные с адаптацией программного обеспечения к изменяемым условиям эксплуатации
	Умеет	Программное обеспечение, имеющее встроенные средства адаптации к изменяемым условиям эксплуатации
	Владеет	Навыками работы в коллективе разработчиков при проектировании программного обеспечения, имеющего встроенные средства адаптации к изменяемым условиям эксплуатации

Для формирования вышеуказанных компетенций в рамках дисциплины «Методы коллективной разработки и верификации программного обеспечения» применяются следующие методы активного/ интерактивного обучения: *семинары, проектный метод.*

I. СТРУКТУРА И СОДЕРЖАНИЕ ТЕОРЕТИЧЕСКОЙ ЧАСТИ КУРСА

Тема 1. Профессиональная деятельность ИТ специалистов с точки зрения разработки программного обеспечения как отрасли промышленности (3 час).

Профессиональные позиции ИТ специалистов и их особенности. Особенности работы в отечественных и международных ИТ компаниях. Понятие коллективной разработки программного обеспечения. Средства поддержки процесса коллективной разработки ПО.

Тема 2. Модели жизненного цикла разработки программных средств (3 час).

Фазы стандартного процесса разработки ПО. Виды проектной документации, сопровождающей разработку ПО. Модели жизненных циклов разработки ПО: водопад, водопад с перекрытиями, водопад с возвратами, инкрементная модель, итеративная модель, итеративно-инкрементная модель, спиральная модель.

Тема 3. Специфика верификации рабочих продуктов при коллективной разработке. Принципы проведения и организации инспекций рабочих продуктов (3 час).

Верификация программных продуктов в процессе их разработки. Принципы проведения инспекций кода, дизайна, тестов, требований. Роли участников инспекций. Процедура организации и проведения формальной инспекции. Метрики по результатам инспекций. Статус и степень серьёзности замечаний по инспекции. Анализ результатов инспекции. Корректировка процесса разработки программного обеспечения на основе результатов инспекции.

Тема 4. Структура и анализ бизнес-процессов коллективной разработки программного обеспечения. Системы сопровождения заданий. Системы отслеживания дефектов (3 час).

Workflow системы отслеживания дефектов и сопровождения задач при разработке программных продуктов. Технологический процесс коллективной разработки программ. Классы подзадач. Основные состояния подзадачи. Переход подзадач из состояния в состояние. Специфика систем сопровождения заданий (issue tracking) и отслеживания дефектов (defect

tracking). Специфика управления проектами с использованием метрик сотрудников.

Тема 5. Структура бизнес-процессов коллективной разработки программного обеспечения: средства и методы сбора метрик сотрудников (3 час).

Особенности бизнес-процессов ИТ предприятий. Общие элементы методики регистрации временных затрат. Проектные и непроjektные виды деятельности. Типы действий различных видов деятельности. Логирование времени сотрудником. Отчеты о затратах времени и уведомления. Специфика управления проектами с использованием метрик сотрудников.

Тема 6. Измерения при разработке и сопровождении программного продукта. Основные метрики эффективности процесса разработки и метрики качества ПС (3 час).

Роль и место измерений при производстве программных средств. Виды измерений, связанные с процессом разработки ПО. Метрики эффективности процесса производства. Метрики качества продуктов. Специфика управления проектами с использованием метрик сотрудников. Структура информационных систем, обеспечивающих программу измерений.

**II. СТРУКТУРА И СОДЕРЖАНИЕ ПРАКТИЧЕСКОЙ ЧАСТИ
КУРСА**

Лабораторные работы (18 час.)

Лабораторная работа №1. Организация коллективной работы разработчиков (2 часа)

1. Объединение в проектные группы, распределение ролей
2. Выбор направленности, типа и конкретной темы проекта
3. Выбор и обоснование инструментов разработки

Лабораторная работа №2. Работа с заказчиком программного продукта (2 часа)

1. Сбор и формулировка основных требований к программному продукту
2. Объектный анализ и концептуальное моделирование конкретной предметной области
3. Выбор и обоснование архитектуры проекта

Лабораторная работа №3. Работа с требованиями к программному продукту (2 часа)

1. Разработка пользовательских требований и внешних спецификаций к программной системе, подготовка документа «Пользовательские требования»

2. Разработка системных требований и верхнего уровня проекта программной системы; подготовка документа «Системные требования»

3. Разработка архитектуры программной системы; подготовка документа «Дизайн проекта»

Лабораторная работа №4. Кодирование программной системы (2 часа)

1. Подготовка документа «Список задач проекта»

2. Разработка политики по управлению конфигурацией проекта

3. Разработка политики по верификации рабочих продуктов на проекте

Лабораторная работа №5. Тестирование программной системы (2 часа)

1. Подготовка документа «План тестирования проекта»

2. Подготовка документа «Матрица покрытия требований»

3. Разработка политики по прогону тестов на проекте (модульное, системное тестирование)

Лабораторная работа №6. Доработка программной системы (2 часа)

1. Разработка списка задач на доработку программной системы по результатам тестирования

2. Подготовка отчета по эффективности обработки задач

Лабораторная работа №7. Метрики качества разработки программной системы (2 часа)

1. Разработка системы метрик для оценки качества программного продукта

2. Сбор метрик

Лабораторная работа №8. Метрики эффективности процесса разработки программной системы (2 часа)

1. Разработка системы метрик для эффективности процесса разработки программного продукта

2. Сбор метрик

Лабораторная работа №9. Выработка рекомендаций по изменению процесса разработки (2 часа)

1. Анализ метрик для оценки качества программного продукта и эффективности процесса разработки

2. Формулировка рекомендаций по трансформации процесса разработки по результатам программы измерений

Ш. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ

Объём самостоятельной работы по дисциплине составляет 72 часа, в том числе 45 часов на подготовку к экзамену. Контроль самостоятельной работы со студентами осуществляется во время семинаров и консультаций.

Учебно-методическое обеспечение самостоятельной работы обучающихся по дисциплине «Методы коллективной разработки и верификации программного обеспечения» представлено в Приложении 1 и включает в себя: план-график выполнения самостоятельной работы по дисциплине, в том числе примерные нормы времени на выполнение по каждому заданию; характеристика заданий для самостоятельной работы обучающихся и методические рекомендации по их выполнению; требования к представлению и оформлению результатов самостоятельной работы; критерии оценки выполнения самостоятельной работы.

IV. КОНТРОЛЬ ДОСТИЖЕНИЯ ЦЕЛЕЙ КУРСА

№ п/п	Контролируемые разделы/темы дисциплины	Коды и этапы формирования компетенций	Оценочные средства - наименование		
			текущий контроль	промежуточная аттестация	
1	Тема 1. Профессиональная деятельность ИТ специалистов с точки зрения разработки программного обеспечения как отрасли промышленности	ОК-2	Знает	Опрос (УО-1)	Экзамен (2, 17)
		ОК-3	Умеет	С/работа (ПР-9)	Экзамен (2, 17)
			Владеет	С/работа (ПР-9)	Экзамен (2, 17)
2	Тема 2. Модели жизненного цикла разработки	ОК-2	Знает	Опрос (УО-1)	Экзамен (1, 17, 19)
		ОК-3	Умеет	С/работа (ПР-9)	Экзамен (1, 17,

	программных средств		Владеет	С/работа (ПР-9)	19) Экзамен (1, 17, 19)
3	Тема 3. Специфика верификации рабочих продуктов при коллективной разработке. Принципы проведения и организации инспекций рабочих продуктов	ПК-6 ПК-11 ПК-14 ПК-15	Знает	Опрос (УО-1)	Экзамен (3, 4)
			Умеет	С/работа (ПР-9)	Экзамен (3, 4)
			Владеет	С/работа (ПР-9)	Экзамен (3, 4)
4	Тема 4. Структура и анализ бизнес-процессов коллективной разработки программного обеспечения. Системы сопровождения заданий. Системы отслеживания дефектов	ПК-6 ПК-11 ПК-14 ПК-15	Знает	Опрос (УО-1)	Экзамен (5, 6)
			Умеет	С/работа (ПР-9)	Экзамен (5, 6)
			Владеет	С/работа (ПР-9)	Экзамен (5, 6)
5	Тема 5. Структура бизнес-процессов коллективной разработки программного обеспечения: средства и методы сбора метрик сотрудников	ПК-6 ПК-11 ПК-14 ПК-15	Знает	Опрос (УО-1)	Экзамен (7, 8)
			Умеет	С/работа (ПР-9)	Экзамен (7, 8)
			Владеет	С/работа (ПР-9)	Экзамен (7, 8)
6	Тема 6. Измерения при разработке и сопровождении программного продукта. Основные метрики эффективности процесса разработки и метрики качества ПС	ПК-6 ПК-11 ПК-14 ПК-15	Знает	Опрос (УО-1)	Экзамен (9, 10)
			Умеет	С/работа (ПР-9)	Экзамен (9, 10)
			Владеет	С/работа (ПР-9)	Экзамен (9, 10)

Типовые контрольные задания, методические материалы, определяющие процедуры оценивания знаний, умений и навыков и (или) опыта деятельности, а также критерии и показатели, необходимые для оценки знаний, умений, навыков и характеризующие этапы формирования

компетенций в процессе освоения образовательной программы, представлены в Приложении 2.

V. СПИСОК УЧЕБНОЙ ЛИТЕРАТУРЫ И ИНФОРМАЦИОННО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Основная литература

(электронные и печатные издания)

1. Котляров, В. П. Основы тестирования программного обеспечения: учебное пособие / В. П. Котляров, Т. В. Коликова. Москва: Интернет-Университет Информационных Технологий: БИНОМ. Лаборатория знаний, 2012. – 285 с. <http://lib.dvfu.ru:8080/lib/item?id=chamo:668103&theme=FEFU>
2. Сеницын, С.В. Верификация программного обеспечения: учебное пособие / С.В. Сеницын, Н.Ю. Налютин. – М.: Интернет-Университет Информационных Технологий : БИНОМ. Лаборатория знаний, 2008. – 367 с. <https://lib.dvfu.ru:8443/lib/item?id=chamo:274428&theme=FEFU>
3. Технология разработки программного обеспечения: Учеб. пос. / Л.Г.Гагарина, Е.В.Кокорева, Б.Д.Виснадул; Под ред. проф. Л.Г.Гагариной - М.: ИД ФОРУМ: НИЦ Инфра-М, 2013. - 400 с. Режим доступа: <http://znanium.com/go.php?id=389963>
4. Введение в архитектуру программного обеспечения: Учебное пособие / Гагарина Л.Г., Федоров А.Р., Федоров П.А. - М.: ИД ФОРУМ, НИЦ ИНФРА-М, 2016. - 320 с.: 60x90 1/16. - (Высшее образование) (Переплёт 7БЦ) ISBN 978-5-8199-0649-1 <http://znanium.com/catalog.php?bookinfo=542665>

Дополнительная литература

(печатные и электронные издания)

1. Соммервилл И. Инженерия программного обеспечения. 6-е издание. М.: Изд. дом Вильямс, 2002. – 624 с. <http://lib.dvfu.ru:8080/lib/item?id=chamo:735&theme=FEFU>
2. Браудэ Э. Технология разработки программного обеспечения, Издательский дом «Питер», 2004. - 656 с. <http://lib.dvfu.ru:8080/lib/item?id=chamo:232388&theme=FEFU>
3. Иванова Г.С. Технология программирования. М.: Изд-во МГУ. 2002. <http://lib.dvfu.ru:8080/lib/item?id=chamo:398553&theme=FEFU>
4. Жоголев Е.А. Технология программирования. М.: Научный мир. 2004. – 215 с.

- <http://lib.dvfu.ru:8080/lib/item?id=chamo:7891&theme=FEFU>
5. Бенкен Е.С. PHP, MySQL, XML программирование для Интернета. СПб: БХВ-Петербург. -2008. Режим доступа: <http://lib.dvfu.ru:8080/lib/item?id=chamo:382736&theme=FEFU>
 6. Скляр Д., Трахтенберг А. PHP. Рецепты программирования. – СПб: Русская редакция БХВ-Петербург. -2007. Режим доступа: <http://lib.dvfu.ru:8080/lib/item?id=chamo:381965&theme=FEFU>
 7. Липаев, В.В. Программная инженерия. Методологические основы [Текст]: Учеб. / В. В. Липаев; Гос. ун-т — Высшая школа экономики. — М. : ТЕИС, 2006. — 608 с. <http://lib.dvfu.ru:8080/lib/item?id=chamo:248067&theme=FEFU>

Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

1. <http://window.edu.ru/resource/711/79711> Липаев В.В. Проектирование и производство сложных заказных программных продуктов. - М.: СИНТЕГ, 2011. - 408 с
2. <http://e.lanbook.com/book/43554> Шопырин, Д.Г. Управление проектами разработки ПО. Дисциплина «Гибкие технологии разработки программного обеспечения». [Электронный ресурс] — Электрон. дан. — СПб.: НИУ ИТМО, 2007. — 131 с.
3. <http://novtex.ru/> Журналы издательства Новые технологии
4. <https://github.com/> Система контроля версий Git

Перечень информационных технологий и программного обеспечения

Интегрированные среды разработчика Eclipse, NetBeans, Visual Studio. Система контроля версий Git. Офисное ПО.

VI. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Особенности проведения лабораторных работ по дисциплине (активные и интерактивные формы)

Лабораторные работы по дисциплине проводятся параллельно с самостоятельной работой студентов в форме деловой игры.

Особенностью учебного процесса по дисциплине является его максимальное приближение к реальным условиям работы на проектах по

коллективной разработке ПО: как коммерческим, так и фрилансовым. Всем аспирантам необходимо создать аккаунт на портале github.com. Разработка проектов ведётся на базе именно этой системы контроля версий с использованием встроенной системы issue трекинга.

В начале семестра студенты разбиваются на команды (по 3-5 человек в каждой). Каждая команда придумывает название своего проекта и определяется с составом (тим-лидер, кодировщики, технический писатель). В том случае, если группа аспирантов немногочисленна (1-2 человека), работа проводится по упрощённой схеме.

В обязанности тим-лидера входит общее руководство проектом и поддержание целостности программного кода путем контроля интеграции всех изменений программного кода.

В обязанности кодировщиков входит текущая работа над проектом, в том числе - написание кода согласно требованиям и инспектирование кода других кодировщиков.

В обязанности технического писателя входит разработка документации и поддержание её целостности.

Структура рабочего продукта по проекту

Тим-лидер каждой команды создаёт репозиторий в системе контроля версий, называя его согласно названию проекта в целом. В корне репозитория должен находиться файл `readme`, содержащий краткое описание проекта. Кроме того, в корне находятся папки `CODE` и `DOCS`.

В папке `CODE` размещается код проекта.

В папке `DOCS` - файлы с документацией.

Папка `DOCS` содержит, в свою очередь, папки:

`PROJECT PLAN` - для размещения документов с планом проекта

`REQUIREMENS` - для размещения документов с требованиями

`DESIGN` - для размещения документов дизайна

`CODING` - для размещения документов кодирования

`TESTING` - для размещения документов тестирования

Работа с системой контроля версий

После того, как тим-лидер создал репозиторий он указывает остальных членов команды его коллабораторами. Таким образом они получают максимум прав для работы с проектом.

Проект должен иметь двухуровневую структуру веток.

Основная ветка (`master`) - содержит стабильную версию, интеграцию в которую осуществляет только (!!) тим-лидер. Все члены команды, в том числе тим-лидер, создают в своём репозитории копии стабильной версии по

мере необходимости. С этих рабочих (work) веток они создают запросы на интеграцию (merge) в основную ветку. Запрос на интеграцию назначается вначале на одного из членов команды, который выполняет роль инспектора. Если инспектор одобряет работу, он переназначает запрос на тим-лидера, который после проверки осуществляет интеграцию изменений. Сам тим-лидер поступает аналогично: после одобрения инспектором запрос на интеграцию возвращается к тим-лидеру обратно.

- **Документация, создаваемая по проекту**
- В рамках работы над проектом создается следующая документация:
- Software Project Plan (SPP) - календарный план работы над проектом.
- Marketing Requirements Specification (MRS) - маркетинговые требования к проекту в виде презентации рекламного характера.
- Software Requirements Specification (SRS) - технические требования к продукту
- User Manual - руководство оператора (пользователя).
- Software Design Specification (SDS) - документы дизайна, раскрывающие архитектуру программного продукта и то КАК именно будут реализованы те или иные требования.
- Code Issue List - список задач, выполняемых на этапе кодирования.
- Software Test Plan - документ, включающий в себя спецификации тестов и процедур тестирования, в том числе список тестов.
- Requirements Tracability Matrix - матрица покрытия тестами требований.
- Test Logs - документ, который заполняется в ходе тестирования и содержит результаты выполнения тестов.
- Вся документация размещается в соответствующих папках проекта. Запрос на интеграцию для документации выполняется аналогично файлам с кодом.

VII. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Лекции проводятся с использованием проектора и внутренней системы портала ДВФУ. Лабораторные занятия проходят в аудиториях, оборудованных компьютерами типа Lenovo C360G-i34164G500UDK с лицензионными программами Microsoft Office 2013 и аудиовизуальными средствами проектор Panasonic DLPPjectorPT-D2110XE, плазма LG

FLATRON M4716CCBAM4716CJ. Для выполнения самостоятельной работы студенты в жилых корпусах ДВФУ обеспечены Wi-Fi.



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«Дальневосточный федеральный университет»
(ДВФУ)

ШКОЛА ЕСТЕСТВЕННЫХ НАУК

**УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ
ОБУЧАЮЩИХСЯ**

по дисциплине «Методы коллективной разработки и верификации программного обеспечения»

Направление подготовки – 09.04.04 Программная инженерия

Магистерская программа «Разработка программно-информационных систем»

Форма подготовки (очная)

Владивосток
2018

План-график выполнения самостоятельной работы по дисциплине

№ п/п	Дата/сроки выполнения	Вид самостоятельной работы	Примерные нормы времени на выполнение	Форма контроля
1	1-2 неделя	Software Project Plan (SPP) – календарный план работы над проектом	3 часов	Собеседование
2	3-4 неделя	Marketing Requirements Specification (MRS) - маркетинговые требования к проекту в виде презентации рекламного характера	3 часов	Проект
3	5-6 неделя	Software Requirements Specification (SRS) - технические требования к продукту	3 часов	Проект
4	7-8 неделя	User Manual - руководство оператора (пользователя).	3 часов	Проект
5	9-10 неделя	Software Design Specification (SDS) - документы дизайна, раскрывающие архитектуру программного продукта и то КАК именно будут реализованы те или иные требования.	3 часов	Проект
6	11-12 неделя	Code Issue List - список задач, выполняемых на этапе кодирования.	3 часов	проект
7	13-14 неделя	Software Test Plan - докумен, включающий в себя спецификации тестов и процедур тестирования, в том числе список тестов.	3 часов	Проект
8	15-16 неделя	Requirements Tracability Matrix - матрица покрытия тестами требований.	3 часов	Проект
9	17-18 неделя	Test Logs - документ, который заполняется в ходе тестирования и содержит результаты выполнения тестов.	3 часов	Проект

Рекомендации по самостоятельной работе студентов

Самостоятельная работа включает в себя разработку итогового проекта, параллельно с лабораторными работами по дисциплине. В начале семестра студенты разбиваются на команды (по 3-5 человек в каждой). Каждая команда придумывает название своего проекта и определяется с составом (тим-лидер, кодировщики, технический писатель).

Рекомендации по планированию и организации времени, отведенного на изучение дисциплины

Освоение дисциплины следует начинать с изучения рабочей учебной программы, которая содержит основные требования к знаниям, умениям и навыкам. Обязательно следует учитывать рекомендации преподавателя, данные в ходе установочных занятий. Затем – приступать к изучению отдельных разделов и тем в порядке, предусмотренном программой.

Получив представление об основном содержании раздела, темы, необходимо изучить материал с помощью рекомендуемой основной литературы. Целесообразно составить краткий конспект или схему, отображающую смысл и связи основных понятий данного раздела и включенных в него тем. Обязательно следует записывать возникшие вопросы, на которые не удалось ответить самостоятельно.

Подготовку к началу обучения включает несколько необходимых пунктов:

1) Необходимо создать для себя рациональный и эмоционально достаточный уровень мотивации к последовательному и планомерному изучению дисциплины.

2) Необходимо изучить список рекомендованной основной и дополнительной литературы и убедиться в её наличии у себя дома или в библиотеке в бумажном или электронном виде.

3) Желательно в самом начале периода обучения возможно тщательнее спланировать время, отводимое на работу с источниками и литературой по дисциплине.

Рекомендации по работе с литературой

1. Всю учебную литературу желательно изучать «под конспект». Цель написания конспекта по дисциплине – сформировать навыки по поиску, отбору, анализу и формулированию учебного материала.

2. При работе над литературой обязательно выявляются и отмечаются трудные для самостоятельного изучения вопросы, с которыми уместно обратиться к преподавателю при посещении установочных лекций и консультаций, либо в индивидуальном порядке.

3. При чтении учебной и научной литературы всегда следить за точным и полным пониманием значения терминов и содержания понятий, используемых в тексте.

Методические указания к самостоятельной работе студентов

Самостоятельная работа включает в себя разработку итогового проекта, параллельно с лабораторными работами по дисциплине. В начале семестра студенты разбиваются на команды (по 3-5 человек в каждой). Каждая

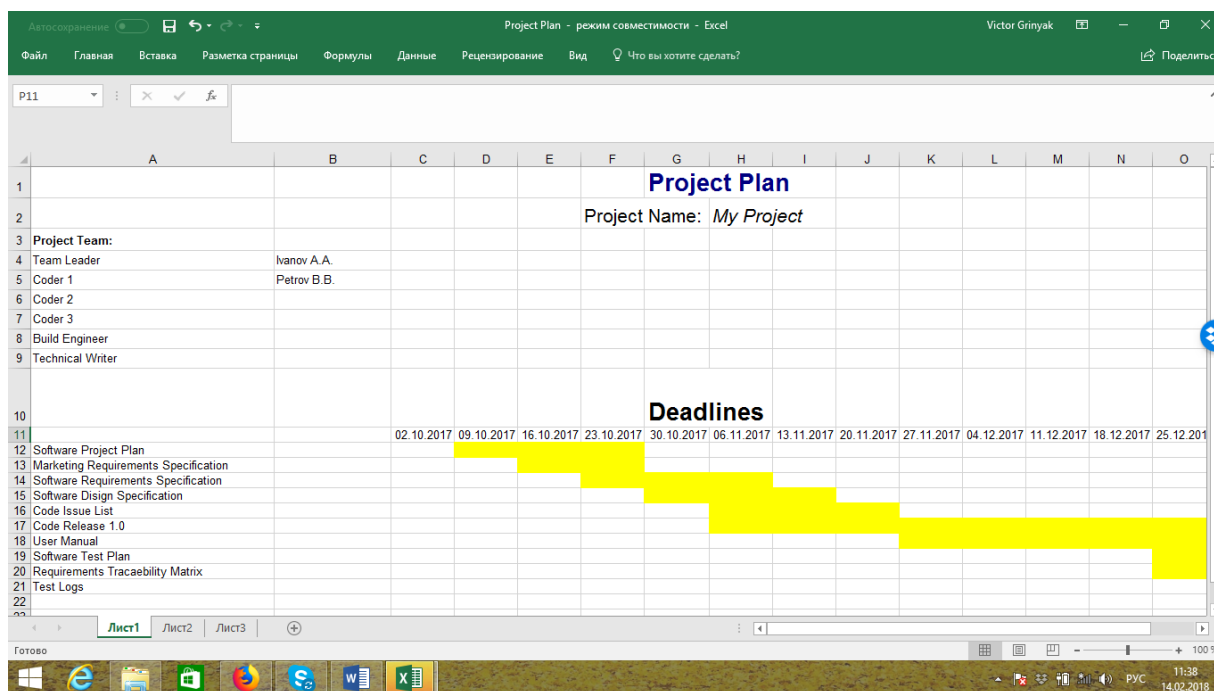
команда придумывает название своего проекта и определяется с составом (тим-лидер, кодировщики, технический писатель).

В рамках работы над проектом создается следующая документация:

- Software Project Plan (SPP) - календарный план работы над проектом.
- Marketing Requirements Specification (MRS) - маркетинговые требования к проекту в виде презентации рекламного характера.
- Software Requirements Specification (SRS) - технические требования к продукту
- User Manual - руководство оператора (пользователя).
- Software Design Specification (SDS) - документы дизайна, раскрывающие архитектуру программного продукта и то КАК именно будут реализованы те или иные требования.
- Code Issue List - список задач, выполняемых на этапе кодирования.
- Software Test Plan - документ, включающий в себя спецификации тестов и процедур тестирования, в том числе список тестов.
- Requirements Tracability Matrix - матрица покрытия тестами требований.
- Test Logs - документ, который заполняется в ходе тестирования и содержит результаты выполнения тестов.

Результаты самостоятельной работы представляются на итоговом семинаре в виде доклада (презентации).

Примерная форма документа Software Project Plan



Примерная форма документа Software Requirements Specification

Программный продукт *название* предназначен для

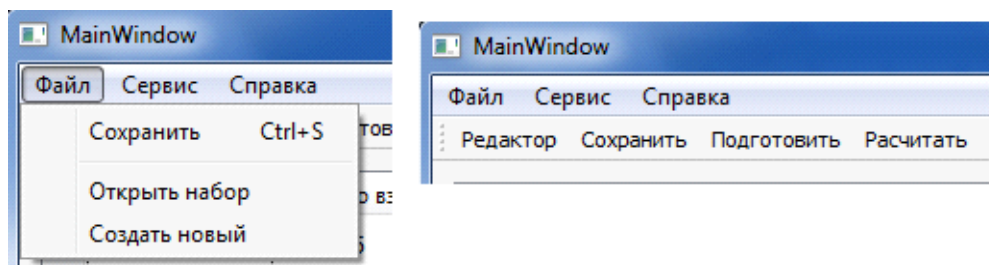
Программный продукт *название* состоит из следующих подсистем:

1. Пользовательский интерфейс
2. Модуль расчета параметров X
3. Модуль расчета параметров Y
4. Подсистема ввода-вывода
5. ...
6. ...

Требования к подсистеме «Пользовательский интерфейс»

Требование REQ_UI_001

Функция сохранения данных должна быть доступна для пользователя тремя способами: кнопка «Сохранить» на панели инструментов, элемент меню «Файл» и нажатием сочетания клавиш «Ctrl + S» (см. Рисунок 10)



Требование REQ_UI_002

Если пользователь ввел данные в строке «+» в одной из ячеек столбцов для отображения обязательной информации, то система должна создать новую строку «+», только если заполнены все ячейки столбцов обязательной информации доступные для редактирования (см. требование REQ_UI_003).

Требование REQ_UI_003

...

Требования к подсистеме «Модуль расчета параметров X»

Требование REQ_CX_001

.....

Требование REQ_CX_002

....

Примерная форма документа Software Disign Specification

Программный продукт *название* предназначен для

Настоящий документ содержит спецификации дизайна, включающие в себя:

1. Архитектурно-контекстную диаграмму системы в целом
2. Архитектурно-контекстную диаграммы подсистем
3. Диаграммы наследования классов (по подсистемам)
4. Диаграммы вызовов объектов (по подсистемам)

1. Архитектурно-контекстная диаграмма системы

Программный продукт *название* состоит из следующих подсистем:

1. Пользовательский интерфейс
2. Модуль расчета параметров X
3. Модуль расчета параметров Y
4. Подсистема ввода-вывода
5. ...
6. ...

2.1. Архитектурно контекстная диаграмма подсистемы «Пользовательский интерфейс»

...

2.2. Диаграмма наследования классов подсистемы «Пользовательский интерфейс»

...

2.3. Диаграммы вызовов объектов подсистемы «Пользовательский интерфейс»

...

3.1. Архитектурно контекстная диаграмма подсистемы «Модуль расчета параметров X»

...

3.2. Диаграмма наследования классов подсистемы «Модуль расчета параметров X»

...

3.3. Диаграммы вызовов объектов подсистемы «Модуль расчета параметров X»

...

Примерная форма документа Requirements Traceability Matrix

Req ID	TEST_UI_001	TEST_UI_002	TEST_CX_001	TEST_CX_002	TEST_SYS_001	TEST_SYS_002
REQ_UI_001	Yes	Yes				
REQ_UI_002	Yes	Yes	Yes		Yes	
REQ_CX_001			Yes		Yes	
REQ_CX_002	Yes				Yes	

Примерная форма документа Software Test Plan

Тесты для тестирования подсистемы «Пользовательский интерфейс»

Тест TEST_UI_001

Тестируемые требования: REQ_UI_001
Описание теста...

Тест TEST_UI_002

Тестируемые требования: REQ_UI_002
Описание теста...

Тесты для тестирования подсистемы «Модуль расчета параметров X»

Тест TEST_CX_001

Тестируемые требования: REQ_CX_001
Описание теста...

Тест TEST_CX_002

Тестируемые требования: REQ_CX_002

Описание теста...

Тесты для тестирования системы в целом (System Test)

Тест TEST_SYS_001

Тестируемые требования: REQ_CX_001, REQ_CX_002, REQ_UI_002

Описание теста...

Тест TEST_SYS_002

Тестируемые требования: REQ_CX_002

Описание теста...

Примерная форма документа Software Test Logs

Тесты для тестирования подсистемы «Пользовательский интерфейс»

Тест TEST_UI_001

Тестируемые требования: REQ_UI_001

Описание теста...

Тестируемая версия продукта: *номер версии из GitHub*

Ожидаемый результат: Показано 5 опций меню

Видимый результат: Показано 5 опций меню

Резюме: **Тест пройден**

Тест TEST_UI_002

Тестируемые требования: REQ_UI_002

Описание теста...

Тестируемая версия продукта: *номер версии из GitHub*

Ожидаемый результат: Показано 2 всплывающих окна

Видимый результат: Показано 1 всплывающее окно

Резюме: **Тест НЕ пройден**

Тесты для тестирования подсистемы «Модуль расчета параметров X»

Тест TEST_CX_001

Тестируемые требования: REQ_CX_001

Описание теста...

Тестируемая версия продукта: *номер версии из GitHub*

Ожидаемый результат:

Видимый результат:

Резюме: **Тест пройден**

Тест TEST_CX_002

Тестируемые требования: REQ_CX_002

Описание теста...

Тестируемая версия продукта: *номер версии из GitHub*

Ожидаемый результат:

Видимый результат:

Резюме: **Тест НЕ пройден**

Критерии оценки выполнения самостоятельной работы

Критерии оценки выполнения самостоятельной работы – достижение правильного результата при разработке программного средства, качество оформления документации и представления работы на семинаре.

Подготовка к промежуточной аттестации по дисциплине: экзамену (зачету)

К аттестации допускаются студенты, которые систематически в течение всего семестра посещали и работали на занятиях и показали уверенные знания в ходе выполнении лабораторных работ и итогового проекта.

Непосредственная подготовка к аттестации осуществляется по вопросам, представленным в рабочей учебной программе. Тщательно изучите формулировку каждого вопроса, вникните в его суть, составьте план ответа.



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«Дальневосточный федеральный университет»
(ДВФУ)

ШКОЛА ЕСТЕСТВЕННЫХ НАУК

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

по дисциплине «Методы коллективной разработки и верификации программного обеспечения»

Направление подготовки – 09.04.04 Программная инженерия

Магистерская программа «Разработка программно-информационных систем»

Форма подготовки (очная)

Владивосток
2018

Паспорт фонда оценочных средств
по дисциплине «Методы коллективной разработки и верификации
программного обеспечения»

Код и формулировка компетенции	Этапы формирования компетенции	
ОК-2 готовностью проявлять качества лидера и организовать работу коллектива, владеть эффективными технологиями решения профессиональных проблем	Знает	Специфику работы коллектива разработчиков программного обеспечения
	Умеет	Ставить задачу членам коллектива разработчиков программного обеспечения согласно их роли
	Владеет	Инструментами поддержки командной работы при разработке программного обеспечения
ОК-3 умением работать в проектных междисциплинарных командах, в том числе в качестве руководителя	Знает	Подходы к формулировке требований к программному обеспечению
	Умеет	Организовать коллективную работу по формулировке требований к программному обеспечению
	Владеет	Инструментами поддержки командной работы при разработке программного обеспечения
ПК-6 пониманием существующих подходов к верификации моделей программного обеспечения	Знает	Методы верификации моделей программного обеспечения
	Умеет	Проводить процедуру верификации моделей программного обеспечения
	Владеет	Методикой обоснования правильности моделей программного обеспечения
ПК-11 способностью проектировать сетевые службы	Знает	Особенности архитектуры сетевых служб
	Умеет	Проводить анализ данных, необходимых для функционирования сетевых служб
	Владеет	Методами сбора и анализа данных, обрабатываемых в процессе функционирования сетевых служб
ПК-14 способностью руководить коллективом разработчиков при разработке проектов информационных систем для автоматизации профессиональной деятельности	Знает	Подходы к организации работы коллектива разработчиков
	Умеет	Разработать систему измерений компании, проанализировать данные системы измерений
	Владеет	Навыками коллективной работы при разработке проектов информационных систем для автоматизации профессиональной деятельности
ПК-15 способностью проектировать программное обеспечение, имеющее встроенные средства адаптации к изменяемым	Знает	Основные приёмы, связанные с адаптацией программного обеспечения к изменяемым условиям эксплуатации
	Умеет	Программное обеспечение, имеющее встроенные средства адаптации к

условиям эксплуатации		изменяемым условиям эксплуатации
	Владеет	Навыками работы в коллективе разработчиков при проектировании программного обеспечения, имеющего встроенные средства адаптации к изменяемым условиям эксплуатации

№ п/п	Контролируемые разделы/темы дисциплины	Коды и этапы формирования компетенций	Оценочные средства - наименование		
			текущий контроль	промежуточная аттестация	
1	Тема 1. Профессиональная деятельность ИТ специалистов с точки зрения разработки программного обеспечения как отрасли промышленности	ОК-2 ОК-3	Знает	Опрос (УО-1)	Экзамен (2, 17)
			Умеет	С/работа (ПР-9)	Экзамен (2, 17)
			Владеет	С/работа (ПР-9)	Экзамен (2, 17)
2	Тема 2. Модели жизненного цикла разработки программных средств	ОК-2 ОК-3	Знает	Опрос (УО-1)	Экзамен (1, 17, 19)
			Умеет	С/работа (ПР-9)	Экзамен (1, 17, 19)
			Владеет	С/работа (ПР-9)	Экзамен (1, 17, 19)
3	Тема 3. Специфика верификации рабочих продуктов при коллективной разработке. Принципы проведения и организации инспекций рабочих продуктов	ПК-6 ПК-11 ПК-14 ПК-15	Знает	Опрос (УО-1)	Экзамен (3, 4)
			Умеет	С/работа (ПР-9)	Экзамен (3, 4)
			Владеет	С/работа (ПР-9)	Экзамен (3, 4)
4	Тема 4. Структура и анализ бизнес-процессов коллективной разработки программного обеспечения. Системы сопровождения заданий. Системы отслеживания	ПК-6 ПК-11 ПК-14 ПК-15	Знает	Опрос (УО-1)	Экзамен (5, 6)
			Умеет	С/работа (ПР-9)	Экзамен (5, 6)
			Владеет	С/работа (ПР-9)	Экзамен (5, 6)

	дефектов				
5	Тема 5. Структура бизнес-процессов коллективной разработки программного обеспечения: средства и методы сбора метрик сотрудников	ПК-6 ПК-11 ПК-14 ПК-15	Знает	Опрос (УО-1)	Экзамен (7, 8)
			Умеет	С/работа (ПР-9)	Экзамен (7, 8)
			Владеет	С/работа (ПР-9)	Экзамен (7, 8)
6	Тема 6. Измерения при разработке и сопровождении программного продукта. Основные метрики эффективности процесса разработки и метрики качества ПС	ПК-6 ПК-11 ПК-14 ПК-15	Знает	Опрос (УО-1)	Экзамен (9, 10)
			Умеет	С/работа (ПР-9)	Экзамен (9, 10)
			Владеет	С/работа (ПР-9)	Экзамен (9, 10)

Шкала оценивания уровня сформированности компетенций

Код и формулировка компетенции	Этапы формирования компетенции		критерии	показатели
ОК-2 готовностью проявлять качества лидера и организовать работу коллектива, владеть эффективными технологиями решения профессиональных проблем	знает (пороговый уровень)	Специфика работы коллектива разработчиков программного обеспечения	Знание ролей в коллективе разработчиков ПО	Способность дать ответы на вопросы
	умеет (продвинутой)	Ставить задачу членам коллектива разработчиков программного обеспечения согласно их роли	Умение выделять отдельные функции в пользовательских историях	Не менее 3 функций на историю
	владеет	Инструм	Владение	Не менее 2

	(высокий)	инструментами поддержки командной работы при разработке программного обеспечения	различными системами для поддержки командной работы	систем
ОК-3 умением работать в проектных междисциплинарных командах, в том числе в качестве руководителя	знает (пороговый уровень)	Подходы к формулировке требований к программному обеспечению	Умение сформулировать требования на основе пользовательских историй	Не менее 3 требований на историю
	умеет (продвинутой)	Организовывать коллективную работу по формулировке требований к программному обеспечению	Умение собирать и формулировать пользовательские истории	Не менее 3 историй за 15 минут
	владеет (высокий)	Инструментами поддержки командной работы при разработке программного обеспечения	Владение различными системами для поддержки командной работы	Не менее 2 систем
ПК-6 понимание	знает	Методы	Знание методов	Не менее 5

существующих подходов к верификации моделей программного обеспечения	(пороговый уровень)	верификации моделей программного обеспечения	верификации моделей ПО	методов
	умеет (продвинутой)	Проводить процедуру верификации моделей программного обеспечения	Умение пользоваться методами верификации	Не менее 5 методов
	владеет (высокий)	Методикой обоснования правильности моделей программного обеспечения	Владение подходами к верификации ПО	Не менее 5 подходов
ПК-11 способность проектировать сетевые службы	знает (пороговый уровень)	Особенности архитектуры сетевых служб	Знание нескольких вариантов архитектуры сетевых служб	Не менее 2 вариантов
	умеет (продвинутой)	Проводить анализ данных, необходимых для функционирования сетевых служб	Знание форматов данных сетевых служб	Не менее 2 форматов
	владеет (высокий)	Методам и сбора и анализа данных, обрабатываемых в процессе функционирования сетевых	Владение инструментами анализа и обработки данных	Не менее 2 инструментов

		служб		
ПК-14 способностью руководить коллективом разработчиков при разработке проектов информационных систем для автоматизации профессиональной деятельности	знает (пороговый уровень)	Подходы к организации работы коллектива разработчиков	Знание нескольких подходов к организации работы коллектива разработчиков	Не менее 2 подходов
	умеет (продвинутой)	Разработать систему измерений компании, проанализировать данные системы измерений	Умение сформулировать метрики эффективности процесса разработки ПО и качества ПО	Не менее 10 метрик
	владеет (высокий)	Навыкам и коллективной работы при разработке проектов в информационных систем для автоматизации профессиональной деятельности	Способность распределить по ролям коллектив разработчиков	Не менее 10 разработчиков и 5 ролей
ПК-15 способностью проектировать программное обеспечение, имеющее встроенные средства адаптации к изменяемым условиям эксплуатации	знает (пороговый уровень)	Основные приёмы, связанные с адаптацией программного обеспечения	Знание типовых приемов адаптации ПО	Не менее 2 приемов

		ния к изменяемым условиям эксплуатации		
	умеет (продвинутой)	Проектировать программное обеспечение, имеющее встроенные средства адаптации к изменяемым условиям эксплуатации	Умение проектировать многослойные программные системы	Не менее 3 слов
	владеет (высокий)	Навыкам и работы в коллективе разработчиков при проектировании программного обеспечения, имеющего встроенные средства адаптации к изменяемым условиям эксплуатации	Способность распределить по ролям коллектив разработчиков	Не менее 10 разработчиков и 5 ролей

Промежуточная аттестация студентов. Промежуточная аттестация студентов проводится в соответствии с локальными нормативными актами ДВФУ и является обязательной.

По дисциплине предусмотрен экзамен, который проводится в устной форме.

Вопросы к экзамену по дисциплине

1. В каких условиях лучше всего применять итеративно-инкрементную модель жизненного цикла ПО и почему.
2. Какой проектной документацией принято сопровождать процесс разработки программного обеспечения.
3. Опишите функции участников инспекции на каждом этапе.
4. Зачем необходимо собирать метрики по результатам инспекций и какие метрики существуют.
5. Опишите состояния, в которых может пребывать подзадача в процессе коллективной разработки программного продукта.
6. Для чего служит система CLEARDDTS.
7. Какие существуют проектные виды деятельности.
8. Назовите этапы регистрации затрат рабочего времени.
9. В чем состоит методика GQ(I)M.
10. Приведите примеры и описание метрик качества продуктов.
11. Опишите сущность модели CMM.
12. Какие ключевые виды деятельности должны иметь место в компании третьего уровня.
13. Зачем необходимы стандарты кодирования.
14. Приведите пример правил и рекомендаций по оформлению заголовков файлов и функций.
15. Опишите основные принципы тестирования программного средства.
16. Основная идея восходящего тестирования, достоинства и недостатки.
17. Что включает в себя стандартный процесс разработки ПО согласно современным парадигмам?
18. Назовите модели жизненных циклов разработки ПО, опишите подробно (основной принцип, достоинства и недостатки) любой из них.
19. Перечислите роли участников инспекции рабочих продуктов и опишите их, напишите функции одного из них на каждом этапе инспекции.

- 20.Опишите способ оценивания инспекций и приведите пример оценивания.
- 21.Опишите систему ClearDdts в 5-8 предложениях.
- 22.Какие понятия лежат в основе технологии Workflow.
- 23.Назовите 10 проектных видов деятельности.
- 24.Для каждого вида деятельности перечислите набор соответствующих им действий.
- 25.Приведите 3 примера метрик эффективности процесса производства.
- 26.Приведите 3 примера метрик качества продуктов.
- 27.Опишите (вкратции) процесс аттестации по модели СММ.
- 28.Опишите сущность модели СММ.
- 29.Напишите пример требования написания на любом языке.
- 30.Назовите наиболее известный стандарт программирования и причину его введения.
- 31.Назовите главный принцип тестирования. Перечислите фазы тестирования.
- 32.Что включает в себя комплексная отладка программного средства.
- 33.Опишите 5 классов проектов и сопровождающую их проектную документацию.
- 34.Опишите модели жизненных циклов разработки ПО Водопад и Водопад с перекрытиями. В чем различия.
- 35.Что такое инспекция.
- 36.Опишите этапы планирования и завершения инспекции в процедуре организации и проведения формальной инспекции.
- 37.Выскажите свое мнение о системе отслеживания дефектов CLEARDDTS
- 38.Какие понятие вложены в основу регистрации времени.
- 39.Перечислите проектные виды деятельности (не менее 8).
- 40.Нарисуйте схему иерархии методики GQ(I)M.
- 41.Примеры метрик качества продуктов.
- 42.Перечень КРА третьего уровня.
- 43.Что такое практики. Перечислите некоторые из них (не менее 3).
- 44.Для чего нужны общие и корпоративные правила и рекомендации.
- 45.Придумайте требование для ПО.
- 46.Каков главный принцип тестирования.
47. Опишите вторую задачу тестирования

Образец экзаменационного билета



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«Дальневосточный федеральный университет»

(ДВФУ)

Школа ШЕН

ОП 09.04.04 программная инженерия (магистратура)
Шифр, наименование направления подготовки (специальности)

Дисциплина Методология научных исследований)

Форма обучения очная

Семестр осенний 2018-2019 учебного года
осенний, весенний

Реализующая кафедра ПММУиПО

Экзаменационный билет №

- Опишите 5 классов проектов и сопровождающую их проектную документацию
-

Заведующий кафедрой

Критерии выставления оценки студенту на экзамене

Баллы (рейтингово й оценки)	Оценка зачета/ экзамена (стандартная)	Требования к сформированным компетенциям
86-100	«зачтено»/ «отлично»	Оценка «отлично» выставляется студенту, если он глубоко и прочно усвоил программный материал, исчерпывающе, последовательно, четко и логически стройно его излагает, умеет тесно увязывать теорию с практикой, свободно справляется с задачами, вопросами и другими видами применения знаний, причем не затрудняется с ответом при видоизменении заданий, правильно обосновывает принятое решение, владеет разносторонними навыками и приемами выполнения практических задач.
76-85	«зачтено»/ «хорошо»	Оценка «хорошо» выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос, правильно применяет теоретические положения при решении практических вопросов и задач, владеет необходимыми навыками и приемами их выполнения.
61-75	«зачтено»/ «удовлетворительно»	Оценка «удовлетворительно» выставляется студенту, если он имеет знания только основного материала, но не усвоил его деталей, допускает неточности, недостаточно правильные формулировки, нарушения логической последовательности в изложении программного материала, испытывает затруднения при выполнении практических работ.
0-60	«не зачтено»/ «неудовлетворительно»	Оценка «неудовлетворительно» выставляется студенту, который не знает значительной части программного материала, допускает существенные ошибки, неуверенно, с большими затруднениями выполняет практические работы. Как правило, оценка «неудовлетворительно» ставится студентам, которые не могут продолжить обучение без дополнительных занятий по соответствующей дисциплине.

Критерии оценки проектов

- 100-86 баллов выставляется, если студент/группа точно определили содержание и составляющие части задания, умеют аргументированно отвечать на вопросы, связанные с заданием. Продемонстрировано знание и владение навыками самостоятельной исследовательской работы по теме. Фактических ошибок, связанных с пониманием проблемы, нет.

- 85-76 - баллов - работа студента/группы характеризуется смысловой цельностью, связностью и последовательностью изложения; допущено не более 1 ошибки при объяснении смысла или содержания проблемы. Продемонстрированы исследовательские умения и навыки. Фактических ошибок, связанных с пониманием проблемы, нет.

- 75-61 балл – проведен достаточно самостоятельный анализ основных этапов и смысловых составляющих проблемы; понимание базовых основ и теоретического обоснования выбранной темы. Привлечены основные источники по рассматриваемой теме. Допущено не более 2 ошибок в смысле или содержании проблемы

- 60-50 баллов - если работа представляет собой пересказанный или полностью переписанный исходный текст без каких бы то ни было комментариев, анализа. Не раскрыта структура и теоретическая составляющая темы. Допущено три или более трех ошибок смыслового содержания раскрываемой проблемы

Шкала оценивания

Менее 60 баллов	незачтено	неудовлетворительно
От 61 до 75 баллов	зачтено	удовлетворительно
От 76 до 85 баллов	зачтено	хорошо
От 86 до 100 баллов	зачтено	отлично

Текущий контроль

Текущий контроль предполагает систематическую проверку усвоения учебного материала, сформированности компетенций или их элементов, регулярно осуществляемую на протяжении изучения дисциплины, в соответствии с ее рабочей программой.

Состоит в проверке правильности выполнения заданий по самостоятельной работе. Задание зачтено, если нет ошибок. По текущим ошибкам даются пояснения.

Тесты предназначены для проверки знаний по компетенциям. Проверка достижения умений и навыков по компетенциям проверяется выполнением практических работ и курсовой работы.

Примерные тесты для проверки сформированности компетенций

ОК-2 готовностью проявлять качества лидера и организовать работу коллектива, владеть эффективными технологиями решения профессиональных проблем	Специфику работы коллектива разработчиков программного обеспечения
1. Классическая технология разработки программного обеспечения подразумевает разделение коллектива на рабочие группы под неформальным руководством тим-лидера численностью	ответы а. 3-7 человек б. не более 3 человек в. не менее 10 человек
2. Технология разработки Agile предполагает	ответы а. многочисленные команды разработчиков под руководством менеджера б. наличие отдельного коллектива тестировщиков в. самоорганизующиеся команды разработчиков без жесткого распределения ролей и задач

ОК-3 умением работать в проектных междисциплинарных командах, в том числе в качестве руководителя	Подходы к формулировке требований к программному обеспечению
1. Модель зрелости организации СММ предполагает	ответы а. разработку ПО на основе управления требованиями б. отсутствие необходимости создавать требования в. постоянное изменение требований по ходу разработки
2. Технология разработки Agile предполагает	ответы а. разработку ПО на основе управления требованиями б. отсутствие необходимости создавать требования в. неизменность требований по ходу разработки

ПК-6 пониманием существующих подходов к верификации моделей программного обеспечения	Методы верификации моделей программного обеспечения
1. Верификация разработанного ПО проходит	ответы а. только на этапе тестирования б. только на этапе интеграции в. на каждом этапе разработки всех рабочих продуктов
2. Наиболее удачным и желательным	ответы

случае выявления ошибки в ПО является	<p>a. её выявление на этапе тестирования</p> <p>б. выявление на том же этапе, когда она была допущена</p> <p>в. выявление на следующем этапе, после того как она была допущена</p>
---------------------------------------	---

ПК-11 способность проектировать сетевые службы	Особенности архитектуры сетевых служб
1. Транзакции в Java persistence API имеют тип	<p>ответы</p> <p>a. RESOURCE_LOCAL</p> <p>b. JTA transactions</p> <p>c. NULL</p> <p>d. LOCAL</p>

ПК-14 способностью руководить коллективом разработчиков при разработке проектов информационных систем для автоматизации профессиональной деятельности	Подходы к организации работы коллектива разработчиков
1. автоматизированное тестирование применяется на этапе	<p>ответы</p> <p>a. Тестирования и интеграции</p> <p>б. Разработки требований</p> <p>в. Планирования проекта</p>
2. автоматизированное тестирование подразумевает	<p>ответы</p> <p>a. использование специальных программных или программно-аппаратных систем TestBench</p> <p>б. автоматизацию разработки тестов</p> <p>в. автоматизацию инспекций рабочих продуктов</p>

ПК-15 способностью проектировать программное обеспечение, имеющее встроенные средства адаптации к изменяемым условиям эксплуатации	Основные приёмы, связанные с адаптацией программного обеспечения к изменяемым условиям эксплуатации
Организация процесса обучения сотрудников предписывается	<p>ответы</p> <p>a. третьим уровнем модели CMM</p> <p>б. вторым уровнем модели CMM</p> <p>в. первым уровнем модели CMM</p>
Популярный подход организации базы знаний при разработке ПО связан с использованием технологии	<p>ответы</p> <p>a. Power Point</p> <p>б. HTML5</p> <p>в. Wiki</p>