

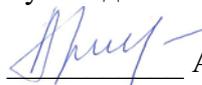


МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«Дальневосточный федеральный университет»
(ДФУ)

ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ (ШКОЛА)

«СОГЛАСОВАНО»

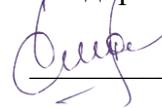
Руководитель ОП

 Артемьева И.Л.

« 26 » января 2022 г.

«Утверждаю»

И.о. директора департамента

 Смагин С.В.


« 26 » января 2022 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Методы коллективной разработки и верификации программного обеспечения

Направление подготовки – 09.04.04 «Программная инженерия»

(Разработка программно-информационных систем)

Форма подготовки очная

курс 2 семестр 3,4
лекции 18 час.
практические занятия 0 час.
лабораторные работы 42 час.
в том числе с использованием МАО лек. 0 / пр. 0 / лаб. 30 час.
всего часов аудиторной нагрузки – 60 час.
в том числе с использованием МАО – 30 час.
самостоятельная работа 84 час
в том числе на подготовку к экзамену 54 час.
контрольные работы (количество) не предусмотрены
курсовая работа / курсовой проект не предусмотрен
зачет не предусмотрен
экзамен 3, 4 семестр

Рабочая программа составлена в соответствии с требованиями Федерального государственного образовательного стандарта по направлению подготовки 09.04.04 **Программная инженерия**, утвержденного приказом Министерства образования и науки РФ от 19.09.2017 № 932 (с изменениями и дополнениями).

Рабочая программа обсуждена на заседании департамента программной инженерии и искусственного интеллекта, протокол № 1.1 от «24» января 2022 г.

И.о. директора департамента программной инженерии и искусственного интеллекта к.т.н. Смагин С.В.

Составитель (ли): профессор департамента ПИИИИ д.т.н. доцент Гриняк В.М.

Владивосток
2022

Оборотная сторона титульного листа РПД

I. **Рабочая программа пересмотрена на заседании департамента:**
Протокол от « ____ » _____ 20 ____ г. № ____
Директор департамента _____
(подпись) (И.О.Фамилия)

II. **Рабочая программа пересмотрена на заседании департамента:**
Протокол от « ____ » _____ 20 ____ г. № ____
Директор департамента _____
(подпись) (И.О.Фамилия)

ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

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 «Вычислительные системы, сети и низкоуровневое программирование» у обучающихся должны быть сформированы предварительные компетенции:

- по проектированию и разработке программ,
- по поиску необходимой информации в интернет.

Универсальные компетенции выпускников и индикаторы их достижения:

Наименование категории (группы) универсальных компетенций	Код и наименование универсальной компетенции (результат освоения)	Код и наименование индикатора достижения компетенции
---	---	--

Командная работа и лидерство	УК-3 Способен организовывать и руководить работой команды, вырабатывая командную стратегию для достижения поставленной цели	УК-3.1 формирует стратегию командной работы на основе совместного обсуждения целей и направлений деятельности для их реализации УК-3.2 организует работу команды с учетом объективных условий (технология, внешние факторы, ограничения), индивидуальных особенностей поведения и возможностей членов команды УК-3.3 обеспечивает выполнение поставленных задач на основе мониторинга командной работы и своевременного реагирования на существенные отклонения
------------------------------	---	---

Код и наименование индикатора достижения компетенции	Наименование показателя оценивания (результата обучения по дисциплине)
УК-3.1 формирует стратегию командной работы на основе совместного обсуждения целей и направлений деятельности для их реализации	Знает принципы формирования команд разработчиков программного обеспечения
	Умеет исполнять основные роли участников команд разработчиков программного обеспечения.
	Владеет навыками исполнения ролей участников команд разработчиков программного обеспечения
УК-3.2 организует работу команды с учетом объективных условий (технология, внешние факторы, ограничения), индивидуальных особенностей поведения и возможностей членов команды	Знает принципы взаимодействия участников команд разработчиков программного обеспечения.
	Умеет использовать инструменты взаимодействия участников команд разработчиков программного обеспечения.
	Владеет навыками работы с инструментами взаимодействия участников команд разработчиков программного обеспечения.
УК-3.3 обеспечивает выполнение поставленных задач на основе мониторинга командной работы и своевременного реагирования на существенные отклонения	Знает принципы взаимодействия участников команд разработчиков программного обеспечения.
	Умеет использовать инструменты взаимодействия участников команд разработчиков программного обеспечения.
	Владеет навыками исполнения ролей участников команд разработчиков программного обеспечения

Профессиональные компетенции выпускников и индикаторы их достижения:

Тип задач	Код и наименование профессиональной компетенции (результат освоения)	Код и наименование индикатора достижения компетенции
научно-исследовательский	ПК-5 Способен выполнить постановку новых задач анализа и синтеза новых проектных решений	ПК-5.1 демонстрирует знание методов постановки новых задач анализа и синтеза новых проектных решений ПК-5.2 использует методы постановки новых задач анализа и синтеза новых

Тип задач	Код и наименование профессиональной компетенции (результат освоения)	Код и наименование индикатора достижения компетенции
		<p>проектных решений</p> <p>ПК5.3 применяет методы разработки постановок задач анализа и синтеза новых проектных решений, требуемых в профессиональной деятельности</p>
	<p>ПК-6 Способен использовать существующие подходы к верификации моделей программного обеспечения</p>	<p>ПК-6.1 демонстрирует знание методов верификации моделей программного обеспечения</p> <p>ПК-6.2 использует методы верификации моделей программного обеспечения</p> <p>ПК-6.3 применяет методы проведения верификации моделей программного обеспечения, требуемых в профессиональной деятельности</p>
проектный	<p>ПК-8. Способен организовать промышленное тестирование создаваемого программного обеспечения</p>	<p>ПК-8.1 демонстрирует знание методов организации промышленного тестирования создаваемого программного обеспечения</p> <p>ПК-8.2 использует методы организации промышленного тестирования создаваемого программного обеспечения</p> <p>ПК-8.3 применяет методы организации тестирования программных средств</p>

Код и наименование индикатора достижения компетенции	Наименование показателя оценивания (результата обучения по дисциплине)
<p>ПК-5.1 демонстрирует знание методов постановки новых задач анализа и синтеза новых проектных решений</p>	Знает современные технологии проектирования и производства программного обеспечения
	Умеет правильно выбрать технологию производства программного обеспечения для конкретного проекта
	Владеет навыками применения технологий производства программного обеспечения
<p>ПК-5.2 использует методы постановки новых задач анализа и синтеза новых проектных решений</p>	Знает современные технологии проектирования и производства программного обеспечения
	Умеет правильно выбрать технологию производства программного обеспечения для конкретного проекта
	Владеет навыками применения технологий производства программного обеспечения
<p>ПК-5.3 применяет методы разработки постановок задач анализа и синтеза новых проектных решений, требуемых в профессиональной деятельности</p>	Знает современные технологии реализации, оценки качества и анализа эффективности программного обеспечения
	Умеет правильно выбрать технологию реализации, оценки качества и анализа эффективности программного обеспечения
	Владеет навыками применения технологий реализации, оценки качества и анализа эффективности программного обеспечения

Код и наименование индикатора достижения компетенции	Наименование показателя оценивания (результата обучения по дисциплине)
ПК-6.1 демонстрирует знание методов верификации моделей программного обеспечения	обеспечения
	Знает основные концептуальные положения функционального, логического, объектно-ориентированного и визуального направлений программирования
	Умеет создавать программы в рамках функционального, логического, объектно-ориентированного и визуального направлений программирования
	Владеет навыками создания программ в рамках функционального, логического, объектно-ориентированного и визуального направлений программирования
ПК-6.2 использует методы верификации моделей программного обеспечения	Знает основные концептуальные положения функционального, логического, объектно-ориентированного и визуального направлений программирования
	Умеет создавать программы в рамках функционального, логического, объектно-ориентированного и визуального направлений программирования
	Владеет навыками создания программ в рамках функционального, логического, объектно-ориентированного и визуального направлений программирования
ПК-6.3 применяет методы проведения верификации моделей программного обеспечения, требуемых в профессиональной деятельности	Знает основные концептуальные положения функционального, логического, объектно-ориентированного и визуального направлений программирования
	Умеет создавать программы в рамках функционального, логического, объектно-ориентированного и визуального направлений программирования
	Владеет навыками создания программных систем в рамках функционального, логического, объектно-ориентированного и визуального направлений программирования
ПК-8.1 демонстрирует знание методов организации промышленного тестирования создаваемого программного обеспечения	Знает методы организации работы в коллективах разработчиков ПО и методы сопровождения ПО.
	Умеет использовать методы организации работы в коллективах разработчиков ПО и методы сопровождения ПО.
	Владеет навыками коллективной разработки и использования ПО
ПК-8.2 использует методы организации промышленного тестирования создаваемого программного обеспечения	Знает методы организации работы в профессиональной деятельности.
	Умеет использовать методы организации работы в профессиональной деятельности.
	Владеет навыками организации работы в профессиональной деятельности.
ПК-8.3 применяет методы организации тестирования программных средств	Знает методы разработки и сопровождения ПО при создании программных средств
	Умеет применять методы разработки и сопровождения ПО при создании программных средств
	Владеет навыками создания программных средств и их сопровождения

Общая трудоемкость дисциплины составляет 4 зачётных единиц (144 академических часа).

(1 зачетная единица соответствует 36 академическим часам)

Видами учебных занятий и работы обучающегося по дисциплине являются:

Обозначение	Виды учебных занятий и работы обучающегося
Лек	Лекционные занятия
Лаб	Лабораторные работы
СР	Самостоятельная работа обучающегося в период теоретического обучения
Контроль	Самостоятельная работа обучающегося и контактная работа обучающегося с преподавателем в период промежуточной аттестации

Структура дисциплины:

Форма обучения – очная.

№	Наименование раздела дисциплины	Семестр	Количество часов по видам учебных занятий и работы обучающегося						Формы промежуточной аттестации
			Лек	Лаб	Пр	ОК	СР	Контроль	
1	Раздел I. Структура и анализ бизнес-процессов разработки программного обеспечения.	3	18	18	-	-	84	27	УК-3, ОПК-8, ПК-6, ПК-7, ПК-9
2	Раздел II. Тестирование и отладка программного средства	4		24					
Итого:			18	28		-	124	27	

I. СТРУКТУРА И СОДЕРЖАНИЕ ТЕОРЕТИЧЕСКОЙ ЧАСТИ КУРСА

Трудоемкость теоретической части курса 18 час.

Семестр 3 (18 часов)

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 И САМОСТОЯТЕЛЬНОЙ РАБОТЫ**

Лабораторные работы (32 час.)

Лабораторная работа №1. Организация коллективной работы разработчиков (4 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 (4 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 (4 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 (4 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 (4 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 (3 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 (3 часа)

1. Разработка системы метрик для оценки качества программного продукта
2. Сбор метрик

Лабораторная работа №8. Метрики эффективности процесса разработки программной системы (3 часа)

1. Разработка системы метрик для эффективности процесса разработки программного продукта
2. Сбор метрик

Лабораторная работа №9. Выработка рекомендаций по изменению процесса разработки (3 часа)

1. Анализ метрик для оценки качества программного продукта и эффективности процесса разработки
- Формулировка рекомендаций по трансформации процесса разработки по результатам программы измерений

Ш. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ

Учебно-методическое обеспечение самостоятельной работы обучающихся по дисциплине включает в себя:

- план-график выполнения самостоятельной работы по дисциплине, в том числе примерные нормы времени на выполнение по каждому заданию;
- требования к представлению и оформлению результатов самостоятельной работы;
- критерии оценки выполнения самостоятельной работы.

План-график выполнения самостоятельной работы по дисциплине

№ п/п	Дата/сроки выполнения	Вид самостоятельной работы	Примерные нормы времени на выполнение	Форма контроля
3 семестр				
1	1-4 неделя	Software Project Plan (SPP) – календарный план работы над проектом	2 час	Собеседование
2	5-9 неделя	Marketing Requirements Specification (MRS) - маркетинговые требования к проекту в виде презентации рекламного характера	2 час	Проект
3	10-14 неделя	Software Requirements Specification (SRS) - технические требования к продукту	2 час	Проект
4	15-18 неделя	User Manual - руководство оператора (пользователя).	3 час	Проект
5	18	Подготовка к экзамену	27 часов	экзамен
4 семестр				
5	1-3 неделя	Software Design Specification (SDS) - документы дизайна, раскрывающие архитектуру программного продукта и то КАК именно будут реализованы те или иные требования.	4 час	Проект
6	4-7 неделя	Code Issue List - список задач, выполняемых на этапе кодирования.	4 час	проект
7	8-11 неделя	Software Test Plan - докумен, включающий в себя спецификации тестов и процедур тестирования, в том числе список тестов.	4 час	Проект

8	12-15 неделя	Requirements Tracaebility Matrix - матрица покрытия тестами требований.	4 час	Проект
9	16-18 неделя	Test Logs - документ, который заполняется в ходе тестирования и содержит результаты выполнения тестов.	5 час	Проект
	18 неделя	Подготовка к экзамену	27 час	экзамен
	ИТОГО		84 час	

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).

Методические указания к самостоятельной работе студентов

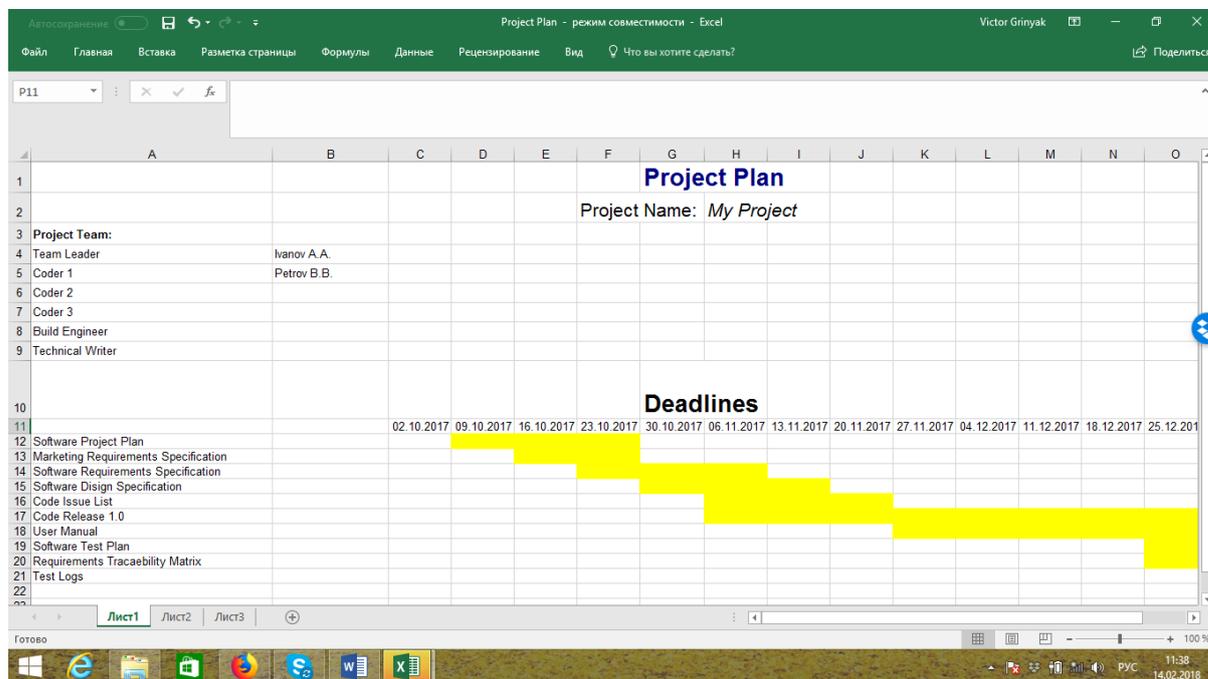
Самостоятельная работа включает в себя разработку итогового проекта, параллельно с лабораторными работами по дисциплине. В начале семестра студенты разбиваются на команды (по 3-5 человек в каждой). Каждая команда придумывает название своего проекта и определяется с составом (тим-лидер, кодировщики, технический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

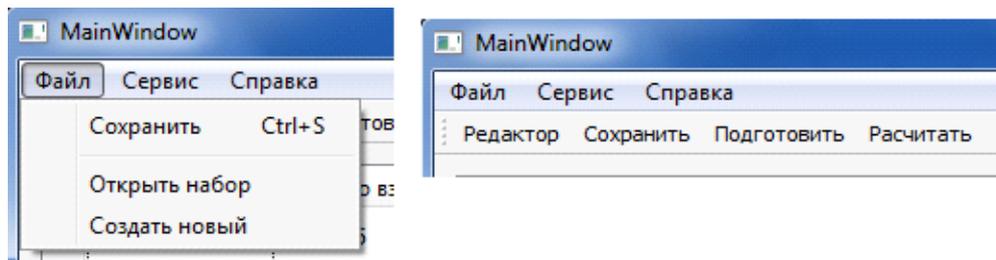
Программный продукт *название* состоит из следующих подсистем:

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_UL_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 Design 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						
REQ_UI_002						
REQ_CX_001						
REQ_CX_002						

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 отчетов по самостоятельной работе

– 100-86 баллов выставляется, если содержание и составляющие части соответствуют выданному заданию. Продемонстрировано владение навыками подготовки документа по теме. Фактических ошибок, связанных с пониманием структуры и содержания задания нет.

– 85-76 - баллов выставляется, если при выполнении задания допущено не более одной ошибки. Продемонстрировано владение навыками подготовки документа по теме. Фактических ошибок, связанных с пониманием структуры и содержания задания нет.

– 75-61 балл выставляется, если при выполнении задания допущено не более двух ошибок. Продемонстрировано навыки подготовки документа по теме. Допущено не более 2 ошибок, связанных с пониманием структуры и содержания задания.

60-50 баллов - если структура и содержание задания не соответствуют требуемым

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

Рекомендации по работе с литературой

Для более эффективного освоения и усвоения материала рекомендуется ознакомиться с теоретическим материалом по той или иной теме до проведения лабораторного занятия. Всю учебную литературу желательно изучать «под конспект».

Цель написания конспекта по дисциплине – сформировать навыки по поиску, отбору, анализу и формулированию учебного материала.

Работу с теоретическим материалом по теме можно проводить по следующей схеме:

- название темы;
- цели и задачи изучения темы;
- основные вопросы темы;
- характеристика основных понятий и определений, необходимых для усвоения данной темы;
- краткие выводы, ориентирующие на определенную совокупность сведений, основных идей, ключевых положений, систему доказательств, которые необходимо усвоить.

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

Подготовка к лабораторным работам

Подготовку к каждой лабораторной работе или к практическому занятию каждый студент должен начать с изучения теоретического материала и ознакомления с планом, который отражает содержание предложенной темы. Все новые понятия по изучаемой теме необходимо выучить наизусть и внести в глоссарий, который целесообразно вести с самого начала изучения курса. Результат такой работы должен проявиться в способности студента свободно ответить на теоретические вопросы по теме задания, правильном его выполнении.

В процессе выполнения лабораторной работы или практического задания студент должен создать требуемый документ с помощью предлагаемого программного средства и выполнить требуемые в задании операции. Задание по лабораторной или практической работе содержит методические указания по подготовке документа, который должен быть получен в результате выполнения работы. При подготовке следует их внимательно прочесть.

Критерии оценки лабораторных работ

- 100-86 - выполнены все задания лабораторной работы, студент четко и без ошибок ответил на все контрольные вопросы.
- 85-76 - выполнены все задания лабораторной работы; студент ответил на все контрольные вопросы с замечаниями.
- 75-61 выполнены все задания лабораторной работы с замечаниями; студент ответил на все контрольные вопросы с замечаниями.

- 60-50 баллов - студент не выполнил или выполнил неправильно задания лабораторной работы; студент ответил на контрольные вопросы с ошибками или не ответил на контрольные вопросы.

IV. КОНТРОЛЬ ДОСТИЖЕНИЯ ЦЕЛЕЙ КУРСА

№ п/п	Контролируемые модули/ разделы / темы дисциплины	Код индикатора достижения компетенции	Результаты обучения	Оценочные средства - наименование	
				текущий контроль	промежуточная аттестация
1	Раздел 1. Структура и анализ бизнес-процессов разработки программного обеспечения.	ПК-5.1 демонстрирует знание методов постановки новых задач анализа и синтеза новых проектных решений.	Знает современные технологии проектирования и производства программного обеспечения Умеет правильно выбрать технологию производства программного обеспечения для конкретного проекта	ПР1 лабораторные работы и задания для самостоятельного выполнения	Зачет, вопросы № 1-26
		Владеет навыками применения технологий производства программного обеспечения	Знает современные технологии проектирования и производства программного обеспечения		
		ПК-5.2 использует методы постановки новых задач анализа и синтеза новых проектных решений	Умеет правильно выбрать технологию производства программного обеспечения для		

	конкретного проекта
	Владеет навыками применения технологий производства программного обеспечения
	Знает современные технологии реализации, оценки качества и анализа эффективности программного обеспечения
ПК5.3 применяет методы разработки постановок задач анализа и синтеза новых проектных решений, требуемых в профессиональной деятельности	Умеет правильно выбрать технологию реализации, оценки качества и анализа эффективности программного обеспечения
	Владеет навыками применения технологий реализации, оценки качества и анализа эффективности программного обеспечения
	Знает основные концептуальные положения функционального, логического, объектно-ориентированного и визуального направлений программирования
ПК-6.1 демонстрирует знание методов верификации	Умеет создавать программы в рамках

		<p>моделей программного обеспечения</p>	<p>функционального, логического, объектно-ориентированного и визуального направлений программирования</p>	
			<p>Владеет навыками создания программ в рамках функционального, логического, объектно-ориентированного и визуального направлений программирования</p>	
			<p>Знает основные концептуальные положения функционального, логического, объектно-ориентированного и визуального направлений программирования</p>	
		<p>ПК-6.2 использует методы верификации моделей программного обеспечения.</p>	<p>Умеет создавать программы в рамках функционального, логического, объектно-ориентированного и визуального направлений программирования</p>	
			<p>Владеет навыками создания программ в рамках функционального, логического, объектно-ориентированно</p>	

			го и визуального направлений программирования		
			Знает основные концептуальные положения функционального, логического, объектно-ориентированного и визуального направлений программирования		
		ПК-6.3 применяет методы проведения верификации моделей программного обеспечения, требуемых в профессиональной деятельности	Умеет создавать программы в рамках функционального, логического, объектно-ориентированного и визуального направлений программирования		
			Владеет навыками создания программных систем в рамках функционального, логического, объектно-ориентированного и визуального направлений программирования		
			Знает современные технологии проектирования и производства программного обеспечения		
2	Раздел 2. Тестирование и отладка программного средства		ПК-8.1 демонстрирует знание методов организации промышленного тестирования создаваемого	Знает методы организации работы в коллективах разработчиков ПО и методы	ПР1 лабораторные работы и задания для самостоятел

		<p>программного обеспечения</p>	<p>сопровождения ПО.</p> <p>Умеет использовать методы организации работы в коллективах разработчиков ПО и методы сопровождения ПО.</p> <p>Владеет навыками коллективной разработки и использования ПО</p>	<p>ьного выполнения</p>	
		<p>ПК-8.2 использует методы организации промышленного тестирования создаваемого программного обеспечения</p>	<p>Знает методы организации работы в профессиональной деятельности.</p> <p>Умеет использовать методы организации работы в профессиональной деятельности.</p> <p>Владеет навыками организации работы в профессиональной деятельности.</p>		
		<p>ПК-8.3 применяет методы организации тестирования программных средств</p>	<p>Знает методы разработки и сопровождения ПО при создании программных средств</p> <p>Умеет применять методы разработки и сопровождения ПО при создании программных средств</p>		

Владеет навыками создания программных средств и их сопровождения

Типовые контрольные задания, методические материалы, определяющие процедуры оценивания знаний, умений и навыков и (или) опыта деятельности, а также качественные критерии оценивания, которые описывают уровень сформированности компетенций, представлены в разделе VIII.

V. СПИСОК УЧЕБНОЙ ЛИТЕРАТУРЫ И ИНФОРМАЦИОННО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Основная литература

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. 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>
3. 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

1. 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>
2. 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>

3. Браудэ Э. Технология разработки программного обеспечения, Издательский дом «Питер», 2004. - 656 с.
<http://lib.dvfu.ru:8080/lib/item?id=chamo:232388&theme=FEFU>
4. Иванова Г.С. Технология программирования. М.: Изд-во МГУ. 2002.
<http://lib.dvfu.ru:8080/lib/item?id=chamo:398553&theme=FEFU>
5. Жоголев Е.А. Технология программирования. М.: Научный мир. 2004. – 215 с. <http://lib.dvfu.ru:8080/lib/item?id=chamo:7891&theme=FEFU>
6. 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>
7. 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>
8. 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. Электронная библиотечная система ZNANIUM.COM
<http://znanium.com/>
2. Студенческая электронная библиотека «Консультант студента»
<http://www.studentlibrary.ru/cgi-bin/mb4x>
3. Научная библиотека ДВФУ. Электронный каталог
<http://lib.dvfu.ru:8080/>
4. <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.
5. <http://novtex.ru/> Журналы издательства Новые технологии
6. <https://github.com/> Система контроля версий Git

Профессиональные базы данных и информационные справочные системы

1. База данных Scopus <http://www.scopus.com/home.url>
2. База данных Web of Science <http://apps.webofknowledge.com/>

3. Научная электронная библиотека <https://www.elibrary.ru/>
4. Общероссийский математический портал Math-Net.Ru <http://www.mathnet.ru>
5. Электронная библиотека диссертаций Российской государственной библиотеки <http://diss.rsl.ru/>
6. Электронная библиотека Европейского математического общества <https://www.emis.de/>
7. Электронные базы данных EBSCO <http://search.ebscohost.com/>

Перечень информационных технологий и программного обеспечения

При осуществлении образовательного процесса студентами и профессорско-преподавательским составом используется следующее программное обеспечение:

1. Microsoft Office (Access, Excel, PowerPoint, Word и т. д).
2. Open Office.
3. Интегрированные среды разработчика Eclipse, NetBeans, Visual Studio.
4. Система контроля версий Git.

При осуществлении образовательного процесса студентами и профессорско-преподавательским составом используются следующие информационно-справочные системы:

1. Научная электронная библиотека eLIBRARY.
2. Электронно-библиотечная система издательства «Лань».
3. Электронная библиотека "Консультант студента".
4. Электронно-библиотечная система IPRbooks.
5. Информационная система "ЕДИНОЕ ОКНО доступа к образовательным ресурсам".
6. Доступ к электронному заказу книг в библиотеке ДВФУ, доступ к нормативным документам ДВФУ, расписанию, рассылке писем.

Лабораторные занятия проводятся в специализированном компьютерном классе.

VI. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Дисциплина изучается в следующих организационных формах: лекционное занятие; лабораторное занятие; самостоятельное изучение теоретического материала; самостоятельное выполнение индивидуального проекта; индивидуальные и групповые консультации.

Основной формой самостоятельной работы студента является изучение конспекта лекций, их дополнение рекомендованной литературой, выполнение проекта, а также активная работа на лабораторных занятиях.

К прослушиванию лекции следует готовиться, для этого необходимо знать программу курса и рекомендованную литературу. Тогда в процессе лекции легче отделить главное от второстепенного, легче сориентироваться: что записать, что самостоятельно проработать, что является трудным для понимания, а что легко усвоить.

Контроль за выполнением самостоятельной работы студента производится в виде контроля каждого этапа работы, отраженного в документации и защиты проекта.

Студент должен планировать график самостоятельной работы по дисциплине и придерживаться его.

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

REQUIREMENTS - 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 Tracaebility 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.

Работа с литературой. Рекомендуется использовать различные возможности работы с литературой: фонды научной библиотеки ДВФУ и электронные библиотеки (<http://www.dvfu.ru/library/>), а также доступные для использования другие научно-библиотечные системы.

Подготовка к экзамену/зачету. К сдаче экзамена/зачета допускаются обучающиеся, выполнившие все задания (лабораторные, самостоятельные), предусмотренные учебной программой дисциплины, посетившие не менее 85% аудиторных занятий.

VII. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Перечень материально-технического и программного обеспечения дисциплины приведен в таблице.

Материально-техническое и программное обеспечение дисциплины

Наименование специальных помещений и помещений для самостоятельной работы	Оснащенность специальных помещений и помещений для самостоятельной работы	Перечень лицензионного программного обеспечения. Реквизиты подтверждающего документа
690922, Приморский край, г. Владивосток, остров Русский, полуостров Саперный, поселок Аякс, 10, корпус D, ауд. D 733,733а. Учебная аудитория для проведения занятий лекционного типа, групповых и	Помещение укомплектовано специализированной учебной мебелью (посадочных мест – 13) Оборудование: ЖК-панель 47", Full HD, LG M4716 CCBA – 1 шт. Доска аудиторная, Моноблок Lenovo C360G-i34164G500UDK с лицензионными программами	1С Предприятия8 (8.2), 7-Zip, ABBYY Lingvo12,Alice 3, Anaconda3,Autodesk,CodeBlocks,CorelDRAW X7,Dia,Directum4.8,DosBox-0.74,Farmanager,Firebird 2.5,FlameRobin,Foxit Reader,Free Pascal,Geany,Ghostscript,Git,Greenfoot,gsview,Inscapе0.91,Java,Java development Kit,Kaspersky,Lazarus,LibreOffice4.4,MatLab R2017b,Maxima 5.37.2,Microsoft Expression,Microsoft Office 2013,Microsoft Silverlight,Microsoft Silverlight 5SDK-русский,MicrosoftSistem Center,Microsoft Visial Studio 2012,MikTeX2.9,MySQL,NetBeans,Notepad++,Oracle VM VirtualBox,PascalABC.NET,PostgreSQL 9.4,PTC Mathcad,Putty,PyQt GPL v5.4.1 for Pythonv 3.4,Python2.7(3.4,3.6),QGIS Brighton,RStudio,SAM CoDeC Pack,SharePoint,Strawberry Perl,Tecnomatix,TeXnicCenter,TortoiseSVN,Unity2017.3.1f1,Veusz,Vim 8.1,Visual Paradigm CE,Visual Studio2013,Windows Kits,Windows Phone SDK8.1,Xilinx Design ToolsAcrobat ReaderDC,AdobeBridge CS3,AdobeDeviceCentralCS3,Adobe ExtendScript Toolkit 2,Adobe Photoshpe CS3,DVD-студия Windows,GoogleChrome,Internet Explorer,ITMOproctor,Mozilla Firefox, Visual Studio Installer,Windows

индивидуальных консультаций, текущего контроля и промежуточной аттестации	Microsoft Office 2013(13 шт.) и аудиовизуальным и средствами проектор Panasonic DLPPjectorPT-D2110XE	Media Center, WinSCP,
---	--	-----------------------

В целях обеспечения специальных условий обучения инвалидов и лиц с ограниченными возможностями здоровья в ДВФУ все здания оборудованы пандусами, лифтами, подъемниками, специализированными местами, оснащёнными туалетными комнатами, табличками информационно-навигационной поддержки.

VIII. ФОНДЫ ОЦЕНОЧНЫХ СРЕДСТВ

Для дисциплины «Вычислительные системы, сети и низкоуровневое программирование» используются следующие оценочные средства:

Устный опрос:

1. Собеседование (УО-1)
2. Презентация / сообщение (УО-3)

Письменные работы:

1. Лабораторная работа (ПР-6)
2. Тест (ПР-1)

Устный опрос

Устный опрос позволяет оценить знания и кругозор студента, умение логически построить ответ, владение монологической речью и иные коммуникативные навыки.

Обучающая функция состоит в выявлении деталей, которые по каким-то причинам оказались недостаточно осмысленными в ходе учебных занятий и при подготовке к зачёту.

Собеседование (УО-1) – средство контроля, организованное как специальная беседа преподавателя с обучающимся на темы, связанные с изучаемой дисциплиной, и рассчитанное на выяснение объема знаний обучающегося по определенному разделу, теме, проблеме и т.п.

Презентация / сообщение (УО-3) – продукт самостоятельной работы обучающегося, представляющий собой публичное выступление по

представлению полученных результатов решения определенной учебно-практической, учебно-исследовательской или научной темы.

Письменные работы

Письменная проверка используется во всех видах контроля и осуществляется как в аудиторной, так и во внеаудиторной работе.

Лабораторная работа (ПР-6) – средство для закрепления и практического освоения материала по определенному разделу.

Тест (ПР-1) - Система стандартизированных заданий, позволяющая автоматизировать процедуру измерения уровня знаний и умений обучающегося.

Методические рекомендации, определяющие процедуры оценивания результатов освоения дисциплины

Оценочные средства для промежуточной аттестации

Промежуточная аттестация студентов проводится в соответствии с локальными нормативными актами ДВФУ и является обязательной. Форма отчётности по дисциплине – экзамен (3, 4 семестр). Оценка на экзамене выставляется согласно рейтинга, набранного студентом по итогам выполнения лабораторных работ.

Критерии выставления оценки студенту

Баллы (рейтинговой 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.
--	--	---

Вопросы к экзамену

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 СММ.
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 (вкратце) 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

Оценочные средства для текущей аттестации

Текущая аттестация студентов по дисциплине проводится в соответствии с локальными нормативными актами ДВФУ и является обязательной.

Текущая аттестация проводится в форме контрольных мероприятий (собеседования, презентации, лабораторных работ, тестов) по оцениванию фактических результатов обучения студентов.

Объектами оценивания выступают:

- степень усвоения теоретических знаний - оценивается в форме собеседования и тестирования;
- уровень овладения практическими умениями и навыками – оценивается в форме защиты лабораторных работ.

Составляется календарный план контрольных мероприятий по дисциплине. Оценка посещаемости, активности обучающихся на занятиях, своевременность выполнения различных видов заданий ведётся на основе журнала, который ведёт преподаватель в течение учебного семестра.

Примерные тесты для проверки сформированности компетенций

ПК-6 Способен выполнить постановку новых задач анализа и синтеза новых проектных решений	демонстрирует знание методов постановки новых задач анализа и синтеза новых проектных решений
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

ПК-7 Способен использовать существующие подходы к верификации моделей программного обеспечения	демонстрирует знание методов верификации моделей программного обеспечения.
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 Верификация разработанного ПО проходит	ответы а. только на этапе тестирования б. только на этапе интеграции в. на каждом этапе разработки всех рабочих продуктов

ПК-9 Способен организовать промышленное тестирование создаваемого программного обеспечения	демонстрирует знание методов организации промышленного тестирования создаваемого программного обеспечения.
1. Наиболее удачным и желательным случаем выявления ошибки в ПО является	ответы а. её выявление на этапе тестирования б. выявление на том же этапе, когда она была допущена в. выявление на следующем этапе, после того как она была допущена
2 Технология разработки Agile предполагает	ответы а. разработку ПО на основе управления требованиями б. отсутствие необходимости создавать требования в. неизменность требований по ходу разработки

Примерные тесты для текущей аттестации

1. Классическая технология разработки программного обеспечения подразумевает разделение коллектива на рабочие группы под неформальным руководством тим-лидера численностью

а. **3-7 человек**

б. не более 3 человек

в. не менее 10 человек

2. Технология разработки Agile предполагает

а. многочисленные команды разработчиков под руководством менеджера

б. наличие отдельного коллектива тестировщиков

в. самоорганизующиеся команды разработчиков без жесткого распределения ролей и задач

3. Модель зрелости организации СММ предполагает

а. разработку ПО на основе управления требованиями

- б. отсутствие необходимости создавать требования
 - в. постоянное изменение требований по ходу разработки
4. Технология разработки Agile предполагает
- а. разработку ПО на основе управления требованиями
 - б. отсутствие необходимости создавать требования**
 - в. неизменность требований по ходу разработки
5. Верификация разработанного ПО проходит
- а. только на этапе тестирования
 - б. только на этапе интеграции
 - в. на каждом этапе разработки всех рабочих продуктов**
6. Наиболее удачным и желательным случаем выявления ошибки в ПО является
- а. её выявление на этапе тестирования
 - б. выявление на том же этапе, когда она была допущена**
 - в. выявление на следующем этапе, после того как она была допущена
7. Основное свойство требований к программному обеспечению
- а. Верифицируемость**
 - б. Краткость
 - с. понятность
 - д. удобочитаемость
8. автоматизированное тестирование применяется на этапе
- а. Тестирования и интеграции**
 - б. Разработки требований
 - в. Планирования проекта
9. автоматизированное тестирование подразумевает
- а. использование специальных программных или программно-аппаратных систем TestBench**
 - б. автоматизацию разработки тестов
 - в. автоматизацию инспекций рабочих продуктов
10. Организация процесса обучения сотрудников предписывается
- а. третьим уровнем модели СММ**
 - б. вторым уровнем модели СММ

в. первым уровнем модели СММ

11. Популярный подход организации базы знаний при разработке ПО связан с использованием технологии

а. Power Point

б. HTML5

в. **Wiki**