Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Максимов Алексей Борисович
Должность: директор департамента по образовательной политике
Дата подписания: 30.10.2023 12:57:21
Уникальный программный ключ:
8db180d1a3f02ac9e60521a5672742735c18b1d6

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение высшего образования
«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
Факультет информационных технологий

УТВЕРЖДАЮ
Декан факультета
«Информационные технологии»
_____/Д.Г.Демидов/
«16» мая 2022

Рабочая программа дисциплины

«Программная инженерия»

Направление подготовки/специальность
09.03.03 Прикладная информатика

Профиль/специализация
«Информационные технологии управления бизнесом»

Квалификация
бакалавр

Формы обучения
очная

Москва, 2022 г.

Разработчик(и):

к.т.н., доцент /В.Ю. Верещагин /

Согласовано:
Заведующий кафедрой «Инфокогнитивные технологии»,
к.т.н., доцент / Е.А. Пухова /



1 Цели, задачи и планируемые результаты обучения по дисциплине

К основным целям освоения дисциплины «Программная инженерия» следует отнести:

формирование знаний о современных принципах, методах и средствах разработки программных систем;

подготовка студентов к деятельности в соответствии с квалификационной характеристикой бакалавра по направлению, в том числе формирование умений по выявлению необходимых усовершенствований и разработке нового программного обеспечения.

К основным задачам освоения дисциплины «Программная инженерия» следует отнести:

освоение методологии, анализа и выбора принципов и методов создания сложного программного обеспечения.

Обучение по дисциплине направлено на формирование у обучающихся следующих компетенций:

Код и наименование компетенций	Индикаторы достижения компетенции
ПК-3. Способен разрабатывать требования и проектировать программное обеспечение	ИПК-3.1. Знает возможности существующей программно-технической архитектуры; возможности современных и перспективных средств разработки программных продуктов, технических средств; методологии разработки программного обеспечения и технологии программирования; методологии и технологии проектирования и использования баз данных; языки формализации функциональных спецификаций; методы и приемы формализации задач; методы и средства проектирования программного обеспечения; методы и средства проектирования программных интерфейсов; методы и средства проектирования баз данных; принципы построения архитектуры программного обеспечения и виды архитектуры программного обеспечения; типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения; методы и средства проектирования программного обеспечения и баз данных; методы и средства проектирования программных интерфейсов. ИПК-3.2. Умеет проводить анализ исполнения требований; вырабатывать

	варианты реализации требований; проводить оценку и обоснование рекомендуемых решений; осуществлять коммуникации с заинтересованными сторонами; выбирать средства реализации требований к программному обеспечению; вырабатывать варианты реализации программного обеспечения; проводить оценку и обоснование рекомендуемых решений; осуществлять коммуникации с заинтересованными сторонами; использовать существующие типовые решения и шаблоны проектирования программного обеспечения; применять методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов; осуществлять коммуникации с заинтересованными сторонами. ИПК-3.3. Владеет современным инструментарием и средами разработки программного кода; современным инструментарием и средами проектирования программного кода, методами тестирования ПО.

2 Место дисциплины в структуре образовательной программы

Дисциплина относится к части, формируемой участниками образовательных отношений блока Б1.2 и междисциплинарно связана с поддерживающими дисциплинами: основы программирования, базы данных, и последующими дисциплинами: Архитектура автоматизированных систем.

3 Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 4 зачетных единицы (144 часа).

3.1 Виды учебной работы и трудоемкость для очной формы обучения

№ п/п	Вид учебной работы	Количество часов	Семестры	
			6	
1	**Аудиторные занятия**	**72**	72	
	В том числе:			
1.1	Лекции	72	72	
1.2	Семинарские/практические занятия			
1.3	Лабораторные занятия			
2	**Самостоятельная работа**	**72**	72	

3	**Промежуточная аттестация**			
	Курсовой проект			
	Дифференцированный зачет		Диф.зачет	
	Итого:	**144**	144	

3.2 Тематический план изучения дисциплины для очной формы обучения

3.2.1 Очная форма обучения

№ п/п	Разделы/темы дисциплины	Трудоемкость, час					
		Всего	Аудиторная работа				Самостоятельная работа
			Лекции	Семинарские/ практические занятия	Лабораторные занятия	Практическая подготовка	
1	Введение. Программная инженерия	9			5		4
2	Эволюция программного обеспечения	9			5		4
3	Проект программной системы	9			5		4
4	Риски программных систем и требования к ним	9			5		4
5	Инженерия качества	9			5		4
6	Программометрика	9			5		4
7	Тестирование программных систем	9			5		4
8	Программные дефекты	9			5		4
9	Сопровождение программных систем	8			4		4
10	Интеграции программ и данных	8			4		4
11	Методологии проектирования программных систем	8			4		4
12	Проектирование программных систем с использованием UML	8			4		4
13	Новые тенденции программирования	10			4		6
14	Прототипирование программных систем	10			4		6
15	Управление ресурсами	10			4		6
16	Сертификация программных продуктов	12			4		8
Итого		**144**			**72**		**72**

3.3 Содержание дисциплины

Введение. Программная инженерия

Предмет, задачи и содержание дисциплины. Структура курса, его место и роль в подготовке специалиста, связь с другими дисциплинами. Программная инженерия. Ее цели и задачи. Категории специалистов, применяющих методы программной инженерии в своей практической деятельности. Отличия программной инженерии от других инженерных дисциплин. Модели процесса разработки программного обеспечения. Программная документация, виды программных документов. Требования к ней. Стандарты документирования. ЕСПД. ГОСТ. Техническое задание на программный продукт. Программы и методики испытаний программного продукта. Руководство оператора программного продукта.

Эволюция программного обеспечения

Программная система и ее подсистемы. Архитектура программной системы. Стадии и этапы разработки программных систем и программной документации. Основные стандарты программной инженерии и их краткая характеристика. Эволюция программного обеспечения. Краткая характеристика реинженерии, реверсной инженерии, рефакторинга. Жизненный цикл программной системы. Стандарты жизненного цикла в программной инженерии и их назначение. Перечень и содержание процессов жизненного цикла. Виды моделей жизненного цикла. Сравнительная оценка модели процессов жизненного цикла стандарта 12207 и областей-процессов ядра знаний SWEBOK.

Проект программной системы

Проект программной системы, программа проектов и портфель проектов. Состав и содержание документов системного проекта программной системы. Выработка проекта контракта с заказчиком на обеспечение жизненного цикла программной системы. Назначение, содержание и состав плана обеспечения жизненного цикла и распределения ресурсов проекта программной системы. Технико-экономическое обоснование проекта программной системы. Разработка документов технико-экономического обоснования проекта программной системы на базе экспертных оценок модели COCOMO.

Риски программных систем и требования к ним

Риски в жизненном цикле программных систем. Идентификация рисков. Риски при формировании требований к программным системам. Качественный и количественный анализ рисков. Требования к программным системам. Инженерия требований, выявление, анализ, спецификация, аттестация и управление требованиями. Выявление, анализ и документирование требований к программной системе. Версии требований. Трассировка требований. Программные средства управления требованиями.

Инженерия качества

Классификация моделей надежности. Данные для оценивания надежности программных систем. Качество программного обеспечения. Характеристики качества: функциональность, надежность, практичность, эффективность, сопровождаемость, мобильность. Стандарты качества программного обеспечения. Оценка качества. Выработка требований к характеристикам качества программных систем. Классификация программных систем по стандарту ISO 12182. Анализ и сравнение требований к их характеристикам качества. Динамические и статические методы обеспечения качества программного обеспечения. Верификация. Статический анализ программного обеспечения и обнаружение ошибок. Рефакторинг и модификация программ.

Программометрика

Объектно-ориентированные метрики: метрики Чидамбера-Кемерера, Лоренца и Кида, MOOD. Метрики сложности и размера программ. Задачи, методы и средства аудита программного кода. Отладка программных систем. Методы и средства отладки. Автоматизация отладки программ. CASE-системы. Методы верификации, кодирования и тестирования программных систем. Их краткая характеристика. Автоматизация верификации.

Тестирование программных систем

Тестирование программных систем: основные концепции тестирования, уровни, техники, метрики тестирования. Управление процессом тестирования. Классификация тестов для проверки программных систем. Организация работ для проведения тестирования. Задачи группы тестировщиков. Анализ и оценивание корректности программ по их покрытию тестами. Функциональное, структурное, динамическое, статическое, модульное тестирование программных систем. Тестирование программных систем. Разбиение на классы эквивалентности. Анализ граничных условий. Отрицательное тестирование. Понятие «белого» и «черного» ящика при тестировании. Тестирование производительности, надежности, безопасности, интеграции, стресс-тестирование. Тестирование приложений с графическим пользовательским интерфейсом. Ключевые факторы, влияющие на проектирование интерфейсов. Тестирование конформности (соответствия) реализации программных систем исходным стандартам.

Программные дефекты

Дефект. Управление дефектами. Возможности автоматического выявления дефектов в программе. Общие особенности дефектов, ошибок и рисков в программных системах. Причины и свойства дефектов, ошибок и модификаций в программных системах.

Сопровождение программных систем

Основные концепции, ключевые вопросы и техники сопровождения программных систем. Организация и методы сопровождения программных систем.

Задачи и процессы переноса программ и данных на иные платформы. Управление конфигурацией, идентификацией, контролем, учетом статуса, версиями программного проекта. Этапы и процедуры при управлении конфигурацией программных средств. Технологическое обеспечение при сопровождении и управлении конфигурацией программных средств.

Интеграции программ и данных

Цели и задачи интеграции программ и данных в программной инженерии. Краткая характеристика технологий интеграции программ и данных: COM, CORBA, .NET и др. Проблемы преобразования данных при замене одной базы данных на другую. Методы переноса данных. Цели и задачи внесения изменений в программные системы в процессе их сопровождения. Методы внесения изменений. Основные проблемы, возникающие при внесении изменений.

Методологии проектирования программных систем

Групповая разработка. Организация коллектива разработчиков. Организация коллектива разработчиков в компании Microsoft. Метод MSF и его основные модели. Управление проектом и рисками в MSF. Цели и задачи проектной группы в MSF. Agile-методология как пример гибкой методологии разработки. Рекомендации Agile по управлению продуктом, командой, контрактами, рисками. Инженерные практики. Контроль и обеспечение качества. Анализ требований. Масштабирование Agile. Гибкие методологии проектирования. Экстремальное программирование (XP). Методология SCRUM. Водопадная модель проектирования и ее разновидности. Спиральная модель проектирования. Ее особенности. Общие черты инкрементной и эволюционной модели.

Проектирование программных систем с использованием UML

Возможности Rational Rose. Проектирование программных систем с использованием UML: назначение UML, его базовые понятия. UML как средство документирования. Диаграммы классов в UML и моделирование предметной области. Ассоциации. Обобщение. Зависимости. Диаграммы последовательности и взаимодействия компонент в UML. Декомпозиция системы средствами UML. Структурный метод проектирования программных систем. Структуры наглядного проектирования языка UML. Унифицированный процесс RUP (Rational Unified Process) компании Rational Software как итеративный метод разработки программного обеспечения: фазы, визуальное моделирование.

Новые тенденции программирования

Пути развития компонентного программирования. Основные понятия и этапы жизненного цикла компонентного программирования. Основные элементы аспектно-ориентированного программирования. Объекты генерирующего программирования и краткая их характеристика.

Прототипирование программных систем

Быстрая разработка приложений. Итерационно-инкрементальные подходы. Методологии разработки программного обеспечения, базирующиеся на модели CMMI. Задачи и особенности объектно-ориентированного проектирования программных систем. Основные понятия и модели объектно-ориентированного проектирования программных систем.

Управление ресурсами

Управление ресурсами в жизненном цикле программных систем. Основные ресурсы обеспечения жизненного цикла программных систем: обеспечение функциональной пригодности, конструктивных характеристик качества, имитации внешней среды для тестирования и испытаний, сопровождения и мониторинга. Ресурсы специалистов для обеспечения жизненного цикла программных систем. Паттерн «Оценка кандидата». Оценка и отбор кандидатов по резюме. Проведение собеседования. Формирование команды. Лидерство и управление. «Правильные» люди. Мотивация. Эффективное взаимодействие.

Сертификация программных продуктов

Процессы сертификации в жизненном цикле программных средств. Организация сертификации программных продуктов. Документирование процессов и результатов сертификации программных продуктов.

3.4 Тематика лабораторных занятий

Введение. Программная инженерия
Эволюция программного обеспечения
Проект программной системы
Риски программных систем и требования к ним
Инженерия качества
Программометрика
Тестирование программных систем
Программные дефекты
Сопровождение программных систем
Интеграции программ и данных
Методологии проектирования программных систем
Проектирование программных систем с использованием UML
Новые тенденции программирования

Прототипирование программных систем
Управление ресурсами
Сертификация программных продуктов

4 Учебно-методическое и информационное обеспечение

4.1 Нормативные документы и ГОСТы

1. Федеральный закон от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации».

2. Приказ Минобрнауки России от 19.09.2017 N 929 (ред. от 08.02.2021) «Об утверждении федерального государственного образовательного стандарта высшего образования – бакалавриат по направлению подготовки 09.03.03 Прикладная информатика, утвержденный приказом Министерства образования и науки Российской Федерации от 19.09.2017 № 922.

3.мАкадемический учебный план Направление подготовки: 09.03.03 Прикладная информатика Профиль: Информационные технологии управления бизнесом Форма обучения: очная.

4.2 Основная литература

Маран, М. М. Программная инженерия : учебное пособие для вузов / М. М. Маран. — 3-е изд., стер. — Санкт-Петербург : Лань, 2022. — 196 с. — ISBN 978-5-8114-9323-4. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/189470 (дата обращения: 28.09.2023). — Режим доступа: для авториз. пользователей.

4.3 Дополнительная литература

Системная и программная инженерия : учебное пособие / А. Н. Миронов, Ю. А. Воронцов, Е. К. Михайлова, С. М. Трушин. — Москва : РТУ МИРЭА, 2022. — 129 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/310997 (дата обращения: 28.09.2023). — Режим доступа: для авториз. пользователей.

4.4 Электронные образовательные ресурсы

https://online.mospolytech.ru/course/view.php?id=1339

4.5 Лицензионное и свободно распространяемое программное обеспечение

1. Visual Studio Code
2. Браузеры Chrome, Edge, Firefox
3. OpenVPN с правами для запуска у студентов
4. FileZilla
5. PuTTy
6. Git
7. Node.js 18
8. Python 3.10
9. Wireshark

5 Материально-техническое обеспечение

Для проведения лабораторных работ и самостоятельной работы студентов подходят аудитории, оснащенные компьютерами с программным обеспечением в соответствии со списком в пункте 4.5 и подключенные к интернету.

Число рабочих мест в аудитории должно быть достаточным для обеспечения индивидуальной работы студентов.

Рабочее место преподавателя должно быть оснащено компьютером с подключенным к нему проектором или иным аналогичным по функциональному назначению оборудованием.

6 Методические рекомендации

6.1 Методические рекомендации для преподавателя по организации обучения

1. При подготовке к занятиям следует предварительно проработать материал занятия, предусмотрев его подачу точно в отведенное для этого время занятия. Следует подготовить необходимые материалы – теоретические сведения, задачи и др. При проведении занятия следует контролировать подачу материала и решение заданий с учетом учебного времени, отведенного для занятия.

2. При проверке работ и отчетов следует учитывать не только правильность выполнения заданий, но и оптимальность выбранных методов решения, правильность выполнения всех его шагов.

6.2 Методические указания для обучающихся по освоению дисциплины

Обучение по дисциплине «Программная инженерия» предполагает изучение курса на аудиторных занятиях и самостоятельной работы обучающихся. Виды самостоятельной работы обучающегося, порядок и сроки ее выполнения:

– подготовка к лекциям, лабораторным работам и практическим занятиям с использованием конспекта лекций, материалов лабораторных занятий и учебнометодического и информационного обеспечения дисциплины (в течение 5 и 6 семестров в соответствии с расписанием занятий);

– оформление отчетов по выполненным лабораторным работам и теоретическая подготовка к их сдаче (в течение 5 и 6 семестров в соответствии с расписанием занятий).

Перечень вопросов для проведения текущего контроля и промежуточной аттестации – в соответствии с тематикой разделов дисциплины.

С целью обеспечения успешного обучения обучающийся должен готовиться к лекции, которая является важнейшей формой организации учебного процесса, поскольку:

− знакомит с новым учебным материалом;

− разъясняет учебные элементы, трудные для понимания;

− систематизирует учебный материал; − ориентирует в учебном процессе.

Подготовка к занятиям лекционного типа заключается в следующем:

− внимательном чтении материала предыдущей лекции;

− предварительное ознакомление с темой предстоящей лекции (по тематическому плану, по информации лектора);

− ознакомление с учебным материалом по учебнику и учебным пособиям;

− уяснение места изучаемой темы в своей профессиональной подготовке;

− подготовке возможных вопросов, которые студент задаст лектору на лекции.

Подготовка к написанию курсового проекта исходит из того, что курсовой проект – это самостоятельная учебно-исследовательская работа обучающегося, где автор раскрывает суть исследуемой проблемы, приводит различные точки зрения, а также собственные взгляды на

нее. Содержание материала должно быть логичным, изложение материала носит проблемно-поисковый характер. Этапы работы над курсовым проектом.

1. Формулирование темы. Тема должна быть не только актуальной по своему значению, но и оригинальной, интересной по содержанию.
2. Подбор и изучение основных источников по теме.
3. Составление библиографии.
4. Обработка и систематизация информации.
5. Разработка плана курсового проекта.
6. Оформление курсового проекта в виде презентации.
7. Публичное выступление с результатами исследования на занятии.

Содержание работы должно отражать.

1. Знание современного состояния проблемы.
2. Обоснование выбранной темы.
3. Использование известных результатов и фактов.
4. Полноту цитируемой литературы, ссылки на работы ученых, занимающихся данной проблемой.
5. Актуальность поставленной проблемы.
6. Материал, подтверждающий научное, либо практическое значение в настоящее время.

Типовая структура курсового проекта.

1. Титульный лист.
2. План (простой или развернутый).
3. Введение.
4. Основная часть.
5. Заключение.
6. Список литературы.

Курсовой проект может рассматриваться как одна из форм контрольных работ. Объем курсового проекта не должен превышать 10 страниц.

Представление курсового проекта осуществляется в форме доклада с предъявлением презентации.

7 Фонд оценочных средств

7.1 Методы контроля и оценивания результатов обучения

Методика преподавания дисциплины «Программная инженерия» и реализация компетентностного подхода в изложении и восприятии материала предусматривает использование следующих активных и интерактивных форм проведения групповых, индивидуальных, аудиторных занятий в сочетании с внеаудиторной работой с целью формирования и развития профессиональных навыков обучающихся:

– подготовка к выполнению лабораторных работ в лабораториях вуза;
– обсуждение и защита рефератов по дисциплине;
– подготовка, представление и обсуждение презентаций на семинарских занятиях;
– организация и проведение текущего контроля знаний студентов в форме бланкового тестирования;
– проведение интерактивных занятий по процедуре подготовки к интернет тестированию;
– использование интерактивных форм текущего контроля в форме аудиторного и внеаудиторного интернет-тестирования;
– проведение мастер-классов экспертов и специалистов по программированию.

Удельный вес занятий, проводимых в интерактивных формах, определен главной целью образовательной программы, особенностью контингента обучающихся и содержанием

дисциплины «Программная инженерия» и в целом по дисциплине составляет 50% аудиторных занятий. Занятия лекционного типа составляют 25% от объема аудиторных занятий.

Промежуточная аттестация обучающихся в форме экзамена проводится по результатам выполнения всех видов учебной работы, предусмотренных учебным планом по данной дисциплине (модулю), при этом учитываются результаты текущего контроля успеваемости в течение семестра. Оценка степени достижения обучающимися планируемых результатов обучения по дисциплине (модулю) проводится преподавателем, ведущим занятия по дисциплине (модулю) методом экспертной оценки. По итогам промежуточной аттестации по дисциплине (модулю) выставляется оценка «отлично», «хорошо», «удовлетворительно» или «неудовлетворительно».

7.2 Шкала и критерии оценивания результатов обучения

Отлично Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует соответствие знаний, умений, навыков приведенным в таблицах показателей, оперирует приобретенными знаниями, умениями, навыками, применяет их в ситуациях повышенной сложности. При этом могут быть допущены незначительные ошибки, неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.

Хорошо Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует неполное, правильное соответствие знаний, умений, навыков приведенным в таблицах показателей, либо если при этом были допущены 2-3 несущественные ошибки.

Удовлетворительно Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует соответствие знаний, в котором освещена основная, наиболее важная часть материала, но при этом допущена одна значительная ошибка или неточность.

Неудовлетворительно Не выполнен один или более видов учебной работы, предусмотренных учебным планом. Студент демонстрирует неполное соответствие знаний, умений, навыков приведенным в таблицах показателей, допускаются значительные ошибки, проявляется отсутствие знаний, умений, навыков по ряду показателей, студент испытывает значительные затруднения при оперировании знаниями и умениями при их переносе на новые ситуации.

Базовый уровень

студент способен разрабатывать, внедрять и адаптировать прикладное программное обеспечение для решения задач научной и проектно-технологической деятельности в стандартных учебных ситуациях.

студент способен принимать участие во внедрении, адаптации и настройке информационных систем в стандартных учебных ситуациях.

студент способен осуществлять ведение базы данных и поддержку информационного обеспечения решения прикладных задач в стандартных учебных ситуациях.

Повышенный уровень

студент способен разрабатывать, внедрять и адаптировать прикладное программное обеспечение для решения задач научной и проектно-технологической деятельности в практической деятельности

студент способен принимать участие во внедрении, адаптации и настройке информационных систем в практической деятельности.

студент способен осуществлять ведение базы данных и поддержку информационного обеспечения решения прикладных задач в практической деятельности.

7.3 Оценочные средства

7.3.1 Текущий контроль

Примерный список вопросов

1. Программная инженерия. Ее цели и задачи.
2. Категории специалистов, применяющих методы программной инженерии в своей практической деятельности.
3. Отличия программной инженерии от других инженерных дисциплин.
4. Модели процесса разработки программного обеспечения.
5. Эволюция программного обеспечения.
6. Краткая характеристика реинженерии, реверсной инженерии, рефакторинга.
7. Жизненный цикл программной системы.
8. Стандарты жизненного цикла программ в программной инженерии и их назначение.
9. Перечень и содержание процессов жизненного цикла программ.
10. Программная система и ее подсистемы.
11. Стадии и этапы разработки программных систем и программной документации.
12. Проект программной системы, программа проектов и портфель проектов.
13. Выработка проекта контракта с заказчиком на обеспечение жизненного цикла программной системы.
14. Назначение, содержание и состав плана обеспечения жизненного цикла и распределения ресурсов проекта программной системы.
15. Технико-экономическое обоснование проекта программной системы.
16. Разработка документов технико-экономического обоснования проекта программной системы на базе экспертных оценок модели COCOMO.
17. Риски в жизненном цикле программных систем.
18. Идентификация рисков.
19. Риски при формировании требований к программным системам.
20. Качественный и количественный анализ рисков.
21. Требования к программным системам.
22. Инженерия требований, выявление, анализ, спецификация, аттестация и управление требованиями.
23. Выявление, анализ и документирование требований к программной системе.
24. Версии требований.
25. Трассировка требований.
26. Программные средства управления требованиями.
27. Инженерия качества.
28. Классификация моделей надежности.
29. Данные для оценивания надежности программных систем.
30. Характеристики качества: функциональность, надежность, практичность, эффективность, сопровождаемость, мобильность.
31. Стандарты качества программного обеспечения.

32. Оценка качества программного обеспечения.
33. Выработка требований к характеристикам качества программных систем.
34. Классификация программных систем по стандарту ISO 12182.
35. Анализ и сравнение требований к их характеристикам качества.
36. Динамические и статические методы обеспечения качества программного обеспечения.
37. Статический анализ программного обеспечения и обнаружение ошибок.
38. Рефакторинг и модификация программ.
39. Программометрика. Объектно-ориентированные метрики: метрики Чидамбера-Кемерера, Лоренца и Кида, MOOD.
40. Метрики сложности и размера программ.
41. Задачи, методы и средства аудита программного кода.
42. Отладка программных систем. Методы и средства отладки.
43. Автоматизация отладки программ. CASE-системы.
44. Методы верификации, кодирования и тестирования программных систем. Их краткая характеристика.
45. Автоматизация верификации программного обеспечения.
46. Тестирование программных систем: основные концепции тестирования, уровни, техники, метрики тестирования.
47. Управление процессом тестирования программного обеспечения.
48. Классификация тестов для проверки программных систем.
49. Организация работ для проведения тестирования.
50. Задачи группы тестировщиков программного обеспечения.
51. Анализ и оценивание корректности программ по их покрытию тестами.
52. Функциональное, структурное, динамическое, статическое, модульное тестирование программных систем.
53. Тестирование программных систем. Разбиение на классы эквивалентности. Анализ граничных условий.
54. Отрицательное тестирование программных систем.
55. Понятие «белого» и «черного» ящика при тестировании программных систем.
56. Тестирование производительности, надежности, безопасности, интеграции, стресс-тестирование.
57. Тестирование приложений с графическим пользовательским интерфейсом.
58. Ключевые факторы, влияющие на проектирование интерфейсов.
59. Дефект. Управление дефектами. Возможности автоматического выявления дефектов в программе.
60. Общие особенности дефектов, ошибок и рисков в программных системах.
61. Причины и свойства дефектов, ошибок и модификаций в программных системах.
62. Тестирование конформности (соответствия) реализации программных систем исходным стандартам.
63. Сопровождение программных систем. Основные концепции, ключевые вопросы и техники сопровождения.
64. Задачи и процессы переноса программ и данных на иные платформы.
65. Программная документация, виды программных документов. Требования к ней.

66. Стандарты документирования. ЕСПД.

67. ГОСТ. Техническое задание на программный продукт.

68. ГОСТ. Программы и методики испытаний программного продукта.

69. ГОСТ. Руководство оператора программного продукта.

70. Управление конфигурацией, идентификацией, контролем, учетом статуса, версиями программного проекта.

71. Этапы и процедуры при управлении конфигурацией программных средств.

72. Технологическое обеспечение при сопровождении и управлении конфигурацией программных средств.

73. Цели и задачи интеграции программ и данных в программной инженерии.

74. Краткая характеристика технологий интеграции программ и данных: COM, CORBA, .NET и др.

75. Проблемы преобразования данных при замене одной базы данных на другую. Методы переноса данных.

76. Цели и задачи внесения изменений в программные системы в процессе их сопровождения.

77. Методы внесения изменений. Основные проблемы, возникающие при внесении изменений.

78.Групповая разработка. Организация коллектива разработчиков. Организация коллектива разработчиков в компании Microsoft.

79. Метод MSF и его основные модели. Управление проектом и рисками в MSF. Цели и задачи проектной группы в MSF.

80. Возможности Rational Rose. Проектирование программных систем с использованием UML: назначение UML, его базовые понятия.

81. UML как средство документирования.

82. Диаграммы классов в UML и моделирование предметной области. Ассоциации. Обобщение. Зависимости.

83. Диаграммы последовательности и взаимодействия компонент в UML.

84. Декомпозиция системы средствами UML.

85. Структурный метод проектирования программных систем. Структуры наглядного проектирования языка UML.

86. Унифицированный процесс RUP (Rational Unified Process) компании Rational Software как итеративный метод разработки программного обеспечения: фазы, визуальное моделирование.

87. Agile-методология как пример гибкой методологии разработки.

88. Рекомендации Agile по управлению продуктом, командой, контрактами, рисками. Инженерные практики. Контроль и обеспечение качества. Анализ требований. Масштабирование Agile.

89. Гибкие методологии проектирования. Экстремальное программирование (XP).

90. Методология SCRUM.

91. Водопадная модель проектирования и ее разновидности.

92. Спиральная модель проектирования. Ее особенности.

93. Общие черты инкрементной и эволюционной модели. Примеры.

94. Сравнительная оценка модели процессов жизненного цикла стандарта 12207 и областей-процессов ядра знаний SWEBOK.

95. Пути развития компонентного программирования. Основные понятия и этапы жизненного цикла компонентного программирования.

96. Основные элементы аспектно-ориентированного программирования.

97. Объекты генерирующего программирования и краткая их характеристика.

98. Прототипирование. Быстрая разработка приложений.

99. Методологии разработки программного обеспечения, базирующиеся на модели CMMI.

100. Задачи и особенности объектно-ориентированного проектирования программных систем. Основные понятия и модели объектноориентированного проектирования программных систем.

101. Управление ресурсами в жизненном цикле программных систем.

102. Основные ресурсы обеспечения жизненного цикла программных систем: обеспечение функциональной пригодности, конструктивных характеристик качества, имитации внешней среды для тестирования и испытаний, сопровождения и мониторинга.

103. Ресурсы специалистов для обеспечения жизненного цикла программных систем. Паттерн «Оценка кандидата».

104. Оценка и отбор кандидатов по резюме. Проведение собеседования.

105. Формирование команды. Лидерство и управление. «Правильные» люди. Мотивация. Эффективное взаимодействие. Примеры.

106. Процессы сертификации в жизненном цикле программных средств.

107. Организация сертификации программных продуктов.

108. Документирование процессов и результатов сертификации программных продуктов.

7.3.2 Промежуточная аттестация

Пример комплекта разноуровневых задач и заданий

Требуется составить техническое задание, программу и методику испытаний и руководство оператора на разработку электронного портала учебного заведения, предусмотрев реализацию основных функций электронного портала. В качестве примера подражания рекомендуется рассмотреть электронный портал Московского политехнического университета.

8 Тематики курсовых проектов

Техническое задание программы, разработанной студентом в семестре.

Программа и методика испытаний программы, разработанной студентом в семестре.

Руководство оператора программы, разработанной студентом в семестре.

Основные стандарты программной инженерии и их краткая характеристика.

Виды моделей жизненного цикла программ.

Архитектура программной системы.

Качество программного обеспечения.

Верификация программного обеспечения.

Организация и методы сопровождения программных систем.

Применение методов программной инженерии для разработки программы студента (индивидуально для каждого обучающегося).