

Министерство науки и высшего образования РФ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Национальный исследовательский университет «МЭИ»

Направление подготовки/специальность: 01.04.02 Прикладная математика и информатика

Наименование образовательной программы: Математическое и программное обеспечение вычислительных машин и компьютерных сетей

Уровень образования: высшее образование - магистратура

Форма обучения: Очная


Рабочая программа дисциплины
CASE-ТЕХНОЛОГИИ РАЗРАБОТКИ ПРОГРАММНЫХ СРЕДСТВ

Блок:	Блок 1 «Дисциплины (модули)»
Часть образовательной программы:	Часть, формируемая участниками образовательных отношений
№ дисциплины по учебному плану:	Б1.Ч.06.01.01
Трудоемкость в зачетных единицах:	3 семестр - 4;
Часов (всего) по учебному плану:	144 часа
Лекции	3 семестр - 16 часов;
Практические занятия	не предусмотрено учебным планом
Лабораторные работы	3 семестр - 32 часа;
Консультации	проводится в рамках часов аудиторных занятий
Самостоятельная работа	3 семестр - 95,7 часа;
в том числе на КП/КР	не предусмотрено учебным планом
Иная контактная работа	проводится в рамках часов аудиторных занятий
включая: Лабораторная работа Семинар	
Промежуточная аттестация:	
Зачет с оценкой	3 семестр - 0,30 часа;

Москва 2024

ПРОГРАММУ СОСТАВИЛ:


Преподаватель

	Подписано электронной подписью ФГБОУ ВО «НИУ «МЭИ»	
	Сведения о владельце ЦЭП МЭИ	
	Владелец	Куриленко И.Е.
	Идентификатор	R73df8d6c-KurilenkoIY-5c331b90

И.Е. Куриленко


СОГЛАСОВАНО:

Руководитель
образовательной
программы

	Подписано электронной подписью ФГБОУ ВО «НИУ «МЭИ»	
	Сведения о владельце ЦЭП МЭИ	
	Владелец	Чернецов А.М.
	Идентификатор	Re594826f-ChernetsovAM-0080e09

А.М. Чернецов

Заведующий выпускающей
кафедрой

	Подписано электронной подписью ФГБОУ ВО «НИУ «МЭИ»	
	Сведения о владельце ЦЭП МЭИ	
	Владелец	Варшавский П.Р.
	Идентификатор	R9a563c96-VarshavskyPR-efb4bbd

П.Р.
Варшавский

1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Цель освоения дисциплины: Изучение современных подходов к разработке программного обеспечения, соответствующих языковых и инструментальных средств.

Задачи дисциплины

- ознакомление с современными методологиями разработки программного обеспечения;
- освоение современных технологий разработки программного обеспечения;
- освоение современных инструментальных систем, используемых при разработке программного обеспечения;
- ознакомление с проблемами, возникающими при реализации крупных программных проектов.

Формируемые у обучающегося **компетенции** и запланированные **результаты обучения** по дисциплине, соотнесенные с **индикаторами достижения компетенций**:

Код и наименование компетенции	Код и наименование индикатора достижения компетенции	Запланированные результаты обучения
ПК-1 Способен выполнять работы на всем жизненном цикле информационных систем в выбранной среде разработки компьютерного ПО	ИД-1 _{ПК-1} Выбирает методы анализа и проектирования ПО с применением CASE-средств	знать: - современные технологии разработки программного обеспечения. уметь: - применять современные CASE-средства.
ПК-2 Способен выполнять работы по внедрению и сопровождению разработанного прикладного ПО	ИД-1 _{ПК-2} Демонстрирует умение выполнять внедрение и сопровождение ПО	знать: - методику организации процесса сопровождения внедренных информационных систем; - современные архитектурные концепции, применяемые при разработке программных систем; - дополнительные средства поддержки жизненного цикла программного обеспечения.. уметь: - создавать и внедрять стандарты кодирования и проектирования; - анализировать архитектуру программных систем, оценивать уместность применения той или иной разновидности архитектурного решения; - фиксировать требования к ПО в системах управления требованиями.
РПК-2 Способен применять методы проектирования для обеспечения реализации результатов анализа	ИД-1 _{РПК-2} Формализует описание бизнес-процессов	знать: - современные методологии разработки программного обеспечения. уметь: - организовывать команду разработки программного продукта.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ ВО

Дисциплина относится к основной профессиональной образовательной программе Математическое и программное обеспечение вычислительных машин и компьютерных сетей (далее – ОПОП), направления подготовки 01.04.02 Прикладная математика и информатика, уровень образования: высшее образование - магистратура.

Требования к входным знаниям и умениям:

- знать основы программирования
- знать языки и методы программирования
- знать структуры данных и методы программирования

Результаты обучения, полученные при освоении дисциплины, необходимы при выполнении выпускной квалификационной работы.

3. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

3.1 Структура дисциплины

Общая трудоемкость дисциплины составляет 4 зачетных единицы, 144 часа.

№ п/п	Разделы/темы дисциплины/формы промежуточной аттестации	Всего часов на раздел	Семестр	Распределение трудоемкости раздела (в часах) по видам учебной работы										Содержание самостоятельной работы/ методические указания
				Контактная работа							СР			
				Лек	Лаб	Пр	Консультация		ИКР		ПА	Работа в семестре	Подготовка к аттестации /контроль	
КПР	ГК	ИККП	ТК											
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	Введение	19	3	3	1	-	-	-	-	-	-	15	-	<u>Самостоятельное изучение теоретического материала:</u> Изучение дополнительного материала по разделу "CASE-технологии"
1.1	CASE-технология. CASE-средства. CASE-системы.	7		1	1	-	-	-	-	-	-	5	-	
1.2	Классификация CASE-средств.	6		1	-	-	-	-	-	-	-	5	-	
1.3	Жизненный цикл программного обеспечения. Модели жизненного цикла программного обеспечения.	6		1	-	-	-	-	-	-	-	5	-	
2	Применение CASE-средств на этапе анализа в жизненном цикле программного обеспечения.	49		5	12	-	-	-	-	-	-	-	32	
2.1	Этап анализа в жизненном цикле программного обеспечения.	8	1	2	-	-	-	-	-	-	5	-		
2.2	Техническое задание.	8	1	2	-	-	-	-	-	-	5	-		
2.3	Структурные методологии и подходы к анализу и проектированию.	9	1	2	-	-	-	-	-	-	6	-		
2.4	Объектно-ориентированный	12	1	3	-	-	-	-	-	-	8	-		

	анализ и проектирование												
2.5	Унифицированный язык моделирования	12	1	3	-	-	-	-	-	-	8	-	
3	Методологии разработки программного обеспечения	75.70	8	19	-	-	-	-	-	-	48.70	-	<p><u>Самостоятельное изучение теоретического материала:</u> Изучение дополнительного материала по разделу "Методологии разработки программного обеспечения"</p> <p><u>Изучение материалов литературных источников:</u></p> <p>[1], с. 158-166 [2], п. 3-18 [3], п. 2 [4], с. 21-27 [5], с. 26-33</p>
3.1	Принципы проектирования сложных систем	11	1	2	-	-	-	-	-	-	8	-	
3.2	Методология Rational Unified Process (RUP)	11	1	2	-	-	-	-	-	-	8	-	
3.3	Методология разработки программных систем MSF (Microsoft Solutions Framework)	11	1	2	-	-	-	-	-	-	8	-	
3.4	Гибкие и адаптивные методики разработки (Agile)	14	2	4	-	-	-	-	-	-	8	-	
3.5	Надежность программного обеспечения (ПО). CASE-средства и надежность ПО. Контроль качества ПО.	12	1	3	-	-	-	-	-	-	8	-	
3.6	Дополнительные средства поддержки жизненного цикла программного обеспечения.	16.70	2	6	-	-	-	-	-	-	8.70	-	
	Зачет с оценкой	0.30	-	-	-	-	-	-	-	0.30	-	-	
	Всего за семестр	144.00	16	32	-	-	-	-	-	0.30	95.70	-	
	Итого за семестр	144.00	16	32	-	-	-	-	-	0.30	95.70	-	

Примечание: Лек – лекции; Лаб – лабораторные работы; Пр – практические занятия; КПП – аудиторные консультации по курсовым проектам/работам; ИККП – индивидуальные консультации по курсовым проектам/работам; ГК- групповые консультации по разделам

дисциплины; СР – самостоятельная работа студента; ИКР – иная контактная работа; ТК – текущий контроль; ПА – промежуточная аттестация

3.2 Краткое содержание разделов

1. Введение

1.1. CASE-технология. CASE-средства. CASE-системы.

Что такое CASE-технология.. Что такое CASE-средство.. Что такое CASE-система.. Области применения CASE-технологий.. Классификация технологии разработки программных систем.. Процесс разработки программного обеспечения с использованием CASE-средств.. Информационная инженерия и обратное проектирование.. Методология быстрой разработки приложений (Rapid Application Development (RAD))..

1.2. Классификация CASE-средств.

Классификация по типам. Классификация по уровням.

1.3. Жизненный цикл программного обеспечения. Модели жизненного цикла программного обеспечения.

Понятие жизненного цикла ПО.. Каскадная модель жизненного цикла.. Спиральная модель.. Итеративная модель жизненного цикла.. Достоинства и недостатки различных моделей жизненного цикла ПО..

2. Применение CASE-средств на этапе анализа в жизненном цикле программного обеспечения.

2.1. Этап анализа в жизненном цикле программного обеспечения.

Этап анализа в жизненном цикле.. Методы сбора и анализа требований.. Методологические аспекты анализа целей и требований к разрабатываемому программному обеспечению..

2.2. Техническое задание.

Что такое техническое задание.. Для чего разрабатывают техническое задание. Выгоды для заказчика и исполнителя.. Структура технического задания..

2.3. Структурные методологии и подходы к анализу и проектированию.

Функционально-ориентированное (структурное) проектирование программного обеспечения.. Диаграммные методологии проектирования программного обеспечения.. CASE-средства поддержки структурных методологий.. Проектирование, ориентированное на данные.. Методология DATARUN проектирования информационных систем..

2.4. Объектно-ориентированный анализ и проектирование

Методики объектно-ориентированного анализа и проектирования.. Классификация, основные этапы и задачи объектно-ориентированных методов анализа и проектирования..

2.5. Унифицированный язык моделирования

Язык моделирования UML. Синтаксис. Семантика. Пакеты.. Канонические диаграммы.. Диаграммы вариантов использования и сценарии.. Диаграммы классов и их использование.. Диаграммы кооперации и диаграммы последовательности.. Диаграммы состояний и диаграммы деятельности.. Диаграммы компонентов и диаграммы развертывания.. Направление развития UML.

3. Методологии разработки программного обеспечения

3.1. Принципы проектирования сложных систем

Принципы проектирования сложных систем.. Декомпозиция.. Абстракция.. Повторное использование.. Правила разбиения системы на подсистемы и модули..

3.2. Методология Rational Unified Process (RUP)

Методология объектно-ориентированной разработки RUP (Ration Unified Process).. Унифицированный процесс Rational.. Фазы и дисциплины.. Жизненный цикл проекта в унифицированном процессе Rational..

3.3. Методология разработки программных систем MSF (Microsoft Solutions Framework)

Методология разработки программных систем MSF (Microsoft Solutions Framework).. Модель процессов в MSF.. Этап анализа в MSF.. Этап планирования в MSF.. Этап разработки в MSF.. Этапы контроля качества и внедрения в MSF.. Модель команды разработчиков в MSF. Преимущества и недостатки.. Управление проектом в MSF. Общий обзор.. Дисциплина управления проектом.. Масштабируемость.. Иерархическая структура работ (WBS).. Оценка сроков разработки.. Составление календарного графика работ в проекте.. Диаграммы вариантов использования системы и сценариев использования системы.. Управление компромиссами в MSF.. Стратегия выпуска версий в MSF..

3.4. Гибкие и адаптивные методики разработки (Agile)

Манифест Agile. Семейство методологий Agile.. Методология XP - «экстремальное программирование» : особенности, преимущества, недостатки.. Методология FDD : особенности, преимущества, недостатки.. Методология OpenUP: особенности, преимущества, недостатки.. Методология Scrum: особенности, преимущества, недостатки..

3.5. Надежность программного обеспечения (ПО). CASE-средства и надежность ПО.

Контроль качества ПО.

Причины возникновения ошибок при разработке программных средств.. CASE-модель жизненного цикла программного обеспечения.. Системы автоматизации тестирования программных средств..

3.6. Дополнительные средства поддержки жизненного цикла программного обеспечения.

Дополнительные средства поддержки жизненного цикла разработки программного обеспечения: классификация инструментальных систем.. Системы отслеживания ошибок (Bug-tracking).. Системы управления задачами и заявками. Основные понятия. Обзор.. Контроль версий. Системы управления версиями. Основные понятия. Обзор. Модели версионирования.. Система управления версиями RCS.. Система управления версиями CVS.. Система управления версиями SVN. Основные возможности. Внутренняя архитектура, компоненты.. Система управления версиями GIT.. Планирование.. Средства управления требованиями, конфигурационного управления, документирования, тестирования, поддержки коллективной разработки.. Системы автоматизации сборки программных средств.

3.3. Темы практических занятий

не предусмотрено

3.4. Темы лабораторных работ

1. Разработка технического задания на программный продукт.;
2. Использование методологии функционального моделирования IDEF0 для выполнения этапа анализа и проектирования программного обеспечения;
3. Использование методологии DFD для выполнения этапа анализа и проектирования

- программного обеспечения;
4. Использование методологии IDEF3 для выполнения этапа анализа и проектирования программного обеспечения;
 5. Использование методологии ARIS для выполнения этапа анализа и проектирования;
 6. Использование методологии онтологического моделирования IDEF5;
 7. Использование методологии проектирования реляционных баз данных IDEF1X;
 8. Использование унифицированного языка моделирования (UML) для объектно-ориентированного анализа и проектирования;
 9. Поддержка средств моделирования и реинжиниринга в Microsoft Visual Studio;
 10. Автоматизация сборки с применением MSBuild;
 11. Практическое применение систем контроля качества кода на основе метрик на примере Microsoft Visual Studio;
 12. Применение средства контроля качества кода на базе правил;
 13. Создание автоматизированных тестов пользовательского интерфейса с применением Microsoft Visual Studio 2019;
 14. Практическое применение средств автоматической генерации документации по исходному коду;
 15. Практическое применение систем контроля версий.

3.5 Консультации

Групповые консультации по разделам дисциплины (ГК)

1. Обсуждение материалов по кейсам раздела "CASE-технологии"
2. Обсуждение материалов по кейсам раздела "Применение CASE-средств на этапе анализа в жизненном цикле программного обеспечения."
3. Обсуждение материалов по кейсам раздела "Методологии разработки программного обеспечения"

3.6 Тематика курсовых проектов/курсовых работ

Курсовой проект/ работа не предусмотрены

3.7. Соответствие разделов дисциплины и формируемых в них компетенций

Запланированные результаты обучения по дисциплине (в соответствии с разделом 1)	Коды индикаторов	Номер раздела дисциплины (в соответствии с п.3.1)			Оценочное средство (тип и наименование)
		1	2	3	
Знать:					
современные технологии разработки программного обеспечения	ИД-1ПК-1			+	Лабораторная работа/Практическое применение методологии IDEF1X. Использование унифицированного языка моделирования (UML) для объектно-ориентированного анализа и проектирования Семинар/Разработка диаграмм IDEF0, разработка диаграммы потоков данных (DFD, от англ. Data flow diagram), разработка диаграммы IDEF3
методику организации процесса сопровождения внедренных информационных систем	ИД-1ПК-2			+	Лабораторная работа/Применение дополнительных средств поддержки жизненного цикла программного обеспечения.
дополнительные средства поддержки жизненного цикла программного обеспечения.	ИД-1ПК-2			+	Лабораторная работа/Применение дополнительных средств поддержки жизненного цикла программного обеспечения.
современные архитектурные концепции, применяемые при разработке программных систем	ИД-1ПК-2			+	Лабораторная работа/Практическое применение методологии IDEF1X. Использование унифицированного языка моделирования (UML) для объектно-ориентированного анализа и проектирования
современные методологии разработки программного обеспечения	ИД-1РПК-2	+			Лабораторная работа/Этап анализа в жизненном цикле программного обеспечения. Методы сбора и анализа требований. Разработка технического задания на программный продукт.
Уметь:					
применять современные CASE-средства	ИД-1ПК-1	+	+		Семинар/Разработка диаграмм IDEF0, разработка диаграммы потоков данных (DFD, от англ. Data flow diagram), разработка диаграммы IDEF3

фиксировать требования к ПО в системах управления требованиями	ИД-1 _{ПК-2}		+		Лабораторная работа/Этап анализа в жизненном цикле программного обеспечения. Методы сбора и анализа требований. Разработка технического задания на программный продукт.
анализировать архитектуру программных систем, оценивать уместность применения той или иной разновидности архитектурного решения	ИД-1 _{ПК-2}		+	+	Лабораторная работа/Практическое применение методологии IDEF1X. Использование унифицированного языка моделирования (UML) для объектно-ориентированного анализа и проектирования
создавать и внедрять стандарты кодирования и проектирования	ИД-1 _{ПК-2}			+	Лабораторная работа/Применение дополнительных средств поддержки жизненного цикла программного обеспечения.
организовывать команду разработки программного продукта	ИД-1 _{РПК-2}			+	Лабораторная работа/Применение дополнительных средств поддержки жизненного цикла программного обеспечения.

4. КОМПЕТЕНТНОСТНО-ОРИЕНТИРОВАННЫЕ ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ КОНТРОЛЯ ОСВОЕНИЯ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ (ТЕКУЩИЙ КОНТРОЛЬ УСПЕВАЕМОСТИ, ПРОМЕЖУТОЧНАЯ АТТЕСТАЦИЯ ПО ДИСЦИПЛИНЕ)

4.1. Текущий контроль успеваемости

3 семестр

Форма реализации: Выполнение задания

1. Разработка диаграмм IDEF0, разработка диаграммы потоков данных (DFD, от англ. Data flow diagram), разработка диаграммы IDEF3 (Семинар)

Форма реализации: Компьютерное задание

1. Практическое применение методологии IDEF1X. Использование унифицированного языка моделирования (UML) для объектно-ориентированного анализа и проектирования (Лабораторная работа)
2. Применение дополнительных средств поддержки жизненного цикла программного обеспечения. (Лабораторная работа)
3. Этап анализа в жизненном цикле программного обеспечения. Методы сбора и анализа требований. Разработка технического задания на программный продукт. (Лабораторная работа)

Балльно-рейтинговая структура дисциплины является приложением А.

4.2 Промежуточная аттестация по дисциплине

Зачет с оценкой (Семестр №3)

В диплом выставляется оценка за 3 семестр.

Примечание: Оценочные материалы по дисциплине приведены в фонде оценочных материалов ОПОП.

5. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

5.1 Печатные и электронные издания:

1. Вендров, А. М. CASE-технологии:Современные методы и средства проектирования информационных систем / А. М. Вендров . – М. : Финансы и статистика, 1998 . – 176 с. – (Прикладные информационные технологии) . - ISBN 5-279-01979-8 : 24.60 .;
2. Куриленко, И. Е. Разработка программного обеспечения с помощью CASE-технологий : методическое пособие по курсам "CASE-технологии разработки программных средств" по направлениям "Прикладная математика и информатика", "Информатика и вычислительная техника" / И. Е. Куриленко, Нац. исслед. ун-т "МЭИ" . – М. : Изд-во МЭИ, 2013 . – 72 с. <http://elibr.mpei.ru/elibr/view.php?id=5701>;
3. Куриленко, И. Е. Модульное тестирование : учебное пособие по курсам "CASE-технологии разработки программных средств", "Технологии разработки программных средств" по направлениям "Прикладная математика и информатика", "Информатика и вычислительная техника", "Информационные системы" / И. Е. Куриленко, П. Р. Варшавский, Моск. энерг. ин-т (МЭИ ТУ) . – М. : Издательский дом МЭИ, 2011 . – 48 с. - ISBN 978-5-383-00632-0 . <http://elibr.mpei.ru/elibr/view.php?id=2858>;

4. Маран М. М.- "Программная инженерия", (2-е изд., стер.), Издательство: "Лань", Санкт-Петербург, 2021 - (196 с.)
<https://e.lanbook.com/book/175503>;
5. Долженко А. И.- "Технологии командной разработки программного обеспечения информационных систем", (2-е изд.), Издательство: "ИНТУИТ", Москва, 2016 - (300 с.)
<https://e.lanbook.com/book/100515>.

5.2 Лицензионное и свободно распространяемое программное обеспечение:

1. СДО "Прометей";
2. Office / Российский пакет офисных программ;
3. Windows / Операционная система семейства Linux;
4. Видеоконференции (Майнд, Сберджаз, ВК и др);
5. Visual Studio;
6. Libre Office;
7. ERwin Data Modeler.

5.3 Интернет-ресурсы, включая профессиональные базы данных и информационно-справочные системы:

1. ЭБС Лань - <https://e.lanbook.com/>
2. Электронная библиотека МЭИ (ЭБ МЭИ) - <http://elib.mpei.ru/login.php>

6. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Тип помещения	Номер аудитории, наименование	Оснащение
Учебные аудитории для проведения лекционных занятий и текущего контроля	Ж-120, Машинный зал ИВЦ	сервер, кондиционер
	Г-404, Учебная аудитория	парта со скамьей, стол преподавателя, стул, доска меловая, компьютерная сеть с выходом в Интернет
Учебные аудитории для проведения лабораторных занятий	М-708, Дисплейный класс каф. "ПМИИ"	стол преподавателя, стол компьютерный, стул, компьютерная сеть с выходом в Интернет, мультимедийный проектор, экран, компьютер персональный, кондиционер
	М-706, Дисплейный класс каф. "ПМИИ"	стол преподавателя, стол компьютерный, стул, компьютерная сеть с выходом в Интернет, мультимедийный проектор, экран, компьютер персональный, кондиционер
Учебные аудитории для проведения промежуточной аттестации	М-708, Дисплейный класс каф. "ПМИИ"	стол преподавателя, стол компьютерный, стул, компьютерная сеть с выходом в Интернет, мультимедийный проектор, экран, компьютер персональный, кондиционер
	М-706, Дисплейный класс каф. "ПМИИ"	стол преподавателя, стол компьютерный, стул, компьютерная сеть с выходом в Интернет, мультимедийный проектор, экран, компьютер персональный, кондиционер
Помещения для самостоятельной работы	НТБ-201, Компьютерный	стол компьютерный, стул, стол письменный, вешалка для одежды,

	читальный зал	компьютерная сеть с выходом в Интернет, компьютер персональный, принтер, кондиционер
Помещения для консультирования	М-704, Преподавательская кафедры ПМИИ	стол, стул, шкаф, тумба, доска меловая, компьютерная сеть с выходом в Интернет, мультимедийный проектор, экран, компьютер персональный, холодильник, кондиционер
Помещения для хранения оборудования и учебного инвентаря	Е-403, Склад	стол для работы с документами, шкаф, шкаф для документов, книги, учебники, пособия, дипломные и курсовые работы студентов

БАЛЛЬНО-РЕЙТИНГОВАЯ СТРУКТУРА ДИСЦИПЛИНЫ

CASE-технологии разработки программных средств

(название дисциплины)

3 семестр

Перечень контрольных мероприятий текущего контроля успеваемости по дисциплине:

- КМ-1 Этап анализа в жизненном цикле программного обеспечения. Методы сбора и анализа требований. Разработка технического задания на программный продукт. (Лабораторная работа)
- КМ-2 Разработка диаграмм IDEF0, разработка диаграммы потоков данных (DFD, от англ. Data flow diagram), разработка диаграммы IDEF3 (Семинар)
- КМ-3 Практическое применение методологии IDEF1X. Использование унифицированного языка моделирования (UML) для объектно-ориентированного анализа и проектирования (Лабораторная работа)
- КМ-4 Применение дополнительных средств поддержки жизненного цикла программного обеспечения. (Лабораторная работа)

Вид промежуточной аттестации – Зачет с оценкой.

Номер раздела	Раздел дисциплины	Индекс КМ:	КМ-1	КМ-2	КМ-3	КМ-4
		Неделя КМ:	4	8	10	15
1	Введение					
1.1	CASE-технология. CASE-средства. CASE-системы.			+		
1.2	Классификация CASE-средств.		+			
1.3	Жизненный цикл программного обеспечения. Модели жизненного цикла программного обеспечения.			+		
2	Применение CASE-средств на этапе анализа в жизненном цикле программного обеспечения.					
2.1	Этап анализа в жизненном цикле программного обеспечения.		+	+		
2.2	Техническое задание.			+		
2.3	Структурные методологии и подходы к анализу и проектированию.			+		
2.4	Объектно-ориентированный анализ и проектирование				+	
2.5	Унифицированный язык моделирования				+	
3	Методологии разработки программного обеспечения					
3.1	Принципы проектирования сложных систем				+	

3.2	Методология Rational Unified Process (RUP)			+	
3.3	Методология разработки программных систем MSF (Microsoft Solutions Framework)		+	+	+
3.4	Гибкие и адаптивные методики разработки (Agile)		+	+	+
3.5	Надежность программного обеспечения (ПО). CASE-средства и надежность ПО. Контроль качества ПО.		+	+	
3.6	Дополнительные средства поддержки жизненного цикла программного обеспечения.				+
Вес КМ, %:		10	20	30	40