

**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

**Оценочные материалы
по дисциплине
CASE-технологии разработки программных средств**

**Москва
2023**

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ РАЗРАБОТАЛ:

Преподаватель

(должность)

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

(подпись)

И.Е.

Куриленко

(расшифровка
подписи)

СОГЛАСОВАНО:

Руководитель
образовательной
программы

(должность, ученая степень,
ученое звание)

	Подписано электронной подписью ФГБОУ ВО «НИУ «МЭИ»	
	Сведения о владельце ЦЭП МЭИ	
	Владелец	Маран М.М.
	Идентификатор	R7be141f2-MaranMM-804b01e2

(подпись)

М.М. Маран

(расшифровка
подписи)

Заведующий
выпускающей кафедры

(должность, ученая степень,
ученое звание)

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

(подпись)

П.Р.

Варшавский

(расшифровка
подписи)

ОБЩАЯ ЧАСТЬ

Оценочные материалы по дисциплине предназначены для оценки: достижения обучающимися запланированных результатов обучения по дисциплине, этапа формирования запланированных компетенций и уровня освоения дисциплины.

Оценочные материалы по дисциплине включают оценочные средства для проведения мероприятий текущего контроля успеваемости и промежуточной аттестации.

Формируемые у обучающегося компетенции:

1. ПК-2 Способен применять методы проектирования для обеспечения реализации результатов анализа

ИД-1 Формализует описания бизнес-процессов

ИД-2 Демонстрирует умение следить за изменениями требований

ИД-3 Формализует новые требования к ПО

2. ПК-4 Способен выполнять все этапы жизненного цикла информационных систем в выбранной программно-технической среде

ИД-1 Выбирает методы анализа и проектирования ПО с применением CASE-средств

3. ПК-6 Способен обеспечивать работоспособность внедренных информационных систем, включая вопросы защиты данных

ИД-1 Демонстрирует умение выполнять работы по сопровождению внедренных информационных систем

и включает:

для текущего контроля успеваемости:

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)

БРС дисциплины

3 семестр

Раздел дисциплины	Веса контрольных мероприятий, %				
	Индекс КМ:	КМ-1	КМ-2	КМ-3	КМ-4
	Срок КМ:	4	8	10	15
Введение					

CASE-технология. CASE-средства. CASE-системы.	+			
Классификация CASE-средств.	+			
Жизненный цикл программного обеспечения. Модели жизненного цикла программного обеспечения.	+			
Применение CASE-средств на этапе анализа в жизненном цикле программного обеспечения.				
Этап анализа в жизненном цикле программного обеспечения.	+			
Техническое задание.	+			
Структурные методологии и подходы к анализу и проектированию.		+		
Объектно-ориентированный анализ и проектирование			+	
Унифицированный язык моделирования			+	
Методологии разработки программного обеспечения				
Принципы проектирования сложных систем			+	
Методология Rational Unified Process (RUP)			+	
Методология разработки программных систем MSF (Microsoft Solutions Framework)				+
Гибкие и адаптивные методики разработки (Agile)				+
Надежность программного обеспечения (ПО). CASE-средства и надежность ПО. Контроль качества ПО.				+
Дополнительные средства поддержки жизненного цикла программного обеспечения.				+
Вес КМ:	10	20	30	40

\$Общая часть/Для промежуточной аттестации\$

СОДЕРЖАНИЕ ОЦЕНОЧНЫХ СРЕДСТВ ТЕКУЩЕГО КОНТРОЛЯ

I. Оценочные средства для оценки запланированных результатов обучения по дисциплине, соотнесенных с индикаторами достижения компетенций

Индекс компетенции	Индикатор	Запланированные результаты обучения по дисциплине	Контрольная точка
ПК-2	ИД-1 _{ПК-2} Формализует описание бизнес-процессов	Знать: современные методологии разработки программного обеспечения Уметь: организовывать команду разработки программного продукта	Этап анализа в жизненном цикле программного обеспечения. Методы сбора и анализа требований. Разработка технического задания на программный продукт. (Лабораторная работа) Применение дополнительных средств поддержки жизненного цикла программного обеспечения. (Лабораторная работа)
ПК-2	ИД-2 _{ПК-2} Демонстрирует умение следить за изменениями требований	Знать: дополнительные средства поддержки жизненного цикла программного обеспечения. Уметь: анализировать архитектуру программных систем, оценивать уместность применения той или иной разновидности архитектурного решения	Практическое применение методологии IDEF1X. Использование унифицированного языка моделирования (UML) для объектно-ориентированного анализа и проектирования (Лабораторная работа) Применение дополнительных средств поддержки жизненного цикла программного обеспечения. (Лабораторная работа)
ПК-2	ИД-3 _{ПК-2} Формализует новые требования к ПО	Знать: современные архитектурные концепции, применяемые при разработке программных	Этап анализа в жизненном цикле программного обеспечения. Методы сбора и анализа требований. Разработка технического задания на программный продукт. (Лабораторная работа) Практическое применение методологии IDEF1X. Использование унифицированного языка моделирования (UML) для объектно-

		систем Уметь: фиксировать требования к ПО в системах управления требованиями	ориентированного анализа и проектирования (Лабораторная работа)
ПК-4	ИД-1 _{ПК-4} Выбирает методы анализа и проектирования ПО с применением CASE-средств	Знать: современные технологии разработки программного обеспечения Уметь: применять современные CASE-средства	Разработка диаграмм IDEF0, разработка диаграммы потоков данных (DFD, от англ. Data flow diagram), разработка диаграммы IDEF3 (Семинар) Применение дополнительных средств поддержки жизненного цикла программного обеспечения. (Лабораторная работа)
ПК-6	ИД-1 _{ПК-6} Демонстрирует умение выполнять работы по сопровождению внедренных информационных систем	Знать: методику организации процесса сопровождения внедренных информационных систем Уметь: создавать и внедрять стандарты кодирования и проектирования	Применение дополнительных средств поддержки жизненного цикла программного обеспечения. (Лабораторная работа)

II. Содержание оценочных средств. Шкала и критерии оценивания

КМ-1. Этап анализа в жизненном цикле программного обеспечения. Методы сбора и анализа требований. Разработка технического задания на программный продукт.

Формы реализации: Компьютерное задание

Тип контрольного мероприятия: Лабораторная работа

Вес контрольного мероприятия в БРС: 10

Процедура проведения контрольного мероприятия: Проводится в форме обсуждения требований к демонстрационному программному продукту на практическом занятии с преподавателем в роли заказчика-источника требований. Затем проводится в форме выполнения и защиты лабораторной работы.

Краткое содержание задания:

Обсудить требований к демонстрационному программному продукту.

Цель работы

Изучить основные принципы и получить базовые навыки подготовки технических заданий на разработку программного обеспечения.

Порядок выполнения работы

1. Подготовить описание предметной области
 - разработать краткое введение в предметную область;
 - выделить основные элементы предметной области и их взаимосвязи;
 - определить особенности и ограничения предметной области;
 - подготовить словарь используемых терминов и сокращений.
2. Сформулировать цель создания системы
 - сформулировать цель создания системы – как ответ на вопрос что за процесс в предметной области будет автоматизирован;
 - определить назначение системы, классифицировать существующие аналоги;
 - определить целевую аудиторию и ожидаемый уровень использования системы.
3. Осуществить детализацию функций системы
 - осуществить изучение потребностей заказчика;
 - выделить категории пользователей;
 - определить функциональные требования для пользователей каждой категории;
 - подготовить описание функций системы.
4. Произвести анализ ограничений
 - провести анализ аппаратных особенностей и ограничений;
 - провести анализ топологии и особенностей развертывания;
 - определить технологические ограничения.
5. Сформировать совокупный список требований к системе
 - если система предполагает интерактивность в общении с пользователем, то определить функциональные требования (описывают в динамике сценарии взаимодействия посетителя с системой) и структуру данных;
 - выделить специфические требования (например, многоязычность, требования к дизайну экранов оператора);
 - сформировать прочие требования (например, какая документация должна быть предоставлена разработчиком);

- сформировать итоговый список требований.
6. Выработать архитектурное решение
 - обосновать выбор технологической платформы;
 - если система должна реализовывать специфическую бизнес-логику, в которой обычно хорошо разбирается заказчик и плохо - исполнитель, эта логика должна быть задокументирована в техническом задании максимально подробно;
 - подготовить модульную структуру системы;
 - подготовить детализированное описание подсистем.
 7. Подготовить календарный план
 - осуществить оценку сложности реализации подсистем;
 - разбить проект на работы, построить сетевой график;
 - осуществить оценку сроков и стоимости выполнения работ.
 8. Скомпоновать из полученных материалов текст технического задания.

Контрольные вопросы/задания:

Знать: современные методологии разработки программного обеспечения	1.Какова структура технического задания 2.Какие модели жизненного цикла ПО вы знаете?
Уметь: фиксировать требования к ПО в системах управления требованиями	1.Перечислите ключевые атрибуты требований.

Описание шкалы оценивания:

Оценка: 5

Нижний порог выполнения задания в процентах: 70

Описание характеристики выполнения знания: Оценка "отлично" выставляется если задание выполнено в полном объеме или выполнено преимущественно верно

Оценка: 4

Нижний порог выполнения задания в процентах: 60

Описание характеристики выполнения знания: Оценка "хорошо" выставляется если большинство вопросов раскрыто. выбрано верное направление для решения задач

Оценка: 3

Нижний порог выполнения задания в процентах: 50

Описание характеристики выполнения знания: Оценка "удовлетворительно" выставляется если задание преимущественно выполнено

КМ-2. Разработка диаграмм IDEF0, разработка диаграммы потоков данных (DFD, от англ. Data flow diagram), разработка диаграммы IDEF3

Формы реализации: Выполнение задания

Тип контрольного мероприятия: Семинар

Вес контрольного мероприятия в БРС: 20

Процедура проведения контрольного мероприятия: Практическое занятие

Краткое содержание задания:

Построить диаграмму IDEF0

Построить диаграмму DFD

Построить диаграмму IDEF3

Контрольные вопросы/задания:

Описание шкалы оценивания:

Оценка: 5

Нижний порог выполнения задания в процентах: 70

Описание характеристики выполнения знания: Оценка "отлично" выставляется если задание выполнено в полном объеме или выполнено преимущественно верно

Оценка: 4

Нижний порог выполнения задания в процентах: 60

Описание характеристики выполнения знания: Оценка "хорошо" выставляется если большинство вопросов раскрыто. выбрано верное направление для решения задач

Оценка: 3

Нижний порог выполнения задания в процентах: 50

Описание характеристики выполнения знания: Оценка "удовлетворительно" выставляется если задание преимущественно выполнено

КМ-3. Практическое применение методологии IDEF1X. Использование унифицированного языка моделирования (UML) для объектно-ориентированного анализа и проектирования

Формы реализации: Компьютерное задание

Тип контрольного мероприятия: Лабораторная работа

Вес контрольного мероприятия в БРС: 30

Процедура проведения контрольного мероприятия: На семинаре строятся примеры основных канонических диаграмм UML

Краткое содержание задания:

Для демонстрационного приложения разработать

Диаграмму классов

Диаграмму компонентов

Диаграмму композитной/составной структуры

Диаграмму развёртывания

Диаграмму объектов

Диаграмму пакетов

Диаграмму деятельности

Диаграмму автомата

Диаграмму вариантов использования

Диаграмму коммуникации и последовательности

Диаграмму обзора взаимодействия

Диаграмму синхронизации

Изучить методологию моделирования IDEF1X и получить навыки ее практического применения.

Порядок выполнения работы

1. Определить сущности, которые будут храниться в базе данных
2. Разместить сущности в логической модели
3. Определить ключевые атрибуты и установить связи между сущностями (с указанием мощностей связей)
4. Установить ограничения (constraints)

5. Переключиться в режим изменения физической модели и установить типы данных
6. Подготовить текстовое описание ограничений и типов данных

Требования

- Наличие текстового описания предметной области (какие в нее входят информационные объекты и какие свойства есть у этих объектов)
- Наличие логической модели
- Наличие на модели всех типов связей (идентифицирующих/не идентифицирующих, категоризации). Модель должна быть включена с отображением мощности связей и отображением ограничений.
- Наличие физической модели. Модель должна быть включена с отображением типов данных.
- Наличие списка ограничений на связях (Parent Delete, Parent Update) с комментариями

Контрольные вопросы/задания:

Знать: современные архитектурные концепции, применяемые при разработке программных систем	1. Чем отличаются физическая и логическая модели данных?
Уметь: анализировать архитектуру программных систем, оценивать уместность применения той или иной разновидности архитектурного решения	1. Перечислите основные элементы диаграммы классов. 2. Перечислите возможные элементы диаграммы компонентов.

Описание шкалы оценивания:

Оценка: 5

Нижний порог выполнения задания в процентах: 70

Описание характеристики выполнения знания: Оценка "отлично" выставляется если задание выполнено в полном объеме или выполнено преимущественно верно

Оценка: 4

Нижний порог выполнения задания в процентах: 60

Описание характеристики выполнения знания: Оценка "хорошо" выставляется если большинство вопросов раскрыто. выбрано верное направление для решения задач

Оценка: 3

Нижний порог выполнения задания в процентах: 50

Описание характеристики выполнения знания: Оценка "удовлетворительно" выставляется если задание преимущественно выполнено

КМ-4. Применение дополнительных средств поддержки жизненного цикла программного обеспечения.

Формы реализации: Компьютерное задание

Тип контрольного мероприятия: Лабораторная работа

Вес контрольного мероприятия в БРС: 40

Процедура проведения контрольного мероприятия: Проводится в форме выполнения и защиты лабораторной работы.

Краткое содержание задания:

1. Завести задачу на разработку демонстрационного продукта, декомпозировать на подзадачи
2. Создать проект
3. Использовать систему контроля версий GIT, опубликовать заготовку.
4. Настроить автоматическую сборку

Контрольные вопросы/задания:

Знать: дополнительные средства поддержки жизненного цикла программного обеспечения.	1.Какие модели версионирования вы знаете?
Знать: современные технологии разработки программного обеспечения	1.Для чего нужны системы управления задачами? 2.Какие типы системы контроля версий вы знаете?
Знать: методику организации процесса сопровождения внедренных информационных систем	1.Как можно отследить изменение в коде программы, которое привело к ошибке?
Уметь: организовывать команду разработки программного продукта	1.Как исправить конфликт слияния?
Уметь: создавать и внедрять стандарты кодирования и проектирования	1.Как создать новый репозиторий git?

Описание шкалы оценивания:

Оценка: 5

Нижний порог выполнения задания в процентах: 70

Описание характеристики выполнения знания: Оценка "отлично" выставляется если задание выполнено в полном объеме или выполнено преимущественно верно

Оценка: 4

Нижний порог выполнения задания в процентах: 60

Описание характеристики выполнения знания: Оценка "хорошо" выставляется если большинство вопросов раскрыто. выбрано верное направление для решения задач

Оценка: 3

Нижний порог выполнения задания в процентах: 50

Описание характеристики выполнения знания: Оценка "удовлетворительно" выставляется если задание преимущественно выполнено

СОДЕРЖАНИЕ ОЦЕНОЧНЫХ СРЕДСТВ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

3 семестр

Форма промежуточной аттестации: Зачет с оценкой

Процедура проведения

Зачет устный, по билетам. Студенту дается 30 минут на подготовку, во время ответа задаются дополнительные вопросы на темы, рассмотренные в рамках курса.

1. Перечень компетенций/индикаторов и контрольных вопросов проверки результатов освоения дисциплины

1. Компетенция/Индикатор: ИД-1ПК-2 Формализует описания бизнес-процессов

Вопросы, задания

1.

Введение. Понятие CASE-систем и CASE-технологий.

1. Основные определения
2. Основная задача профессионального программирования
3. Методология программирования
4. Базовая последовательность решения основной задачи профессионального программирования
5. Понятие жизненного цикла программного обеспечения
6. Особенности разработки современных крупных программных систем
7. CASE-технология и CASE-средства
8. Особенности современных CASE-средств
9. Эволюция CASE-средств

2.

Классификация CASE-средств.

1. Варианты классификации CASE-средств
2. Классификация CASE-средств по типам - средства анализа, средства проектирования приложений, средства проектирования баз данных, средства разработки, средства реинжиниринга, средства управления
3. Классификация CASE-средств по уровням - низкоуровневые (утилиты), средние (поддержка этапов анализа требований и проектирования спецификаций и структуры программного обеспечения), верхнеуровневые (средства планирования)
4. Классификация CASE-средств по категориям
5. Классификация по типу решаемых задач

3.

Интегрированные CASE-средства.

1. Понятие интегрированного CASE-средства
2. Особенности интегрированных CASE-средств
3. Структура современного интегрированного CASE-средства

4.

Жизненный цикл программного обеспечения.

1. Жизненный цикл программного обеспечения

2. Модели жизненного цикла программного обеспечения
3. Каскадная модель
4. Поэтапная модель с возвратами
5. Спиральная модель
6. Итеративная модель
7. Модель процессов Rational Unified Process
8. Модель процессов Microsoft Solutions Framework
9. Модели жизненных циклов программного обеспечения по гибким (agile) методологиям
10. Причины возникновения ошибок при разработке программных средств.
11. CASE-модель жизненного цикла программного обеспечения.

5.

[Области применения CASE-технологий.](#)

1. Классификация технологии разработки программных систем.
2. Процесс разработки программного обеспечения с использованием CASE-средств.

6.

[Методология Microsoft Solutions Framework.](#)

1. Методология разработки программных систем MSF (Microsoft Solutions Framework). Обзор, основные концепции.
2. Модель процессов в MSF.
3. Этап анализа в MSF.
4. Этап планирования в MSF.
5. Этап разработки в MSF.
6. Этапы контроля качества и внедрения в MSF.
7. Модель команды разработчиков в MSF. Преимущества и недостатки.
8. Управление проектом в MSF. Общий обзор. Дисциплина управления проектом. Масштабируемость.
9. Иерархическая структура работ (WBS).
10. Оценка сроков разработки. Составление календарного графика работ в проекте.
11. Диаграммы вариантов использования системы и сценариев использования системы.
12. Управление компромиссами в MSF.
13. Стратегия выпуска версий в MSF.

7.

[Гибкие и адаптивные методологии разработки программного обеспечения.](#)

1. Введение
2. Принципы и ценности agile, включенные в манифест agile
3. Жизненный цикл agile
4. Экстремальное программирование (Методология XP) : особенности, преимущества, недостатки.
5. Методология OpenUP
6. Методология FDD
7. Методология SCRUM

Материалы для проверки остаточных знаний

1. Поясните различие методологии и технологии.
2. Что такое жизненный цикл программного обеспечения?

3. Какие виды жизненных циклов программного обеспечения вы знаете?

2. Компетенция/Индикатор: ИД-2ПК-2 Демонстрирует умение следить за изменениями требований

Вопросы, задания

1.

Моделирование

1. Задача моделирования
2. Виды моделей
3. Процесс моделирования

2.

Унифицированный язык моделирования.

1. Язык моделирования UML. Синтаксис. Семантика. Пакеты. Канонические диаграммы.
2. Диаграммы вариантов использования и сценарии.
3. Диаграммы классов и их использование.
4. Диаграммы кооперации и диаграммы последовательности.
5. Диаграммы состояний и диаграммы деятельности.
6. Диаграммы компонентов и диаграммы развертывания.
7. Обзор UML 2.0

3.

Принципы проектирования сложных систем.

1. Принципы работы с сложными системами.
2. Абстракция.
3. Уточнение.
4. Модульная разработка. Принципы разбиения системы на модули. Выделение интерфейсов и сокрытие информации. Адекватность, полнота, минимальность и простота интерфейсов. Разделение ответственности. Слабая связность модулей и сильное родство функций в одном модуле.
5. Повторное использование.

Материалы для проверки остаточных знаний

1. В чем состоит основная задача профессионального программирования?

3. Компетенция/Индикатор: ИД-3ПК-2 Формализует новые требования к ПО

Вопросы, задания

1.

Техническое задание на программный продукт.

1. Что такое техническое задание
2. Для чего нужно разрабатывать техническое задание
3. Структура технического задания
4. Порядок разработки технического задания на программный продукт

2.

Технология внедрения CASE-средств.

1. Определение потребностей в CASE-средствах.
2. Оценка и выбор CASE-средств.
3. Выполнение пилотного проекта.
4. Практическое внедрение CASE-средств.

Материалы для проверки остаточных знаний

1. Перечислите особенности разработки современных крупных программных систем
2. Перечислите особенности современных CASE-средств

4. Компетенция/Индикатор: ИД-1_{ПК-4} Выбирает методы анализа и проектирования ПО с применением CASE-средств

Вопросы, задания

1.

[Этап анализа в жизненном цикле программного обеспечения.](#)

1. Цель этапа анализа
2. Требование. Виды требований
3. Методологические аспекты анализа целей и требований к разрабатываемому программному обеспечению.
4. Основные предметы анализа требований
5. Результаты стадии анализа
6. Этапы работы с требованиями
7. Механизмы извлечения требований
8. Механизмы анализа требований
9. Спецификация требований
10. Механизмы проверки требований
11. Проблемы, с которыми сталкивается системный аналитик
12. Подходы к анализу и проектированию
13. Особенности анализа и проектирования крупных систем

2.

[Обзор методологий анализа и проектирования.](#)

1. Методология. Роль методологии.
2. Проектирование, ориентированное на данные.
3. Функционально-ориентированное (структурное) проектирование программного обеспечения.
4. Структурная методология (SADT).
5. Объектно-ориентированная методология (ООАП)
6. Диаграммные методологии проектирования программного обеспечения.
7. Агентно-ориентированная методология
8. Сервис-ориентированная методология

3.

[Методология SADT.](#)

1. Введение в методологию структурного анализа и проектирования SADT
2. Процесс моделирования в SADT
3. Принципы функционального моделирования
4. SADT-модель. Свойства SADT-модели
5. Структурные методологии

4.

[Методология IDEF0.](#)

1. Методология функционального моделирования IDEF0
2. Элементы диаграмм IDEF0. Типы связей
3. Понятие туннелирования

5.

[Методология DFD.](#)

1. Введение в методологию моделирования потоков данных (DFD)
2. Диаграммы потоков данных (Data Flow Diagram)
3. Необходимость построения модели потоков данных
4. Модель DFD. Элементы диаграмм DFD
5. Принцип построения моделей DFD
6. Построение иерархии диаграмм потоков данных. Критерии завершения детализации
7. Проверка контекстных диаграмм
8. Процесс моделирования в DFD

6.

[Методология IDEF3.](#)

1. Методология IDEF3
2. Отличие IDEF0 от IDEF3
3. Возможности IDEF3
4. Два типа диаграмм IDEF3
5. Диаграмма описания последовательности этапов процесса (Process Flow Description Diagrams, PFDD)
6. Диаграммами состояния объекта в и его трансформаций процессе (Object State Transition Network, OSTN)

7.

[Методология ARIS.](#)

1. Обзор методологии ARIS
2. Основные группы моделей и нотации ARIS
3. Сравнение методологии ARIS с методологиями IDEF3, IDEF0 и DFD

[Нотация ARIS eEPC.](#)

1. Основные элементы нотации ARIS eEPC
2. Правила построения моделей в нотации ARIS eEPC
3. Рекомендации по построению диаграмм ARIS eEPC

[Нотация ARIS InformationFlow.](#)

[Нотация ARIS Application System Type.](#)

8.

[Методология IDEF1X.](#)

1. Нормализация данных.
2. Обзор графического языка IDEF1X

9.

[Объектно-ориентированная методология разработки программного обеспечения.](#)

1. Методики объектно-ориентированного анализа и проектирования.
2. Классификация, основные этапы и задачи объектно-ориентированных методов анализа и проектирования.
3. Развитие объектно-ориентированной методологии

[Методология онтологического моделирования IDEF5.](#)

[Современные технологии объектно-ориентированного анализа и проектирования программных систем.](#)

10.

[Методология Rational Unified Process \(RUP\).](#)

Материалы для проверки остаточных знаний

1. Перечислите методы анализа требований

5. Компетенция/Индикатор: ИД-1_{ПК-6} Демонстрирует умение выполнять работы по сопровождению внедренных информационных систем

Вопросы, задания

1.

[Влияние CASE-средств на надежность программных средств.](#)

1. Надежность программного обеспечения (ПО).
2. CASE-средства и надежность программного обеспечения.
3. Контроль качества программного обеспечения.

2.

[Дополнительные средства поддержки жизненного цикла разработки программного обеспечения. Классификация инструментальных систем.](#)

1. Обзор дополнительных средств поддержки жизненного цикла разработки программного обеспечения.
2. Классификация инструментальных систем.

3.

[Системы управления задачами и заявками.](#)

1. Предпосылки появления систем управления задачами и заявками.
2. Системы отслеживания ошибок (Bug-tracking). Основные понятия. Обзор.
3. Система отслеживания ошибок Bugzilla.
4. Системы управления задачами и заявками. Основные понятия. Обзор.
5. Система управления задачами JIRA.
6. Система управления задачами TracStudio.

4.

[Системы управления версиями.](#)

1. Системы управления версиями. Основные понятия. Обзор.
2. Модели версионирования.
3. Revision Control System (RCS).
4. Система управления версиями CVS.
5. Система управления версиями Subversion (SVN). Основные возможности. Внутренняя архитектура, компоненты.
6. Система управления версиями Perforce.

7. Система управления версиями GIT.
8. Система управления проектом Borland StarTeam. Обзор. Контроль версий. Обзор. Планирование работ. Накопление знаний.

5.

Системы автоматизации сборки программных средств.

1. Автоматизация процесса сборки программного продукта.
2. Сборка с помощью командных файлов
3. Apache Ant.
4. MSBuild
5. Компонентная сборка.
6. Apache Maven 2

Материалы для проверки остаточных знаний

1. Чем отличаются императивные системы автоматической сборки от декларативных?

II. Описание шкалы оценивания

Оценка: 5

Нижний порог выполнения задания в процентах: 70

Описание характеристики выполнения знания: Работа выполнена в рамках "продвинутого" уровня. Ответы даны верно, четко сформулированные особенности практических решений

Оценка: 4

Нижний порог выполнения задания в процентах: 60

Описание характеристики выполнения знания: Работа выполнена в рамках "базового" уровня. Большинство ответов даны верно. В части материала есть незначительные недостатки

Оценка: 3

Нижний порог выполнения задания в процентах: 50

Описание характеристики выполнения знания: Работа выполнена в рамках "порогового" уровня. Основная часть задания выполнена верно. на вопросы углубленного уровня

III. Правила выставления итоговой оценки по курсу