

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«Национальный исследовательский ядерный университет «МИФИ»  
ИНСТИТУТ ИНТЕЛЛЕКТУАЛЬНЫХ КИБЕРНЕТИЧЕСКИХ СИСТЕМ  
КАФЕДРА КИБЕРНЕТИКИ

ОДОБРЕНО УМС ИИКС

Протокол № 8/1/2025

от 25.08.2025 г.

**РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ**

**ТЕХНОЛОГИЯ ПРОГРАММИРОВАНИЯ КИБЕРНЕТИЧЕСКИХ СИСТЕМ (УПРАВЛЕНИЕ  
ПРОГРАММНЫМИ ПРОЕКТАМИ)**

Направление подготовки  
(специальность)

[1] 09.03.04 Программная инженерия

Семестр	Трудоемкость, кред.	Общий объем курса, час.	Лекции, час.	Практич. занятия, час.	Лаборат. работы, час.	В форме практической подготовки/ В	СРС, час.	КСР, час.	Форма(ы) контроля, экз./зач./КР/КП
7	3-4	108- 144	32	32	0		8-44	0	Э
8	2	72	16	16	0		13	0	Э
Итого	5-6	180- 216	48	48	0	16	21-57	0	

## АННОТАЦИЯ

Дисциплина призвана обеспечить освоение студентами базовых теоретических знаний и практических приемов, необходимых для организации процесса коллективной разработки сложных программных систем. Основное внимание в курсе уделяется аспектам разработки, связанным с разработкой сертифицируемого программного обеспечения.

### 1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

Целями освоения учебной дисциплины являются:

- усвоение основных процессов жизненного Цикла сертифицируемых программных разработок;
- получение практических навыков разработки и заполнения необходимых технологических документов.

### 2. МЕСТО УЧЕБНОЙ ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП ВО

Дисциплина относится к факультативной дисциплиной для студента.

Для успешного обучения требуются знания в объеме первых трех курсов обучения, в том числе:

- Информатика;
- Языки программирования и методы программирования; основы архитектуры параллельных вычислительных систем;
- Операционные системы;

### 3. ФОРМИРУЕМЫЕ КОМПЕТЕНЦИИ И ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

Универсальные и(или) общепрофессиональные компетенции:

Код и наименование компетенции	Код и наименование индикатора достижения компетенции
--------------------------------	--

Профессиональные компетенции в соответствии с задачами и объектами (областями знаний) профессиональной деятельности:

Задача профессиональной деятельности (ЗПД)	Объект или область знания	Код и наименование профессиональной компетенции; Основание (профессиональный стандарт-ПС, анализ опыта)	Код и наименование индикатора достижения профессиональной компетенции
научно-исследовательский			
- участие в проведении научных исследований (экспериментов, наблюдений и количественных	- программный проект (проект разработки программного продукта) -	ПК-11 [1] - способен к формализации в своей предметной области с учетом ограничений используемых методов	З-ПК-11[1] - Знать методы формализации в своей предметной области с учетом ограничений

измерений), связанных с объектами профессиональной деятельности (программными продуктами, проектами, процессами, методами и инструментами программной инженерии), в соответствии с утвержденными заданиями и методиками; - построение моделей объектов профессиональной деятельности с использованием инструментальных средств компьютерного моделирования; - составление описания проводимых исследований, подготовка данных для составления обзоров и отчетов;	программный продукт (создаваемое программное обеспечение) - процессы жизненного цикла программного продукта - методы и инструменты разработки программного продукта	исследования  <i>Основание:</i> Профессиональный стандарт: 40.011	используемых методов исследования; У-ПК-11[1] - Уметь формализовать в своей предметной области ; В-ПК-11[1] - Владеть методами формализации в своей предметной области с учетом ограничений используемых методов исследования
--	---	--	---

#### 4. ВОСПИТАТЕЛЬНЫЙ ПОТЕНЦИАЛ ДИСЦИПЛИНЫ

Направления/цели воспитания	Задачи воспитания (код)	Воспитательный потенциал дисциплин
Профессиональное воспитание	Создание условий, обеспечивающих, формирование ответственности за профессиональный выбор, профессиональное развитие и профессиональные решения (B18)	Использование воспитательного потенциала дисциплин профессионального модуля для формирования у студентов ответственности за свое профессиональное развитие посредством выбора студентами индивидуальных образовательных траекторий, организации системы общения между всеми участниками образовательного процесса, в том числе с использованием новых информационных технологий.
Профессиональное воспитание	Создание условий, обеспечивающих, формирование научного	1.Использование воспитательного потенциала дисциплин/практик «Научно-исследовательская

	<p>мировоззрения, культуры поиска нестандартных научно-технических/практических решений, критического отношения к исследованиям лженаучного толка (B19)</p>	<p>работа», «Проектная практика», «Научный семинар» для:</p> <ul style="list-style-type: none"> <li>- формирования понимания основных принципов и способов научного познания мира, развития исследовательских качеств студентов посредством их вовлечения в исследовательские проекты по областям научных исследований.</li> </ul> <p>2.Использование воспитательного потенциала дисциплин "История науки и инженерии", "Критическое мышление и основы научной коммуникации", "Введение в специальность", "Научно-исследовательская работа", "Научный семинар" для:</p> <ul style="list-style-type: none"> <li>- формирования способности отделять настоящие научные исследования от лженаучных посредством проведения со студентами занятий и регулярных бесед;</li> <li>- формирования критического мышления, умения рассматривать различные исследования с экспертной позиции посредством обсуждения со студентами современных исследований, исторических предпосылок появления тех или иных открытий и теорий.</li> </ul>
Профессиональное воспитание	<p>Создание условий, обеспечивающих, формирование профессионально значимых установок: не производить, не копировать и не использовать программные и технические средства, не приобретённые на законных основаниях; не нарушать признанные нормы авторского права; не нарушать тайны передачи сообщений, не практиковать вскрытие информационных систем и сетей передачи данных; соблюдать конфиденциальность доверенной информации (B40)</p>	<p>1. Использование воспитательного потенциала дисциплин "Информатика (Основы программирования)", Программирование (Объектно-ориентированное программирование)", "Программирование (Алгоритмы и структуры данных)" для формирования культуры написания и оформления программ, а также привития навыков командной работы за счет использования систем управления проектами и контроля версий.</p> <p>2.Использование воспитательного потенциала дисциплины "Проектная практика" для формирования культуры решения</p>

		<p>изобретательских задач, развития логического мышления, путем погружения студентов в научную и инновационную деятельность института и вовлечения в проектную работу.</p> <p>3.Использование воспитательного потенциала профильных дисциплин для формирования навыков цифровой гигиены, а также системности и гибкости мышления, посредством изучения методологических и технологических основ обеспечения информационной безопасности и кибербезопасности при выполнении и защите результатов учебных заданий и лабораторных работ по криптографическим методам защиты информации в компьютерных системах и сетях.</p> <p>4.Использование воспитательного потенциала дисциплин "</p> <p>"Информатика (Основы программирования)",</p> <p>Программирование (Объектно-ориентированное программирование)",</p> <p>"Программирование (Алгоритмы и структуры данных)" для формирования культуры безопасного программирования посредством тематического акцентирования в содержании дисциплин и учебных заданий.</p> <p>5.Использование воспитательного потенциала дисциплины</p> <p>"Проектная практика" для формирования системного подхода по обеспечению информационной безопасности и кибербезопасности в различных сферах деятельности посредством исследования и перенятия опыта постановки и решения научно-практических задач организациями-партнерами.</p>
--	--	---

## 5. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

Разделы учебной дисциплины, их объем, сроки изучения и формы контроля:

№ п.п	Наименование раздела учебной дисциплины	Недели	Лекции/ Практ. (семинары) / Лабораторные работы, час.	Обязат. текущий контроль (форма*, неделя)	Максимальный балл за раздел**	Аттестация раздела (форма*, неделя)	Индикаторы освоения компетенции
	<i>7 Семестр</i>						
1	Раздел1	1-8	16/16/0		25	КИ-8	3-ПК-11, У-ПК-11, В-ПК-11
2	Раздел2	9-16	16/16/0		25	КИ-16	3-ПК-11, У-ПК-11, В-ПК-11
	<i>Итого за 7 Семестр</i>		32/32/0		50		
	<b>Контрольные мероприятия за 7 Семестр</b>				50	Э	3-ПК-11, У-ПК-11, В-ПК-11
	<i>8 Семестр</i>						
1	Раздел1	1-4	8/8/0		25	КИ-4	3-ПК-11, У-ПК-11, В-ПК-11
2	Раздел2	5-8	8/8/0		25	КИ-8	3-ПК-11, У-ПК-11, В-ПК-11
	<i>Итого за 8 Семестр</i>		16/16/0		50		
	<b>Контрольные мероприятия за 8 Семестр</b>				50	Э	3-ПК-11, У-ПК-11, В-ПК-11

\* – сокращенное наименование формы контроля

\*\* – сумма максимальных баллов должна быть равна 100 за семестр, включая зачет и (или) экзамен

Сокращение наименований форм текущего контроля и аттестации разделов:

Обозначение	Полное наименование
КИ	Контроль по итогам
Э	Экзамен

## КАЛЕНДАРНЫЙ ПЛАН

Недели	Темы занятий / Содержание	Лек., час.	Пр./сем., час.	Лаб., час.
	<i>7 Семестр</i>	32	32	0
<b>1-8</b>	<b>Раздел1</b>	16	16	0
1	<b>Введение</b> История развития технологий и инструментов	Всего аудиторных часов		
		2	2	0

	программирования. Проблемы качества программного обеспечения (ПО). Основные характеристики программных систем. Основные характеристики процесса разработки ПО. Проблемы разработки ПО. Взаимосвязь характеристик ПО и процесса его разработки.	Онлайн		
		0	0	0
2 - 3	<b>Технология программирования как инженерная дисциплина</b> Основные понятия технологии программирования. Свод знаний о программной инженерии (SWEBOOK). Цели и история проекта SWEBOOK. Области знаний SWEBOOK.	Всего аудиторных часов		
		4	4	0
		Онлайн		
		0	0	0
4 - 5	<b>Жизненный цикл коллективной разработки ПО</b> Определение понятия жизненного цикла. Процессы жизненного цикла. Проблемы коллективной разработки ПО. Принципы коллективной разработки ПО и модели команд разработчиков.	Всего аудиторных часов		
		4	4	0
		Онлайн		
		0	0	0
6	<b>Процессы жизненного цикла ПО</b> Процессная модель разработки ПО. Основные процессы разработки ПО: планирование, определение требований, проектирование, кодирование, интеграция, верификация, управление конфигурацией. Процессы и проблемы сертификации ПО.	Всего аудиторных часов		
		2	2	0
		Онлайн		
		0	0	0
7 - 8	<b>Основы управления проектами</b> Концепции проекта и управления проектом. Проект как система. Цели, требования, среда, участники проекта. Жизненный цикл проекта. Планирование, оценка, управление рисками, обеспечение качества проекта	Всего аудиторных часов		
		4	4	0
		Онлайн		
		0	0	0
9-16	<b>Раздел2</b>	16	16	0
9 - 11	<b>Модели жизненного цикла ПО</b> Каскадные, эволюционные и итерационные версии моделей жизненного цикла. Agile технологии. Понятие промышленного программирования.	Всего аудиторных часов		
		6	6	0
		Онлайн		
		0	0	0
12 - 13	<b>Стандарты разработки ПО</b> Стандарты IEEE, ISO, ГОСТ. Использование стандартов в процессах разработки ПО. Стандарты предприятия и проекта.	Всего аудиторных часов		
		4	4	0
		Онлайн		
		0	0	0
14 - 16	<b>Требования к ПО</b> Виды требований в процессе разработки ПО. Бизнес-требования, пользовательские требования, системные требования, требования высокого уровня, требования низкого уровня. Трассировка требований различных уровнях. Процесс разработки требований. Проблемы разработки требований. Управление требованиями. Спецификации требований. Стандарты спецификации требований.	Всего аудиторных часов		
		6	6	0
		Онлайн		
		0	0	0
	8 Семестр	16	16	0
1-4	<b>Раздел1</b>	8	8	0
1	<b>Процесс проектирования ПО</b> Цели и мероприятия процесса проектирования ПО. Методологии проектирования. Критерии проектирования. Стандарты проектирования ПО. Описание проекта ПО. Архитектура ПО. Требования низкого уровня.	Всего аудиторных часов		
		2	2	0
		Онлайн		
		0	0	0
2 - 3	<b>Стандарты разработки сертифицируемого ПО (DO-</b>	Всего аудиторных часов		

	<b>178В/С, КТ-178В)</b> Цели, содержание и структура документов DO-178В/С. Процессы жизненного цикла, определенные в DO-178В/С: процесс планирования ПО, процессы разработки ПО, интегральные процессы, процесс сертификации. Цели и мероприятия процессов жизненного цикла. Уровни ПО. Данные жизненного цикла ПО. Потoki данных жизненного цикла.	4	4	0
		Онлайн		
		0	0	0
4	<b>Управление конфигурациями</b> Цели и мероприятия процесса управления конфигурациями. Идентификация конфигурации. Трассируемость элементов конфигурации. Категории контроля данных. Контроль версий. Базовые версии. Индексы конфигурации. Управление изменениями, сообщения о проблемах, запросы на изменение. Системы управления конфигурациями. Информационная модель процесса разработки ПО.	Всего аудиторных часов		
		2	2	0
		Онлайн		
5-8	<b>Раздел2</b>	8	8	0
		Всего аудиторных часов		
		2	2	0
5	<b>Процесс верификации ПО</b> Цели и мероприятия процесса верификации ПО. План верификации. Рассмотрения, анализы и испытания как виды деятельности в процессе верификации.	Онлайн		
		0	0	0
		Всего аудиторных часов		
6	<b>Рассмотрения и анализы ПО</b> Рассмотрения и анализы в процессе верификации. Различие между рассмотрениями и анализами. Рассмотрения и анализы требований высокого уровня. Рассмотрения и анализы требований низкого уровня. Рассмотрения и анализы архитектуры ПО. Рассмотрения и анализы исходного кода ПО.	2	2	0
		Онлайн		
		0	0	0
7 - 8	<b>Тестирование ПО</b> Тестирование в процессе верификации. Способы тестирования. Тестирование черного ящика, белого ящика, серого ящика. Тестовые планы и тестовые процедуры. Среда тестирования. Тесты на основе требований. Робастные тесты. Интеграционные и низкоуровневые тесты. Анализ структурного покрытия ПО. Критерии структурного покрытия. Рассмотрения и анализы тестовых примеров, тестовых процедур и результатов тестов.	Всего аудиторных часов		
		4	4	0
		Онлайн		
		0	0	0

Сокращенные наименования онлайн опций:

Обозначение	Полное наименование
ЭК	Электронный курс
ПМ	Полнотекстовый материал
ПЛ	Полнотекстовые лекции
ВМ	Видео-материалы
АМ	Аудио-материалы
Прз	Презентации
Т	Тесты
ЭСМ	Электронные справочные материалы
ИС	Интерактивный сайт

## ТЕМЫ ЛАБОРАТОРНЫХ РАБОТ

Недели	Темы занятий / Содержание
	<i>7 Семестр</i>
1 - 8	<b>Раздел 1</b> На практических и лабораторных занятиях проводится коллективная разработка учебного проекта по выбору студентов (преподавателем предлагается на выбор несколько проектов). Реализуется модель промышленной разработки ПО. Из студентов создается коллектив разработчиков, между ними распределяются роли. Коллективу ставится общая задача, которая под контролем преподавателей детализируется на более мелкие. Проводится контроль и оценка выполнения частных и общей задач отдельными студентами и коллективом в целом. Занятия проводятся каждую неделю.
9 - 16	<b>Раздел 2</b> На практических и лабораторных занятиях проводится коллективная разработка учебного проекта по выбору студентов (преподавателем предлагается на выбор несколько проектов). Реализуется модель промышленной разработки ПО. Из студентов создается коллектив разработчиков, между ними распределяются роли. Коллективу ставится общая задача, которая под контролем преподавателей детализируется на более мелкие. Проводится контроль и оценка выполнения частных и общей задач отдельными студентами и коллективом в целом. Занятия проводятся каждую неделю.
	<i>8 Семестр</i>
1 - 8	<b>Раздел 1</b> На практических и лабораторных занятиях проводится коллективная разработка учебного проекта по выбору студентов (преподавателем предлагается на выбор несколько проектов). Реализуется модель промышленной разработки ПО. Из студентов создается коллектив разработчиков, между ними распределяются роли. Коллективу ставится общая задача, которая под контролем преподавателей детализируется на более мелкие. Проводится контроль и оценка выполнения частных и общей задач отдельными студентами и коллективом в целом. Занятия проводятся каждую неделю.
9 - 16	<b>Раздел 2</b> На практических и лабораторных занятиях проводится коллективная разработка учебного проекта по выбору студентов (преподавателем предлагается на выбор несколько проектов). Реализуется модель промышленной разработки ПО. Из студентов создается коллектив разработчиков, между ними распределяются роли. Коллективу ставится общая задача, которая под контролем преподавателей детализируется на более мелкие. Проводится контроль и оценка выполнения частных и общей задач отдельными студентами и коллективом в целом. Занятия проводятся каждую неделю.

## ТЕМЫ ПРАКТИЧЕСКИХ ЗАНЯТИЙ

Недели	Темы занятий / Содержание
	<i>7 Семестр</i>
1 - 8	<b>Раздел 1</b> На практических и лабораторных занятиях проводится коллективная разработка учебного проекта по выбору студентов (преподавателем предлагается на выбор несколько проектов). Реализуется модель промышленной разработки ПО. Из студентов создается коллектив разработчиков, между ними распределяются роли. Коллективу ставится общая задача, которая под контролем преподавателей

	детализируется на более мелкие. Проводится контроль и оценка выполнения частных и общей задач отдельными студентами и коллективом в целом. Занятия проводятся каждую неделю.
9 - 16	<b>Раздел2</b> На практических и лабораторных занятиях проводится коллективная разработка учебного проекта по выбору студентов (преподавателем предлагается на выбор несколько проектов). Реализуется модель промышленной разработки ПО. Из студентов создается коллектив разработчиков, между ними распределяются роли. Коллективу ставится общая задача, которая под контролем преподавателей детализируется на более мелкие. Проводится контроль и оценка выполнения частных и общей задач отдельными студентами и коллективом в целом. Занятия проводятся каждую неделю.
	<i>8 Семестр</i>
1 - 8	<b>Раздел1</b> На практических и лабораторных занятиях проводится коллективная разработка учебного проекта по выбору студентов (преподавателем предлагается на выбор несколько проектов). Реализуется модель промышленной разработки ПО. Из студентов создается коллектив разработчиков, между ними распределяются роли. Коллективу ставится общая задача, которая под контролем преподавателей детализируется на более мелкие. Проводится контроль и оценка выполнения частных и общей задач отдельными студентами и коллективом в целом. Занятия проводятся каждую неделю.
9 - 16	<b>Раздел2</b> На практических и лабораторных занятиях проводится коллективная разработка учебного проекта по выбору студентов (преподавателем предлагается на выбор несколько проектов). Реализуется модель промышленной разработки ПО. Из студентов создается коллектив разработчиков, между ними распределяются роли. Коллективу ставится общая задача, которая под контролем преподавателей детализируется на более мелкие. Проводится контроль и оценка выполнения частных и общей задач отдельными студентами и коллективом в целом. Занятия проводятся каждую неделю.

## 6. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

### 1. Лекционные занятия:

- a. комплект электронных презентаций/слайдов,
- b. аудитория, оснащенная презентационной техникой (проектор, экран, компьютер/ноутбук)

### 2. Практические занятия:

- a. компьютерный класс,
- b. презентационная техника (проектор, экран, компьютер/ноутбук),
- c. стандартный пакет программ Microsoft Office.

## 7. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

Фонд оценочных средств по дисциплине обеспечивает проверку освоения планируемых результатов обучения (компетенций и их индикаторов) посредством мероприятий текущего, рубежного и промежуточного контроля по дисциплине.

Связь между формируемыми компетенциями и формами контроля их освоения представлена в следующей таблице:

Компетенция	Индикаторы освоения	Аттестационное мероприятие (КП 1)	Аттестационное мероприятие (КП 2)
ПК-11	З-ПК-11	Э, КИ-8, КИ-16	Э, КИ-4, КИ-8
	У-ПК-11	Э, КИ-8, КИ-16	Э, КИ-4, КИ-8
	В-ПК-11	Э, КИ-8, КИ-16	Э, КИ-4, КИ-8

### Шкалы оценки образовательных достижений

Шкала каждого контрольного мероприятия лежит в пределах от 0 до установленного максимального балла включительно. Итоговая аттестация по дисциплине оценивается по 100-балльной шкале и представляет собой сумму баллов, заработанных студентом при выполнении заданий в рамках текущего и промежуточного контроля.

Итоговая оценка выставляется в соответствии со следующей шкалой:

Сумма баллов	Оценка по 4-ех балльной шкале	Оценка ECTS	Требования к уровню освоению учебной дисциплины
90-100	5 – «отлично»	A	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, использует в ответе материал монографической литературы.
85-89	4 – «хорошо»	B	Оценка «хорошо» выставляется студенту, если он твёрдо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос.
75-84		C	
70-74		D	
65-69	3 – «удовлетворительно»	E	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.
60-64			
Ниже 60	2 – «неудовлетворительно»	F	Оценка «неудовлетворительно» выставляется студенту, который не знает значительной части программного материала, допускает существенные ошибки. Как правило, оценка «неудовлетворительно» ставится студентам, которые не могут продолжить обучение без дополнительных занятий по соответствующей дисциплине.

## **8. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ**

### **ОСНОВНАЯ ЛИТЕРАТУРА:**

1. ЭИ S34 Agile Software Development Teams : , Schmidt, Christoph. , Cham: Springer International Publishing, 2016
2. ЭИ S70 Software Quality. The Future of Systems- and Software Development : 8th International Conference, SWQD 2016, Vienna, Austria, January 18-21, 2016, Proceedings, , Cham: Springer International Publishing, 2016

### **ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА:**

1. 004 Ф28 UML. Основы : краткое рук-во по унифицированному языку моделирования, Фаулер М., Скотт К., СПб: Символ, 2002
2. 681.5 Т47 Введение в проектирование систем управления : , Тищенко Н.М., М.: Энергоатомиздат, 1986
3. 004 С38 Верификация программного обеспечения : учебное пособие, Налютин Н.Ю., Сеницын С.В., Москва: Интернет-Университет информационных технологий; Бином. Лаборатория знаний, 2008
4. 681.3 Б89 Как проектируются и создаются программные комплексы : Мифический человеко-месяц. Очерки по системному программированию, Брукс Ф.П., М.: Наука, 1979
5. ЭИ С79 Методы объектно-ориентированного описания систем и моделирования на языке UML : учеб. пособие, Тимофеев А.В., Степанова Е.Б., Москва: МИФИ, 2006
6. 004 С38 Операционные системы : учеб. пособие, Налютин Н.Ю., Сеницын С.В., Москва: МИФИ, 2006
7. 004 О-66 Технологии разработки программного обеспечения : Разработка сложных программных систем: Учеб. пособие для вузов, Орлов С.А., М.и др.: Питер, 2003
8. 004 Б42 Экстремальное программирование. Разработка через тестирование : , Бек К., М.и др.: Питер, 2003
9. 681.3 С38 Языки управления заданиями в операционных системах : Текст лекций, Сеницын С.В., М.: МИФИ, 1989

### **ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ:**

Специальное программное обеспечение не требуется

### **LMS И ИНТЕРНЕТ-РЕСУРСЫ:**

<https://online.mephi.ru/>

## **9. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ**

Специальное материально-техническое обеспечение не требуется

## **10. УЧЕБНО-МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ДЛЯ СТУДЕНТОВ**

Семинарские и лабораторные работы по курсу проходят в режиме имитации реальной программной разработки. Ведущий занятия преподаватель играет роль Заказчика.

Группа студентов образуют команду разработчиков. Роли в группе студенты выбирают и назначают исполнителей самостоятельно. Вмешательство в этот процесс преподавателей должно быть минимизировано.

Тема общего проекта выбирается студентами самостоятельно. При этом преподаватель должен помогать группе в этом путём участия в анализе предлагаемых тем и предсказания областей риска в каждой из них.

У выбираемых тем есть единственное ограничение: тема не должна "умещаться в одну голову", т.е. она должна быть настолько большой, чтобы все её спецификации не могли быть запомнены даже самым талантливым членом группы.

Возможным вариантом следует признать вариант, при котором группа продолжает проект, начатый другой группой ранее и не завершённый в связи с окончанием курса.

Следует приветствовать выбор студентами в качестве темы своего проекта одну из компонент среды сертифицируемых программных разработок (конфигурационное управление, тестирование, включая валидацию и верификацию,...).

Ещё более должно быть интересным, если один из "работающих" студентов приносит тему из места своей работы. В последнем случае этот студент автоматически становится "проводником" между заинтересованными лицами (своими работодателями) и исполнителями. При этом ведущий занятия преподаватель должен иметь право на включение собственных требований по совершенствованию разрабатываемого продукта в его спецификации.

Предложения об оплате деятельности группы следует сдержанно (без энтузиазма) приветствовать, но при этом жёстко предупредить заказчиков, что назначение этих занятий – не зарабатывание денег, а обучение студентов практике коллективной работы. Также следует поддерживать студентов в их намерении завершить свою разработку после завершения данного курса, превратив её результат в коробочный продукт. До настоящего времени было две попытки (предложения оплачивать работу группы), но обе они завершились "ничем" сразу после такого предупреждения. Была одна попытка доведения результатов проекта до уровня коробочного продукта, однако она ничем не завершилась.

Выбор студентами собственной технологии работы и используемого инструментария следует всячески поощрять и поддерживать, добиваясь реального опробования нескольких вариантов. На эту деятельность ведущему преподавателю не следует жалеть времени (ни своего, ни студенческого).

В самом начале курса следует жёстко потребовать быстрого создания сайта проекта, на котором хранить все материалы проекта, включая протоколы общих собраний и принятых решений. Следует приветствовать решения, связанные с использованием SVN, Subversion или иных подобных компонент для управления версиями создаваемых материалов. Этот сайт

следует использовать в качестве общего хранилища данных проекта и быстрого обмена сведениями между участниками разработки.

В ходе разработки ведущий преподаватель должен, с одной стороны, всячески мотивировать студентов пробовать разные роли (меняться ими) в ходе проекта и вносить полезные (и не полезные) предложения по изменению применяемой технологии, включая используемые инструменты.

Поскольку главная цель курса – научить студентов работать командой в современных российских условиях – ведущий преподаватель должен время от времени обоснованно менять требования к разрабатываемому продукту, заставляя студентов переделывать уже принятые решения.

## **11. УЧЕБНО-МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ДЛЯ ПРЕПОДАВАТЕЛЕЙ**

Семинарские и лабораторные работы по курсу проходят в режиме имитации реальной программной разработки. Ведущий занятия преподаватель играет роль Заказчика.

Группа студентов образуют команду разработчиков. Роли в группе студенты выбирают и назначают исполнителей самостоятельно. Вмешательство в этот процесс преподавателей должно быть минимизировано.

Тема общего проекта выбирается студентами самостоятельно. При этом преподаватель должен помогать группе в этом путём участия в анализе предлагаемых тем и предсказания областей риска в каждой из них.

У выбираемых тем есть единственное ограничение: тема не должна "умещаться в одну голову", т.е. она должна быть настолько большой, чтобы все её спецификации не могли быть запомнены даже самым талантливым членом группы.

Возможным вариантом следует признать вариант, при котором группа продолжает проект, начатый другой группой ранее и не завершённый в связи с окончанием курса.

Следует приветствовать выбор студентами в качестве темы своего проекта одну из компонент среды сертифицируемых программных разработок (конфигурационное управление, тестирование, включая валидацию и верификацию,...).

Ещё более должно быть интересным, если один из "работающих" студентов приносит тему из места своей работы. В последнем случае этот студент автоматически становится "проводником" между заинтересованными лицами (своими работодателями) и исполнителями. При этом ведущий занятия преподаватель должен иметь право на включение собственных требований по совершенствованию разрабатываемого продукта в его спецификации.

Предложения об оплате деятельности группы следует сдержанно (без энтузиазма) приветствовать, но при этом жёстко предупредить заказчиков, что назначение этих занятий – не зарабатывание денег, а обучение студентов практике коллективной работы. Также следует поддерживать студентов в их намерении завершить свою разработку после завершения данного курса, превратив её результат в коробочный продукт. До настоящего времени было две попытки (предложения оплачивать работу группы), но обе они завершились "ничем" сразу после такого предупреждения. Была одна попытка доведения результатов проекта до уровня коробочного продукта, однако она ничем не завершилась.

Выбор студентами собственной технологии работы и используемого инструментария следует всячески поощрять и поддерживать, добиваясь реального опробования нескольких вариантов. На эту деятельность ведущему преподавателю не следует жалеть времени (ни своего, ни студенческого).

В самом начале курса следует жёстко потребовать быстрого создания сайта проекта, на котором хранить все материалы проекта, включая протоколы общих собраний и принятых решений. Следует приветствовать решения, связанные с использованием SVN, Subversion или иных подобных компонент для управления версиями создаваемых материалов. Этот сайт следует использовать в качестве общего хранилища данных проекта и быстрого обмена сведениями между участниками разработки.

В ходе разработки ведущий преподаватель должен, с одной стороны, всячески мотивировать студентов пробовать разные роли (меняться ими) в ходе проекта и вносить полезные (и не полезные) предложения по изменению применяемой технологии, включая используемые инструменты.

Поскольку главная цель курса – научить студентов работать командой в современных российских условиях – ведущий преподаватель должен время от времени обоснованно менять требования к разрабатываемому продукту, заставляя студентов переделывать уже принятые решения.

Автор(ы):

Козырев Владимир Петрович, к.т.н.