

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное образовательное  
учреждение  
высшего профессионального образования  
«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Факультет математики, механики и компьютерных наук

Рассмотрено и рекомендовано  
на заседании кафедры

ЮФУ  
2007 г.  
Протокол № \_\_\_\_\_ от \_\_\_\_\_  
Зав. кафедрой \_\_\_\_\_.

УТВЕРЖДАЮ

Декан мехмата

\_\_\_\_\_ Ерусалимский Я.М.  
« \_\_\_\_\_ » \_\_\_\_\_ 2007 г.

**УЧЕБНО-МЕТОДИЧЕСКИЙ КОМПЛЕКС**

Специального курса  
«Программная инженерия»

для направления 010400 «Информационные технологии»  
квалификация – бакалавр информационных технологий

Составитель: к.т.н. Адигеев М.Г.

г.Ростов-на-Дону  
2007 г.

**Программа специального курса**  
**«Параллельные алгоритмы»**  
**по направлению «Информационные технологии»**

**Число часов: 68** – II семестр IV курса

**Всего: 68**

**Аудиторно:** лекции – 34

**Самостоятельная работа:** 34

## **ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

Курс предназначен для студентов, обучающихся по направлению «информационные технологии». Требования к знаниям и навыкам: от слушателей курса требуются базовые навыки разработки и тестирования программ.

**Цель и задачи курса:** ознакомить студентов современными методиками, применяемыми при индустриальной разработке программного обеспечения. В результате прохождения курса студенты приобретут представление о жизненном цикле программного продукта (ПП), технологиях и инструментах, применяемых на каждом этапе разработки ПП, освоить базовые понятия и принципы проектного менеджмента. Полученные знания и навыки позволят выпускникам факультета математики, механики и компьютерных наук быстрее вливаться в команды разработки ПО и успешнее строить карьеру в софтверном бизнесе.

Курс соответствует стандарту по направлению «Информационные технологии» (бакалавриат), а также международным рекомендациям Computer Curricula 2001.

Курс включает в себя следующие разделы:

- Введение в дисциплину.
- Обзор методологий разработки ПО.
- Управление командой разработчиков.

- Работа с требованиями к ПО.
- Планирование и проектирование архитектуры ПО.
- Планирование и контроль процесса разработки ПО.
- Методы управления рисками и сроками в процессе разработки ПО.
- Организация и методы тестирования.
- Разработка документации.
- Развертывание и сопровождение программного продукта.
- Обзор популярных методологий разработки: RUP, MSF, XP.
- Обзор отраслевых стандартов.

В качестве самостоятельной работы студентам предлагается выбрать пример программного продукта и проработать на этом примере все вопросы, разбираемые в рамках курса. Данный учебно-методический комплекс содержит вопросы и пояснения, помогающие студентам в самостоятельной работе.

В завершении курса студенты должны сдать экзамен на знание рассмотренных в курсе вопросов.

## **Учебно-тематический план курса**

### ***Раздел 1. Введение в дисциплину***

#### *Лекция 1. (2 часа)*

Классификация проектов разработки ПП. Отличие индустриальной разработки ПП от «кустарной». Разработка ПП как инженерная дисциплина. Жизненный цикл программного обеспечения. Показатели качества ПП: практичность, отказоустойчивость, надежность. Качество проекта разработки ПП. Треугольник «сроки-качество-стоимость». Основные причины неудач в проектах разработки ПП. Есть ли «серебряная пуля»? [, ,,]

#### *Лекция 2. (2 часа)*

Обзор методологий разработки ПП. Аспекты методологии: персонал, процесс, проект и продукт. Виды процессов: каскадный, спиральный, итерационный. Виды методологий: формальные и гибкие (agile). Примеры: RUP, MSF, XP (краткая характеристика). [, ,,,,]

## ***Раздел 2. Составные части методологии***

### *Лекция 3. (2 часа)*

Состав команды разработки ПО. Подбор персонала, распределение ролей. Принципы организации коммуникаций и совместной работы. [,,]

### *Лекция 4. (2 часа)*

Сбор и согласование требований к ПО. Виды требований: С- и D-требования. Методы выявления требований заказчика, управление изменениями в требованиях. Уточнение требований, прототипирование. Контроль качества требований. Структура технического задания, стандарты. [, ]

### *Лекция 5. (2 часа)*

Способы описания требований к ПО. Краткий обзор нотации UML. Программные инструменты для управления требованиями. [, ]

### *Лекции 6. (2 часа)*

Проектирование архитектуры ПО. Методы проектирования. Краткий обзор принципов объектно-ориентированного проектирования. [,]

### *Лекции 7. (2 часа)*

Способы описания архитектуры ПО. Программные средства для описания и валидации архитектуры ПО [,]

### *Лекция 8. (2 часа)*

Разработка ПО. Планирование итераций, распределение задач. Контроль качества: стандарты программирования, метрики для исходного кода. Организация и инструментальные средства коллективной разработки. []

### *Лекция 9. (2 часа)*

Управление рисками и сроками. Основные виды и источники рисков в программных проектах, методы анализа и управления рисками. Методы календарного планирования и управления отклонениями.[]

*Лекция 10. (2 часа)*

Тестирование ПО. Принципы: функциональное/модульное тестирование, регрессивное тестирование, нагрузочное тестирование. Приемы и технологии тестирования. [,,]

*Лекция 11. (2 часа)*

Планирование тестирования, взаимодействие тестировщиков и разработчиков. Тестирование документации к ПП. Интеграция, верификация и валидация системы. Инструментальные средства поддержки тестирования. [,]

*Лекция 12. (2 часа)*

Разработка документации к ПП. Пользовательская и техническая документация: структура, стандарты. []

*Лекция 13. (2 часа)*

Развертывание ПО. Планирование, управление развертыванием. Сопровождение ПО. []

### **Раздел 3. Популярныe методологии разработки ПП**

*Лекция 14. (2 часа)*

Методология RUP (Rational Unified Process). Этапы RUP: начало работ, уточнение, проектирование, передача. Дисциплины: бизнес-моделирование, управление требованиями, анализ и проектирование, реализация, тестирование, развертывание. Соотношение дисциплин и этапов. Программные инструменты, поддерживающие RUP. [,]

*Лекция 15. (2 часа)*

Методология MSF (Microsoft Solutions Framework). Основные принципы: фазы и вехи; итерации, версии ПП; целостный взгляд на разработку и внедрение. Фазы: выработка концепции, планирование, разработка,

стабилизация, внедрение. Другие составляющие MSF: модели управление командой, рисками, проектом, подготовкой. []

*Лекция 16. (2 часа)*

Методология «Экстремальное программирование» (eXtreme Programming, XP). Двенадцать принципов экстремального программирования. Методы планирования итераций, тестирования, управления изменениями. Достоинства, недостатки и область применения методологии XP. []

*Лекция 17. (2 часа)*

Отраслевые стандарты: ISO, IEEE. Модели CMM, PSP, TSP. []

## **СПИСОК ЛИТЕРАТУРЫ**

### **Основная литература**

1. Адигеев М.Г. Жизненный цикл программного обеспечения. Метод. указания. // *Готовится к публикации в рамках данной темы..*
2. Брукс Ф. Мифический человеко-месяц или как создаются программные системы. — СПб.: Символ-Плюс, 1999. — 304 с.
3. Брауде Э.Д. Технология разработки программного обеспечения. — СПб.: Питер, 2004. — 656 с.
4. Кантор М. Управление программными проектами: Практическое руководство по разработке успешного программного обеспечения. — М.: Вильямс, 2002. — 174 с.
5. Канер С., Фолк Дж., Енг Кек Нгуен. Тестирование программного обеспечения. — К.: «ДиаСофт», 2000. — 544 с.
6. Леффингуэлл Д., Уидриг Д. Принципы работы с требованиями к программному обеспечению. Унифицированный подход. — М.: Вильямс, 2002. — 448 с.
7. Бек К. Экстремальное программирование. — СПб.: Питер, 2002. — 224 с.

8. Документация по методологии Microsoft Solutions Framework (<http://www.microsoft.com/rus/msdn/msf/default.mspx>).

### **Дополнительная литература**

- Бек К., Фаулер М. Экстремальное программирование: планирование. — СПб.: Питер, 2003. — 144 с.
- Ауэр К., Миллер Р. Экстремальное программирование: постановка процесса. С первых шагов до победного конца. — СПб.: Питер, 2003. — 368 с.
- Йордон Э. Путь камикадзе. — М.: Лори, 2003.

### **Ссылки на ресурсы Интернета**

- Персональная страница Алистера Кокберна — специалиста по организации и методологии управления проектами по разработке ПО.  
<http://alistair.cockburn.us/>
- Персональная страница Мартина Фаулера — известного специалиста в области технологии и методологии разработки ПО.  
<http://www.martinfowler.com/>
- Сайт Agile Alliance — организации по развитию гибких методологий  
<http://www.AgileAlliance.org>

### **Экзаменационные вопросы**

1. Модели разработки. Основные фазы жизненного цикла ПП в методологиях RUP, MSF, XP.
2. Качество ПП: показатели качества, метрики, способы контроля, стандарты.

3. Принципы формирования команды разработчиков. Роли участников проекта.
4. Классификация требований. Методы выявления и документирования требований.
5. Способы описания требований к ПО: классификация, достоинства и недостатки.
6. Нотация UML: вид диаграмм и блоков, принципы построения.
7. Принципы объектно-ориентированного проектирования.
8. Стандарты программирования. Метрики исходного кода.
9. Виды и источники рисков в проектах разработки ПО. Принципы управления рисками.
10. Виды и приемы тестирования ПП.
11. Виды документации к ПП. Требования к документации. Тестирование документации.
12. Этапы RUP, содержание каждого этапа.
13. Дисциплины RUP. Соотношение дисциплин с ролями и фазами.
14. Базовые принципы MSF.
15. Фазы MSF, содержание каждой фазы.
16. Модель управления командой в технологии MSF.
17. Модель управления рисками в технологии MSF.
18. Модель управления проектом в технологии MSF.
19. Двенадцать принципов экстремального программирования.
20. Методы планирования итераций, тестирования, управления изменениями в технологии XP.

## **УЧЕБНО-ПРАКТИЧЕСКОЕ ПОСОБИЕ**

### ***Конспект лекций***

#### *Лекция 1.*



Вводные слова о курсе.

Отличие индустриальной разработки ПП от «кустарной» []:

- «просто программа» — завершённый продукт, пригодный для запуска своим автором на системе, на которой была разработана;
- «программный продукт» — программа, которую любой человек может запускать, тестировать, исправлять и развивать.

Программный продукт должен быть достаточно универсальным (в рамках решаемых им задач), устойчиво работать в различных ситуациях (в т.ч. при «некорректной» работе пользователя, «защита от дурака»), иметь достаточно полную документацию. По оценке Фредерика Брукса, программный продукт обычно стоит как минимум вдвое дороже, чем просто отлаженная программа с такой же функциональностью.

Таким образом, для разработки ПП необходим другой подход, нежели для разработки «просто программы». Нужен индустриальный подход, с участием команды разработчиков, более-менее формализованных регламентов. Программирование становится инженерной дисциплиной.

Будем различать термины «программное обеспечение» (ПО) — исполнимые файлы, библиотеки и прочие файлы, необходимые для работы программы, и «программный продукт» — кроме ПО включает в себя документацию и, возможно, услуги по сопровождению.

Жизненный цикл программного обеспечения — в разных методологиях описывается немного по-разному, но есть общее. Обобщённые этапы:

1. Определение требований.
2. Проектирование.
3. Разработка, тестирование.
4. Развёртывание, сопровождение.

Даже если ПП полностью протестирован и снабжен подробной документацией, он может быть неуспешным. Т.е. некачественным. Какое ПП можно назвать качественным? Показатели качества ПП []:

- Практичность — соответствие нуждам и ожиданиям *заинтересованных лиц*. Классификация заинтересованных лиц: пользователи, покупатели, инвесторы.
- Отказоустойчивость: мощность, масштабируемость.
- Надежность: средняя наработка на отказ, работоспособность, способность к восстановлению.

Качество проекта разработки ПП: качественный ПП в заданные сроки и в пределах бюджета. Треугольник «сроки-качество-стоимость».

Основные причины неудач в проектах разработки ПП:

- Заниженные оценки сроков («оптимизм разработчиков»).
- Неправильно понятые требования.
- Прочие риски.