

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования

«Новосибирский национальный исследовательский государственный
университет»

Факультет информационных технологий

УТВЕРЖДАЮ

« ___ » _____ 20__ г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Практика управления проектами

(наименование дисциплины)

Магистерская программа

Технология разработки программных систем

(наименование магистерской программы)

НАПРАВЛЕНИЕ ПОДГОТОВКИ 230100 «ИНФОРМАТИКА И
ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА»

Квалификация (степень) выпускника

Магистр

Форма обучения очная

Новосибирск

2012

Программа дисциплины «Практика управления проектами» составлена в соответствии с требованиями ФГОС ВПО к структуре и результатам освоения основных образовательных программ магистратуры по Профессиональному циклу по направлению подготовки «Информатика и вычислительная техника», а также задачами, стоящими перед Новосибирским государственным университетом по реализации Программы развития НГУ.

Автор: Мухортов Вячеслав Викторович

Факультет Информационных Технологий

Кафедра Общей Информатики

1. Цели освоения дисциплины (курса)

Необходимость для выпускников «доучиваться» на первом месте работы давно и повсеместно является узким местом высшей школы, как в России, так и за рубежом. Часто приходится слышать избитые фразы о том, что «единственное, чему учит ВУЗ – это учиться», причем проблема заключается не в недостатке у выпускника теоретических знаний, а в отсутствии навыка их применения на практике, в неумении вписать свои обширные познания в реальный технологический процесс.

Применительно к сфере информационных технологий, проблему следует уточнить следующим образом: выпускник специальности «информатика и вычислительная техника», если и знаком с производственным процессом, используемым в индустрии разработки программного обеспечения (при наличии в программах кафедр соответствующих спецкурсов), то никогда не применял их на *практике* в процессе *коллективной работы над проектом*.

Проблема усугубляется сложившейся практикой выставления оценок в ВУЗе: оценки строго индивидуальны, студент мотивирован только на личные достижения, привыкает работать в одиночку. По этой же причине дипломный проект, как правило, выполняется отнюдь не в «проектном» режиме – студент решает строго определенную задачу и успешность его «проекта» не зависит от успешности работ других членов коллектива, то есть дипломная *практика* не дает будущему специалисту один из необходимейших *практических* навыков: умение работать в команде. В то же время в реальном производстве время программистов-одиночек давно прошло, и для успеха проекта имеет значение только конечный результат коллективной работы.

Целью курса «Практика управления проектами» является получение студентом магистратуры первого *практического* навыка управления *проектом* по разработке программного обеспечения, в соответствии с технологическим процессом, принятым в индустрии.

Уникальность курса заключается в соединении практических занятий бакалавров, проходящих курс «Коллективная разработка программного обеспечения» и ма-

гистрантов данного курса: магистранты участвуют в учебном проекте в роли менеджеров, архитекторов, специалистов по качеству, бакалавры – в ролях разработчик, тестировщик, дизайнер пользовательского интерфейса и т.д.

Результаты освоения дисциплины могут и должны быть использованы студентом при работе над магистерской диссертацией, приветствуется выполнение студентами в рамках курса проектов по разработке ПО, непосредственно связанного с темой магистерской диссертации.

Курс также направлен на формирование общекультурных и профессиональных компетенций в соответствии с ФГОС ВПО по данному направлению подготовки.

2. Место дисциплины в структуре образовательной программы

Дисциплина относится к разделу дисциплин по выбору М2.ДВ3 «Вариативная часть» образовательной программы.

Курс проводится в 3 семестре магистратуры, что даёт магистрантам возможность самостоятельно выбрать наиболее подходящие для выполнения курсового проекта языки программирования и методологии разработки программ, изученные за прошлые годы обучения на курсах: «Программирование», «Объектно-ориентированное программирование», «Объектно-ориентированный анализ и проектирование», «Коллективная разработка программного обеспечения».

Материал курса «Объектно-Ориентированное программирование» необходим в полном объеме, т.к практическая часть курса требует от студента знания современных технологий разработки ПО (C++/Java/C#). Теоретическая часть курса предполагает предварительное знакомство студентов с современными процессами разработки и моделирования ПО, преподаваемыми в курсе «Объектно-Ориентированный Анализ и Дизайн».

Материал курса «Практика управления проектами» является необходимым для успешного прохождения производственной практики, а также дальнейшей работы выпускника в должностях «ведущий программист», «Senior Java-разработчик», «Senior C++-разработчик», «Senior C#-разработчик», «Senior Web-разработчик», «системный архитектор», «ведущий бизнес-аналитик», «менеджер проектов».

3. Компетенции обучающегося, формируемые в результате освоения дисциплины

В результате освоения дисциплины у обучающегося формируются следующие компетенции:

ОК-4 использует на практике умения и навыки в организации исследовательских и проектных работ, в управлении коллективом

ОК-5 способен проявлять инициативу, в том числе в ситуациях риска, брать на себя всю полноту ответственности

ОК-8 способен использовать методы, инструменты проектного управления для планирования, организации научной и практической деятельности в форме проектов

ОК-9 способен применять на практике полученные знания и навыки для разработки методик, учебных материалов, научных публикаций и докладов, отчетов, технической документации, презентаций

ПК-4 формировать технические задания и участвовать в разработке аппаратных и/или программных средств вычислительной техники

ПК-5 выбирать методы и разрабатывать алгоритмы решения задач управления и проектирования объектов автоматизации

ПК-6 применять современные технологии разработки программных комплексов с использованием CASE-средств, контролировать качество разрабатываемых программных продуктов

ПК-7 организовывать работу и руководить коллективами разработчиков аппаратных и/или программных средств информационных и автоматизированных систем

В результате освоения дисциплины студент должен:

Знать:

- основные фазы процесса разработки ПО;
- распределение ролей в проектной команде;
- методы оценки трудоемкости проектов;
- методы оценки проектных рисков;
- методы контроля за ходом проекта;

Уметь:

- распределять роли в проектной команде;
- проводить декомпозицию проекта на задачи;
- составлять план проекта;
- проводить оценку трудозатрат и рисков;
- выбирать стратегию управления рисками проекта

Владеть:

- средством планирования проектов MS Project или OpenProj;
- инструментом контроля версий SVN;

4. Структура и содержание дисциплины

Структура дисциплины

Общая трудоемкость дисциплины составляет 4 зачетных единицы, 144 часа, из них 64 аудиторных часов – лекции и семинары, и 43 часа самостоятельной работы.

№ п/ п	Раздел дисциплины	Семестр	Неделя семестра	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)				Формы текущего контроля успеваемости <i>(по неделям семестра)</i> Форма промежуточной аттестации <i>(по семестрам)</i>
				Лекции	Семинары	Самостоятельная работа	КСР	
1	Введение	3	1	2				
2	Выбор курсового проекта и формирование команд	3	2		4	2		Утверждение проекта
3	Инструменты коллективной разработки ПО	3	2	2				
4	Планирование проекта	3	3	4	8			
5	Управление рисками	3	4	2	8			
6	Обеспечение качества: организация процессов тестирования и ревью	3	5	2				Утверждение технического задания преподавателем

7	Методы текущего контроля хода выполнения проекта	3	6	4				
8	Обсуждение проектов на семинарах	3	6-18		24			
9	Реализация проекта	3	6-18		4	43	1	Защита проекта
	Экзамен						36	Экзамен
	Итого по курсу			16	48	43	37	СУММА: 144

Содержание курса лекций

1. Введение в управление проектами
 - a. Понятие проекта, проектное управление как область знаний, терминология PMI
 - b. Модели жизненного цикла ИТ проекта: каскадная, итеративная и инкрементная модели. Agile и RUP-подобные методологии.
 - c. Цикл управления ИТ проектом
 - d. Авторское право в контексте ИТ
2. Инструменты коллективной разработки ПО:
 - a. Системы контроля версий (SVN, CVS)
 - b. Системы багтрекинга (Bugzilla, JIRA)
3. Планирование проекта
 - a. Понятие плана, задачи процесса планирования, мета-план: план составления плана
 - b. Представление планов: сетевые и Гантт-диаграммы, связи и ограничения задач, распределение ресурсов
 - c. Методы проведения оценки задач проекта: аналоговая, параметрическая, экспертная оценки
 - d. PERT-анализ
 - e. Критический путь проекта, управление критическим путем

4. Обеспечение качества
 - a. Роли тест-дизайнера и тестировщика, разработка тест-сценариев
 - b. Важность ревью кода и архитектуры в обеспечении качества ПО
5. Управление рисками
 - a. Определение риска по PMI, типы и характеристики рисков
 - b. Типичные риски ИТ разработки
 - c. Понятие управления риском – планирование срывов плана
 - d. Методы идентификации, качественной и количественной оценки рисков ИТ проекта
 - e. Стратегии управления риском: сдерживание, избегание, принятие, передача. Влияние стратегий на бюджет проекта
6. Методы текущего контроля хода выполнения проекта
 - a. Задачи контроля, контроль темпов работ и бюджета проекта
 - b. Контроль темпов работ: Tracking Gantt, понятие baseline
 - c. Комплексный метод контроля (темпы и бюджет): отчеты по сметной стоимости, отклонение стоимости и расписания, индексы выполнения бюджета и расписания CPI и SPI
 - d. Ловушка 90% завершенности проекта

5. Образовательные технологии

В курсе использованы следующие образовательные технологии:

1. Лекции. Презентации лекций доступны студентам на сайте курса. Приветствуется предварительная распечатка студентом презентаций и добавление комментариев по ходу лекции непосредственно на распечатку слайдов, что позволяет не тратить время на переписывание и перерисовывание в конспекты диаграмм, графиков и тезисов.
2. Семинары. В ходе семинаров производится обсуждение проектов с преподавателем и с другими студентами группы. Задачей преподавателя на семинарах является помощь в выборе подходящей технологии и установке правильного

объема функциональных требований, реализуемого командой в течение 1 семестра.

3. Самостоятельная работа. В процессе самостоятельной работы студенты выполняют проект группами по 2-3 студента магистратуры и 4-5 студентов бакалавриата. В ходе работы над проектом студенты обучаются инструментам организации коллективной работы, учатся документировать проектное решение (для составления документации студентам предлагаются шаблоны проектной документации, составленные на основе используемых в индустрии), для обсуждения проекта студенты организуют Google Groups.

6. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины и учебно-методическое обеспечение самостоятельной работы студентов

В целях обеспечения текущего контроля успеваемости студентов в течение семестра работа над проектом разбивается на этапы, с приемкой каждого этапа преподавателем на семинарах:

1. Видение проекта
2. Техническое задание, включающее описание требований, план проекта, анализ рисков, оценку трудозатрат.
3. Защита проекта

Тема проекта предлагается группой студентов (проектная команда) и согласуется с преподавателем на семинаре. При согласовании проекта преподаватель, основываясь на своем опыте в разработке ИТ проектов, ограничивает объем проекта до уровня, реализуемого командой в пределах 1 семестра.

На каждом из этапов производится оценка преподавателем достигнутых результатов, выдача рекомендаций по исправлению допущенных недочетов. Итоговая оценка выставляется после защиты проектной командой завершеного проекта, на основании оценки личного вклада студента в реализацию проекта и ответов на теоретические вопросы.

Примерный перечень теоретических вопросов к экзамену:

1. Управление конфигурацией. Задачи, методы и технические средства.

2. Определение проекта по PMI. Планирование. Задачи планирования. WBS. Связи задач.
3. Критический путь проекта. Определение. Методы управления.
4. PERT анализ. Применение при расчете бюджета проекта.
5. PERT анализ. Расчет запаса для управления критическим путем.
6. Управление рисками. Определение, классификация рисков по PMI. Типичные риски ИТ-проекта и типичные стратегии управления.
7. Авторские права разработчика и заказчика ПО.
8. Контроль хода выполнения проекта. Отчеты по сметной стоимости выполненных работ. Методика расчета, применение.

7. Учебно-методическое и информационное обеспечение дисциплины

а) основная литература:

1. PMBOK – Project Management Body of Knowledge, USA, 2005
2. Лекции по управлению ИТ проектами. <http://inteks.ru/PM/> , Мухортов В.В
3. Лекции по инструментам управления конфигурацией, Иртегов Д.В, <http://parallels.nsu.ru/~fat/subversion.ppt>
4. Rational Unified Process, www.rational.com
5. Управление версиями в Subversion, <http://svnbook.red-bean.com/>

б) дополнительная литература:

1. Project Management, Clifford F.Gray, Erik W.Larson, Irwin McGraw-Hill, USA, 2000
2. Structured Computer Project Management, W.Roetzheim, Prentice Hall, 1988

в) программное обеспечение и Интернет-ресурсы:

1. UML редактор ASTAH Community edition (бесплатное ПО)
2. Microsoft Office Word (коммерческое ПО) или Open Office (бесплатное ПО)

3. Microsoft Project (коммерческое ПО) или OpenProj (бесплатное ПО)
4. SVN-сервер на UNIX-сервере ФИТ (бесплатное ПО)
5. Tortoise SVN-клиент (бесплатное ПО)
6. сайт с материалами курса: <http://www.inteks.ru/PM/>

8. Материально-техническое обеспечение дисциплины

Для проведения курса требуются следующие технические ресурсы:

1. Терминальный класс с компьютерами на базе процессоров не хуже P-IV 3Hz, 1Gb RAM
2. Программное обеспечение компьютеров терминального класса:
 - a. операционная система Microsoft Windows (XP, Vista, Win7) или Linux
 - b. Программное обеспечение J2SE 1.6 (бесплатное ПО)
 - c. UML редактор ASTAH Community edition (бесплатное ПО)
 - d. Microsoft Office Word (коммерческое ПО) или Open Office (бесплатное ПО)
 - e. Microsoft Project (коммерческое ПО) или OpenProj (бесплатное ПО)
 - f. Tortoise SVN-клиент на компьютерах терминального класса (бесплатное ПО)
3. SVN-сервер на UNIX-сервере ФИТ (бесплатное ПО)

Рецензент (ы) _____

Программа одобрена на заседании Методической комиссии ФИТ

от _____ года, протокол № _____.