ПОДХОД К СОЗДАНИЮ КОМПЛЕКСНОЙ УЧЕБНОЙ ПРОГРАММЫ ДЛЯ ПОДГОТОВКИ ИТ-МЕНЕДЖЕРОВ

Е.М. Лаврищева, Д.Л.Безуглый, С.М.Молдавский

Институт Программных Систем НАН Украины, Адрес: просп. Глушкова, 40, Киев, 03187, Тел.: (+38) (044) 266 3470, E-mail: lem@isofts.kiev.ua

Украинский Учебно-Практический Центр Программной Инженерии УкрСофтПро, Адрес: б-р Л.Украинки, 26, Ин-т "Дипромисто", к.315, Киев, 01133, Тел.: (+38) (044) 249 3958, Факс: (+38) (044) 295 0915, E-mail: info@ukrsoftpro.com.ua

Рассматриваются подходи к обучению программной инженерии в мировой практике, а также принципы, которые используются при построении комплексной дистанционной учебной программы «Профессионал управления программными проектами» (“Software Project Management Professional”, SPMP). Обосновывается набор методик и учебных курсов.

Розлядаються підходи до навчання програмної інженерії у світової практиці, а також принципи, що використані при побудові комплексної дистанційної учбової програми «Професіонал Управління Програмними Проектами» (“Software Project Management Professional”, SPMP). Обгрунтовується вибор методики та учбових курсів.

The article describes approaches to training Software Engineering in the world practice and so major principles used while creating the complex distance education program “Software Project Management Professional” (SPMP). The justification for the selected curricula and methodology is also being considered.

1 Подходы к обучению новой специальности – программной инженерии

1.1 Анализ состояния

Вопрос обучения специалистов, занимающихся разработкой и внедрением информационных технологий (ИТ- специалистов), дискутируется много лет [1–9]. Считается, что обучение должно быть поставлено таким образом, чтобы ИТ - специалисты обладали не только фундаментальными знаниями в области компьютерных наук (computer science), но и достигали баланса между эффективными и жесткими сроками, широтой функциональных возможностей продукта и его низкой ценой, многообразием потребностей пользователей и ограничениями архитектур, постоянно повышали свою квалификацию. Во всем мире сохраняется высокая потребность в квалифицированных специалистах в области ИТ (информационных технологий). Сфера образования должна готовить специалистов, которые могли бы приступать к производственной деятельности сразу после получения диплома для удовлетворения этой потребности.

Одной из наиболее серьезных проблем, является подбор специалистов на руководящие должности: директоров информационных служб, руководителей проектов и технологов. Руководители бизнес–проектов отмечают, что выпускники технических специальностей университетов не имеют навыков управления и знаний современных технологии, что не позволяет им управлять проектами по созданию проектов в области ИТ. Специалисты, обладающие навыками управления, как правило, не обладают актуальными фундаментальными знаниями, не знают специфики информационных и компьютерных ресурсов и зачастую пытаются решить проблемы проекта с помощью стандартных инженерных или управленческих методов, что приводит к серьезным управленческим ошибкам. Как свидетельствует статистика [9], в итоге более 42% проектов проваливаются или не укладываются в заданную стоимость и сроки.

Другой серьезной проблемой является ряд отличий программной инженерии от классических инженерных дисциплин. Постоянное изменение предмета изучения - несмотря на большой объем фундаментальных знаний, знания необходимые для непосредственного создания информационных систем постоянно изменяются. Современный ИТ-специалист в первую очередь сам использует богатый арсенал программных средств, в результате производительность труда возросла многократно, и значительно возросла ответственность лежащая на каждом специалисте и требования к его квалификации. С другой стороны информационные системы тоже изменились, они стали значительно сложнее, объем требуемых знаний для построения и использования настолько возрос, что потребовалась значительная специализация ИТ-специалистов для обеспечения достижения необходимых результатов. Как правило, большинство систем требует эффективной работы большого коллектива высококвалифицированных специалистов.

Таким образом, ИТ-специалисты должны знать основополагающие принципы и системные подходы к решению вопросов разработки, внедрения и вывода информационных систем из эксплуатации, управления требованиями [10], рисками, качеством [11-12] и конфигурацией, интеграцией информации/компонентов, тестированием и метрическим анализом готовых систем [11]. А также уметь эффективно работать в команде специалистов, выбирать адекватные задачам процессы и технологии. Для распределенных проектов особые требования предъявляются к навыкам невербальной коммуникации.

Для ИТ-специалистов занимающих руководящие должности является необходимым знание общих методов, стандартов жизненных циклов (ЖЦ) программного обеспечения (ПО) [10], инструментов

проектирования информационных систем [11–20], а также инфраструктуры организации-разработчика (оборудование, связи, интерфейсы и т.п.), уровней зрелости организации (коллектива) в соответствии с моделью CMM (Capability Maturity Model [13, 14]), базовых понятий программной инженерии ядра SWEBOK [15] и других действующих отечественных и международных стандартов.

Ключевыми дисциплинами получения знаний студентов на факультетах информатики, ориентированных на разработку ПО, являются: программирование и языки программирования, как формальные математические объекты, дискретная математика Кнута, основы математической логики, введение в формальную семантику Хоара, теория алгоритмов и основы построения ЯП. Получаемый ими этот базис знаний способствует формированию математического мышления и формального подхода к процессам разработки ПО.

С точки зрения когнитивной психологии требуется не только преподавание этих дисциплин, но и применение теоретических знаний на практике. Например, проведение силами студентов разработки некоторого прототипа проекта с использованием идей объектно-ориентированного или компонентного проектирования с повторным использованием готовых объектов и компонентов по полному ЖЦ создания программного продукта. Прототип может быть собран из готовых компонентов и на нем отработаны функции, определенные в требованиях к системе. При реализации долговременного проекта, в процессе разработки могут быть определены стандарты на компоненты, модели качества, новые методы проверки надежности компонентов, а также подходы к определению уровня зрелости по модели CMM, соответствующей оценки деятельности разработчиков этого проекта. Таким образом, в процессе обучения студентами приобретаются не только теоретические знания, но и умение их использовать при выполнении различных ролей в группе разработчиков ПО, создании отдельных версий проекта, проведении их качественного анализа и измерения заданных в требованиях показателей качества.

1.2 Подходы к обучению программной инженерии

К настоящему времени, когда программная инженерия проникла во все сферы деятельности, разработка ПО и сущность инженерной деятельности в программировании приблизилась к энциклопедическому определению инженерной деятельности, поэтому ее становление как специальности не вызывает сомнения. Одним из дополнительных отличий, к указанным выше, программной инженерии состоит в большой сложности и недостаточной специализации большинства видов деятельности по отдельным направлениям, введения научной системы классификации и каталогизации готовых решений.

Перечисленные отличия требуют значительных усилий для их нивелирования и превращения программной инженерии в специальность. В настоящее время мировая компьютерная общественность признала целесообразность и своевременность таких усилий. Были созданы мощные профессиональные объединения – широко известное американское объединение компьютерных специалистов ACM (Association for Computing Machinery) и компьютерный союз при институте инженеров по электронике и электротехнике (IEEE Computer Society). Объединенными усилиями подкомиссий этого комитета были созданы: ядро SWEBOK (1999г., 2001г.) [15], этический кодекс профессионалов по программной инженерии (1999г.) [28]; руководства для обучения программной инженерии [4]; программа обучения Computing Curricula (CC) 2001 [2]. Кроме того, в США работает комитет по сертификации учебных заведений (Computing Accreditation Commission of the Accreditation Board for Engineering and Technology [3].

Таким образом, начались работы по преобразованию программной инженерии в специальность с зафиксированными признаками для ее распознания, официального признания и обучения.

В рамках работ по определению этой специальности опубликовано ряд учебников и монографий, отображающих разные ее аспекты и сформировалось несколько подходов к обучению и подготовки соответствующих специалистов [24–27]:

1) введение в программы обучения ИТ–специалистов отдельных элементов программной инженерии;

2) создание самостоятельной специальности «программная инженерия» и обучение ей студентов всех курсов;

3) сертифицированное обучение программной инженерии как профессии на курсах подготовки или переподготовки ИТ–специалистов.

Подход 1. Обучение этой дисциплине фактически уже проводиться на факультетах информатики в виде отдельных курсов, отражающих аспекты программной инженерии: модульное, объектно-ориентированное, компонентное программирование, управление данными, тестирование ПО и др. Дипломированные специалисты, прошедшие изучение некоторых из этих аспектов, не пользуются большим спросом на рынке труда, так как не имеют знаний и опыта в организации планирования и управления деятельностью разработчиков ПО, а также знаний в вопросах распределения ресурсов (людских, аппаратных, программных), оценки трудозатрат, повышения качества и др. важных моментов ведения крупных промышленных проектов. Они могут использоваться как программисты, либо повышать знания до уровня менеджера проекта или инженера в области программной инженерии на курсах повышения квалификации.

Подход 2. Наибольшее развитие в международной практике получил подход, ориентированный на создание самостоятельной специальности «программная инженерия» на факультетах информатики. Данный подход поднимает престиж учебного заведения, требует дополнительных вложений на его оборудование и привлечение соответствующего преподавательского состава. Учебный план факультета информатики

предусматривает программы по информатике и программной инженерии. Согласно [4] эти учебные программы относятся как 50:50. При этом треть предметов факультета связаны с программной инженерией, а две трети – с информатикой.

Другая программа обучения СС–2001 [2] рекомендует это соотношение, как 90 к 10. На факультетах информатики курс программной инженерии должен занимать 10%, т.е. 24–30 учебных часов в семестр. При этом 10% преподавателей факультета должны учить студентов дисциплине “программная инженерия”, а 90%–15 (пятнадцати) базовым курсам по специальности “Информатика” (дискретные системы, основы программирования, алгоритмы и теория сложности, ОС, теория алгоритмов, языки программирования и др.). Отведенный для программной инженерии диапазон часов соответствует не только базовым требованиям СС–2001, но и требованиям к курсам информатики, перечисленных в документах комитета по сертификации учебных заведений [3], готовящих инженерных специалистов. Программная инженерия, как дисциплина изучает теорию, сумму знаний, отображенных в ядре SWEBOK, и практику эффективного построения ПО на всех этапах ЖЦ. Если на факультете информатике работает 30 преподавателей, то по программной инженерии их должно быть не менее 5. СС–2001 рекомендует типовой факультативный учебный план по программной инженерии, включающий 12 тем:

1. Проектирование ПО.
2. Интерфейсы приложений.
3. Программные средства и окружение.
4. Процессы разработки ПО
5. Требования к ПО.
6. Проверка соответствия ПО.
7. Методы эволюции ПО.
8. Управление программными проектами.
9. Компонентно-ориентированная разработка (не обязательная).
10. Формальные методы.
11. Надежность и качество ПО.
12. Подходы к разработке специализированных систем (не обязательная).

Многие из этих тем, нашли отражение в ядре SWEBOK [15]. Темы 2, 9, 12 относятся к важным в проблематике разработки ПО, они обобщают целые эпохи формирования и применения общих методов (от модульного до компонентного) программирования в процессе создания ПО и специализированных систем различного назначения с использованием средств (GUI, Corba, COM – интерфейсы и др.). По этим темам к настоящему времени накоплен огромный опыт и знания, которые слабо представлены в ядре SWEBOK. Эти темы включены в программу обучения СС–2001 как не обязательные темы, хотя проблемы интерфейса и компонентного подхода сейчас являются перспективными направлениями дальнейшего развития современного программирования и производства ПО.

Подход 3. Обучение сформировавшихся специалистов программной инженерии на специальных курсах повышения квалификации (с отрывом и без отрыва от производства) представляет значительный интерес для действующих компаний, специализирующихся в разработке ПО и крупных программных проектов. Для усовершенствования квалификации персонала фирм объявляются или заказываются отдельные темы программной инженерии или целый курс. Самыми широко распространенными сертифицированными курсами являются: программная инженерия [25-27], управление требованиями, повторное использование [16], управление программными проектами [18, 22], моделирование в UML и Case Rational Rose [19, 20] и др. Именно этого подхода к обучению придерживаются авторы данной работы. Далее рассматривается опыт разработки дистанционной программы обучения ИТ-специалистов, ориентированной на повышение квалификации и получения профессиональных знаний по управлению программными проектами в Украинском Учебно-практическом Центре Программной инженерии – www. ukrsoftpro [21].

2. Концепция разработки дистанционной учебной программы подготовки ИТ-менеджеров

2.1 Идея создания Украинского Учебно-практического Центра Программной инженерии

Одним из направлений вхождения Украины в систему международного распределения ресурсов является производство ПО и информационных систем. На сегодня украинские компании разработчиков ПО выполняют прямые заказы для зарубежных фирм: Xerox, Delta Airlines, Motorola, World Bank, Apple, Gateway, Microsoft и др. Важным препятствием на пути широкого привлечения западных компаний к сотрудничеству с украинскими институтами и организациями разработчиков ПО является отсутствие законодательных и общегосударственных инициатив по прорыву на международный рынок услуг и продвижения высоких технологий.. Для успешного решения этих вопросов необходимо:

– улучшить подготовку выпускников профильных ВУЗов и факультетов (проектный менеджмент, бизнес-аналитики, программный инжиниринг и др.) со знаниями английского языка;

– создать систему представительств украинских ИТ-компаний за рубежом;

– организовать изучение компаниями–разработчиками ПО опыта работы по международной стандартизации и сертификации для создания собственных производственных и организационных процессов ПО;

– проводить маркетинговые мероприятия (участие в зарубежных выставках, учреждение и проведение собственных выставок и конференций и др.).

Для повышения конкурентоспособности собственного ПО и обеспечения отрасли инженерии ПО высококвалифицированными ИТ–специалистами в Киеве создан Украинский Учебно–практический Центр Программной инженерии – UkrSoftPro по инициативе и в рамках Украинской Ассоциации Производителей ПО (УАППО). В развертывании работы этого центра приняли также участие:

- IEEE Software Engineering Standards Committee,
- UkrSPIN - украинское отделение SEI Software Process Improvement Network,
- Eurasia Foundation (финансовая поддержка),
- Украинская Ассоциация Управления Проектами,
- Институт Программных Систем Национальной Академии Наук Украины,
- Компания "ИЛС-Украина",
- Компания "РИАЦ ИНТЕК - Украина".

При их участии в марте 2003г. УАППО провел круглый стол «Перспективы развития программной инженерии и управление программными проектами в Украине». В ходе заседания была поддержана инициатива создания этого Центра, как образовательной базы для повышения уровня знаний профессионалов в области программной инженерии.

2.2 Цель учебной программы

Основная цель учебной программы Центра – подготовка профессионалов управления программными проектами, руководителей проектов и технологических групп для коллективной разработки ПО и компьютерных систем. Повышение качества реализации программных проектов должно основываться на повышении точности управленческих решений за счет овладения знаниями, представленных в SWEBOK, а также знаний по стратегиям и методам современного программирования .

Предполагается, что учащимися Центра смогут овладеть такими знаниями и навыками:

• знания о современных методологиях производства ПО;

• знания о современных стандартах в области создания Информационных систем;

• знания по управлению программными проектами на стадиях ЖЦ в условиях ограниченного времени, ресурсов и противоречивых требований к системе;

• навыки создания и использования основных артефактов процесса производства ПО;

• навыки определения необходимого объема работ, коммуникации важных идей и защиты своей точки зрения;

• умение эффективно организовывать работу команды и отслеживать коммуникационные потоки в команде разработки ПО, как внутренние, так и внешние;

• понимание процесса определения требований пользователя к системе и преобразования их в требования к ПО;

• навыки проектирования приложений в одной или нескольких предметных областях;

• понимание основных концепций управления качеством, управления конфигурацией, а также верификации и валидации требований и их соответствия готовому ПО;

• навыки построения системы качества и унифицированного процесса производства ПО, также совершенствования процесса разработки;

• общие навыки создания документации на готовый программный продукт;

• навыки изучения новых моделей, техник и технологий производства ПО (например, экстремальное программирование, RUP-процессы и др.).

В основу построения учебной программы положены:

• концепции профессионального развития для софтверных организаций, разработанные под руководством Стива МакКоннелла в возглавляемой им компании Construx Software (Professional Development for Software Organizations. Version 3);

• указания по программно-инженерному обучению, разработанные в Software Engineering Institute (D. Bagert et al. Guidelines for Software Engineering Education. Version 1.0);

• учебный план по курсу программной инженерии CC–2001.

При построении программы обучения принимались во внимание такие классификационные аспекты:

• бизнес-роль учащегося - менеджер проекта, менеджер технологической группы;

• предметные области для производства ПО - системы баз данных, web-приложения, встроенные системы и т.п.;

• современные методологии разработки - Rational Unified Process, гибкие «agile»-технологии и Microsoft Solutions Framework –MSF – методология проектирования приложений, UML– моделирование, с помощью которых могут реализовываться практические задания выбранных тем данной учебной программы;

• стандарты и модели производства ПО (стандарты ISO серии 9000, программно-инженерные стандарты ISO, программно-инженерные стандарты IEEE, SEI CMM, управление качеством – ISO 9126-98, ЖЦ – ISO/IEC 12207-95 и др;

• основные области знаний программной инженерии SWEBOK (управление требованиями, проектирование ПО, построение ПО, тестирование, сопровождение, управление конфигурацией, управление проектом, совершенствование процессов разработки, инструменты и методы и управление качеством);

• «проектный» и «процессный» взгляд на разработку ПО.

Общей задачей программы является ознакомление учащихся Центра со спектром перечисленных концепций и аспектов, учитывая практическую полезность приобретения знаний и навыков на рабочем месте конкретного обучающегося.

Особенностью этой учебной программы является:

1) гибкая «настройка» программы под конкретного учащегося;

2) отказ от унифицированного подхода к делению программы на отдельные курсы в соответствии с методологиями разработки или областями знаний SWEBOK. Принято решение о применении «гибридного» подхода, при котором часть курсов знакомит учащихся с методологиями с упором на конкретные разделы знаний, а часть тем курса знакомит с важнейшими областями знаний применительно к выбранной методологии.

Для обучения по данной программе учащийся избирает специализацию, связанную с бизнес-ролью и выбранной методологией (менеджер проектов или менеджер технологической группы, Rational Unified Process или гибкие методологии), и выбирает набор курсов в соответствии со своими имеющимися знаниями и предпочтениями.

Помимо этого, учащийся должен выбрать близкую ему предметную область или данные реального проекта, в рамках которой он будет выполнять практические задания (обработка данных и информационные системы, web-приложения, графические и мультимедийные приложения, финансовые приложения и электронная коммерция, системы, критичные по отношению к надежности и безопасности, встроенные системы и системы реального времени, интеллектуальные системы, телекоммуникационные системы и т.п.).

2.3 Структура программы обучения

Данная учебная программа предполагает прохождение учащимся нескольких курсов (до 5) из имеющихся 10-ти. При этом некоторые курсы являются обязательными для каждого учащегося.

Ниже в приводится список курсов программы (в скобках даны разделы знаний для углубленного изучения):

1. Введение в программную инженерию (обзор областей знаний, инструменты и методы программной инженерии);
2. Стандарты программной инженерии (обязательный) (ЖЦ, документирование, управление качеством);
3. Менеджмент проектов (обязательный) (планирование, контроль и тайм-менеджмент, управление рисками, управление командой разработки и коммуникациями, инструменты программной инженерии – пакеты по управлению проектами);
4. Управление требованиями к ПО (анализ и управление требованиями);
5. Проектирование ПО (моделирование и проектирование архитектуры, данных и интерфейсов);
6. Управление качеством и конфигурацией (валидация и верификация, тестирование, обеспечение качества, управление изменениями и конфигурацией);
7. Rational Unified Process (построение ПО, сопровождение и эволюция ПО, инструменты – CASE-пакеты, управление процессами разработки);
8. Экстремальное программирование и гибкие (agile) методологии (построение ПО, сопровождение и эволюция ПО, управление процессами разработки);
9. Microsoft Solutions Framework (построение ПО, сопровождение и эволюция ПО, управление процессами разработки, управление рисками);
10. SEI Capability Maturity Model Integration (модель зрелости коллективов разработки ПО) (управление процессами разработки, совершенствование процессов разработки).

Согласно цели программы обязательными являются курсы №1-3, поскольку он должен получить сертификат ИТ-менеджера. Курсы № 4 –6 достаточно подробно изложены в ядре SWEBOK и, если нет ориентации на качество разработки требований и проекта в целом, можно ограничиться получением знаний из ядра по этим вопросам.

Требования к поступающим на курсы повышения квалификации. В качестве предварительных условий предполагается наличие у обучающихся таких базовых знаний и навыков:

• английского языка на уровне чтения технической литературы;

• процедурной и объектно-ориентированной парадигм программирования;

• один или несколько современных ЯП;

• опыт работы в одной или нескольких предметных областях разработки ПО.

Обучающийся должен иметь доступ к Интернету и электронной почте. Желательным (но не обязательным) является доступность хотя бы одного CASE-пакета (например, Rational Rose) и пакета управления проектами (желательно Microsoft Project).

Схема процесса обучения. После выбора специализации и предметной области обучающемуся высылается по обычной почте базовая литература по всем выбранным курсам, а по электронной – краткое содержание первой темы выбранного курса.

Процесс обучения состоит из следующих шагов:

• получение по электронной почте учебных материалов по каждой теме, содержащих очень краткое освещение темы, ссылки на ресурсы и литературу, а также варианты практических заданий;

• изучение темы , ее ресурсов и учебной литературы;

• обращение к преподавателю курса (или куратора) с вопросами по данной теме по почте или с помощью web-форума;

• общение с другими учащимися и преподавателями с помощью web-форума;

• выполнение практических заданий;

• консультации по электронной почте с преподавателем/куратором ;

• отправка преподавателю практического задания и получение предварительной оценки по выполненному заданию;

• получение оценки по окончательной версии выполненного задания;

• автоматическое получение по электронной почте материалов следующей темы.

Учебная программа рассчитана примерно на год и предполагает изучение каждой темы в течение примерно 3-х недель. Однако обучающийся никак не ограничен по времени, и в принципе может увеличить или сократить период обучения темы.

3. Анализ результатов дистанционного обучения курсам учебной программы

Обучением охвачено в настоящий момент 35 человек. Первоначально было 5 человек (руководители групп разработчиков ПО, инженеры, программисты и др.), они не были знакомы с базисом дисциплины – программная инженерия и все начали обучение с этого курса. Содержание тем по замыслу составлялось очень коротким, чтобы учащийся самостоятельно изучал дополнительную литературу. В частности по этому курсу предлагалась в качестве основной литературы – оригинал материала SWEBOK – knowledge areas.

Анализ коммуникации между преподавателями и учащимися позволил выделить следующие факты:

- Большой разброс в уровне подготовки учащихся, значительная часть показала отсутствие необходимых коммуникативных навыков и базовых знаний, с другой стороны для отдельных учащихся сложность заданий для повышения квалификации была явна недостаточной.
- Высокую эффективность в обучении формальному представлению результатов работы и управления объемом работ. Учащиеся справившиеся с предложенными заданиями продемонстрировали значительный рост качества выполнения заданий в более краткие сроки.
- Высокую эффективность по выявлению учащихся не имеющих склонности (способностей) к управленческой деятельности в ИТ сфере.

По результатам обучения были сделаны следующие выводы:

- Целесообразно введение программы для базового повышения квалификации до уровня необходимого для начала изучения основной программы.
- Постепенное наращивание сложности практических заданий для выработки необходимых навыков без потери мотивации учащимися.
- Целесообразно незначительное увеличение объема вводных лекций для каждой темы .
- Необходимо предоставление шаблонов выполнения практических заданий.
- Формирование библиотеки материалов на CD для снижения стоимости учебных пособий.
- Качество материалов подготовленных отдельными учащимися даже на первых курсах позволяет построить схему обучения сотрудников компании совмещенную с консалтингом по улучшению процессов разработки компании.

При анализе накопленного опыта необходимо учитывать, что количество учащихся не может быть критерием для определения спроса на данный вид обучения, в силу высокой эластичности спроса, низкой платежеспособности потенциальных учащихся и вынужденной высокой стоимостью обучения, по причине работы проекта в режиме самоокупаемости.

Значительный эффект может оказать организация специальных фондов при общественных организациях для оказания материальной помощи в оплате обучения для отдельных студентов.

Литература:

1. *Bagert D., et al*., Guidelines for Software Engineering Education? Version 1.0, Carnegie Mellon Univ., Pittsburgh, pa., 1999.
2 . http://computer.org/education/cc2001 или http://se.math.spbu.ru/cc2001- русский вариант
3. ABET – Criteria for According Computing Programs, 2002, www.abet.org/criteria.htm
4. *Shaw M.* Software Engineering Education: A Roadmap. The future of Software Engineering, A finkkelstein, ed., ACM press, New York, 2000.
5. *Lethbridge T.*, What Knowledge is important to Software Professional?, Computer, vol33, N5, May, 2000.
6. *Каллахан Д., Педиго Б.* Подготовка ИТ-профессионалов для бизнеса. Открытые системы, февраль, 2003.–С.35–38.
7. *Терехов А., Павлов В.*. Перспективы развития ИТ–образования. Там же. С. 44-48.
8. *Драгомирзян И., Иванников В.* Подготовка кадров для ИТ. Там же с. 53-54.
9. Программы следующего десятилетия. Открытые системы.–Декабрь, 2001.–с.60-71.
10. *Jackson M.* Software requirement & specifications.– Wokingham, England: Addison–Wesley, ACM Press Books, 1995. –228p.
11. *Андон Ф.И., Лаврищева Е.М.* Методы инженерии распределенных компьютерных систем, Киев, Изд. «Наукова думка», 1997г.–228с.
12. *Андон Ф.И., Коваль Г.И., Коротун Т.М., Суслов В.Ю.* Основы инженерии качества программных систем.–К: Академпериодика, 2002.–502с.
13. Capability Maturity Model for Software, version 1.1/M.Paulk, B Curtis at al// CMU–SEI–93–24, Soft.Engin. Institute, Pittsburg PA 15213, Feb.– Pittsburg.–82p.
14. *Андон Ф.И. , Коротун Т.М.* и др. Модель технологической зрелости организаций–разработчиков ПО. Проблемы программирования, 1998. №4.–с.44-57.
15. http:/www.swebok.org.html
16. *Jacobson I., Griss M., Jonsson P.* Software Reuse.–N.–Y.– Addison–Wesley, 1997. – 497 p.
17. *MeyerB.* Object-oriented Software Construction. – 2nd. ed., Prentice Hall, 1997.–531p.
18. *Thayer R.H. , ed.*, Software Engineering Project Management, 2nd.ed., IEEE CS Press, Los Alamitos, Calif. 1997.–391p.
19. *Фаулер М., Скотт К.* UML в кратком изложении//М.-"Мир".–1999.–200с.
20. *Jacobson I ., Booch G., Rumbaugh J.* The Unified Software Development Process, N.–Y.– Addison-Wesley, 1999. – 463 p.
21. = УкрСофтПро = Украинский Учебно-Практический Центр Программной Инженерии http://www.ukrsoftpro.com.ua
22. *Duncan B.* . A Guide to the Project Management Body of Knowledge//PMBOK GUIDE.–2000.– Edition/ www.pmi.org/publication/download/2000welcome.html.
23. 12297/FPDAM1.2 – Software Engineering –Life Cycle Processes // ISO/IEC JTC/SC7 N 2413, Software & Software Engineering Secretariat, Canada.–2001.– 62p.
24. *Pfleeger S.L.* Software Engineering. Theory and practice. – Printice Hall: Upper Saddenle River, New Jersey 07458, 1998. – 576p.
25. *Jacobson I.* Object-Oriented Software Engineering. A use Case Driven Approach, Revised Printing. – New York: Addison-Wesley Publ.Co., – 1994.– 529 p.
26. *Соммервил И.* Инженерия программного обеспечения. 6 -издание.–Москва–Санкт–Петербург–Киев, 2002.–623с.
27. *Бабенко Л.П., Лаврищева Е.М.* Основы програмной инженерии. Учебник (укр.язык). – Киев: Знання, 2001. – 269 с.
28. Jotterbarn D., Miller K., Rogerson S. Software Engineering CODE of Ethic is Approved//Com. of the ACM .v. 42.–N 10.–1999.–P. 102–107.
29. *Лаврищева Е.М.* Пути стандартизации программной инженерии как специальности// Материалы междун. научно–практ. конф. “Теория и практика стандартизации образования” часть 1.– Минск, 18–19 января. 2001г.–с.8–13.
30. Программа Профессионал Управления Программными Проектами. http://www.ukrsoftpro.com.ua/spmp.shtml