ОСВІТА ТА ІНЖЕНЕРІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

УДК 378.147

***М.О. Сидоров, *І.Б.Мендзебровский, **А.А.Орехов**

«ПРОФЕСІЙНА ПРАКТИКА ПРОГРАМНОЇ ІНЖЕНЕРІЇ» – ДОСВІД ВИКЛАДАННЯ

*****Національний авіаційний університет
**«Майкрософт Україна»

Розглядається досвід викладання дисципліни «Професійна практика програмної інженерії», яка вивчається студентами напрямку навчання «Програмна інженерія» згідно галузевого стандарту та навчального плану Національного авіаційного університету. Особливість досвіду полягає в тому, що навчання було організовано на підприємстві з застосуванням проектного методу.

Рассматривается опыт преподавания дисциплины «Профессиональная практика программной инженерии», которая изучается студентами направления обучения «Программная инженерия» в соответствии с отраслевым стандартом и учебным планом Национального авиационного университета. Особенностью опыта являться то, что обучение было организовано на предприятии с использованием проектного метода.

Experience of learning subject by name "Professional practic of Software engineering" in National Aviation University is presented. The specify properties of the Experience consist in learning subject on the base of the software developer firms with the help of the project method.

Ключові слова: інженерія програмного забезпечення, професійна практика, метод проектів, навчання.

Вступ

Напрям підготовки 6.050103 «Програмна інженерія» було введено постановою Кабінету міністрів в 2006 році [1]. В навчальні плани університетів, які готують фахівців за цим напрямом згідно галузевого стандарту, входить дисципліна «Професійна практика програмної інженерії» [2]. Наявність такої дисципліни є вимога міжнародного документу, щодо навчання інженерії програмного забезпечення [3].

Дисципліна відрізняється від інших двома ознаками:

1). Має чітку практичну та професійну спрямованість;

2). Вперше, за навчання потребує інтеграції знань та умінь набутих студентами за дисциплінами, які є попередниками цієї дисципліни згідно структурно-логічної схеми навчання.

Відносно першої ознаки – студенти повинні налаштуватися до конкретної праці, яка максимально наближена до реальної, а викладачі повинні бути практично та професійно (з точки зору розробки програмного забезпечення) підготовлені. Тому, бажано, наступне: навчання проводити на території організації, яка є розробником програмного забезпечення; в викладанні дисципліни повинні приймати участь професійні розробники; практичні заняття повинні бути організовані відповідно сучасним вимогам.

Згідно другої ознаки – студенти повинні інтегрувати власні знання та уміння і спрямувати результат на розробку програмного забезпечення, а викладачі, застосовуючи відповідний навчально-методичний матеріал, повинні спростити цю інтеграцію.

Загальні питання підготовки фахівців за інженерією програмного забезпечення розглядаються в роботах [4-7].

Застосування методу проектів для навчання програмуванню розглядається в роботі [8].

В статті розглядається досвід викладання дисципліни «Професійна практика програмної інженерії», який був отриманий в Національному авіаційному університеті на кафедрі інженерії програмного забезпечення.

В першій частині статті розглядається метод проектів, як засіб вирішення проблеми практичної підготовки студентів. В другій частині наводиться місто дисципліни в структурно-логічної схемі навчального плану і навчально-методичне забезпечення дисципліни. В третій частині описано організацію навчання за дисципліною. В четвертій частині представлені результати викладання дисципліни за допомогою компаній "Itera group of Ukraine" та "Microsoft Україна".

1. Метод проектів

Метод проектів – дидактичний прийом, сутність якого полягає в організації навчання шляхом спрямування студентів на конкретний результат, який досягається побудовою продукту (технології) [9, 10]. Проект, це самостійна діяльність студентів (індивідуальна, або групова), яка планується і виконується студентами з метою отримання конкретного, «відчутного» результату готового до використання. Проект характеризується наступним [10]:

- наявністю задачі, вирішення якої потребує інтегрованого знання та пошуку її вирішення;
- практичною, теоретичною, або пізнавальною значимістю результату;
- плануванням (проектуванням) проекту, як за діяльністю, так і за результатом;
- застосуванням методів та засобів для реалізації проекту;
- обов'язковим оформленням результату і загалом його презентації.

За типологією проекти з дисципліни «Професійна практика програмної інженерії» повинні мати наступні ознаки:

- домінуюча проектна діяльність – прикладна;
- предметна галузь – будь-яка, але бажано, не вимагати глибоких знань у студентів з галузі, яку вони не вивчали;
- характер координації проекту – прихований, тому, що проект повинен виконуватися як телекомунікаційний;
- характер контактів – серед студентів навчальної групи;
- кількість учасників проекту – залежить від кількості студентів в групі;
- термін проекту – залежить від терміну навчання.

Для ефективного застосування методу проектів дуже важливо структурування (проектування) проекту і організація оцінки результату, при якої обов'язковим є колективний захист, обговорення і презентація проекту – результату і процесу виконання.

Для оцінки проектів доцільно використовувати наступні критерії [10]:

- актуальність та значимість задачі (якщо задача запропонована виконавцями проекту);
- коректність методів, що застосовані, для вирішення задачі;
- активність кожного учасника проекту;
- «колективність» рішень, що приймаються, якщо проект був колективний;
- характер спілкування і взаємодопомоги, взаємодоповніності учасників проекту;
- глибина вивчення задачі (предметної галузі);
- доказаність рішень, що сприймаються, їх аргументованість;
- естетика оформлення результатів проекту;
- лаконічна, аргументована поведінка членів проекту при захисті.

2. Навчально-методичне забезпечення дисципліни

Згідно навчального плану Національного авіаційного університету дисципліна викладається в п'ятому семестрі. Структурно-логічна схема зв'язку дисципліни з тими дисциплінами, які вже надано студентам наведено на рис. 1.

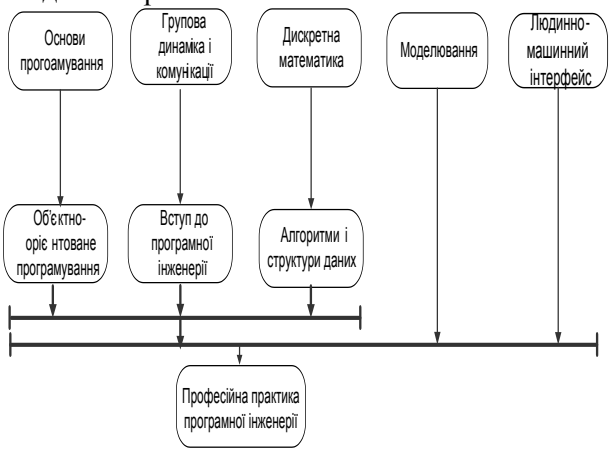

Рис. 1 Структуро-логічна схема навчання

Керуючись рекомендаціями з викладання цієї дисципліни було розроблено навчальні програми [3, 11, 12]. Зміст навчальної програми [11] приведено в Додатку 1. Положенням, яке було покладено в основу програми дисципліни є таке, що зміст дисципліни повинен враховувати два аспекти – теоретичний і практичний. В теоретичному аспекті зміст спрямовано на надання знань, які б дозволили випускникам працювати в інженерії програмного забезпечення з почуттям відповідних професійних коренів (щодо походження), та як членів професійних спільнот. В практичному аспекті зміст спрямовано на набуття перших навичок практичного застосування отриманих раніше знань в умовах колективної праці з використанням відповідних методів і інструментів.

Тому перші три теми програми (див. Додаток 1) присвячено історії та культурі інженерії програмного забезпечення, безпосередньо професіоналізму та забезпеченню якості. Остання тема охоплює практичні професійні питання на прикладі Agale - методології (для навчання було обрано технологію Scrum Microsoft Framework Solution), тому, що ця методологія спрямована саме на невеликі колективи розробників, тісно пов'язані між собою. Це є тою моделлю, яку можна використовувати в умовах студентської групи, розбиваючи її на маленькі колективи та застосовуючи метод проектів. В цій темі також відтворено відповідну професійну лексику.

Лекційний матеріал за професійною спрямованістю містить такі теми:

- практика управління проектами програмного забезпечення, – пов'язана з застосуванням методу проектів;
- введення в ERP, – знайомить з процесним підходом, щодо моделювання програмного забезпечення з метою застосування в проекті;
- забезпечення якості проекту – пов'язано з необхідністю досягнення якісних властивостей проектів;
- введення в CRM, - знайомить з особливостями систем, цього типу.

Лабораторні роботи були спрямовані на набуття вмінь користування засобом Microsoft Team Foundation Server та навичок колективної розробки застосувань у ньому, які були представлені в лекційному матеріалі. Теми доповідей на конференції були також запропоновані студентами і пов'язані з темою проекту.

3. Організація навчання

Приймаючи до уваги професійну спрямованість дисципліни було прийнято рішення доручити її викладання фахівцям компанії – розробнику програмного забезпечення під методичним «доглядом» викладача кафедри. Також, було прийнято рішення викладати дисципліну в умовах наближених до професійних, і тому на території компанії.

З цими вимогами погодились компанії Асоціації "IT України" – "Itera group of Ukraine" та "Microsoft Україна", які, так склалось, ще реально були розташовані в одній будівлі, на одному і тому ж поверсі.

Для навчання було обрано одну групу студентів третього курсу англійського проекту (навчання повністю англійською мовою). Якісні та кількісні показники групи наведено в табл. 1.

Таблиця 1. Показники групи

№ п/п	Показник	Значення	
		За оцінкою студента	Фактично
1	Кількість студентів	-	18
2	Жіноче/чоловіче співвідношення	-	5/13
3	Середня успішність	-	4.4
4	Володіння мовами		
4.1	C++	****	***
4.2	C#	****	**
4.3	Java	***	*

Навчання було організовано за видами, що наведено в табл. 2

Таблиця 2. Види навчальних робіт

№ п/п	Вид роботи	Код роботи
1	Лекції	Л
2	Лабораторні роботи	О
3	Практичні заняття	П
4	Лабораторні роботи	К
5	Конференції	Ф
6	Зустріч з керівництвом	З
7	Зустріч з розробниками	Р
8	Модульний контроль	Мк
9	Підсумковий контроль	Ік

Тривалість різних форм навчання наведено на рис. 2

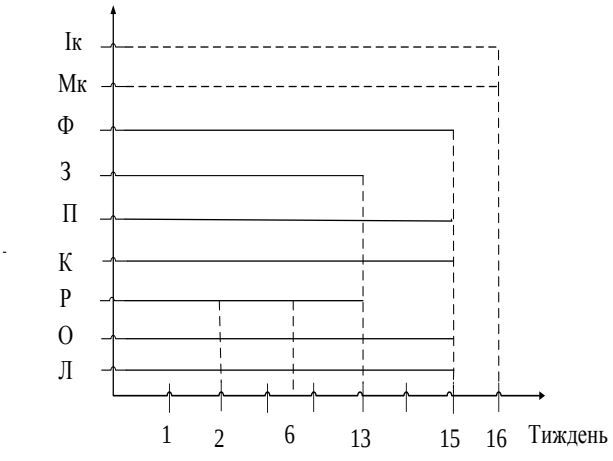

Рис. 2 Графік навчального процесу за дисципліною

Лекції було проведено за штатною формою. Лабораторні роботи були передбачені трьох

видів – для студента однака (О), для колективу (К, проект), практичні заняття (П).

Конференції (Ф) - це зустрічі студентів з метою презентації проектів за попередньо зазначеними темами.

Зустріч з керівництвом (З) – присвячено зустрічі з топ-менеджером компанії.

Зустріч з розробниками (Р) – присвяченого відповідям співробітників компаній на запитання студентів, щодо технологій, методів і інструментів.

Рис.2. Практичні заняття студентів

Лабораторні роботи було спрямовано на засвоєння наступного:

- організації і управління проектами;
- колективної праці;
- технологій;
- інструментів.

Модульний та підсумковий контроль призначався згідно положень робочої навчальної програми.

4. Навчання і досвід

Детальніше, за формами навчальних робіт (табл. 2) навчання було організовано наступним чином.

Лекційні заняття проводилися відповідно до змісту програми дисципліни (див. Додаток 1) і обов'язково містили дві складові – теоретичну, в якій надавалися загальні знання відносно теми лекції і практичну складову, в якій викладач – досвідчений співробітник фірми, на конкретних прикладах власного досвіду демонстрував практичне застосування теоретичного матеріалу. Наприклад, розглядаючи тему культури інженерії програмного забезпечення викладач демонстрував особливості моделі культури, яка склалася на даному підприємстві. Лекції обов'язково супроводжувалися наочним матеріалом у вигляді комп'ютерних презентацій.

Лабораторні роботи для одного студента (таб. 2, 0) виконувалися за наступними темами, що спрямовані на вивчення Microsoft Team Foundation Server, який було обрано як технічний засіб для колективного виконання проекту:

– робочі елементи (Work Items);
– сценарії, задачі й помилки;
– запити до робочих елементів;
– збірка (Team Build);
– система контролю версій;
– аналіз стану проекту за допомогою звітів;
– аналіз коду, профілювання застосування.

Скорочений приклад змісту завдань лабораторної роботи – робочі елементи (Work Items) наведено в додатку 2.

Практичні заняття були спрямовані на здобуття студентами навичок створення документації на прикладі розробки специфікації вимог. Ці навички повинні були забезпечити документування проекту (процесів і продуктів).

Рис.3. Студенти на лекції

Проекти виконували команди студентів, які було сформовано за бажаннями. Задача, яка вирішуватиметься колективами студентів – розробити програмне забезпечення інтернет-магазину велосипедів і аксесуарів до них. Сформованим командам надавалося наступне:

– мінімальні сценарії для функціональності (деталі необхідно було уточнювати у замовника, «запитуючи правильно»);

– організаційні умови відносно деталей виконання проекту (опис, тестування, завершення);

– технічні умови відносно технологій, з використанням яких повинно бути побудоване застосування.

Скорочений приклад завдання для колективної розробки наведено в Додатку 3.

У студентів відбулася одна зустріч з керівниками компанії і декілька – з співробітниками. Керівник, розповів про компанію і власний досвід в цьому бізнесі.

Співробітники відповідали на різні запитання – відносно організації праці,

особливостей компанії, а також стосовно проектів, які виконувалися студентами.

З метою здобуття навичок презентувати виконану роботу студенти прийняли участь в конференції.

В цілому з боку студентів та з боку викладачів можуть бути відзначені певні позитивні зрушення.

Наприклад, з боку студентів надходить розуміння про наступне:

– в яких реальних обставинах створюється програмне забезпечення, і як наслідок, зростає розуміння ролі виробничої культури;

– важливість колективної праці і необхідність вибору власної ролі в команді згідно власним схильностям та вмінням (навичкам) і необхідність в спеціалізації у відповідній ролі;

– важливість знань і навичок набутих (чи не набутих) при вивчені інших дисциплін структурно-логічної схеми;

– різниця між власним професійним рівнем та професійним рівнем співробітників підприємства і бачення відповідної перспективи.

З боку викладачів з'явилося розуміння наступного:

– реальних обставин, у яких створюється програмне забезпечення;

– важливість власної технологічної підготовки;

– важливість попередньої підготовки студентів до самостійного виконання робіт в колективах.

Висновки

В навчальному плані є дисципліни, мета яких надати студентам знання та первинний досвід, які необхідні для початку професійної діяльності. Найбільш привабливими властивостями інженерів з програмного забезпечення є навички комунікації, чіткість, навички, щодо роботи в команді, навички міжособистих відношень, мотивація і ініціатива, оцінка роботи. Деякі з цих властивостей набуваються в результаті вивчення дисциплін подібних той, що розглядається в статті. На підготованість студентів впливають також участь в конференціях, обговореннях, зустрічах зі «старшими товаришами» по професії.

Метод проектів вимагає особливого підходу до його застосування. Викладачам університету необхідні додаткові знання, час і кропітка праця. З іншого боку, в будь якій компанії виконання робіт здійснюється виключно за проектами, тому викладання цієї дисципліни в умовах компанії дуже ефективно.

Досвід навчання показав також, що викладання цієї дисципліни потребує знань з дисциплін, які стосуються конкретних доменів, наприклад «Фінансові системи», «Науково-дослідні системи», а це можливо вимагає змін в навчальному плані.

Додаток 1

Зміст навчальної програми

Тема 1: Історія інженерії програмного забезпечення

Криза програмного забезпечення. Перша та друга конференції NATO з інженерії програмного забезпечення. Від програмування «в малому» до програмування «в великому». Розвиток інженерії програмного забезпечення в Радянському Союзі і Україні. Розвиток мов програмування і програмних конструкцій (підпрограма, модуль, клас). Життєвий цикл програмного забезпечення – історія розвитку. Економіка програмного забезпечення історичний погляд. Повторне використання програмного забезпечення. Екологія програмного забезпечення. Витоки і розвиток. Культура інженерії програмного забезпечення.

Тема 2: Стандарти і якість програмного забезпечення

Походження і значення стандартів програмного забезпечення. Стандарти розробки і супроводження програмного забезпечення: огляд. Суспільна зацікавленість в якості програмного продукту. Вартість якості і втрати від поганої якості. Роль документування в програмному забезпеченні. Культура якості при розробці програмного забезпечення. Якість процесу та якість продукту – взаємозв'язок. Стандарти якості та сертифікація якості в Україні.

Тема 3: Професіоналізм

Акредитація, сертифікація і ліцензування інженерів з програмного забезпечення. Кодекс етики і професійної поведінки інженера з програмного забезпечення. Походження професійних спільнот і їх роль у професійної діяльності інженера з програмного забезпечення. Професійна діяльність в галузі програмного забезпечення, принципи професійної діяльності. Розробка і супроводження програмного забезпечення – зв'язок з професіями.

Тема 4: Професійна практика. Процеси

Вибір технологій та шаблонів процесів при розробці програмного забезпечення. Процеси/ фази розробки проекту. Формування бачення проекту (Envision). Фаза планування й управління вимогами (Plan). Створення сценарію користувача. Оцінка сценаріїв користувача.

Оцінка швидкості роботи групи. Виконання плану реліза. Інкрементне планування. Фаза написання коду і збірки проекту (Build). Поетапна розробка та розгортання проекту. Фаза стабілізації та забезпечення якості (Stabilize). Проведення заходів по забезпеченню якості програмного продукту на кожному етапі його життєвого циклу та тестуванню. Фаза розгортання / поставки продукту (Deploy). Операційне управління (Operational Management). Навчання персоналу (Governance). Взаємозв'язок між фазами й інтерактивною розробкою програмного забезпечення.

Тема 5: Професійна практика. Артефакти

Артефакти. Робочі елементи Work Items та Workflow. Сценарії користувача (User Story). Задача (Task). Тестовий сценарій (Test Case). Колективні кроки (Shared Steps). Помилка (Bug). Проблема (Issue). Поля робочого елементу (Work Item Fields). Командні запити. Контрольні панелі моніторингу (Dashboards). Персональна контрольна панель (My Dashboard). Тренд об'єму виконаної роботи (Burndown Dashboard). Тренд помилок (Bugs Dashboard). Табло стану зборок (Build Dashboard). Табло стану якості (Quality Dashboard). Табло стану по тестуванню (Test Dashboard). Робочі журнали (Workbooks). Беклог продукту (Product Backlog Workbook). Беклог ітерації (Iteration Backlog Workbook). Беклог проблем (Issues Workbook). Журнал змін (Triage Workbook). Звіти (документація при розробці програмного забезпечення). Звіт про статус помилок (Bug Status Report). Звіт про стан помилок, їх кількість (Bug Trends Report). Звіт про ефективність розв'язання помилок (Reactivations Report). Звіт про якість зборки (Build Quality Indicators Report). Детальний звіт по кожній збірці за кожен день (Build Success Over Time Report). Сумарний звіт по збірці (Build Summary Report). Звіт по об'єму виконаної роботи та кількості робочих елементів, що вирішено й закрито (Burndown and Burn Rate Report). Звіт по об'єму роботи, що залишилися виконати для спринту(Remaining Work Report). Звіт про статус на всіх ітераціях (Status on All Iterations Report). Оглядовий звіт по сценаріях (Stories Overview Report). Звіт по сценаріях у порядку їх важливості (Stories Progress Report). Звіт про кількість і готовність сценаріїв з тестування (Test Case Readiness Report). Звіт про виконання плану тестування (Test Plan Progress Report). Ролі. Роль власника продукту (Product Owner Role). Роль команди (Team Role). Наради. Нарада з планування (Planning Meeting). Робоча нарада (Meeting). Звітування перед клієнтом (Customer Review Meeting). Ретроспективна нарада (Retrospective Meeting).

Додаток 2

Робочі елементи Work Items

Завдання 1. Ітерації і області роботи

Використовуючи Visual Studio 2008 підключіться до сервера.

Додайте проект NAU Agile Demo. Створіть Area для своєї діяльності (використовуйте своє прізвище для її назви).

Створіть вкладені Area для планування, розробки, архітектури, тестування і т.д.

Створіть Iteration для своєї діяльності (використовуйте своє прізвище для її назви).

Створіть вкладені ітерації: старт проекту, кілька ітерацій на розробку, фінальну ітерацію впровадження проекту.

Частина завдання виконаєте, використовуючи TFS Web Access.

Завдання 2. Робота з сценаріями, завданнями та багами

Для виконання цього завдання використовуйте або Visual Studio, або TFS Web. Access

Створіть завдання для збору вимог, презентації клієнту, розгортання Dev & Test Enviroments. Частина з цих завдань призначте вашому сусідові. Задайте відповідні Area і Iterations.

Опишіть 5-7 сценаріїв для проекту «Форум».

Відмітьте завдання планування як виконані.

Створіть 4-5 багів для сценаріїв вашого сусіда (їх потрібно призначити йому і пов'язати з його сценаріями).

Вирішіть 2 / 3 призначених вам багів.

Половину багів, які вирішив ваш сусід закрийте, а другу половину реактивуйте.

Завдання 3. Робота із запитами до робочих елементів

Створіть Work Item Query для відображення створених вами робочих елементів.

Перегляньте її в TFS Web Access.

Додаток 3

Завдання для проекту

Команда повинна розробити інтернет-магазин велосипедів і аксесуарів до них. Застосування - Інтернет магазин повинен мати інтерфейс для відвідувачів і менеджерів.

Мінімальні сценарії наступні:

– відвідувачі переглядають каталог товарів з цінами, властивостями і описом товарів, а також

їх фотографіями. Товари повинні групуватися/фільтруватися за категоріями;
– відвідувачі додають товари у кошик;
– відвідувачі виконують замовлення товарів з корзини з їх оплатою за допомогою кредитної картки (механізм повинен стимулювати такі дії, без реальної інтеграції);
– відвідувачі можуть реєструватися і зберігати історію своїх покупок;
– менеджери переглядають список замовлень;
– менеджери позначають замовлення як виконання.

Деталі потрібно уточнювати у замовника задаючи правильні питання.

Організаційні умови

- Проект повинен вестися з використанням Team Foundation Server;
- Усі функціональні можливості повинні бути описані у вигляді сценаріїв;
- Застосування повинно бути протестовано автоматично за допомогою модульних, Web і навантажувальних тестів;
- Замовник може уточнювати і змінювати свої вимоги;
- Умовою успішного завершення є прийняття замовників.

Технічні умови

Web - застосування має бути побудовано з використанням наступних технологій Microsoft:

- Internet Information Services 7.0;
- SQL Server 2008;
- ASP.NET 3.5;
- Silverlight;
- ASP.NET MVC;
- RIA Services;

Звичайно, що технологічні умови вимагають від студентів відповідальних знань і навичок.

Список літератури

1. Постанови Кабінету Міністрів України «Про перелік напрямів, за якими здійснюється підготовка фахівців у вищих навчальних закладах за освітньо-кваліфікаційним рівнем бакалавр» від 13.12.06 №1719
2. *М.Ф.Бондаренко, М. Сидоров, Т. Морозова* Модель випускника бакалаврату програмна інженерія. – Вища школа. - № 4. – 2009. – с. 50-61
3. Joint Task Force on Computing Curricula, Software Engineering 2004. – Curriculum Guidelines for Undergraduate Degree Programs on. -234 p.
4. *M.B. Bloke*. A Student-Enacted Simulation Approach to Software Engineering Education.- IEEE Transactions on Education.- v. 46.- No 1.- 2003.- p. 124-132.
5. *D. Parnas*. Software Engineering Programs Are Not Computer Sience Programs.- IEEE Software.- v.16.- N6.-1999.- p. 19-30.
6. *M.* Shaw. Prospects for on Engineering Discipline of Software.- IEEE Software.- vol. 7.- no. 6.- 1990.- pp. 15-24.
7. *Є.Г. Кириленко, О.В. Лучшева* Обоснование содержания обучения в рамках методологии преподавания профессионально-ориентированной дисциплины «Групповая динамика и коммуникация». – Інженерія програмного забеспечення. - №1. – 2010. – с.71-78
8. *Е.М. Давыдова, Р.В. Мещеряков, А.А. Шелупанов.* Проектное обучение – парадигма элитного инженерного образования в России в условиях стратегии инновационного развития.- Образование в России. -2. -2008. –с. 9-15
9. *W.H. Kilpatnic.* The Project Method.- Teachers College Record.- 1918.- 19.- Sept.- p. 319-334.
10. Новые педагогические и информационные технологии в системе образования.- под. Ред. Е.С. Полат.- М.- Издательский центр «Академия» 1999.
11. *М.О.Сидоров, М.А. Безверха*. Навчальна програма «Професійна практика програмної інженерії».- К.- НАУ: 2009. -7 с.
12. *М.О. Сидоров, М.А. Безверха*. Робоча-навчальна програма «Професійна практика програмної інженерії».- К.- НАУ: 2009. -12 с.

Відомості про авторів

Сидоров Николай Александрович, д.т.н., проф., декан факультету комп'ютерних наук, завідувач кафедри інженерії програмного забезпечення Національного авіаційного університету, наукові інтереси – інженерія програмного забезпечення, навчання
E-mail: nikolay.sidorov@livenau.net

Мендзебровський Ігор Борисович, президент компанії «Itera Group of Ukraine», аспірант Національного авіаційного університету. Наукові інтереси – розробка програмного забезпечення, навчання інженерії програмного забезпечення
E-mail: igor.mendzebrovski@iteraconsulting.com, imend@softservecom.com

Орехов Олександр Арсенійович, к.т.н., доцент, Компанія «Microsoft Україна» . Наукові інтереси – інформаційні технології, навчання
E-mail: oryekhov@microsoft.com

Стаття надійшла до редакції 20.05.2010 р.