

УДК 004.9:37.1

Бублик В. В., Глибовець М. М., Олецький О. В.

## ШЛЯХИ ТРАНСФОРМАЦІЇ ІТ-ОСВІТИ В НАПРЯМІ ПРОГРАМНОЇ ІНЖЕНЕРІЇ: ДОСВІД ФАКУЛЬТЕТУ ІНФОРМАТИКИ НаУКМА

*Розглянуто проблеми, пов'язані з наближенням навчальних програм із інформатики до вимог ІТ-індустрії в контексті розгортання бакалаврської програми «Програмна інженерія» на основі існуючих навчальних програм із комп'ютерних наук. Особливу увагу приділено впровадженню в навчальний процес практики виконання навчальних групових проектів із розробки програмних систем, а також організації і підтримці взаємодії «Бізнес-Освіта».*

Рух людства до постіндустріального, інформаційного суспільства набуває дедалі більш стрімкого та революційного характеру. Особливо швидкими й помітними є зміни в царині інформаційно-комп'ютерних технологій. Це призводить до того, що існуючі методики підготовки фахівців у цій галузі практично вичерпали свої ресурси, не відповідають сучасним вимогам і потребують кардинальної трансформації. При цьому ключовим принципом і стратегічним напрямом такої трансформації має стати **максимальне наближення навчального процесу до вимог ІТ-індустрії** [1]. Проте необхідним є органічне поєднання фундаментальних та прикладних знань і забезпечення достатнього фундамен-

тально-освітнього та наукового рівня навчального процесу – інакше університет просто перетвориться на навчальний центр при рекрутинговому агентстві.

Про існування серйозної кризи свідчить, зокрема, таке спостереження: кожні 2–3 роки спеціаліст у галузі інформаційно-комп'ютерних технологій повинен підвищувати свою кваліфікацію та перенавчатися з метою оновлення своїх знань. Це означає, що значна частина інформації, яку студент із напрямку «Комп'ютерні науки» засвоює на молодших курсах, на момент отримання ним диплому вже встигає застаріти.

Іншим викликом є необхідність якомога більш ранньої участі студентів у серйозних практичних

проектах, а можливостей сучасної університетської освіти для цього очевидно недостатньо. Загальна ситуація, за якої студенти влаштовуються на роботу на 2-му – 3-му курсах для того, щоб забезпечувати себе засобами для існування (зокрема, щоб мати можливість сплатити за своє навчання), ще більше поглиблює й загострює цю проблему.

Нарешті, сучасна ІТ-індустрія вимагає дедалі більшої кількості фахівців, здатних до проектування складних програмних продуктів і колективної роботи над ними, до створення якісних, гнучких і надійних програмних компонентів.

З цього випливає, що перехід до спеціальності «Програмна інженерія» вимагає від університету серйозних змін у системі освіти. Очевидно, що характер цих змін має безпосередньо визначатися моделями взаємодії основних учасників цього процесу, як-от:

- система вищої освіти, зокрема університет, його професорсько-викладацький склад, адміністрація та допоміжні підсистеми, які повинні забезпечувати навчальний процес на належному рівні;

- ринок праці як основний споживач результатів роботи університету; у разі інформаційно-комп'ютерних технологій це власне ІТ-індустрія, а також організації, які зацікавлені в інноваційних та наукових дослідженнях. Сюди також слід віднести й систему освіти, оскільки окремі випускники можуть залишатися у вищих навчальних закладах для викладацької та наукової роботи, а також у школах, середніх спеціальних навчальних закладах тощо;

- студенти, які мають набути необхідних знань і перетворитися на кваліфікованих фахівців.

Сьогодні ще не існує цілковитої ясності стосовно змісту й цілей навчальних програм, які становлять напрями підготовки в галузі інформаційно-комп'ютерних технологій. З огляду на це, в рекомендаціях АС М СС2005 [2] пропонується провести чітке розмежування й у рамках загального напрямку *Computing* («Інформатика і обчислювальна техніка») виділити такі частини:

- *Computer Science* («Комп'ютерні науки»); ця спеціальність має більш науково-дослідницький характер і фокусується на створенні нового знання; студенти повинні бути підготовлені до роботи в широкому діапазоні напрямів: від теоретичної роботи до розроблення програмного забезпечення;

- *Software Engineering* («Програмна інженерія») – фокусується на методах надійного проектування і створення програмних систем, що працюють відповідно до потреб і специфікацій;

випускники мають бути готовими підтримувати процес створення і експлуатації програмного забезпечення на всіх етапах його життєвого циклу;

- *Information Systems* («Інформаційні системи») – фокусується на інтеграції рішень у галузі інформаційних технологій і бізнес-процесів;

- *Information Technology* («Інформаційні технології») – готує студентів до застосування комп'ютерних технологій для потреб бізнесу, управлінської діяльності, охорони здоров'я, освіти та інших організаційних потреб;

- *Computer Engineering* – фокусується на проектуванні та впровадженні систем, які передбачають інтеграцію програмного та апаратного забезпечення.

Різницю між цими напрямками часто зображують у вигляді двовимірних діаграм, одна шкала яких варіює від теоретичних до прикладних знань, а інша відображає різні аспекти знань (організаційні питання та інформаційні системи; технології застосування; методи і технології розроблення програмного забезпечення; системна інфраструктура; апаратне забезпечення та архітектура комп'ютерів). Із цих діаграм, зокрема, випливає, що спеціальність «Програмна інженерія» має значний перетин зі спеціальністю «Комп'ютерні науки», але характеризується більш прагматичним і прикладним характером. Очевидним є й великий перетин із напрямком «Інформаційні системи».

В українській системі освіти, відповідно до постанови Кабінету Міністрів України від 13 грудня 2006 р., виділяються напрями «Системні науки та прикладна математика» та «Інформатика та обчислювальна техніка» з піднапрямами «Комп'ютерні науки», «Комп'ютерна інженерія» та «Програмна інженерія».

У вересні 2006 р. на факультеті інформатики НаУКМА відбувся перший набір студентів за спеціальністю «Програмна інженерія», і необхідність реорганізації навчального процесу та навчальних планів перейшла в практичну площину.

Рекомендація з викладання програмної інженерії SE2004 [3] акцентує увагу на таких ключових моментах навчального процесу:

- розробники навчальних планів і викладачі мають бути орієнтовані на кінцевий результат;

- курси мають бути достатньо гнучкими, і треба залишати простір для інновацій;

- планованість і послідовність у викладанні; навчальний матеріал має бути структурований таким чином, щоб студенти повною мірою оцінили важливість і мотивацію базових принципів;

– студенти повинні також вивчити певну предметну область, що не належить до програмної інженерії;

– курси з програмної інженерії мають демонструвати належність цієї дисципліни як до інформатики, так і до інженерії;

– крім власне знань, студенти повинні отримати ряд інших навичок, необхідних для професійної роботи, як-от: вміння вчитися, вміння працювати в команді, комунікабельність, креативність тощо;

– програмну інженерію мають викладати як проблемно-орієнтовану дисципліну;

– найбільший акцент слід зробити на фундаментальних і незмінних принципах програмної інженерії, а не на новітніх або спеціалізованих засобах. Проте саме викладання мають вести з використанням адекватних програмних засобів;

– матеріал, що викладається, повинен мати строге математичне чи теоретичне обґрунтування або бути загальноприйнятою професійною практикою;

– навчальний план має ґрунтуватися на реальному практичному досвіді, передбачати навчальні приклади, проектно-орієнтовані курси, дипломні проекти тощо;

– вивчення програмної інженерії має вийти за межі лекційного формату й передбачати впровадження нових форм навчання;

– навчальний план і курси, що до нього належать, мають постійно оновлюватися й розвиватися.

Як магістральний напрям на бакалаврській програмі факультету мають бути передбачені такі основні групи курсів з програмування та програмної інженерії:

– основи алгоритмічного програмування (на базі Паскалю);

– методи об'єктно-орієнтованого програмування та проектування;

– власне програмна інженерія;

– методи та засоби роботи в команді.

Очевидно, що ряд фундаментальних понять із програмної інженерії слід вводити якомога раніше. Зокрема, видається, що вже на початку навчання студенти повинні отримати базові уявлення про UML (так, як раніше вони отримували знання про блок-схеми). Існуючі курси мають робити основний акцент не стільки на конструкціях мов програмування, скільки на архітектурних рішеннях та методах проектування програмних комплексів. Відповідно до рекомендацій [3] має бути також забезпечена фундаментальна підготовка з базових математичних і комп'ютерних наук (таких, як дискретна математика, алгорит-

ми і структури даних; основи баз даних і знань; прикладна теорія ймовірностей і статистика тощо).

Крім того, рекомендації [3] заохочують включення до навчальних планів інших нормативних і вибіркового курсів – зокрема наукових або орієнтованих на певні прикладні застосування. Як деякі можливі напрями можна відмітити проектування й програмування розподілених та веб-орієнтованих інформаційних систем; захист інформації; фінансові системи та електронну комерцію; мультимедійні системи; системи для мобільних платформ; інтелектуальні агенти; проектування інтелектуальних інформаційних систем; телекомунікаційні системи та ін. Такий курс, зокрема, міг би називатися «Програмна інженерія в галузі...».

Одна з ключових проблем, яку необхідно вирішити на шляху реформування підготовки ІТ-спеціалістів, особливо в контексті переходу до спеціальності «Програмна інженерія», – якомога більш раннє залучення студентів до виконання практичних проектів в умовах, якомога більш наближених до реальних. Ідеться й про індивідуальну роботу, і про участь студентів у великих групових проектах, оскільки саме цього вимагає сучасна програмна інженерія, і саме такою роботою випускникам найвірогідніше доведеться займатися в повсякденній професійній діяльності.

Серйозний крок до цього зроблено в рамках сумісного європейського проекту ITSoftTeam, який виконується на факультеті інформатики НаУКМА [4]. Так, до навчального плану включено новий курс «Групова розробка програмних систем (груповий проект)». Але цього недостатньо, і, крім власне внесення змін до навчального плану, видається доцільним впроваджувати виконання практичних проектів, максимально наближених до реальної роботи, до програм окремих курсів. Зокрема, йдеться про виконання колективних проектів командами з 3–4 студентів. З одного боку, робота в команді сприяє створенню більш повноцінного продукту, а з іншого – кожний студент-учасник команди отримує більш широкі можливості для навчання, розкриття свого потенціалу, а також набуває досвіду. Важливим видається й залучення до оцінювання таких проектів інших студентів.

Суттєвого значення набуває наближення до індустріальних стандартів. Зазначимо один із таких: група з 10 розробників протягом року має написати від 300 000 до 500 000 рядків коду (таким чином, продуктивність роботи програміста має становити 20–30 рядків на годину). Коли

йдеться про студентський груповий проект протягом самостійної роботи (орієнтовно 139 год), то команда з 3 студентів у разі продуктивності роботи 20 рядків на годину має написати 8000 рядків. Ключовим є запитання: наскільки це реально і як цього досягти?

Тут виникає ще одна проблема, а саме – проблема ефективності [5]. Якщо програма з 10 рядків повністю еквівалентна програмі, що налічує 1000 рядків, то перша програма однозначно є кращою за другу, і програміст, який її створив, має вважатися більш ефективним і отримувати більш високу заробітну платню. Але це вже інше питання.

Крім того, перехід до спеціальності «Програмна інженерія» висуває нові, більш серйозні вимоги до розробників навчальних планів та викладачів. Зокрема, рекомендації [3] визначають такі основні вимоги:

- наявність знань із програмної інженерії;
- практичний досвід роботи в програмній інженерії;
- визнання знань у сфері програмної інженерії, отримане або шляхом публікації робіт із даної тематики, або шляхом участі в професійних співтовариствах;
- постійне ознайомлення з новими галузями програмної інженерії;
- наявність мотивованості та можливостей для підтримки високої обізнаності про останні напрацювання.

У цьому контексті набуває особливо великого значення тісна взаємодія університету та бізнесових структур і забезпечення зворотного зв'язку з боку IT-бізнесу. Слід зазначити, що інвестиції в знання стають дедалі більш привабливими для бізнесових кіл, і IT-індустрія зацікавлена в збільшенні кількості кваліфікованих випускників і відповідно – в реорганізації навчального процесу в галузі IT-освіти та в покращенні його якості.

З цього безпосередньо випливає те, що необхідно говорити про становлення й розвиток повноцінної системи Private-Public Partnership у секторі «Бізнес-Освіта». В ідеалі може йтися про організацію аналітично-дослідницького центру знань, основними завданнями якого мають бути:

- вивчення потреб IT-бізнесу; вироблення на основі цього рекомендацій та здійснення практичних кроків у напрямі модифікації вузівських програм;
- забезпечення для IT-бізнесу та інших зацікавлених кіл можливості безпосередньо впливати на процес підготовки майбутніх фахівців, а також брати безпосередню участь у цьому процесі;

- акумуляція та поширення досвіду; проведення семінарів і конференцій, у тому числі віртуальних;

- популяризація та просування ідей щодо розвитку IT-технологій;

- реалізація елементів Job-центру; підтримка бази даних з виконаних проектів та їх виконавців;

- співпраця зі школами; в першу чергу – з метою впровадження окремих елементів у шкільні програми;

- співпраця з науковими установами.

Можна розглядати такі конкретні форми взаємодії університету та IT-індустрії:

- внесення представниками IT-індустрії пропозицій щодо окремих проектів;

- залучення представників IT-індустрії до керування курсовими, читання лекцій; заслуговує на увагу ідея можливого подвійного керування курсовими роботами;

- залучення професорсько-викладацького складу до семінарів, тренінгів, а також до участі в реальних практичних проектах у ролі програмних архітекторів, наукових консультантів тощо;
- підвищення кваліфікації викладачів.

Важливим, ефективним і постійно діючим інструментом для забезпечення ефективної взаємодії «Бізнес-Освіта» має стати Інтернет-портал з такими основними елементами:

- віртуальне співтовариство як основа порталу;

- власне порталні риси: надання необхідної інформації;

- реєстр виконаних проектів та їх виконавців;

- засоби для внесення пропозицій щодо нових проектів та оцінювання й обговорення цих пропозицій;

- середовище для підтримки групової роботи над проектами;

- підтримка концепції віртуального робочого кабінету як розвитку e-folio («електронного портфелю» [6]); крім власне доступу до особистих документів, такий кабінет має забезпечувати зручне персоналізоване середовище для роботи;

- реєстр власне навчальних матеріалів і на основі цього – засобів підтримки навчального процесу з орієнтацією на ідеологію колаборативного електронного навчання [7] і blended learning [8];

- електронний журнал; система підтримки віртуальних конференцій;

- інші елементи, зокрема форум.

Певні елементи такого середовища створено на факультеті в межах більш ранніх міжнародних

проектів [4]. Нині працює міжуніверситетський гурток, у рамках якого обговорюється концепція майбутнього порталу й ведеться робота над створенням його прототипу.

1. [www.career-space.com](http://www.career-space.com).
2. Computing Curricula 2005.— <http://www.acm.org/education>.
3. Рекомендації по преподаванію программной инженерии и информатики в университетах.— М.: ИНТУИТ-РУ, 2007.— 462 с.
4. Бублик М. М., Глибовець М. М., Олецький О. В. Моделі трансформації інформаційної освіти в контексті руху до інформаційного суспільства: досвід факультету інформатики НаУКМА // Наукові праці. Науково-методичний журнал.— Т.71.— Вип. 58. Педагогічні науки.— Миколаїв: Вид-во МДГУ ім. П. Могили, 2007.— С. 60–64.
5. Макконнелл С. Совершенный код. Мастер-класс.— М.: Русская редакция; СПб.: Питер, 2007.— 896 с.
6. Глибовець М. М., Бублик В. В. EMERECU – темпусівський грант створення електронного мультимедійного центру навчальних матеріалів для України. Наукові праці.— Миколаїв: Вид-во МДГУ ім. П. Могили, 2006.— Т. 46.— Вип. 33. Педагогічні науки.— С. 194–195.
7. Boublik V., Hesser W., Schmidt-Braul I. Towards cooperative e-teaching and e-learning. // Теоретичні та прикладні аспекти побудови програмних систем. Матеріали міжнародної конференції TAAPSD'2006.— К., 2006.— С. 231–235.
8. Hartung M., Hesser W., Koch K. Aufbau von Blended Learning mit der open source E-Lernplattform ILIAS an einer Campus-Universität. Digitaler Campus.— <http://beat.doebe.li/bibliothek/b01455.html>.

V. Boublik, M. Glybovets, O. Oletskyi

#### THE IT-LEARNING TRANSFORMATION WAYS IN THE DIRECTION OF PROGRAM ENGINEERING: THE EXPERIENCE OF THE COMPUTER SCIENCE FACULTY AT THE NaUKMA

*The problems connected with the alignment of the educational programs in computer sciences with the demands of the IT-industry in the context of the progress of bachelor program «Program engineering» based on existing computer science educational programs. The article focuses on the introduction of the practice of group software development project realization, and the organization and support of interaction between Business and Education.*