

УДК 378.147:004.4

Є. В. Тимошенко,

кандидат технічних наук

(Бердянський державний педагогічний університет)

МЕТОДИЧНІ ПІДХОДИ ДО НАВЧАННЯ ПРОГРАМНОЇ ІНЖЕНЕРІЇ У ВИЩИХ НАВЧАЛЬНИХ ЗАКЛАДАХ

Постановка проблеми. За останній час з'явилося багато ефективних методологій і середовищ розробки програмного забезпечення (ПЗ), структурне та об'єктно–орієнтоване програмування, моделі вдосконалювання програмних процесів, засоби автоматизації розробки ПЗ та мови четвертого покоління, а тому все більше виникає дискусій [1; 4; 6; 11] про стратегії навчання програмній інженерії (ПІ). Вважається, що навчальний заклад має готувати фахівців, здатних приступати до виробничої діяльності одразу після одержання диплома. Тому навчання має бути організоване так, щоб студенти володіли не тільки фундаментальними знаннями в галузі комп'ютерних наук, але й були здатні знаходити ефективні варіанти проектних рішень в умовах жорстких вимог до строків реалізації, широти функціональних можливостей програмного продукту, доступності ціни, різноманіття потреб користувачів і обмеження архітектури ОТ, а також могли самостійно підвищували свою кваліфікацію. Висока потреба у кваліфікованих інженерах ПЗ спостерігається в усьому світі, Україна тут не виняток. Вітчизняний ринок праці вимагає компетентних фахівців у галузі інформаційних технологій (ІТ), і завдання освітніх установ сприяти якісному задоволенню цього попиту.

Аналіз досліджень і публікацій. Дослідженнями в галузі ПІ в період кінця ХХ ст. – початку ХХІ ст. займаються як окремі вчені (З. Дудар, Е. Лавріщева, В. Ліпав, С. Орлік, М. Brooy, Т Lethbridge, В. Meyer, I. Sommerville, Н. van Vliet) [1; 3-7; 11], так і авторські колективи: Харківський національний університет радіоелектроніки, Association for

Computing Machinery, Institute of Electrical and Electronics Engineers [2; 8; 9; 10]. Результатом цих наукових досліджень є низка підручників та наукових статей, які відображають різні теоретичні, методичні та професійні аспекти.

У контексті підготовки фахівців у галузі ПІ виділяють такі підходи:

1. Введення в навчальні програми з певних комп'ютерних дисциплін окремих елементів ПІ.
2. Створення самостійної спеціальності “Програмна інженерія”.
3. Сертифіковане навчання ПІ як професії на курсах підготовки або перепідготовки спеціалістів у галузі ІТ.

Ці підходи до підготовки фахівців у галузі ПІ використовуються в США, Канаді, Великобританії та багатьох країнах ЄС. До недавнього викладання ПІ на факультетах інформатики українських ВНЗ зводилося до навчання мовам програмування (*C++*, *Pascal*, *Basic* та ін.), теорії алгоритмів, основам операційних систем, СУБД, інформаційним технологіям та системам. Отже, системне викладання цієї дисципліни майже відсутнє.

Мета статті – провести порівняльний аналіз методичних підходів до навчання ПІ та сформулювати методичні рекомендації щодо вдосконалення процесу підготовки фахівців цієї галузі знань у навчальних закладах України.

Сьогодні знання й досвід, накопичені в індустрії ПЗ за попередні десятиріччя, оформилися в окрему дисципліну ПІ. Завдяки розпочатій ще в 1993 р. спільній роботі Асоціації з обчислювальної техніки (*Association of Computer Machinery, ACM*) та Інституту інженерів з електротехніки й електроніки (*Institute for Electrical and Electronic Engineers, IEEE*) в 2004 р. були розроблені дві фундаментальні програми з програмної інженерії (*Software Engineering*):

1. *Guide to the Software Engineering Body of Knowledge (SWEBOK), IEEE 2004 Version* – Керівництво із отримання знань з програмної інженерії [9];
2. *Software Engineering 2004. Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering (SE2004)* – Рекомендації з викладання програмної інженерії в університетах [10].

Керівництво *SWEBOOK* містить лише базові визначення й описи змістових ліній (ЗЛ), однак не охоплює всіх питань створення ПЗ. Опис ЗЛ в *SWEBOOK* побудовано за ієрархічним принципом. При цьому деталізація ЗЛ виконана тільки на тому рівні розуміння природи відповідних тем, який дає можливість самостійного знаходження джерел компетенції, довідкових даних і матеріалів. *SWEBOOK* описує 10 ЗЛ:

- *Software requirements* – програмні вимоги.
- *Software design* – дизайн (архітектура).
- *Software construction* – конструювання програмного забезпечення.
- *Software testing* – тестування.
- *Software maintenance* – експлуатація (підтримка) ПЗ.
- *Software configuration management* – конфігураційне керування.
- *Software engineering management* – управління в програмній інженерії.
- *Software engineering process* – процеси програмної інженерії.
- *Software engineering tools and methods* – інструменти й методи.
- *Software quality* – якість програмного забезпечення.

Метою *SWEBOOK* являється саме визначення й систематизація тих аспектів діяльності, які є підґрунтям професійної підготовки інженера-програміста.

У рекомендаціях *SE2004* зазначено, що ПЗ є одночасно обчислювальною й інженерною дисципліною. У *SE2004* наводяться зразки навчальних планів із спеціальності ПЗ; подано принципи викладання курсу ПЗ, перелік знань, умінь та навичок (*Software Engineering Education Knowledge, SEEK*), якими має оволодіти студент при вивченні цього курсу:

- уміти індивідуально й у групі працювати над створенням програм;
- володіти знаннями й навичками, достатніми для практичної роботи;
- знаходити оптимальні проектні рішення в умовах обмежень (техніко-технологічних, часових, фінансових, людських тощо);
- проектувати в декількох предметних галузях;
- освоювати нові моделі, методи й технології;

- розуміти й застосовувати теорії, моделі та методи ПЗ;
- мати навички міжособистісного спілкування, колективної роботи й лідерства.

Навчальний матеріал *SEEK* поділений на десять ЗЛ, кожна з яких має кілька модулів, розбитих за темами:

- Основи ІТ (інформатика; технології та інструментальні засоби створення програм) (172 години).

- Основи математики та інженерії (дискретна математика, графіка, числові помилки, статистика, виміри; емпіричні методи та методики експериментів; пошук рішення, взаємоприйнятне для учасників проекту, розміщення пріоритетів, аналіз компромісів) (89 годин).

- Професійна практика (групова динаміка, психологічні принципи; навички спілкування; етика, професійне поведіння, юридичні питання, професійні суспільства, стандарти) (35 годин).

- Моделювання та аналіз ПЗ (принципи та мови моделювання, синтаксис і семантика; поведінкове, структурне та доменне моделювання; аналіз правильності, коректності і якості; інженерія вимог) (53 години).

- Проектування програмного забезпечення (принципи, компроміси, завдання, шаблони, архітектура, стратегії, проектування людино-машинного інтерфейсу, детальне проектування, інструментарій, оцінка) (45 годин).

- Верифікація та атестація ПЗ (мети, параметри й показники, огляди, контрольний аналіз; тестування; аналіз людино-машинного інтерфейсу, аналіз проблем і підготовка відповідних звітів) (42 години).

- Еволюція програмного забезпечення (процеси, моделі, закони еволюції, планування еволюції, розуміння програм і зворотна інженерія, аналіз впливів, перенос, рефакторинг) (10 годин).

- Процеси розробки програмного забезпечення (моделі циклу життя, стандарти, окремі процеси; інфраструктура процесів, моделювання процесів, аналіз якості процесів і контроль), 13 годин.

- Якість програмного забезпечення (зацікавленість суспільства в якості,

атрибути якості, витрати й наслідки поганої якості; процеси й стандарти якості; керування якістю процесів і продуктів) (16 годин).

- Управління програмними проектами (моделі управління, планування, організація, контроль; управління випуском і конфігурацією) (19 годин).

Зазначені ЗЛ носять характер методичних рекомендацій, які викладач може використовувати при розробці робочої навчальної програми, де визначає пропорцію модулів усього курсу. Кожному навчальному модулю освітня установа відводить мінімальне число лекційних годин та самостійно визначає кількість навчальних годин на практичні та лабораторні заняття.

Висновки. Відповідно до вимог підготовки фахівців з напрямку 050103 “Програмна інженерія” [2] програми *SWEBOOK* та *SE2004* мають такі взаємодоповнюючі особливості:

1. *SWEBOOK* надає систему знань, якою має володіти досвідчений фахівець; *SE2004* надає знання, якими необхідно оволодіти студентам.

2. *SE2004* має більший діапазон тем (професіоналізм, управління проектами, інформатика та математика), які в *SWEBOOK* відсутні.

3. *SE2004* розглядає певні ЗЛ як основні (проекування людино-машинного інтерфейсу), а *SWEBOOK* відносить їх до суміжних дисциплін.

4. ЗЛ *SE2004*, які спочатку були схожі із ЗЛ *SWEBOOK*, піддалися істотним змінам.

У контексті аналізу світової практики навчання ПІ ми пропонуємо такі методичні рекомендації щодо вдосконалення процесу підготовки фахівців цієї галузі знань у навчальних закладах України:

1. Розробляючи навчальні програми з певних курсів, необхідно приділяти увагу таким темам *SE2004*: групова динаміка, професійна практика, еволюція ПЗ, проектування людино-машинного інтерфейсу, інженерна економіка ПЗ.

2. Змінити відповідно до рекомендацій *SE2004* підходи до виконання курсових та дипломних проектів, а саме: орієнтуватись на групові проекти, які мають носити виробничий характер.

3. Потрібно розвивати співробітництво відповідних університетських кафедр із підприємствами, що розробляють ПЗ: організація виробничої практики для студентів, спільні проекти, залучення співробітників компаній до викладання дисциплін у вишах.

Перспективи подальших пошуків. Розробити навчально-методичний комплекс з дисципліни ПІ на основі світових програм *SWEBOOK* та *SE2004* відповідно до вимог підготовки фахівців з цього напрямку.

ЛІТЕРАТУРА

1. Брой М. Теория и практика программной инженерии [Электронный ресурс] / Манфред Брой // Открытые системы. – 2011. – № 9. – Режим доступа : <http://www.osp.ru/os/2011/09/13011564/>

2. Галузевий стандарт вищої освіти України. Освітньо-кваліфікаційна характеристика бакалавр, напрямку підготовки 050103 “Програмна інженерія”. – К. : МОН України, 2008. – 24 с.

3. Лаврищева Е.М. Методы и средства инженерии программного обеспечения : учеб. пособ. / Е. М. Лаврищева, В. А. Петрухин. – М. : МФТИ, 2006. – 304 с.

4. Летбридж Т. SE2004: рекомендации по обучению специальности “Программная инженерия” [Электронный ресурс] / Тимоти Летбридж, Ричард Лебланк, Энн Келли Собел // Открытые системы. – 2006. – № 10. – Режим доступа : <http://www.osp.ru/os/2006/10/3910113/>

5. Липаев В.В. Программная инженерия. Методологические основы: учеб. пособ. / В. В. Липав. – М. : ТЕИС, 2006. – 608 с.

6. Мейер Б. Программная инженерия как предмет обучения / Бертран Майер // Открытые системы. – 2001. – № 7, 8. – С. 80 – 86.

7. Соммервиль И. Инженерия программного обеспечения / Иан Соммервиль. – 6-е изд. перераб. и доп.; пер. с англ. – М. : Издательский дом “Вильямс”, 2002. – 624 с.

8. Computing Curricula 2005. The Overview Report [Электронный ресурс].

– Режим доступа : <http://www.acm.org/education/curricula.html>

9. Guide to Software Engineering Base of Knowledge (SWEBOK)
[Электронный ресурс]. – 2004. – Режим доступа : <http://www.swebok.org/>

10 Software Engineering 2004. Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering [Электронный ресурс]. – 2004. – Режим доступа : <http://sites.computer.org/ccse/>

11. H. van Vliet, Reflections on Software Engineering Education, IEEE Software, May/June, 2006. – pp. 55-61.