УДК 372.862

Г.С. ПОГРОМСЬКА,
кандидат педагогічних наук, доцент, доцент кафедри прикладних математики і механіки та інформатики Миколаївського національного університету імені В.О. Сухомлинського

Н.А. МАХРОВСЬКА,
кандидат фізико-математичних наук, доцент кафедри прикладних математики і механіки та інформатики Миколаївського національного університету імені В.О. Сухомлинського

МЕТОДИЧНІ АСПЕКТИ ВИКЛАДАННЯ ДИСЦИПЛІНИ «ІНЖЕНЕРІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ» ДЛЯ СТУДЕНТІВ-ІНФОРМАТИКІВ

У статті описано методичні аспекти викладання дисципліни «Інженерія програмного забезпечення» для студентів, які навчаються за напрямом «Системні науки та кібернетика». Надано рекомендації щодо створення навчально-методичного комплексу, наведено змістове наповнення теоретичного курсу із зазначеної дисципліни та рекомендовано форми проведення лекційних занять. З огляду на те, що курс «Інженерія програмного забезпечення» через його зв'язок із практичними потребами замовників має високу динамічність, обумовлену специфікою предметної галузі, що базується на сучасних інформаційних системах і технологіях їх побудови, особлива увага приділена розкриттю методики проведення лабораторних занять з курсу.

Ключові слова: інженерія програмного забезпечення, лабораторна робота, лекція, програмна інженерія, програмне забезпечення, стандарт, форма навчання, SWEBOK.

Постановка проблеми. Формування у студента усвідомлення соціальної значущості майбутньої професії, мотивації до отримання професійних знань, розуміння та освоєння основних концепцій і змісту програмної інженерії як методології індустріального проектування прикладних програмних продуктів є одними із завдань дисципліни «Інженерія програмного забезпечення». Зазначене є невід'ємною умовою при підготовці сучасних спеціалістів у ВНЗ для кар'єри у промисловості і науці з акцентом на формування компетенцій аналізу, проектування, перевірки, впровадження, застосування і підтримки системи програмного забезпечення.

Ця стаття є логічним продовженням матеріалу, викладеного в публікації авторів під назвою «Проблемні питання методики викладання дисципліни «Інженерія програмного забезпечення» для спеціальності «Інформатика». У статті було здійснено аналіз стандартів з програмної інженерії. Досвід викладання програмної інженерії до теперішнього часу акумульовано у вигляді двох міжнародних стандартів:

1. Guide to the Software Engineering Body of Knowledge (SWEBOK) – сукупність знань з програмної інженерії, далі просто SWEBOK [7].

2. Software Engineering. Curriculum Guidelines for Undergraduate Degree Programsin Software Engineering – Рекомендації з викладання програмної інженерії та інформатики в університетах [10].

Розглянуто проблемні питання викладання дисципліни «Інженерія програмного забезпечення» для спеціальності «Інформатика». Серед останніх виділено:

1. Необхідність будувати методику викладання дисципліни «Інженерія ПЗ» на базі промислового проекту. Останнє призводить до зайвого навантаження на студентів та до

© *Г.С. Погромська, Н.А. Махровська, 2015*

доцільності концентрувати увагу максимум на декількох проблемах в кожному спеціально підібраному середовищі, але в загальних рисах розкривати всі необхідні теми курсу, при тому що навчальний проект охоплює лише декілька ключових проблем.

2. Акцент на подібності курсу «Інженерія ПЗ» до інших інженерних дисциплін. Загальна ідея полягає в тому, що подібності превалюють, але, звичайно, існують і суттєві відмінності.

3. Планування в програмній інженерії здійснюється гірше, ніж в інших галузях.

4. Інтегрування адекватних методів проектування інтерфейсу користувача в процесі розробки ПЗ.

5. Відставання або ж випередження SWEBOK і SE 2004 в окремих галузях від практики.

6. Неоднорідність інженерії програмного забезпечення і відповідне ускладнення галузевої практики.

Аналіз останніх досліджень. Дидактичні та педагогічні питання викладання інженерії ПЗ висвітлюються у науково-дослідницьких працях авторів Л.-Б. Раккуна, М. Шоу, Д. Херіслеба, С. Лю, К. Такахаші, Т. Хаясі, В. Блік, Т.Ю. Морозової, З.С. Сейдаметова, В.А. Темненко, М.О. Сидорова, О.А. Орехова, І.Б. Мендзебровського та ін. [2–5; 8]. Авторами А. Келлі, Е. Річмондом, Е. Робертсом, Х. Влітом, Д. Лі, Ю. Ченом, М. Шоу, М. Луцьким, М. Сидоровим [1; 6; 9] та ін. порушуються питання методики та підходи, що використовуються у системі викладання інженерії ПЗ.

Формулювання мети. Метою цієї статті є опис окремих методичних аспектів викладання дисципліни «Інженерія програмного забезпечення» для студентів, які навчаються за напрямом «Системні науки та кібернетика» спеціальності «Інформатика».

Виклад основного матеріалу. У першу чергу необхідне методичне забезпечення дисципліни. З цією метою на кафедрі «Прикладні математика, механіка та інформатика» авторами статті розроблено навчально-методичний комплекс дисципліни, який включає такі компоненти:

1. Навчальну програму дисципліни.

2. Робочу програму дисципліни.

3. Методичні матеріали з вивчення дисципліни для студентів у вигляді методичних рекомендацій, які містять: навчально-методичні матеріали для лекцій, навчально-методичні матеріали для лабораторних робіт, навчально-методичні матеріали для самостійної роботи.

4. Методичні матеріали для поточного контролю успішності студентів: завдання для виконання модульних контрольних робіт (у формі тестування) та комплексна контрольна робота.

5. Методичні матеріали для атестації студентів: питання та завдання до іспиту.

Теоретичні заняття у вигляді лекцій є традиційними для університетської освіти. Лекційний курс побудований на базі останнього стандарту в області програмної інженерії – SWEBOK 2004. Тематика лекцій відповідає назвам базових галузей знань програмної інженерії відповідно до проекту SWEBOK: 1) Software Requirements (Програмні вимоги); 2) Software Design (Проектування ПЗ); 3) Software Construction (Конструювання ПЗ); 4) Software Testing (Тестування ПЗ); 5) Software Maintenance (Супровід ПЗ); 6) Software Configuration Management (Конфігураційне управління ПЗ); 7) Software Engineering Management (Управління ПІ); 8) Software Engineering Process (Процеси ПІ); 9) Software Engineering Models and Methods (Моделі і методи програмної інженерії); 10) Software Quality (Якість ПЗ).

Відповідно до окреслених тем зміст лекцій є таким:

Тема 1. Вступ до інженерії ПЗ. Життєвий цикл ПЗ. (Поняття програмної інженерії. Основні визначення: інформатика, системотехніка, бізнес-реінжиніринг. Програмне забезпечення: визначення, властивості. Поняття процесу розробки ПЗ. Універсальний процес. Поточний процес. Конкретний процес. Стандартний процес. Удосконалення процесу. Pull / Push стратегії. Класичні моделі процесу: водоспадна модель, спіральна модель. Фази і види діяльності).

Тема 2. Робочий продукт, дисципліна обов'язків, проект. (Робочий продукт. Дисципліна зобов'язань. Проект. Управління проектами).

Тема 3. Архітектура ПЗ. (Архітектура ПЗ. Точка зору і характеристики точок зору. Множинність точок зору при розробці ПЗ, компонент архітектури, подання архітектури, сцена-

рій використання, методи оцінки архітектури / UML. Діаграми класів, діаграми взаємодії, діаграми сценаріїв, діаграми компонентів, діаграми розгортання).

Тема 4. Аналіз предметної області та вимоги до ПЗ. (Аналіз предметної галузі, схема Захмана, моделі предметної області, діаграми потоків даних, діаграми сутностей і зв'язків, функції ПЗ, вимоги до ПЗ, варіанти використання, дійові особи, діаграми варіантів використання. Види вимог: функціональні вимоги, нефункціональні вимоги. Властивості вимог: ясність і недвозначність, повнота і несуперечність, необхідний рівень деталізації, простежуваність, тестованість, перевірюваність, модифікованість. Формалізація вимог. Цикл роботи з вимогами).

Тема 5. Конфігураційне управління. (Поняття конфігураційного управління. Управління версіями. Поняття «гілки» проекту. Управління збірками. Засоби версійного контролю. Одиниці конфігураційного управління. Поняття baseline).

Тема 6. Якість ПЗ та методи його контролю. (Якість ПЗ, функціональність, надійність, зручність використання, продуктивність, зручність супроводу, можливість переносу, методи контролю якості, тестування, перевірка властивостей ПО на моделях, помилки в ПЗ. Стандартизація якості. Методи забезпечення якості ПЗ. Поняття тестування. Тестування чорної скрині. Тестування білої скрині. Інструменти тестування. Критерії тестування. Види тестування. Робота з помилками. Засоби контролю помилок – bug tracking systems).

Тема 7. Діаграмні техніки при роботі зі знаннями. (Випадки використання. Робота з вимогами. Випадки використання в управлінні розробкою. Ітеративний цикл автор / рецензент. Карти пам'яті).

Тема 8. MSF. CMMI. (IT-рішення. Основні принципи MSF. Модель команди: основні принципи, рольові кластери. Масштабування команди MSF. Модель процесу. Управління компромісами. Поняття CMMI. Рівні зрілості процесів за CMMI. Області удосконалення).

Тема 9. «Гнучкі» методи розробки. Уніфікований процес розробки та XP. (Загальний опис «гнучких» методів розробки ПЗ. Extreme Programming: загальний опис, основні принципи організації процесу. Scrum: загальний опис, ролі, практики. «Важкі» процеси розробки, «живі» методи розробки, уніфікований процес Rational, екстремальне програмування, моделі ПЗ).

Виклад кожної лекційної теми ведеться на такому рівні, щоб забезпечити студенту розуміння природи загальноприйнятих тем курсу «Інженерія ПЗ» і подальше успішне знаходження необхідних довідкових матеріалів. Необхідні студентам знання містяться в довідкових матеріалах, а не в керівництві SWEBOK. Проблему малого обсягу лекційних годин і великого обсягу теоретичного матеріалу долаємо таким чином: краще щось не встигнути, ніж щось не зрозуміти. Тому ми при необхідності зменшуємо обсяг теоретичних знань в курсі, які не піддаються практичному освоєнню, і більш поглиблено опрацьовуємо матеріал, що залишився. Використовуємо велику кількість «живих» прикладів у лекціях, застосовуючи відеоролики, комп'ютерні презентації. Під час проведення лекцій доцільним є застосування інтерактивної форми навчання: розбір конкретних ситуацій на прикладах під час вивчення та закріплення візуального моделювання та проектування.

Підтримується постійний зворотний зв'язок від студентів під час читання лекцій і проведення лабораторних занять. Особлива організація іспиту – в практико-орієнтованій формі.

Основний акцент в освоєнні курсу «Інженерія ПЗ» зроблено на лабораторних заняттях. Методика проведення лабораторних занять добре відпрацьована і проходить відповідно до практикуму з програмної інженерії, що оформлено у вигляді методичних рекомендацій до курсу.

Дисципліна через її зв'язок із практичними потребами замовників має високу динамічність, обумовлену специфікою предметної галузі, що базується на сучасних інформаційних системах і технологіях їх побудови. Як наслідок, для більш повного задоволення потреби особистості у здобутті освіти, адекватної потребам сучасного суспільства, тематика лабораторних робіт передбачає використання різних інструментальних засобів для розробки за вибором студента, також припустимі і вітаються особисті ініціативи студентів у пропонуванні тематики розробки програмного забезпечення.

Тематика лабораторних робіт передбачається така:

1. Аналіз завдання. Метою діяльності є сформувати навички роботи з реальними замовниками програмних систем; ідентифікації зацікавлених осіб та інтерв'ю з ними; аналі-

зу отриманого матеріалу; формулювання проблеми, її актуальності і потреб зацікавлених осіб.

2. Розробка моделі варіантів використання та їх специфікацій. За мету ставить сформувати навички розробки моделі варіантів використання та специфікації варіантів використання.

3. Розробка технічного завдання (Стадія «Технічне завдання»). Мета роботи – ознайомитися з правилами написання технічного завдання, сформувати навички розробки технічного завдання відповідно до ГОСТ 34.602-89.

4. Стадія «Ескізний проект». Мета – навчитися створювати формальні моделі і на їх основі визначати специфікації розроблюваного програмного забезпечення.

5. Етапи розробки ПЗ. Стадія «Реалізація». Мета – розробка програмного продукту відповідно до заданого варіанта.

6. Тестування програм методом «білої скрині». Метою роботи є вивчення методів тестування логіки програми, формалізовані описи результатів тестування і стандарти зі складання схем програм.

Пропонується наскрізна тема на всі виконувані лабораторні роботи. Як тему робіт можна використовувати одну з наданих викладачем або тему курсової роботи з об'єктно-орієнтованого програмування і для неї виконати етапи: бізнес-аналізу, бізнес-моделювання, проектування архітектури, і, звичайно ж, документування зазначених етапів.

Вправ у комп'ютерному класі не достатньо для формування повноцінного спеціаліста, тому передбачено індивідуальні завдання для самостійної роботи. Індивідуальні завдання являють собою індивідуальне дослідження, виконане студентом за заданою тематикою в години самостійної роботи із застосуванням засобів і можливостей інформаційних систем і технологій. Завдання для самостійної роботи має на меті:

– вироблення у студентів навичок самостійної роботи з наявною вихідною інформацією;

– практичну реалізацію студентами теоретичних знань з використанням інструментальних засобів інформаційних систем і технологій;

– формування і розвиток у студентів науково-дослідницьких навичок пошуку, вибору, аналізу та обробки технологічної інформації;

– комплексне застосування компетенцій, теоретичних знань, практичних навичок і вмінь, набутих студентами при вивченні дисципліни.

Професійно-орієнтоване індивідуальне завдання виконується за допомогою CASE-засобів CA ERwin Data Modeler, AllFusion Process Modeler 7 (BPwin), Rational Rose, середовище програмування (обирається студентом самостійно), наприклад MS Visual Studio, Borland C++ Builder, Borland Delphi, NetBeans, Eclipse PDT та ін. Завдання включає в себе в спрощеному вигляді весь комплекс робіт з проектування, створення та налагодження предметно-орієнтованої бази даних та програмного забезпечення з певної предметної галузі.

Звітними документами з лабораторного курсу є: 1) пояснювальна записка; 2) електронний файл програмного забезпечення, виконаний строго за своїм варіантом завдання.

Підсумковий звіт оформлюється у вигляді пояснювальної записки (шаблон структури документа надається викладачем) з додатками, якими виступають документи, розроблені при виконанні всіх лабораторних робіт.

Проміжні звіти (по кожній лабораторній роботі) і підсумковий звіт здаються в електронному та паперовому вигляді, за винятком деяких об'ємних документів, які допустимо включати в підсумковий звіт в електронному вигляді.

Звіт повинен містити документацію розробки програмного продукту. Для кожного варіанта завдання наводиться мінімальний набір характеристик предметної галузі автоматизованої інформаційної системи.

У підсумку зазначимо, що, виходячи з розглянутих проблемних питань та методичних аспектів викладання дисципліни «Інженерія програмного забезпечення», можна виділити узагальнені етапи для покращання системи викладання зазначеної дисципліни:

– викладання дисципліни з використанням новітніх методик навчання і додаванням більшого обсягу практичних проектів;

– систематична інтеграція прикладних та експериментальних досліджень з галузі інженерії програмного забезпечення у навчальний процес;

– зміцнення та посилення взаємодії освітніх закладів та професійних організацій для проведення практичних занять та досліджень;

– залучення студентів до міждисциплінарних досліджень та розробок;

– популяризація постійного та плідного самонавчання та вдосконалення власних знань студентів;

– постійне оновлення навчального плану.

Висновки. Підсумовуючи все вищесказане, можна зазначити, що програмна інженерія, зокрема курс «Інженерія програмного забезпечення», – це те, чому навчають студентів в університеті та продовжують навчати цьому ж на виробництві. Доречними з цього приводу є колективні роботи та наступність курсових робіт з переходом у дипломні роботи. Тому до перспектив подальшого дослідження віднесемо удосконалення групової методики при проведенні лабораторних робіт з дисципліни «Інженерія програмного забезпечення».

Список використаних джерел

1. Луцький М. Інформатизація процесів підготовки фахівців з програмного забезпечення / М. Луцький, М. Сидоров. – Київ: НАУ, 2010. – 57 с.

2. Морозова Т.Ю. Модель класифікації освітніх спеціальностей в IT-сфері / Т.Ю. Морозова // Інженерія програмного забезпечення. – Луганськ: Східноукраїнський університет ім. Володимира Даля. – 2010. – № 2. – С. 81–85.

3. Сейдаметова З.С. Факторы, влияющие на IT-образование: рынок труда, образовательные стандарты, языки программирования / З.С. Сейдаметова, В.А. Темненко Інженерія програмного забезпечення. – Київ: Національний авіаційний університет. – 2010. – № 1.– С. 62–71.

4. Сидоров М.О. Професійна практика програмної інженерії – досвід викладання / М.О. Сидоров, І.Б. Мендзебровський, О.А. Орехов Інженерія програмного забезпечення. – Київ: Національний авіаційний університет. – 2010. – № 2. – С. 56–65.

5. Bleek W.-G. Weaving Experiences from Software Engineering Training in Industry into Mass University Education / Wolf-Gideon Bleek, Carola Lilienthal, Axel Schmolitzky. – Department of Informatics: University of Hamburg Hamburg, Germany, 2004.

6. Kelley Ann E. A Software Engineering Curriculum Incorporating an Operational Formal Method / Ann E. Kelley. – Sobel Systems Analysis Department: Miami University Oxford, 2000.

7. Knowledge (SWEBOK), IEEE Computer Society 2004 Version [SWEBOK, 2004].

8. Raccoon L. A Leadership Primer for Software Engineers / L. Raccoon, 2012.

9. Shaw M. Software Engineering Education: a Roadmap. Institute for Software Research / Mary Shaw. – International Carnegie Mellon University Pittsburgh, 2000.

10. Software Engineering 2004. Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering. A Volume of the Computing Curricula Series IEEE Computer Society, Association for Computing Machinery, 2004 [SE, 2004].

References

1. Lutskyi, M. Sydorov, M. (2010). *Informatyzatsia protsesiv pidhotovky fakhivtsiv z prohramnoho zabezpechennia* [The computerization of the training software]. NAU, Kyiv. 57 p. (In Ukrainian).

2. Morozova, T.Yu (2010). *Model klasyfikatsii osvitnikh spetsialnostei v IT-sferi* [The model classification of educational professions in the IT field], *Inzheneriia prohramnoho zabezpechennia* [Software Engineering]. Vol 2, pp. 81-85. (In Ukrainian).

3. Sejdametova, Z.S., Temnenko, V.A. (2010). *Faktory, vly'yayushhy'e na IT-obrazovany'e: ry'nok truda, obrazovatel'ny'e standarty', yazy'ky' programmy'rovany'ya)* [Factors affecting the IT-education: the labor market, educational standards, programming languages]. *Inzheneriya programnogo zabezpechennya* [Software Engineering]. Kyiv National Aviation University. Eastern University. Vladimir Dal, Kyiv. Vol. 1, pp. 62–71. (In Russian).

4. Sydorov, M.O., Mendzebrovskyi, I.B., Orekhov, O.A. (2010). *Profesiina praktyka prohramnoi inzhenerii – dosvid vykladannia* [Professional practice of software engineering - teaching experience], *Inzheneriia prohramnoho zabezpechennia* [Software Engineering],

Natsionalnyi aviatsiinyi universytet [The National Aviation University]. Kyiv, vol. 2, pp. 56–65. (In Ukrainian).

5. Bleek, W.-G., Lilienthal, C., Schmolitzky, A. (2004). Weaving Experiences from Software Engineering Training in Industry into Mass University Education / Wolf-Gideon Bleek, Carola Lilienthal, Axel Schmolitzky. Department of Informatics: University of Hamburg Hamburg, Germany.

6. Kelley, Ann E. (2000). A Software Engineering Curriculum Incorporating an Operational Formal Method. Sobel Systems Analysis Department: Miami University. Oxford.

7. Knowledge (SWEBOK), IEEE Computer Society 2004 Version [SWEBOK, 2004].

8. Raccoon, L. A. (2012). Leadership Primer for Software Engineers.

9. Shaw, M. (2000). Software Engineering Education: a Roadmap. Institute for Software Research. International Carnegie Mellon University Pittsburgh.

10. Software Engineering (2004). Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering. A Volume of the Computing Curricula Series IEEE Computer Society, Association for Computing Machinery.

В статье описаны методические аспекты преподавания дисциплины «Инженерия программного обеспечения» для студентов, обучающихся по направлению «Системные науки и кибернетика». Даны рекомендации по созданию учебно-методического комплекса, приведены содержательное наполнение теоретического курса по указанной дисциплине и рекомендуемые формы проведения лекционных занятий. Учитывая то, что курс «Инженерия программного обеспечения» в силу его связи с практическими потребностями заказчиков имеет высокую динамичность, обусловленную спецификой предметной области, которая базируется на современных информационных системах и технологиях их построения, особое внимание уделено раскрытию методики проведения лабораторных занятий по курсу.

Ключевые слова: инженерия программного обеспечения, лабораторная работа, лекция, программная инженерия, программное обеспечение, стандарт, форма обучения, SWEBOK.

This article describes the methodological aspects of teaching students studying the discipline «System Sciences and Cybernetics». The recommendations for setting the educational and methodical complex for the discipline «Software Engineering» are presented. The semantic content of the theoretical course for a specified discipline and recommended forms of lectures are given. Taking into account the fact that the discipline and its relations to the practical needs of customers are highly dynamic due to the specifics of the subject based on modern information systems and technologies of their construction, special attention is paid to the description of methods of laboratory studies in the course.

Key words: software engineering, laboratory work, lectures, software engineering, software, standard, form of study, SWEBOK.

Одержано 9.10.2015.