

I. ПРОБЛЕМИ МЕТОДИКИ НАВЧАННЯ МАТЕМАТИЧНИХ ДИСЦИПЛІН

УДК 371.315

О.Ф. Баранюк

*Кіровоградський державний педагогічний університет
імені Володимира Винниченка*

ПРОБЛЕМНО-ОРІЄНТОВАНЕ НАВЧАННЯ У ПРОГРАМНІЙ ІНЖЕНЕРІЇ

Спостерігається певний розрив між рівнем підготовки фахівців з програмування і вимогами з боку роботодавців. Випускникам бракує навичок комунікації, командної роботи, досвіду реалізації реальних проєктів. Традиційна лекційно-лабораторна система викладання передбачає трансляцію знань, а не формування необхідних випускнику навичок. На цьому наголошують нормативні документи галузі комп'ютерних наук та програмної інженерії. Потрібні активні методи навчання, засновані на природній цікавості молоді до нових знань та досліджень.

Пропонується при вивченні курсу «Проектування програмних систем» застосовувати елементи проблемно-орієнтованого навчання. При цьому активну роль переходить від викладача до студента, який сам керує процесом навчання на основі власного досвіду і запропонованого викладачем набору проблем. Проблема виступає як рушійна сила процесу навчання.

***Ключові слова:** активне навчання, взаємне навчання, командна робота, методи навчання, проблемно-орієнтоване навчання, програмна інженерія, проектування програмних систем, самонавчання.*

Постановка проблеми. Розробка комп'ютерного програмного забезпечення є високотехнологічною і дуже динамічною сферою професійної діяльності. Хоча нинішні розробники програмного забезпечення провадять діяльність, схожу на своїх попередників, проте вони роблять це у більш складних умовах. Останнім часом спостерігається певний розрив між рівнем підготовки фахівців в галузі програмної інженерії і вимогами до кандидатів на посади з боку роботодавців. Інакше кажучи, випускникам вищих навчальних закладів бракує професіоналізму.

Опитування роботодавців, проведене Національною асоціацією коледжів і роботодавців NACE (National Association of Colleges and Employers) у 2013 році показало, що роботодавці найбільше очікують від кандидатів на посади навичок комунікації, здатності до командної роботи, навичок вирішення проблем, навички планування і організації, здатність знаходити і оброблювати інформацію, здатність аналізувати кількісну інформацію та ін. [10]. Діалог автора з місцевими представниками індустрії розробки програмного забезпечення підтверджує тезу про те, що випускникам бракує не стільки знань з програмування, скільки бачення кінцевої мети програмування і здатності довести програмний продукт до практичного використання, навичок командної роботи та комунікації (у тому числі англійською мовою), вміння формулювати та аргументувати свої думки.

Постає питання про те, щоб освітні програми вищої школи стали більше зорієнтовані на потреби і вимоги професійних розробників програмного забезпечення. У першу чергу, навчальний процес потрібно побудувати так, щоб студенти мали змогу здобути якомога більше практичних навичок розробки реального програмного забезпечення.

Керівництво з розробки навчальних планів в галузі комп'ютерних наук 2013 року (Computer Science Curricula 2013) визнає, що освіта повинна підготувати випускника до роботи більш цілісно, ніж просто озброїти технічними фактами. Визнається, що критичну роль на робочому місці відіграють навички міжособистісної взаємодії: командна робота, усна і письмова комунікація, тайм-менеджмент, вирішення проблем, гнучкість, а також особисті якості: толерантність до ризиків, колегіальність, терпимість, виробнича етика, почуття соціальної відповідальності, сприйняття різноманітності тощо [5].

Не випадково, навчальний план згідно [5] повинен включати питання керування проектами (командна робота, керування ризиками, толерантність до неоднозначностей), професійної комунікації (вербальна і невербальна), соціальної відповідальності, які об'єднуються поняттям професійного досвіду. До елементів професійного досвіду, які визначають здобутки студента і мають входити до будь-якого навчального плану, відносять [5]:

Професійні знання. Знання і навички з програмної інженерії, професійні стандарти, необхідні для роботи розробником програмного забезпечення.

Технічні знання. Розуміння і застосування теорій, моделей і методик, які дають базу для ідентифікації й аналізу проблем, проектування, розробки, впровадження, верифікації і документування програмного забезпечення.

Колективна робота. Вміння працювати як самостійно так і в команді у напрямі розробки і постачання якісних програмних артефактів.

Орієнтація на кінцевого споживача. Розуміння і визнання важливості перемовин, навичок ефективної роботи, лідерства, навичок комунікації із зацікавленими особами в типовому середовищі розробки програм.

Контекстні проектні рішення. Здатність проектувати відповідні рішення в різних предметних областях на основі підходів програмної інженерії, що інтегрують етичні, соціальні, правові та економічні аспекти.

Виробничі компроміси. Вміння узгоджувати конфліктні проектні цілі, пошук прийнятних компромісів за умов обмеження коштів, часу, знань, існуючих систем та організацій.

Постійний професійний розвиток. Вивчення нових моделей, методик і технологій в міру їх виникнення, усвідомлюючи необхідність постійного професійного розвитку.

Очевидно, що більшість вітчизняних вищих навчальних закладів мають за мету підготовку фахівців найвищої кваліфікації. Проте, традиційна лекційно-зорієнтована модель організації навчальних занять більшою мірою зосереджена на передачі знань від викладача до студента, ніж на розвитку професійних якостей майбутніх розробників програмного забезпечення. Лабораторні заняття і самостійна робота студентів покликані сприяти формуванню професійних практичних навичок, проте їх результативність значною мірою залежить від змістового наповнення навчальних завдань.

Аналіз останніх досліджень і публікацій. Програмування відноситься до складних дисциплін. Студенти, які вивчають програмування відчують суттєві труднощі в навчанні, особливо на початкових етапах, коли вони змінюють шкільне оточення на доросле університетське середовище. Гомес і Мендес [7] відзначають, що традиційні методи навчання не можна вважати адекватними щоб навчити ефективно програмувати. Для цього є, принаймні, кілька причин:

Навчання не персоналізоване. Було б добре, аби викладач був завжди доступний для студента і міг підтримати його індивідуальний темп навчання. Миттєвий зворотний зв'язок під час розв'язування задач і детальне пояснення незрозумілих моментів було б корисним для студента. Проте, насправді неможливо забезпечити таку підтримку з боку викладача через часові обмеження і рамки курсу.

Стратегія викладача не відповідає стилю навчання студентів. Люди навчаються різними способами і по різному сприймають новий матеріал. Хтось віддає перевагу навчанню наодинці, а комусь більше імпонує дискусія з колегами. Окремі теми вимагають особливого підходу, якщо не спрямувати студентів певним чином, вони будуть діяти у знайомий їм спосіб. За традиційної системи всі студенти змушені вчитися в єдиному темпі, який визначається стилем викладання.

Викладання динамічних концепцій через статичні матеріали. Програмування включає кілька динамічних концепцій (цикли, сортування, пошук тощо), які часто подаються через статичні образи (презентації, діаграми, схеми, текст). Для деяких студентів це становить проблему, було б легше сприймати такий матеріал через динамічні образи (анімацію).

Викладачі зосереджені більше на синтаксичних моментах, ніж на розв'язуванні реальних задач. Викладачі намагаються розвинути у студентів навички програмування, при цьому зосереджуються на особливостях синтаксису мови програмування, забуваючи, що мова має бути лише інструментом для вираження ідей і алгоритмів. Часто студентам дають надмірну кількість мовних деталей замість розвитку навичок алгоритмізації і розв'язування задач. Вибір мови програмування має бути підпорядкований скоріше педагогічним задачам, ніж популярністю серед професійних програмістів.

Ситуація ускладнюється самою природою програмування, яке вимагає високого рівня абстрактного мислення, аналітичних здібностей, здатності узагальнювати та критично мислити. Мови програмування мають досить складний синтаксис, вони розробляються переважно для професійного використання, а не для навчання. Надмірна кількість синтаксичних деталей, яку повинні запам'ятати студенти, заважає сконцентруватися на алгоритмічних особливостях конкретної задачі.

Курси з програмування поєднують теорію з практикою і потребують як передачі знань, так і розвитку навичок. Комп'ютерні науки вимагають скоріше активного навчання, ніж пасивного. Активне навчання відоме як один із семи принципів гарної практики у вищій освіті [4]. Аби активно навчатися, студенти повинні не лише слухати, а ще й читати, писати, дискутувати, залучатися до розв'язування проблем.

Активне навчання характеризується наступними моментами [3]:

- студенти залучаються до більшого, ніж просте слухання;
- більше уваги приділяється розвитку навичок, а не передачі інформації;

- залучення до високорівневого мислення (аналіз, синтез, оцінка);
- студенти включаються в діяльність (читання, дискусія, писання);
- наголос на дослідженні студентами власної позиції і цінностей.

Методів активного навчання досить багато, зокрема: активне аудіювання (Active Listening), активне писання (Active Writing), візуальне активне навчання (Visual-based Active Learning), мозковий штурм (Brainstorming), навчання через співпрацю (Collaborative Learning), взаємне навчання (Peer Teaching), рольові ігри (Role Playing), проблемно-орієнтоване навчання (Problem-Based Learning), навчальні ситуації (Case Studies), дискусія (Class Discussions) та ін. [8].

Одним із найбільш популярних методів навчання стало проблемно-орієнтоване навчання (ПОН). Елементи того, що сьогодні називають проблемним навчанням, можна знайти ще у Сократа, Галілея, Ж.-Ж. Руссо, А. Дістервега та інших відомих мислителів і педагогів [1]. Питанням ПОН приділяли увагу радянські вчені Т.В. Кудрявцев., І.Я. Лернер, М.І. Махмутов, О.М. Матюшкін, В.О. Онищук, польський педагог В. Оконь та багато інших дослідників.

В кінці 60-х років ХХ століття проблемно-орієнтоване навчання було започатковане у медичній школі канадського університету МакМастер для підготовки лікарів у маленьких групах на реальних проблемних ситуаціях. Досить швидко воно поширилося в інших медичних університетах, а згодом і в багатьох інших навчальних закладах.

Проблемно-орієнтоване навчання кардинально змінює навчальний процес і навчальні плани. Замість традиційної передачі знань від викладача до студента на перше місце ставиться проблема, яку пропонується розв'язати студентам. ПОН засноване на ідеях конструктивізму, коли студент не одержує готові знання, а сам творить (конструює) свої знання шляхом активної навчальної діяльності. Важливу роль у цьому процесі відіграє мотивація студентів до навчання, яка породжується природною цікавістю, змагальністю, честолюбством та іншими рисами характеру. Проблема виступає як рушійна сила процесу навчання. Роль викладача стає другорядною, він стає консультантом, помічником (facilitator), який скеровує навчальну діяльність студентів.

Модель проблемно-орієнтованого навчання включає шість основних характеристик [9]:

1. Навчання, орієнтоване на студента. Викладач формує проблемну ситуацію і допомагає студенту. Студент визначає, що він повинен вивчити, конструює свої знання і бере на себе відповідальність за свої знання.

2. Навчання відбувається в малих студентських групах. Студенти в групі розподіляють обов'язки, враховують можливості кожного, діють активно, співпрацюють, взаємно навчаються, допомагають один одному.

3. Викладач виконує роль помічника. Викладач не читає лекції, не виступає експертом, а допомагає студентам дослідити проблему, спрямувати хід думок, використати наявний досвід, знайти необхідний матеріал.

4. Проблема представляється на початку, без підготовки чи попереднього навчання. ПОН використовує реальні життєві ситуації, студентам подаються складні і багатопланові проблеми, які не мають очевидного вирішення (open-ended problems). Студенти самі вирішують, що вони знають, а що вони мають вивчити для вирішення проблеми.

5. Проблема використовується як інструмент для здобуття необхідних професійних знань і навичок. Студенти отримують навички командної роботи: пошуку консенсусу, прийняття рішень, ведення діалогу, підтримки дискусії, розв'язання конфліктів, лідерських навичок.

6. Нова інформація здобувається через самонавчання. Студентам потрібно визначити свої навчальні потреби і розробити стратегію і навички для задоволення цих потреб. Студенти менш залежні від викладача, вони знаходять інші джерела інформації, розвивають своє критичне мислення, викладач лише допомагає студентам.

Проблемно-орієнтоване навчання в чистому вигляді не позбавлене і недоліків, до яких можна віднести наступне: зміст курсу повинен бути зменшений у порівнянні з традиційним лекційним викладом, навчальний процес повинен бути забезпечений ретельно підібраними реальними проблемами, для ефективного застосування ПОН необхідно більше викладацького складу, іноді до навчання залучають навіть студентів-старшокурсників, потрібні додаткові аудиторії для розміщення невеликих груп, студенти гірше засвоюють декларативні знання і менш впевнено почувають себе на екзаменах [6].

Питанням використання проблемно-орієнтованого навчання проектуванню програмних систем приділяється недостатньо уваги і вони потребують подальшого вивчення і узагальнення.

Метою даної статі є обґрунтування доцільності та методика використання проблемно-орієнтованого навчання на заняттях з проектування програмних систем.

Виклад основного матеріалу дослідження. Викладання предметів, пов'язаних з програмуванням потребує нових підходів, методів навчання, удосконалення навчальних планів і організації навчального процесу. У першу чергу це викликано необхідністю підвищити рівень професійності придатності і компетентності майбутніх випускників-програмістів, про що свідчать контакти з представниками ІТ-бізнесу і колишніми випускниками.

Одним із способів розвитку і удосконалення навичок розробки програмного забезпечення у студентів може стати проблемно-орієнтоване навчання, зокрема при вивченні курсу «Проектування програмних систем». Навчальний процес у вищих навчальних закладах України на сучасному етапі залишається досить консервативним і не сприяє всебічному запровадженню нових технологій і методів навчання. Проблемно-орієнтоване навчання «в чистому вигляді» (насправді цей термін досить умовний) потребує кардинальної реконструкції навчального процесу, тому на даному етапі можна говорити лише про застосування елементів проблемно-орієнтованого навчання в умовах існуючої лекційно-лабораторної системи викладання.

Пропонується при викладанні курсу «Проектування програмних систем» лабораторний цикл побудувати як серію проблем в межах одного навчального проекту. Справа в тому, що основні етапи навчальної діяльності згідно концепції проблемно-орієнтованого навчання (аналіз проблеми, розподіл робіт, навчання та дослідження, вирішення проблем, аналіз результатів) дуже нагадують життєвий цикл розробки програмних систем. Причому, завдяки активній аудиторній і позааудиторній навчальній діяльності студентів в групах вдається змодельовати еволюційний процес розробки програмного забезпечення.

В межах запропонованого підходу перші два лабораторних заняття присвячуються організаційним питанням: основам керування проектами, організації студентських команд та створенню середовища для командної роботи над проектом на основі одного з численних онлайн-сервісів для керування проектами. Студенти реєструються в системі, знайомляться з середовищем, випробовують засоби планування, взаємодії та комунікації.

Подальший процес реалізації міні-проекту з розробки програмного забезпечення розбивається на ряд проблем згідно з тематикою лабораторних занять. Наступні заняття починаються із постановки проблеми, де викладач подає чергову так звану «погано структуровану» (ill-structured) або відкриту (open-ended) проблему. Це означає, що вирішення цієї проблеми неочевидне, в процесі розв'язання можливі різні варіанти розвитку подій та кінцевого результату. Саме так буває при реалізації реальних програмних проектів.

Студенти в командах обговорюють проблему, визначають рівень власних знань і готовності до розв'язання проблеми, виявляють брак знань у певних питаннях і складають плани персонального розвитку та плани чергового етапу проекту. На вирішення проблеми зазвичай відводиться два тижні, тому студенти мають достатньо часу для самостійної та колективної роботи.

Запропоновані студентам проблеми охоплюють основні етапи розробки програмних систем, а саме: виявлення та специфікація вимог до системи, моделювання предметної області, розробка проектної моделі системи, програмна реалізація, використання шаблонних класів та шаблонів проектування, тестування та передача (презентація) системи. Моделювання системи здійснюється за допомогою діаграм UML (випадків використання, класів, пакетів, послідовностей, станів та ін.).

На всіх етапах реалізації міні-проекту студенти отримують поради від викладача щодо пошуку інформації, використання літератури, вибору засобів та необхідного інструментарію для моделювання та програмування. При розробці вимог до системи викладач виконує функції віртуального замовника.

У процесі проблемно-орієнтованого навчання студенти навчаються оцінювати проблему та власний рівень підготовки, аналізувати проблему, шукати можливі варіанти розв'язання проблем, формулювати та висловлювати свої думки, переконувати колег, ділитися знаннями один з одним та колективно працювати над розв'язанням проблеми, планувати свою діяльність, координувати свої дії з членами команди, презентувати результати своєї роботи. Перший досвід автора у використанні даного підходу на заняттях з проектування програмних систем переконує в тому, що студенти здатні самостійно і досить ефективно здобувати знання і формувати вміння та навички, яких чекають від них на виробництві.

Висновки. Останнім часом доводиться чути від роботодавців у сфері інформаційних технологій нарікання на недостатній рівень підготовки випускників вищої школи. Причому відзначають не стільки брак знань, стільки відсутність навичок командної роботи, комунікації, досвіду реалізації реальних програмних проектів. Однією з причин є існуюча лекційно-лабораторна система організації навчальних занять. Потрібні методи навчання, які активізують навчальну діяльність студента, включають елементи

командної роботи та знайомлять із практичними прийомами розробки програмних продуктів. Таким вимогам відповідає, зокрема, проблемно-орієнтоване навчання.

На даному етапі пропонується запровадити елементи проблемно-орієнтованого навчання при виконанні циклу лабораторних занять з дисципліни «Проектування програмних систем». Проведені експерименти показують високу ефективність обраного підходу при виконанні навчальних міні-проектів і демонструють значний прогрес студентів завдяки активній творчій роботі над проектом. Подальші пошуки слід зосередити в напрямі розробки системи навчальних проблем, які могли б забезпечити високий ступінь покриття змісту навчальної дисципліни.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Педагогіка вищої школи : навч. посіб. / З.Н. Курлянд, Р.І. Хмелюк, А.В. Семенова та ін.; За ред. З.Н. Курлянд. – 3-тє вид., перероб. і доп. – К. : Знання, 2007. – 495 с.
2. A handbook for teaching and learning in higher education : enhancing academic practice / [edited by] Heather Fry, Steve Ketteridge, Stephanie Marshall. –3rd ed. – New York : Routledge, 2009. – 544 pp.
3. Bonwell, Ch. C., James A. E. Active Learning : Creating Excitement in the Classroom / ASHE; ERIC // Higher Education Report. 1991. – No. 1. – Washington D. C. : The George Washington University, 1991. – 118 pp.
4. Chickering A.W., Gamson Z.F. Seven Principles for Good Practice in Undergraduate Education / A.W. Chickering, Z.F. Gamson // AAHE Bulletin. – Washington : American Association for Higher Education, 1987. – Mar. – pp. 3–7.
5. Computer Science Curricula 2013 : Curriculum Guidelines for Undergraduate Degree Programs in Computer Science / The Joint Task Force on Computing Curricula Association for Computing Machinery. – Los Vaqueros : IEEE Computer Society, 2013. – 514 pp.
6. Davis B.G. Tools for Teaching / Barbara Gross Davis. – 2nd ed. – San Francisco : Jossey-Bass, 2009. – 592 pp.
7. Gomes A. Learning to Program – Difficulties and Solutions / A. Gomes, A.J. Mendes // International Conference on Engineering Education. – ICEE, 2007. –pp. 283–287.
8. Instruction at FSU : A Guide to Teaching and Learning Practices / Florida State University. – Florida : Florida State University, 2011. – 224 pp.
9. Richardson I., Delaney Y. Problem Based Learning in the Software Engineering Classroom / Ita Richardson, Yvonne Delaney // Software Engineering Education and Training. – Hyderabad : IEEE, 2009. – pp. 174–181.
10. Software Engineering 2014 : Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering / Joint Task Force on Computing Curricula Association for Computing Machinery. – Los Vaqueros : IEEE Computer Society, 2015. – 133 pp.

Baranyuk O.F.

Kirovohrad Volodymyr Vynnychenko State Pedagogical University
PROBLEM-BASED LEARNING IN SOFTWARE ENGINEERING

There is a gap between the level of training of graduates in field of programming and the requirements from employers. Graduates do not have enough communication skills, teamwork skills, and experience in implementing real projects. Traditional lecture-laboratory training system provides rather knowledge translation from lecturer to students than the formation of the skills necessary to graduates. Curriculum guidelines for undergraduate degree programs in computer science and software engineering also pay attention to this problem. It's required active learning methods based on the natural interest of young people in new knowledge and research.

Active learning is based on ideas of constructivism, when student doesn't get ready knowledge, but creates (constructs) knowledge through active learning activities and is responsible for own learning. There are some active learning techniques including problem-based

learning as well-known and quite popular technique. PBL is an instructional method in which ill-structured (open-ended) problems are introduced at the beginning of each class. Students need to analyze problem, to get needed knowledge, to find appropriate solution and evaluate whether it meet the problem specification. Thus, problem is the driving force of the learning process.

It is proposed to use elements of problem-based learning during learning of the course «Design of Software Systems». The main steps of problem-based learning are quite similar to appropriate steps of software development life cycle (problem analysis, works distribution, problem solving, results analysis etc.). It's possible to model evolutionary software development process due to active class and out-of-class learning activity of students in their groups. Here active role is transferred from teacher to student, which controls the learning process on the basis of their own experience and problems proposed by the teacher.

With this approach laboratory session is organized as a series of problems solving within general mini-project proposed to the students. The students work in small groups, they get ill-structured (open-ended) problem according to teaching plan. Each problem is solved during approximately two weeks. First experience in PBL shows that students can demonstrate more creativity, responsibility, and success using problem-based learning.

Key words: active learning, cooperative learning, group work, teaching methods, problem-based learning, software engineering, software systems design, self-learning.

Баранюк А. Ф.

Кировоградский государственный педагогический университет
имени Владимира Винниченко

ПРОБЛЕМНО-ОРИЕНТИРОВАННОЕ ОБУЧЕНИЕ В ПРОГРАММНОЙ ИНЖЕНЕРИИ

Наблюдается определенный разрыв между уровнем подготовки специалистов по программированию и требованиями со стороны работодателей. Выпускникам не хватает навыков коммуникации, командной работы, опыта реализации реальных проектов. Традиционная лекционно-лабораторная система организации занятий предусматривает скорее трансляцию знаний, чем формирование необходимых выпускнику навыков. На это обращают внимание нормативные документы в области компьютерных наук и программной инженерии. Нужны активные методы обучения, основанные на природном интересе молодежи к новым знаниям и исследованию.

Предлагается при изучении курса «Проектирование программных систем» применять элементы проблемно-ориентированного обучения. При этом активная роль переходит от преподавателя к студенту, который сам управляет процессом обучения на основе собственного опыта и предложенного преподавателем набора проблем. Проблема выступает как движущая сила процесса обучения.

Ключевые слова: активное обучение, взаимное обучение, командная работа, методы обучения, проблемно-ориентированное обучение, программная инженерия, проектирование программных систем, самообучение.

ВІДОМОСТІ ПРО АВТОРА

Баранюк Олександр Філімонович – доцент кафедри інформатики Кіровоградського державного педагогічного університету імені Володимира Винниченка, кандидат технічних наук.

Коло наукових інтересів: моделювання інформаційних систем, проблеми викладання комп'ютерних наук у вищій школі.