

ФОРМУВАННЯ ПРОФЕСІЙНИХ КОМПЕТЕНЦІЙ СУЧАСНОГО ІНЖЕНЕРА-ПРОГРАМІСТА

У статті запропонована ідея інтегрованого навчання технічним, когнітивним і соціальним навикам, необхідних сучасному інженеру-програмісту при використанні передових методів розробки програмного забезпечення на практиці. Таке інтегроване навчання може бути організовано на основі методів активного навчання. Описано чотири техніки навчання – ролева гра, кластер, мозковий штурм і дебати.

The idea of the integrated teaching to technical, cognitive and social skills necessary to the modern engineer-programmer at the use of front-rank methods of software development in practice is offered in the article. The such integrated teaching can be organized on the basis of methods of the active teaching. Four techniques of teaching are described is role game, cluster, cerebral assault and debates.

Компетенції – це набір чинників, що включає ділові і особистісні якості, знання, вміння і навички, необхідні фахівцям для успішного виконання своєї роботи, тобто показники, за якими оцінюється ефективність роботи співробітника. Професійна компетенція розглядається як системна, інтеграційна єдність, синтез інтелектуальних і навикових складових (загальні знання, вміння і навички), особистісних характеристик (ціннісні орієнтації, здібності, риси вдачі, готовність до здійснення своєї діяльності) і досвіду, що дозволяє використовувати свій потенціал, здійснювати складні види діяльності, оперативно і успішно адаптуватися в суспільстві і професійній діяльності. Такими компетенціями є:

- 1) готовність і здатність спонукати інших людей працювати спільно заради досягнення поставленої мети;
- 2) готовність і здатність робити що-небудь ефективніше, ніж раніше;
- 3) готовність і здатність вирішувати складні питання;
- 4) готовність і здатність виступати зі своїми новими ідеями;
- 5) готовність і здатність посідати керівні посади;
- 6) готовність і здатність вчитися все життя як основа безперервної підготовки в професійному плані;
- 7) готовність і здатність навчатися самостійно;
- 8) готовність і здатність орієнтуватися в інформаційному полі.

Розробка і визначення складу професійних компетенцій бере свій початок в моделі фахівця, при цьому проводиться виявлення рівнів формування професійних компетенцій як мети освітнього процесу (рис. 1).

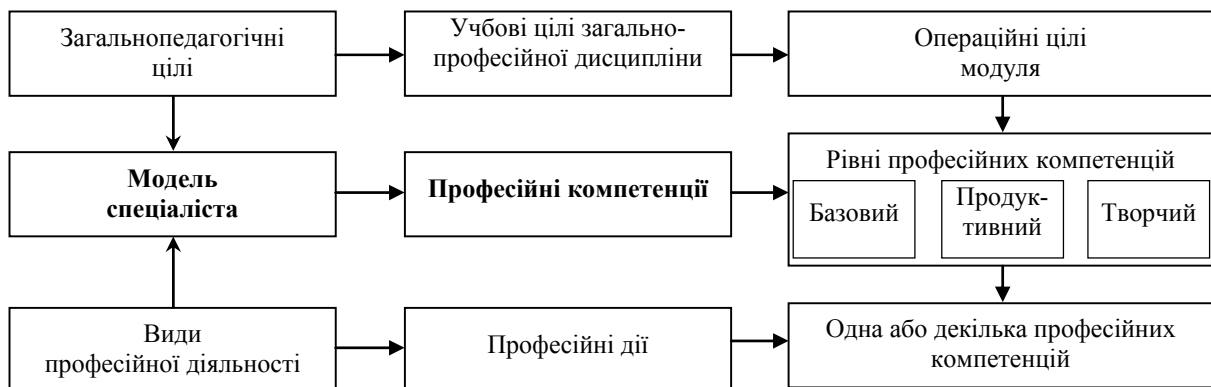


Рис. 1. Виявлення рівнів формування професійних компетенцій

Суспільство на всіх етапах свого історичного розвитку пред'являло високі вимоги до будь-яких фахівців, не є виключенням і наша сучасність. Нині підготовка фахівця для будь-якої сфери діяльності повинна забезпечувати високий рівень його компетентності, мобільності, максимально сприятливих умов для розвитку особистості. Наявність компетенцій у фахівця набуває все більшого значення у зв'язку з ускладненням і розширенням соціального досвіду, виникненням нових різноманітних форм пред'явлення і переробки інформації, зі все зростаючим рівнем тих запитів, які пред'являє до фахівця суспільство.

Розвиток сучасної науки і техніки зв'язаний з використанням інформаційних технологій в різних галузях. Розробка програмного забезпечення (ПЗ) в цьому сенсі є найнеоднозначнішою областю в сучасній галузі інформаційних технологій. Щоб забезпечити стабільний розвиток галузі виробництва ПЗ на перспективу і відповідність її можливостей потребам сучасного суспільства, необхідно готувати фахівців, які володіють необхідними знаннями і навиками для вирішення задач такого роду.

Проте деяка інертність, традиційно властива вищій освіті, в сукупності зі швидким розвитком

технологій і мов програмування, засобів розробки та різних методологій нині призводить до певних спотворень в професійній підготовці інженерів-програмістів [1]. Підготовка типового інженера-програміста, здійснювана в рамках вищого навчального закладу, як правило, включає вивчення низки базових дисциплін (вища математика, фізика, електротехніка, основи алгоритмізації та програмування, структури і алгоритми обробки даних тощо) в поєднанні з деяким набором актуальних на даний момент технологій, часто на прикладі конкретних мов і засобів розробки (об'єктно-орієнтоване програмування, веб-програмування, візуальне програмування тощо). Таким чином, припускається, що після закінчення університетського курсу випускник матиме деякі фундаментальні знання, що дозволять йому надалі вдосконалювати свою професійну підготовку, а також деякі практичні навички у вигляді знання конкретних технологій і мов програмування, які дозволять йому відразу після закінчення вузу здійснювати професійну діяльність.

Важливо відзначити, що суть проблеми полягає не в якості університетської освіти як такої, а, скоріше, в розриві між традиційними програмами навчання фахівців і вимогами галузі, що пред'являються на практиці. Проект SWEBOK (Software Engineering Body Knowledge) [2], ініційований Університетом Квебека (Монреаль, Канада) і координований такими авторитетними міжнародними організаціями, як ACM (Association for Computing Machinery) і IEEE (Institute Electrical and Electronics Engineers), на основі аналізу накопиченого досвіду програмної індустрії дозволив виділити ядро знань програмної інженерії, тобто знання і навички, що залишаються актуальними протягом останніх 20 – 30 років. SWEBOK визначає десять областей знань, що дозволяють отримати знання і навички професійного інженера з розробки ПЗ:

- 1) Software Requirements (вимоги до ПЗ);
- 2) Software Design (проектування ПЗ);
- 3) Software Construction (конструювання ПЗ);
- 4) Software Testing (тестування ПЗ);
- 5) Software Maintenance (супроводження ПЗ);
- 6) Software Configuration Management (управління конфігурацією);
- 7) Software Engineering Management (управління в програмній інженерії);
- 8) Software Engineering Process (процес програмної інженерії);
- 9) Software Engineering Tools and Methods (методи та інструменти);
- 10) Software Quality (якість ПЗ).

Враховуючи масштаби досліджень, проведених в рамках проекту SWEBOK, можна сміливо вважати області знань, рекомендовані ним, необхідним освітнім мінімумом для інженера, що професійно займається розробкою якісного ПЗ на практиці.

Проте практика показує, що є види людської діяльності, включені в розробку ПЗ, які не враховуються в SWEBOK, але які виконують ключову роль в успішності програмних проектів. Рішення, що приймаються всіма розробниками на різних етапах, і здатність членів проектної команди ефективно обмінюватися інформацією як всередині команди, так і з іншими зацікавленими особами, відіграють в успішності проекту більшу роль, ніж техніка розробки, що використовується в ньому. Навички ухвалення рішень в умовах неповноти інформації і уміння ефективно спілкуватися важливі не лише для людей, що виконують ключові ролі в проекті (керівник проекту, бізнес-аналітик, архітектор), але і для звичайних розробників. Багато разів в ході проекту їм доводиться приймати міні-рішення про те, чи реалізувати якусь функціональність у вигляді одного компоненту або двох, чи використовувати той або інший алгоритм, і, звичайно ж, їм дуже часто доводиться спілкуватися між собою, з архітектором, тестувальниками і керівником проекту, а інколи і з користувачами. Сфери діяльності, в яких повинен розбиратися розробник ПЗ для створення програм високої якості, представлені на рис. 2 [3].

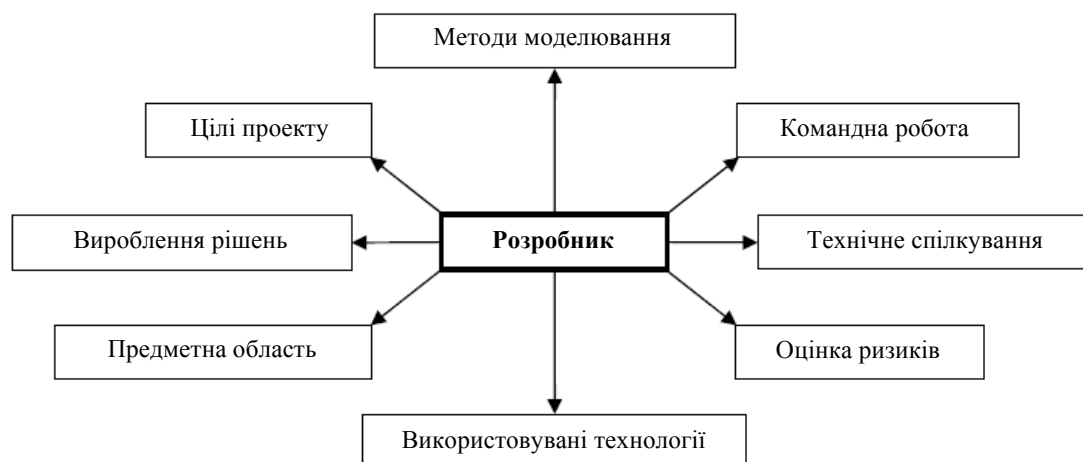


Рис. 2. Сфери компетенції сучасного інженера-програміста

Навички, необхідні будь-якому розробнику ПЗ, можуть бути (не надто строго) розділені на три види.

Технічні навички, які найчастіше розглядаються як єдино необхідні. Вони характеризують знання

розробником основних методів і техніки розробки програм, а також уміння використовувати їх на практиці в контексті вирішуваних задач.

Когнітивні навички, пов'язані з розумінням вимог і обмежень в рамках нової предметної області, умінням відділяти суттєве від несуттєвого для даної задачі, впорядкуванням задач за їх пріоритетністю, побудовою рішень на основі неповної інформації про вимоги і вибором з них найбільш придатного.

Соціальні навички полягають в умінні підтримувати ефективне спілкування з іншими людьми, розуміти цілі проекту, умінні працювати в команді і координувати свою роботу з потребами її членів. Соціальні навички також визначають стиль спілкування людини – те, як вона слухає інших людей, реагує на почуте, бере участь в обговореннях і висуває аргументи на користь власної точки зору.

Увага до соціальних навичок була вперше привернута в 50 – 60 рр. в США усвідомленням того факту, що більшість людей живе і працює в групах, але часто не задумується над тим, як вони в них беруть участь, якими їх бачать інші люди, які реакції викликає у оточуючих їх поведінка. Однією з перших робіт, в яких ця ідея розвинулась як додаток до розробки ПЗ, є робота [4].

Проблеми оволодіння такими навичками досить актуальні в ширшому контексті. Для ефективного навчання є надзвичайно важливими такі соціальні уміння, як уміння діяти впевнено, уміння спрямовувати власні дії на досягнення потрібних результатів і уміння ухвалювати ризиковані рішення, а також таких когнітивних навичок, як здатність вчитися, пристосовуватися до оточення, швидко сприймати необхідні нові навички і знаходити потрібну інформацію в широкому інформаційному потоці. Це особливо актуально в тих випадках, коли в програмному проєкті використовуються нові підходи до розробки ПЗ, наприклад, формальні методи [5]. Це викликано значно складнішими пізнавальними проблемами в рамках таких проєктів – потрібно не лише вникнути в предметну область і зрозуміти задачі, які потрібно вирішити, але і викласти отримані знання в новій, незвичній формі, необхідній для ефективного використання формальних підходів. Крім того, саме розуміння задач повинне бути набагато глибшим і детальнішим, інакше формальні методи стають набагато важчими у використанні, ніж традиційні.

Таким чином, успішність створення ПЗ пов'язана не лише з технічними знаннями учасників проєктної команди, але і з їх умінням адаптувати свої особисті якості і навички до контексту проєкту і поєднувати їх з інтересами колективу. Тому для забезпечення успіху при використанні формальних методів на практиці потрібно навчати персонал не лише їх методологічним основам, але і прищеплювати йому необхідні когнітивні і соціальні навички.

У роботі [5] стверджується, що активні методи навчання є набагато ефективнішим інструментом навчання передовим методологіям розробки ПЗ, ніж традиційні підходи до навчання. Активні методи ефективно виробляють у студентів необхідні когнітивні та соціальні навички за рахунок акценту на автономну роботу, самостійний вибір напрямку дій, самостійне ухвалення рішень, необхідність спілкування з іншими студентами, необхідність підлаштовуватись під зміни в рамках освітнього процесу. В той же час класичні підходи до навчання підштовхують студентів до пасивного сприйняття інформації і перетворюють їх на добре інформованих людей, нездатних без зовнішнього керівництва використовувати свої величезні знання на практиці.

Активні методи навчання – це методи, спрямовані не на викладення готових знань і запам'ятовування їх студентами, а на спонукання студентів до активної розумової і практичної діяльності в процесі оволодіння навчальним матеріалом. Можна виділити наступні основні ознаки активних методів навчання:

- 1) примусова активізація мислення, що змушує студента бути активним незалежно від його бажання;
- 2) залучення студентів у навчальний процес протягом досить тривалого часу, оскільки їх активність повинна бути не короткочасною і епізодичною, а стійкою і тривалою;
- 3) самостійне творче вироблення рішень і підвищений ступінь мотивації.

Наведемо декілька прикладів таких методів, які підштовхують студентів до активного мислення і використання одержаних знань в практичній діяльності:

- 1) ролева гра;
- 2) кластер;
- 3) мозковий штурм;
- 4) дебати (дискусія).

Ролева гра націлена на вироблення навичок самостійного вирішення проблем, хоча в навчанні технічним і інженерним предметам вона використовується не дуже часто. Ролева гра – це штучно сконструйована послідовність ситуацій, яка імітує ті чи інші важливі сторони реальної діяльності. Під час гри її учасники виконують певні ролі і поведуться, виходячи з відповідних точок зору на певну ситуацію. Мета такої гри полягає в тому, щоб знайти можливі вирішення проблем, які виникають в даних ситуаціях.

У роботі [3] наводиться приклад ролевої гри "Генерація тестової послідовності в результаті обходу графової моделі тестованої системи як невідомої місцевості" (табл. 1).

Ролеві ігри дозволяють підвищити інтерес до матеріалу, що вивчається, виявити прогалини в знаннях і розвинути соціальні навички. Проте, ролева гра ведеться за жорсткими правилами і не може відобразити різноманіття ситуацій і динамізм, які властиві реальному життю. Дуже важко скласти набір ролевих ігор, які дозволять відпрацювати більшість важливих реальних ситуацій в даній області.

Ефективність ролевих ігор можна підвищити за рахунок використання методу "розвиваючого дискомфорту". Дієвість цього методу заснована на психологічній перебудові особистості під впливом стихійно

виникаючих або спеціально організованих емоційно негативних ситуацій. В результаті такої перебудови міняється характер активності, і людина набуває уміння, з одного боку, успішно виходити з критичної ситуації, а з іншого – позитивно відображати ці ситуації в своєму соціальному досвіді. Інакше кажучи, відповідно до цього методу треба не лише не боятися складних, негативних ситуацій, але й використовувати їх для поступального розвитку у студентів навиків ухвалення рішень і впевненості у власних силах.

Таблиця 1

Ролева гра з метою оцінки цінності запропонованої наукової ідеї

Пропагандист	Теоретик	Технічний експерт	Менеджер	Інвестор
Графові моделі складних систем можуть бути представлені дуже компактно.	Явні графові моделі дають можливість краще контролювати хід тестування.	Ідея актуальна як досить надійний метод контролю якості.	Ідея обіцяє деяке підвищення якості ПЗ.	Ідея може бути реалізована у вигляді продукту, лише якщо продукт буде націлений на використання при розробці ПЗ з підвищеними вимогами до надійності.
Досить складні ситуації можуть бути протестовані автоматично.	Може використовуватися як спосіб побудови адаптивних тестових послідовностей.	Неявні графові моделі важче описати, особливо тим, хто раніше з ними не мав справи.	Немає ясного зв'язку між тестуванням і вимогами – навіщо нам тестувати всі ці ситуації?	Потрібно запропонувати менш трудомісткий спосіб опису графів.
Якість такого тестування дуже висока.		Цей підхід слід зробити зручнішим для використання.	Щоб навчитися використовувати цей метод, тестувальникам необхідний спеціальний тренінг.	Ідея може використовуватися для створення модулів тестування, які підключаються до середовищ розробки.
		Необхідний інструмент з графічним інтерфейсом користувача, який допомагатиме будувати моделі такого роду.		

Методи активного навчання передбачають не лише створення нової методики навчання і зміну критеріїв оцінки підготовки студентів, але, перш за все, усвідомлення викладачем своєї нової ролі, в якій головним є допомога студентам у самостійному оволодінні предметом, а не його викладання. Активні методи навчання змушують студентів вести активну роботу зі знанням, замислюючись над ним, аналізуючи його і порівнюючи із знаннями, що вже у них є. При активному навчанні не повинен підтримуватися підхід до навчання деяких студентів, коли вони говорять: “А ви скажіть як потрібно, ми так і зробимо”.

Деякі способи зміни відношення до навчального матеріалу засновані на особистому прийнятті знань, в результаті якого одержане знання стає не переказом якого-небудь підручника, а власним знанням студента. Для прив'язки знань до особистого досвіду можна використовувати **кластер** [3]. Кластер створює умови для побудови асоціативних зв'язків між новим матеріалом і наявним досвідом, причому ці зв'язки засновані не лише на подібності, але і на емоційному сприйнятті, інтуїції, а також на основі припущень (можливо, помилкових).

Перед вивченням нової теми кожний студент малює кластер на цю тему. Послідовність дій при малюванні кластера наступна:

- 1) посередині чистого аркуша записується ключове слово або фраза, ключова для даної теми;
- 2) навколо нього без якого-небудь плану або впорядкованості записуються слова або фрази, що виражають ідеї, факти, образи, які пов'язані з даною темою (або здаються пов'язаними);
- 3) по мірі запису слова, що з'явилися, з'єднуються зв'язками з ключовим поняттям (таке з'єднання може бути ієрархічним); при цьому треба дотримуватись наступних правил:
 - 3.1) не потрібно боятися записувати все, що спадає на думку, треба дати волю уяві і інтуїції;
 - 3.2) продовжувати роботу, поки не закінчиться час або не вичерпаються ідеї;
 - 3.3) прагнути побудувати якомога більше зв'язків.

На рис. 3 наведений приклад кластера поняття “Формальні специфікації”.

Перед початком малювання кластера викладач повинен чітко сказати, скільки часу відводиться на цю роботу. Коли кластер у кожного намальований, потрібно попросити одного студента (лише за бажанням!) намальовати свій кластер на дошці. Якщо охочих малювати нема, то викладач малює свій власний кластер. Обговорювати нічого не потрібно. Кожний самостійно порівнює свій кластер з тим, який намальований на дошці і при бажанні може внести зміни в свій кластер.

Дуже корисно організувати при нагоді обговорення кластера в малих групах або в парах. Результатом такого обговорення мають стати нез'ясовані питання, відповіді на які повинні бути одержані на наступних заняттях. Крім того, таке обговорення дозволяє підвищити комунікабельність студентів.



Рис. 3. Приклад кластера поняття “Формальні специфікації”

Для студентів старших курсів, що починають дослідницьку діяльність, дуже важливо мати деякий набір ідей як відправну точку досліджень. Для отримання стартового набору ідей може використовуватися **мозковий штурм**. Мозковий штурм організовується у декілька етапів.

Етап 1. Учасники штурму мають бути ознайомлені з темою досліджень студента. Ця частина може бути організована наступним чином.

- 1) студент та його керівник готують конспект з теми, проблем досліджуваної області і методів, що використовуються в ній;
- 2) кожний учасник штурму читає підготовлений конспект;
- 3) учасники формулюють питання, що стосуються проблем в даній області (питання мають бути записані);
- 4) студент робить доповідь на задану тему, в якій повинні бути сформульовані його інтереси в досліджуваній області і дані відповіді на питання учасників штурму;
- 5) після доповіді питання, на які були отримані відповіді (те, що відповідь на питання одержана, повинен підтвердити учасник, який поставив це питання), відмічаються знаком “+”.

Етап 2. Учасники пропонують ідеї можливих напрямів подальшого дослідження. При цьому повинні дотримуватися наступні правила.

- 1) кожний учасник має право на виступ протягом 30 с; під час такого виступу повинно бути сформульовано не більше однієї ідеї;
- 2) кількість виступів одного учасника необмежена;
- 3) виступ не можна переривати;
- 4) кожний учасник має право нічого не говорити;
- 5) ідея виступу повинна бути сформульована по можливості чітко (бажано одним реченням);
- 6) під час виступу не можна обговорювати, критикувати і оцінювати попередні виступи;
- 7) порядок і коректність виступів забезпечує ведучий штурму;
- 8) всі запропоновані ідеї записуються ведучим.

Етап 3. Учасники оцінюють усі запропоновані ідеї. Цей процес може бути організований таким чином.

Для кожної запропонованої ідеї і кожного учасника готуються аркуші паперу, що містять назву ідеї і три колонки для записів. В першу колонку вписуються аргументи на підтримку ідеї, в другу – проти неї, в третю колонку записуються цікаві аспекти, що стосуються даної ідеї. В іншій схемі разом з плюсами і мінусами формулюються можливі наслідки використання ідеї на практиці. Кожний учасник отримує аркуші зі всіма ідеями і заповнює їх. Необов'язково оцінювати всі ідеї. Всі листи збираються для складання інтегральних оцінок.

Етап 4. Студент і його науковий керівник підводять підсумки розгляду всіх ідей і визначають найперспективніші ідеї.

Одним з важливих навиків для розробки ПЗ є уміння вислухати думку, відмінну від власної, зрозуміти, на чому вона ґрунтується, і взяти це до уваги в подальшій роботі. Дебати дозволяють відпрацювати даний

навик. На початку визначається спірне твердження, яке обговорюватиметься. Кожний учасник окремо (або кожна група учасників) висуває аргументи, що підтримують або спростовують дане твердження. Кожний аргумент може бути, в свою чергу, підтриманий іншими твердженнями. Метою дебатів є надання кожному учаснику матеріалу для формування власної точки зору на обговорювану тему. Іншою метою є тренування навиків у відстоюванні своєї точки зору і розумінні інших, відмінних від неї. Студенти знають, що їм можуть заперечити, вони психологічно готові до цього, вони тренуються концентруватися на суттєвих деталях, аналізувати і оцінювати різні точки зору на предмет обговорення і відноситися з повагою до протилежних і, можливо, непопулярних точок зору.

Таким чином, у статті запропонована ідея інтегрованого навчання технічним, когнітивним і соціальним навикам, які є необхідними при використанні передових методів розробки ПЗ на практиці. Таке інтегроване навчання може бути організоване на основі методів активного навчання.

Список використаних джерел

1. Макконнелл С. Профессиональная разработка программного обеспечения / Макконнелл С. ; [пер. с англ.]. – СПб. : Символ-Плюс, 2006. – 240 с.
2. Guide to the Software Engineering Body of Knowledge (SWEBOOK) [Electronic resource] // Portal “SWEBOOK. IEEE Computer Society”. – URL : <http://www2.computer.org/portal/web/swebok> (date of appeal: 21.09.2009).
3. Кулямин В. В. Формирование профессиональных компетенций современного разработчика ПЗ [Электронный ресурс] / В. В. Кулямин, В. А. Омельченко, О. Л. Петренко. – Режим доступа : <http://panda.ispras.ru/~kuliamin/docs/Learning-2006-ru.pdf> (дата обращения: 21.09.2009).
4. Брукс Ф. Мифический человеко-месяц или как создаются программные системы / Брукс Ф. ; [пер. с англ.]. – СПб. : Символ-Плюс, 2001. – 304 с.
5. Демарко Т. Человеческий фактор: успешные проекты и команды / Т. Демарко, Т. Листер ; [пер. с англ.]. – 2-е изд. – СПб. : Символ-Плюс, 2005. – 256 с.