

УДК 378.1:004:303.732.4(477)

Н.М. Сидорова

Навчання інженерії програмного забезпечення – систематичний огляд літератури

Національний авіаційний
університет

Кафедра педагогіки та
психології професійної
освіти

Науковий керівник –
Лузік Е.В., д.п.н.,
професор

Розробка програмного забезпечення є в Україні відносно новою індустрією, що постійно розвивається. Отже, виникає необхідність у спеціалістах різного рівня кваліфікації. Все більш актуальним стає завдання з розробки комплексного системно-особистісного підходу до формування професійної компетентності фахівців з інженерії програмного забезпечення. Навчання діяльності у сфері програмної інженерії має бути інноваційним, оскільки неможливо при традиційному навчанні забезпечити за період підготовки спеціаліста у вищому навчальному закладі максимальний розвиток спеціальних здібностей у вирішенні проблем у цій сфері діяльності.

Разработка программного обеспечения является в Украине относительно молодой индустрией, которая постоянно развивается. Соответственно, возникает потребность в специалистах разного уровня квалификации. Все более актуальной становится задача по разработке комплексного системно-личностного подхода к формированию профессиональной компетентности специалистов по инженерии программного обеспечения. Преподавание инженерии программного обеспечения должно быть инновационным, поскольку при традиционных методах обучения за период подготовки специалиста в вузе невозможно обеспечить максимальный уровень развития специальных способностей по решению проблем в данной сфере.

Software engineering in Ukraine is a relatively young industry that is steadily developing. That is why the need in specialists of different competence level has arisen. It is essential to develop a complex systematic approach to the formation of professional competence of specialists in software engineering. Education in this area shall be innovative as far as it is impossible within the period of studying in the university to develop essential skills for the successful operation in the area of software engineering using traditional methods of education.

Ключові слова: викладання інженерії програмного забезпечення, стандарти ІТ-професій, кваліфікація, професійна компетенція

Вступ

Розробка програмного забезпечення в Україні є індустрією, що постійно розвивається. Отже, виникає необхідність у фахівцях різних спеціальностей. У 2006 році в Україні було засновано новий напрям навчання з підготовки інженерів програмного забезпечення «Програма інженерія». Розроблено відповідні стандарти та розпочато викладання дисциплін. Чотири роки існування напряму показали, що є певні проблеми, які пов'язані з формуванням професійної компетентності бакалаврів з інженерії програмного забезпечення. Наприклад, актуальним стає завдання розробки інтегрованого підходу до формування готовності майбутніх

бакалаврів до професійної комунікації з інженерії програмного забезпечення.

Систематичний огляд літератури – це один з методів дослідження, що останнім часом набуває все більшого поширення у сфері дослідження інженерії програмного забезпечення [1, 5]. *Systematic mapping study* - це метод, що спочатку був дуже поширеним у сфері медичних досліджень, однак, на жаль, не використовується значною мірою при проведенні досліджень в інженерії програмного забезпечення. На думку вченого Бейлі існує лише один чіткий приклад *systematic mapping study* в межах інженерії програмного забезпечення [1, 1]. *Systematic mapping study* являє собою структурний тип дослідження надрукованих публікацій та наукових робіт,

а також належним чином їх класифікувати. Таким чином, можна побудувати свого роду візуальний підсумок-огляд, так звану карту результатів. Використання методу *systematic mapping study* можна провести набагато точніше та детальніше дослідження за коротший час.

Systematic mapping study – це більш «розгорнута» форма систематичного огляду літератури спрямована на повне планове дослідження, не на пошук конкретних відповідей на детальні питання та завдання початкового дослідження [1, 7]. Таке дослідження спрямоване на заповнення «огріх» або свого роду «нестач» у ряді початкових досліджень, де необхідним є проведення новітньої науково-дослідної роботи, а також на пошук спірних питань, що потребують додаткових уточнювальних досліджень [1, 3]. А *systematic mapping study* дозволяє проводити дуже точні та детальні дослідження у необхідній галузі [1, 7], [1, 1].

Переваги використання методу *Systematic mapping study*

Існує ціла низка приводів для використання у дослідженні саме *systematic mapping study*, найбільш очевидними з яких є наступні:

- Підсумувати існуючі результати досліджень, а також емпіричний досвід у галузі, означити переваги та недоліки певних методів.
- Визначити недоліки та «нестачі» у досліджуваній галузі, підготувати базу для проведення майбутніх науково-дослідних робіт.
- Побудувати структуру та порядок проведення нових досліджень [1, 7].

Ознаки *systematic mapping studies*

Деякими ознаками, що відрізняють традиційний систематичний огляд літератури від *systematic review* є наступні:

- *Systematic mapping study* традиційно починається з побудови оглядового протоколу, що визначає завдання та питання дослідження, виходячи з яких підбираються відповідні методики проведення науково-дослідної роботи.
- *Systematic mapping study* засновується на певній визначеній стратегії, що спрямована на пошук якомога більшої кількості значимої літератури.
- *Systematic mapping study* надає читачам повний та цілісний опис всього процесу дослідження (слід враховувати те, що процес пошуку літератури в електронних бібліотеках майже неможливо відтворити).
- *Systematic mapping study* потребують використання повних і чітких якісних критеріїв включення і виключення початково знайдених наукових робіт.
- *Systematic mapping study* конкретизують

інформацію, отриману в результаті кожного первинного дослідження, у тому числі якісні критерії, за якими оцінюється кожне первинне дослідження.

- *Systematic mapping study* - передумова для кількісного мета-аналізу [1, 6].

Форма *systematic mapping study*

Вчені Petticrew та Roberts пропонують використання даного методу систематичного аналізу літератури з метою з'ясування, “які саме науково-дослідницькі роботи відповідають критеріям пошуку, коли вони були надруковані, в яких базах даних (бібліотеках) зберігаються, яких результатів було досягнуто, а також дослідники яких країн займаються дослідженням поставлених питань” [1, 10].

Перші етапи *systematic mapping study* є дуже схожими з етапами традиційних систематичних оглядів літератури:

1. Пошук. Визначення низки науково-дослідницьких робіт, в яких можуть розкриватися поставлені питання/ завдання дослідження.

2. Включення/виключення науково-дослідницьких робіт із дослідницького процесу. Відбір літератури для проведення подальшого дослідження.

3. Відповідність питанням/ завданням дослідження. Якщо необхідно, проведення якісного оцінювання віднайдених науково-дослідницьких робіт [1, 3].

Визначення питань / завдань дослідження

Основною метою *systematic mapping studies* є надання широкого та повного огляду рівня дослідженості заданої сфери, а також визначення кількості та якості надрукованих науково-дослідницьких робіт із даної теми. Вторинною метою є надання чіткого графічного звіту про проведений *systematic mapping studies* [1, 5]. У нашому дослідженні ми визначили три дослідницькі питання (ДП) для проведення систематичного огляду науково-дослідницьких робіт у сфері освіти інженерії програмного забезпечення:

(ДП1) Якими є головні характеристики існуючої системи освіти інженерії програмного забезпечення?

(ДП2) Якими є основні дидактичні та педагогічні питання освіти інженерії програмного забезпечення? Наскільки дослідженими є питання професійної компетентності, акредитації та ліцензування в інженерії програмного забезпечення?

(ДП3) Які можливі вирішення існуючих проблем та шляхи покращення системи освіти

інженерії програмного забезпечення пропонується?

Критерії включення / виключення науково-дослідних робіт у дослідження

Одним із найважливіших етапів у систематичному огляді літератури є визначення якісних критеріїв включення/ виключення робіт у дослідження.

Наступними є критерії включення робіт до нашого дослідження:

(1) Дослідження, надруковані англійською, російською та українською мовою;

(2) Дослідження, які містять у назві або тексті науково-дослідницької роботи терміни «освіта інженерії програмного забезпечення», «професійна компетентність інженера програмного забезпечення»;

(3) Кандидатські та докторські роботи з навчання інженерії програмного забезпечення;

(4) Навчання з нефіксованим часовим періодом;

(5) «сіра література» [1, 10], включаючи доповіді, надруковані незалежно від освітніх та професійних організацій.

Наступними є критерії виключення робіт з нашого дослідження:

(1) Науково-дослідницькі роботи, що повторюються у декількох електронних бібліотеках (базах даних). У таких випадках, тільки одна робота включається до наукового дослідження;

(2) Науково-дослідницькі роботи, що мають однакові результати дослідження. У такому випадку, до уваги береться лише одна найбільш повна робота;

(3) Науково-дослідницькі роботи та книги, що є недоступними до завантаження;

(4) Наукові роботи, що не містять емпіричних досліджень, або література, що є доступною лише у вигляді окремих абзаців (параграфів), або представлена у форматі Power point презентацій.

(5) Роботи, основною темою дослідження яких не є навчання інженерії програмного забезпечення, або задана тематика лише згадується у декількох реченнях \ абзацах.

Далі ми надаємо результати проведеного *systematic mapping studies*. Відповідно до Kitchenham [1, 7], оптимальним є використання у процесі пошуку як електронної (автоматичної) методики пошуку, так і неавтоматичної. У нашому дослідженні ми використовували обидві. В процесі неавтоматичного пошуку ми перевірили роботи, представлені у ряді наукових конференцій, на сайтах певних вчених-дослідників, а також усно опитали досвідчених

спеціалістів у галузі освіти інженерії програмного забезпечення. Рішення не використовувати у процесі дослідження лише автоматичну методику пошуку літератури було прийняте згідно із дослідженнями вчених [1, 8], які стверджують що максимальної об'єктивності та повноти дослідження можна досягти лише використовуючи обидва методи – як автоматичний, так і неавтоматичний. Неавтоматичний пошук літератури також гарантує більшу адекватність знайдених науково-дослідницьких робіт та їх відповідність поставлених дослідницьких питань. Електронний автоматичний пошук науково-дослідницьких робіт було здійснено у наступних електронних базах даних (бібліотеках): IEEEExplore Digital Library, ACM Digital Library, Elsevier ScienceDirect, EI Compendex, Scopus and Web of Science. Ці електронні бібліотеки є вивереними та надійними джерелами літератури в галузі програмної інженерії і комп'ютерних наук в цілому. Нашим завданням першопочатково було визначення ключових термінів та їх синонімів з поставлених питань / завдань дослідження. Також було побудовано наступну стратегію пошуку:

(1) Визначити основні терміни для здійснення пошуку;

(2) Перевірити низку визначених ключових слів у науково-дослідницьких роботах, що вже були знайдені нами;

(3) пошук та визначення альтернативних та додаткових синонімів та термінів для здійснення дослідження.

Також ми використовували логічні оператори OR та AND у процесі пошуку науково-дослідницьких робіт для більш чіткого та повного дослідження.

Результати дослідження

У нашому дослідженні ми визначили наступні етапи пошуку науково-дослідницьких робіт. Спочатку ми визначили 6 електронних баз даних (бібліотек), зазначених вище, для здійснення автоматичного пошуку літератури. Результатом пошуку в електронних базах даних стали 9.711 науково-дослідницьких робіт, однак лише 630 з них були доступні для завантаження. Після завантаження лише 113 робіт були включені у подальше дослідження після оцінки їх відповідно до визначених якісних критеріїв включення / виключення. Оскільки ми використовували декілька електронних баз даних, багато робіт мали дублікати. Остаточну кількість наукових робіт було скорочено до 40. У процесі неавтоматичного пошуку літератури спочатку ми знайшли 358 публікацій. Однак

лише 145 з них були доступні до завантаження, і 126 були включені до нашого дослідження. Підсумовуючи результати неавтоматичного (електронного) та автоматичного пошуку науково-дослідницьких робіт було знайдено 199 публікацій. Однак, після перевірки результатів обох пошуків кількість робіт, включених у остаточне дослідження було зменшене з 239 до 184 через наявність дублікатів. Після цього

останнім етапом нашого систематичного огляду літератури було якісне оцінювання знайдених публікацій, в результаті чого було відібрано 93 науково-дослідницьких робіт, що повністю відповідають поставленим критеріям пошуку.

У табл. 1 надано підсумки процесу пошуку науково-дослідницьких робіт на кожному етапі дослідження.

Табл. 1. Процес відбору відповідних наукових досліджень.

	<i>Етап I</i>		<i>Етап II</i>		<i>Етап III</i>		<i>Етап IV</i>
<i>Автоматичний (електронний) пошук</i>	9.711	<i>Крок I</i>	630	<i>Крок 2</i>	113	<i>Крок 3</i>	40
<i>Неавтоматичний пошук</i>	358		145		126		53
<i>Загальна кількість робіт</i>	10.069		485		239		93

У процесі синтезу та аналізу знайдених науково-дослідницьких робіт ми користувалися методиками, запропонованими вченим Дуба [1, 4]. Ми використовували методику порівняння робіт поєднану із якісним їх аналізом. Публікації та параграфи робіт, що відповідають дослід-

ницьким завданням були ретельно досліджені згідно з поставленими питаннями та критеріями [1, 11].

На рис. 1 показано класифікацію віднайдених науково-дослідницьких робіт відповідно до року їх публікації.

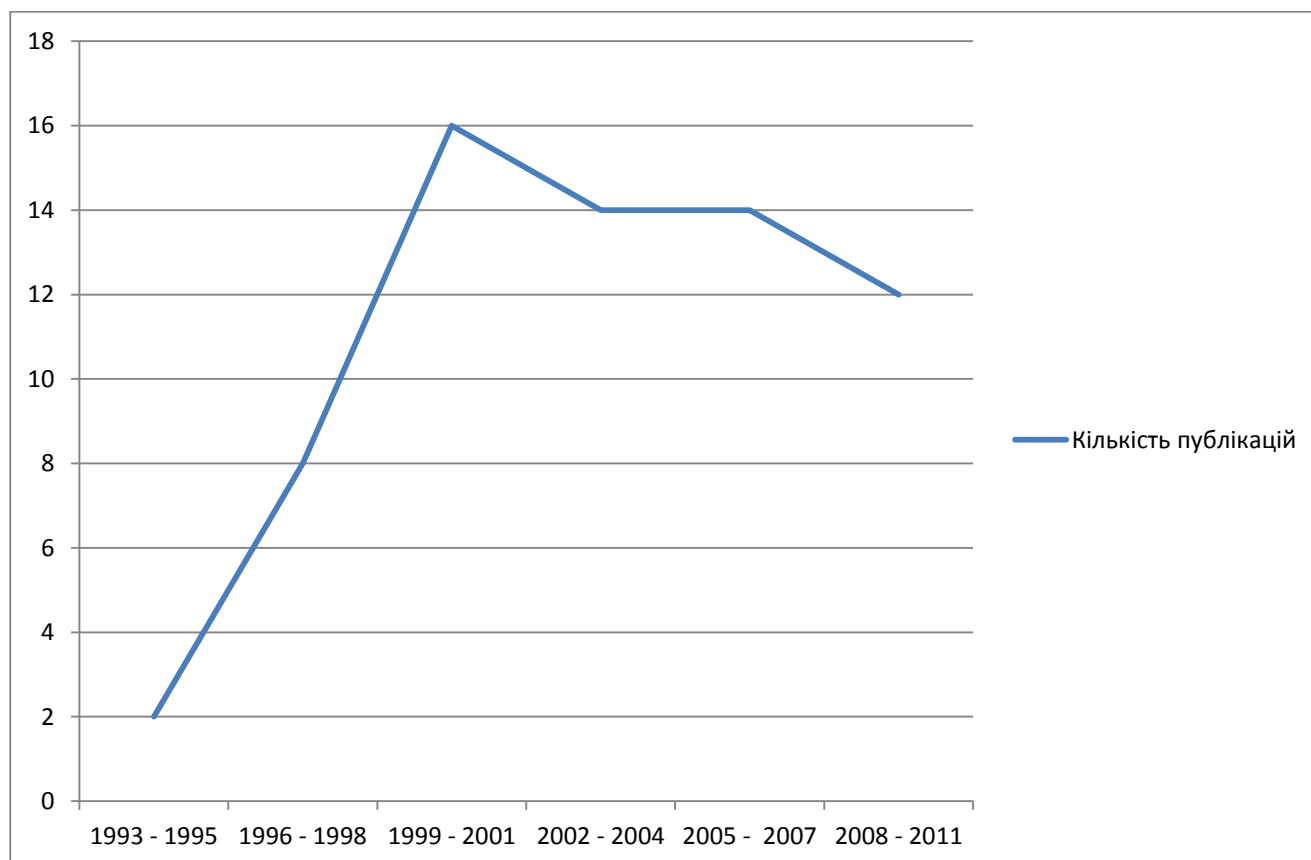


Рис. 1. Класифікація робіт відповідно до року їх публікації

Згідно з діаграмою період з 1999 по 2001 роки був найпродуктивнішим з погляду дослідження проблем освіти інженерії програмного забезпечення. На жаль, у першому десятиріччі 21 сторіччя помічаємо не значення зниження у кількості публікацій з даної теми, що підтверджує актуальність та необхідність проведення подальших досліджень в даній галузі. Найдавніша знайдена публікація з питань навчання інженерії програмного забезпечення датується 1976 роком.

На рис. 2 показано класифікацію знайдених досліджень, згідно з якою можемо бачити, що з 77 науково-дослідницьких робіт, 76% є емпіричними, тобто є заснованими на проведених експериментальних дослідженнях. 24% є теоретичними або поняттєвими роботами, що базуються на розумінні галузі освіти інженерії програмного забезпечення із власного досвіду вченого (дослідника). Незначна кількість робіт (3%) є оглядами літератури або вторинними дослідженнями, що розглядають вже подані у інших роботах результати.



Рис. 2. Типи знайдених науково-дослідницьких робіт

Надалі ми розглядаємо основні поставлені дослідницькі питання та результати проведеного дослідження.

(ДП1) Якими є головні характеристики існуючої системи освіти інженерії програмного забезпечення?

Наприкінці 90-х р. минулого століття знання і досвід, що були накопичені в індустрії програмного забезпечення за попередні 30-35 років, а також більш ніж 15-літніх спроб застосування різних моделей розробки, усе це, нарешті, оформилося в те, що прийнято називати дисципліною програмної інженерії – *Software Engineering*. Якоюсь мірою, таке формування дисципліни на основі широко розповсюдженого практичного досвіду нагадує ті

процеси, що відбувалися в управлінні проектами. Виникали і розвивалися професійні асоціації, спеціалізовані інститути, комітети зі стандартизації й інші утворення, що, зрештою, прийшли до спільної думки про необхідність зведення професійних знань по відповідним областях і стандартизації відповідних програм навчання.

У табл. 2 подано класифікацію розглянутих науково-дослідницьких робіт, що присвячені основним параметрам освітньої системи в галузі інженерії програмного забезпечення – загальне значення проведення досліджень у даній галузі, структура та недоліки існуючої освітньої системи.

Табл. 2. Загальна характеристика системи викладання інженерії програмного забезпечення

	<i>Досліджені питання</i>	<i>Науково-дослідницькі роботи</i>
1	Значення проведення досліджень в галузі навчання інженерії програмного забезпечення	8, 11, 19, 24, 31, 39, 40, 51, 52, 55, 59, 61, 63, 65, 67, 70
2	Структура існуючої системи викладання інженерії програмного забезпечення	11, 12, 13, 17, 19, 20, 25, 29, 31, 38, 39, 40, 51, 52, 53, 58, 59, 65, 68
3	Недоліки існуючої системи викладання інженерії програмного забезпечення	10, 11, 12, 19, 20, 24, 31, 32, 39, 40, 42, 49, 51, 52, 58, 59, 65, 67, 68, 69

(ПД2) Якими є основні дидактичні та педагогічні питання освіти інженерії програмного забезпечення? Наскільки дослідженими є питання професійної компетентності, акредитації та ліцензування в інженерії програмного забезпечення?

Інженерія програмного забезпечення – це комплекс процесів, методів, і інструментів, що використовуються для створення і технічної підтримки комп'ютерного програмного забезпечення з певним рівнем якості, в передбачуваній вартості, за передбачуваним графіком. Бакалаврам з інженерії програмного забезпечення необхідно бути підготовленими для кар'єри у промисловості і науці з акцентом на здатності проаналізувати, спроектувати, перевіряти, впроваджувати, застосовувати, і підтримувати системи програмного забезпечення.

Інженери програмного забезпечення - спеціалісти, які створюють надійні, рентабельні системи програмного забезпечення для клієнтів. Свідоцтво або сертифікат - декларація того, що спеціаліст є абсолютно компетентним в галузі. Сертифікат – це реквізит професійних учбових програм. Одна з переваг професійного технічного акредитування - це встановлення рівня мінімальної компетентності, що гарантує, що спеціаліст має тверде розуміння як практичних, так і теоретичних аспектів в області інженерії програмного забезпечення. Професійна сертифікація також пропагує етику і відповідальність серед практикуючих фахівців.

У табл. 3 надано основні педагогічні питання, що висвітлюються у науково-дослідницьких роботах у сфері викладання інженерії програмного забезпечення.

Табл. 3. Педагогічні питання.

	<i>Досліджені питання</i>	<i>Науково-дослідницькі роботи</i>
1	Дидактичні та педагогічні питання викладання інженерії програмного забезпечення	11, 19, 33, 39, 53, 58, 59, 68, 81 - 85
2	Методи та підходи, що використовуються у системі викладання інженерії програмного забезпечення	2, 4, 8, 10, 11, 12, 13, 15, 16, 17, 18, 19, 20, 22, 28, 31, 39, 40, 41, 48, 50, 51, 52, 53, 58, 59, 65, 67, 70, 76, 79
3	Проблеми професійної компетентності інженерів програмного забезпечення	11, 19, 20, 23, 30, 40, 50, 51, 52, 58, 59, 62, 63, 71, 72, 77, 78, 84 - 93
4	Уміння та знання, необхідні для професійного компетентного інженера програмного забезпечення	19, 34, 38, 39, 40, 43, 50, 51, 52, 58, 59, 63, 64, 68, 77, 78
5	Питання ліцензування, сертифікації та акредитації в галузі інженерії програмного забезпечення	19, 30, 34, 50, 51, 52, 54, 59, 60, 62, 63, 70, 81 - 85

(ПД3) Які можливі вирішення існуючих проблем та шляхи покращення системи освіти інженерії програмного забезпечення пропонуються?

Програмне забезпечення має все більш і більш значну суспільну важливість. Користувачі бажають та заслуговують на гарантії якості програмного забезпечення. Ми можемо придбати довіру в якість продукту безпосередньо маючи попередню довіру у інженерів програмного забезпечення, що його створюють.

Розробка програмного забезпечення і його підтримка вимагає багато навиків, у тому числі

дизайн, управління, програмування, аналіз, оцінювання, документація, системна інтеграція, а також дизайн для безпеки і надійності. Окрім ідеального знання теоретичних питань успішний інженер програмного забезпечення повинен володіти певними додатковими навиками.

Табл. 4 пропонує найбільш широко поширені на сьогоднішній день підходи та методи покращення якості викладання інженерії програмного забезпечення для підвищення професійної компетентності спеціалістів даної галузі.

Табл. 4. Підходи та методики покращення якості викладання інженерії програмного забезпечення

	Досліджені питання	Науково-дослідницькі роботи
1	Навчання, поєднане із практикою на підприємствах	1, 6, 11, 12, 19, 24, 31, 39, 40, 51, 52, 55, 59, 61, 63
2	Групові методики навчання	1, 3, 5, 7, 9, 10, 14, 15, 21, 26, 27, 28, 33, 36, 57, 46, 47, 48, 52, 57, 58, 66, 69
3	Участь у довільних проектах	1, 19, 39, 40, 58
4	Самонавчання	19, 22, 33, 49, 55

Наскільки нам відомо, проведене нами дослідження є першим в галузі освіти та викладання інженерії програмного забезпечення. Проведення подальших вторинних досліджень та літературних оглядів є необхідними для майбутнього розвитку освітньої системи.

Узагальненими етапами та процедурами для покращення системи викладання інженерії програмного забезпечення можуть вважатися наступні:

- Перебудова структури навчального процесу та плану з використанням новітніх методик навчання і додаванням більшого об'єму практичних проектів.
- Систематична інтеграція прикладних та експериментальних досліджень в інженерії програмного забезпечення у навчальний процес.
- Укріплення 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 учбового плану.

Заключення

Чотири роки існування напряму 6.050103 «Програмна інженерія» показали, що є певні проблеми, які пов'язані з формуванням ключових компетентностей бакалаврів з інженерії програмного забезпечення, особливо в напрямку їхньої готовності до професійної комунікації. Виходячи з того, що розробка програмного

забезпечення є командною справою, групова динаміка і професійні комунікації мають велике значення для ефективного виконання процесів життєвого циклу програмного забезпечення, а саме: оволодіння продуктивними знаннями та інтегрованими вміннями професійної комунікації, особливостями вербальних і невербальних засобів комунікації, оволодіння навичками культури спілкування та вироблення умінь застосувати їх на практиці, що є неможливою без формування належного рівня професійної комунікації майбутніх розробників програмного забезпечення. Це, в свою чергу, вимагає від майбутніх бакалаврів з інженерії програмного забезпечення формування здібностей до алгоритмічного мислення, сукупності знань та навичок у сфері інформаційно-комп'ютерних технологій, здатності використовувати їх у професійній діяльності, а також мотиваційної потреби у постійному професійному самонавчанні та самовдосконаленні. Таким чином, актуальним стає завдання розробки інтегрованого підходу до формування готовності майбутніх бакалаврів з інженерії програмного забезпечення до професійної комунікації.

Спираючись на результати проведеного аналізу навчальної, монографічної та наукової літератури, з урахуванням практичних вимог щодо професійної комунікації майбутніх бакалаврів з інженерії програмного забезпечення, надалі буде виявлено інваріантний склад технічних і гуманітарних компетенцій, які необхідно сформулювати у майбутніх фахівців для того, щоб забезпечити їх готовність до професійної комунікації в процесі професійної діяльності.

I. СИСТЕМАТИЧНИЙ ОГЛЯД ЛІТЕРАТУРИ

- 1) Bailey, J., Budgen, D., Turner, M., Kitchenham, B., Brereton, P. & Linkman, S. (2007), Evidence relating to object-oriented software design: A survey, *in* 'Proc. of the 1st Int. Symp. on Empirical Software Engineering and Measurement (ESEM 2007)', pp. 482-484.
- 2) Budgen, David, Stuart Charters, Mark Turner, Pearl Brereton, Barbara Kitchenham and Stephen Linkman Investigating the Applicability of the Evidence-Based Paradigm to Software Engineering, Proceedings of WISER Workshop, ICSE 2006, 7-13, May 2006, ACM Press
- 3) David Budgen, Mark Turner, Pearl Brereton, and Barbara Kitchenham. Using Mapping Studies in Software Education. – Department of Computer Science, Durham University, School of Computing and Maths, Keele University, 2011.
- 4) Dyba, T. Dingsoyr, T. Empirical studies of agile software development: A systematic review. *Information and Software Technology*. – 2008, vol. 50, pp. 833—859.
- 5) Kai Petersen, Feldt R., Mujtaba, S., Mattsson, M. Systematic Mapping Studies in Software Engineering. In: 12th International Conference on Evaluation and Assessment in Software Engineering. (2008).
- 6) Kitchenham, B. Procedures for Performing Systematic Reviews. Joint Technical Report, July 2004.
- 7) Kitchenham, B. Guidelines for performing Systematic Literature Reviews in Software Engineering. vol 2. 3 EBSE Technical Report. (2007)
- 8) Kitchenham, B., Brereton P., Budgen, D., Turner M., Bailey J., Linkman, S. Systematic literature reviews in software engineering - A systematic literature review, *Information and Software Technology*, vol. 51, Issue 1. (2009)
- 9) Olavo Barbosa, Carina Alves. A systematic mapping study on software ecosystems. Proceedings of the Third International Workshop on Software Ecosystems – IWSECO-2011, Brussels, Belgium. – June 7th, 2011. – Vol.746. – pp. 15 – 26.
- 10) Petticrew, M., Roberts, H. Systematic Reviews in the Social Sciences: A Practical Guide (2008)
- 11) Seaman, C. B. Qualitative Methods in Empirical Studies of Software Engineering. *IEEE Transactions on Software Engineering*, 25(4), pp. 557--572. (1999)
- 12) B. Beizer. Qualities of a good tester: a baker's dozen. *Quality Techniques Newsletter*, Software Research, Inc., March 2002. Available at

"<http://www.soft.com/News/Online/qtnmar02.html>".

QTN-

II. ВИКЛАДАННЯ ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

- 1) Allen H. Dutoit, Timo Wolf, Barbara Paech, Lars Borner, Jurgen Ruckert. Using Rationale for Software Engineering Education. 2005
- 2) Ann E. Kelley. A Software Engineering Curriculum Incorporating an Operational Formal Method. Sobel Systems Analysis Department Miami University Oxford. 2000.
- 3) Ardis, Mark A. and Ford Gary A., 1989 SEI Report on Graduate Software Engineering Education, Technical Report CMU/SEI-89-TR-2, Carnegie Mellon University, Pittsburgh PA, 1989.
- 4) Atsuo HAZEYAMA, Akiko NAKANO Web-based Inspection Process Support Environment for Software Engineering Education. Department of Mathematics and Information Science Tokyo Gakugei University. 2001.
- 5) Bagert, Donald J.; Mengel, Susan A. and Gregory, James M, "Web-Based Software Engineering Student Projects", Proceedings of the American Society for Engineering Education Gulf-Southwest Section Annual Conference, Las Cruces NM, 5-8 April 2000, 8 pp.
- 6) Barry Boehm , Alexander Egyed, Dan Port, Archita Shah, Julie Kwan and Ray Madachy. A stakeholder win-win approach to software engineering Education. *Annals of Software Engineering* 6 (1998) 295-321.
- 7) Boehm, B. (1996) "Anchoring the Software Process", *IEEE Software*, July, pp. 73 – 82.
- 8) Colin J Burgess. The Role of Formal Methods in Software Engineering Education and Industry. University of Bristol, Department of Computer Science University of Bristol Bristol.
- 9) Donald J. Bagert, James M. Gregory, Susan A. Mengel, Lloyd R. Heinze. Engineering Education Innovation with Software Engineering Projects. Proceedings of Frontiers in Education 2002, November 6-9, 2002, Boston, MA.
- 10) Elmer Raydean Richmond. SOFTWARE ENGINEERING EDUCATION IN THE ASSOCIATE-DEGREE-LEVEL VOCATIONAL/TECHNICAL COMPUTER SCIENCE PROGRAM'. Computer Science Department Tarrant County Junior College Fort Worth, Texas. CIS Educator Forum Volume 2, Number 3. 1993
- 11) Engineering Education. Designing an Adaptive System. National Academy of Sciences - National Research Council, Washington, DC. Commission on Engineering and Technical Systems. National Academy Press, 2101

- Constitution Ave., N.W., Box 285, Washington, DC. 1995
- 12) Eric Roberts. Computing Education and the Information Technology Workforce (with the endorsement of the ACM Education Board). March 1, 2000
- 13) First Draft of the Software Engineering Education Knowledge (SEEK)/ Edited by Ann E.K. Sobel CCSE Knowledge Area Chair. August 28, 2002
- 14) Gates A.Q.; Delgado N.; Mondragon O.; "A structured approach for managing a practical SE course", 30th Frontiers in Education conf., IEEE, 2000, pp. TIC/21 – TIC/26
- 15) GEORGE G. MITCHELL and JAMES DECLAN DELANEY. An Assessment Strategy to Determine Learning Outcomes in a Software Engineering Problem-based Learning Course. Int. J. Engng Ed. Vol. 20, No. 3, pp. 494-502, 2004 Printed in Great Britain.
- 16) Ghazi Alkhatib, Ghassan Issa, Aiman Turani, M. Ibrahim Zaroor. Incorporating Innovative Practices in Software Engineering Education. April 04 - 06, 2011, IEEE EDUCON Education Engineering 2011 - Learning Environments and Ecosystems in Engineering Education. Amman, Jordan.
- 17) Giancarlo Succi, Romana Spasojevic. A Survey on the Effectiveness of the Internet-Based Facilities in Software Engineering Education. Department of Electrical and Computer Engineering, University of Calgary, Calgary
- 18) Gilda Pour. Software Engineering Education for the Pervasive Human-Centric Computing Era. Department of Computer Engineering, San Jose State University San Jose. Global J. of Engng. Educ, Vol.11, No.1, 2007. Published in Australia.
- 19) Graduate. Employability. Skills. Prepared for the. Business, Industry and. Higher Education. Collaboration Council. August 2007
- 20) Hans van Vliet. Reflections on Software Engineering Education. IEEE Proceedings. Vrije Universiteit Amsterdam. May – June. 2006.
- 21) Hayes J. Huffman "Energizing SE Education through Real-World Projects as Experimental Studies", in Proceedings of the 15th Conference on SE Education and Training (CSEET), Covington, KY, February 2002.
- 22) Heidi J. C. Ellis. Student-Directed Learning in a Graduate Software Engineering Course. Department of Engineering and Science, Rensselaer (RPI) Hartford Hartford, October 20 - 23, 2004, Savannah, GA.
- 23) J. B. Thompson. Closing the Circle on Software Engineering Professionalism and Free Movement of Labour. School of Computing, Engineering and Technology University of Sunderland, Sunderland, UK. 2002.
- 24) J. Fernando Naveda, Michael J. Lutz, Thomas B. Hillburn, Linda M. Northrup, Michael C. Stinson. The Role of Software Engineering in Undergraduate Education. 1997.
- 25) James Cusick. Lessons Learned from Teaching Software Engineering to Adult Students. AT&T Labs Shannon Laboratory. 2000
- 26) Jane Huffman Hayes, Timothy C. Lethbridge, Deniel Port. Evaluating Individual Contribution Toward Group Software Engineering Projects (10.1.1.9.9114)
- 27) Jarzabek, Stan (editor), "Teaching Software Project Courses" (special issue topic containing 13 papers), Forum for Advancing Software engineering Education (FASE), Vol. 11 No. 6 (June 2001). <http://www.cs.ttu.edu/fase/v11n06.txt>
- 28) JIAN CHEN. An Approach for Assessing Students' Performance in Software Engineering Capstone Projects. School of Computer Science Shaanxi Normal University South Changan Rd., Xi'an, CHINA. 1999.
- 29) John D. Tvedt, Roseanne Tesoriero, Kevin A. Gary. The Software Factory: Combining Undergraduate Computer Science and Software Engineering Education. 2001.
- 30) John R. Speed. What do you mean I can't call myself a software engineer? Texas Board of professional engineers. IEEE. November/December 1999.
- 31) Jonathan Lee, Yu Chin Cheng. Change the Face of Software Engineering Education: a Field Report from Taiwan. Department of Computer Science and Information Engineering, National Central University, Jhongli, Taiwan. 2011.
- 32) Judy Sheard, Martin Dick, Ian Macdonald, Meagan Walsh. Cheating and Plagiarism: Perceptions and Practice of 1st Year IT Students. ITiCSE '02 Proceedings of the 7th annual conference on Innovation and technology in computer science education. 2002.
- 33) L. B. S. Raccoon. A Leadership Primer for Software Engineers. листопад 27, 2012.
- 34) Leonard R. Tripp. Professionalization of Software Engineering: Next Steps. IEEE Software Society. March 1999.
- 35) Letizia Jaccheri. On the Importance of Dialogue with Industry about Software Engineering Education. Department of Computer and Information Science Norwegian University of Science and Technology, Norway. 2006.
- 36) M. Brian Blake and Todd Cornett. Teaching an Object-Oriented Software Development Lifecycle in Undergraduate Software Engineering

- Education. Department of Computer Science Georgetown University Washington, DC. 2002.
- 37) Maria Isabel Alfonso and Francisco Mora. Learning Software Engineering with Group Work. Dept. of Computer Science and Artificial Intelligence. University of Alicante. Spain. 2009
- 38) Mark A. Ardis, Peter B. Henderson. Software Engineering Education (SEEd). ACM SIGSOFT Software Engineering Notes. November 2009 Volume 34 Number 6.
- 39) Mary Shaw, Jim Herbsleb and Ipek Ozkaya. Methods: Deciding What to Design. Course home page. Sept. 2005- December 2005. School of Computer Science, Carnegie-Mellon University.
- 40) Mary Shaw. Software Engineering Education: a Roadmap. Institute for Software Research, International Carnegie Mellon University Pittsburgh. 2000.
- 41) Michael Barker, Katsuro Inoue. IT SPIRAL: A Case Study in Scalable Software Engineering. Nara Institute of Science and Technology, Osaka University. 2009.
- 42) Michael J. Lutz. Alloy, Software Engineering, and Undergraduate Education. Department of Software Engineering Rochester Institute of Technology. 2008.
- 43) Muhammad Raza Ali . Imparting Effective Software Engine. ACM SIGSOFT Software Engineering Notes. Volume 31, Number 4, July, 2006
- 44) Naoki Ohsugi. EASE Project: Introducing Empirical Software Engineering into Japanese Industry. Graduate School of Information Science, Nara Institute of Science and Technology. Kansai Science City, Japan. 2004.
- 45) Nor Azliana Akmal Jamaludin, Shamsul Sahibuddin. Readiness of Requirement Engineering towards Global Collaboration and Communication. Faculty of Computer Science and Information System Universiti Teknologi Malaysia. International Journal of Computer Trends and Technology- July to Aug Issue 2011.
- 46) Olly Gote, Vidya Kulkarni, Long Chrea Neak, Christelle Scharff, and Sopheap Seng. Introducing Global Supply Chains into Software Engineering Education. 2005.
- 47) Owen, Cherry; Bagert, Donald J. and Gregory, James M., "QUICK: Using Web Technology To Facilitate Class Scheduling", Proceedings of the American Society for Engineering Education Gulf-Southwest Section Annual Conference, Las Cruces NM, 5-8 April 2000, 13 pp.
- 48) P. Brereton, S. Lees and M. Gumbley, C. Boldyreff and S. Drummond, P. Layzell, L. Macaulay and R. Young. Distributed Group Working in Software Engineering Education. 1998.
- 49) Pankaj Kamthan. A Social Web Perspective of Software Engineering Education. Concordia University, Canada. 2010.
- 50) Peter J. Denning. PROFESSIONAL SOFTWARE ENGINEERING EDUCATION. Computer Science Department George Mason University. 1998.
- 51) Robert L. Baber. Software engineering education: issues and alternatives. Department of Computer Science University of the Witwatersrand, Johannesburg. 1999
- 52) SE2004. Curriculum Guidelines for Undergraduate Degree programs in SE. A volume of the Computing Curricula Series. August 23, 2004.
- 53) Shaoying Liu, Kazuhiro Takahashi, Toshinori Hayashi, Toshihiro Nakayama. Teaching Formal Methods in the Context of Software Engineering. 2009.
- 54) Shari L. Pfleeger, Joann M. Atlee. The future of software engineering: Theory and practice. 4th edition. 2011
- 55) Shihong Huang, Damiano Distanto. On Practice-Oriented Software Engineering Education. 2006.
- 56) Silvije Jovalekic. Project-Oriented Approach to Software Engineering Education in a Multidisciplinary Environment: Objectives, Rtization, Evaluation. Information Technology and Engineering Fachhochschule Albstadt-Sigmaringen, Germany. 1996
- 57) Sims-Knight J., Upchurch R., Powers T.A., Haden S, Topciu R. "Teams in SE Engineering"/ Proceedings of Frontiers in Education 2002, November 6-9, 2002, Boston, MA, pp. S3G17 – S3G22.
- 58) Software engineering education in the modern age: software education and training sessions at the International Conference on Software Engineering, ICSE 2005, St. Louis, MO, USA, May 15-21, 2005
- 59) Software Engineering: Effective Teaching and Learning Approaches ...Heidi J. C. Ellis, Steven A. Demurjian, J. Fernando Naveda, Na J. Fernando. 2008.
- 60) T. H. Tse. Software Engineering Professionalism: Is the End of Constraints and Conflicts in Sight? Department of Computer Science and Information Systems The University of Hong Kong Pokfulam Road, Hong Kong. Proceedings of the 26 th Annual International Computer Software and Applications Conference (COMPSAC'02). 2002.
- 61) T. Lethbridge, B. Probert, D. Ionescu, D. Gibbons, J. Raymond, L. Orozco-Barbosa, S.

Szpakowicz, (1997) "Proposal for a Software Engineering Program", School of Information Technology and Engineering, University of Ottawa.

62) Task Force Report and Recommendation for Computer and Software Engineering Licensure Path. Prepared by the Professional Engineers in Industry - Computer and Software Engineering (PEI-CSE) Task Force for the NSPE Licensure & Qualifications for Practice (L&QP) Committee. August 10, 2006.

63) The Software Engineering Body Of Knowledge (SWEBOK). 2004

64) Thomas B. Hilburn, Dale Oexmann, Susan Mengel. Software Engineering Across Computing Curricula. 1998.

65) Vasudeva Varma, Kirti Garg, Vamsi. A Case Study Initiative for Software Engineering Education. International Institute of Information Technology, Gachibowli, Hyderabad, India. 2005.

66) Wasserman, Anthony I. and Peter Freeman (editors), Software Engineering Education: Needs and Objectives, Springer-Verlag New York (1976).

67) Wm. Arthur Conklin, Glenn Dietrich. Secure Software Engineering: A New Paradigm. Proceedings of the 40th Hawaii International Conference on System Sciences - 2007

68) Wolf-Gideon Bleek, Carola Lilienthal, Axel Schmolitzky Weaving Experiences from Software Engineering Training in Industry into Mass University Education. Department of Informatics, University of Hamburg Hamburg, Germany. 2004

69) Zaigham Mahmood. A Framework for Software Engineering Education: A Group Projects Approach. INTERNATIONAL JOURNAL OF EDUCATION AND INFORMATION TECHNOLOGIES Issue 3, Volume 1, 2007.

70) C. Wohlin and B. Regnell, "Achieving Industrial Relevance in Software Engineering Education", Proceedings Conference on Software Engineering Education & Training, pp. 16-25, New Orleans, Louisiana, USA, 1999.

ІІІ. РОБОТИ ВІТЧИЗНЯНИХ НАУКОВОЦІВ

71) Бондаренко М., Сидоров М., Морозова Т., Мендзєбровський І. Модель випускника бакалаврату "Програмна інженерія" (3 досвіду роботи науково-методичної підкомісії 050103) / М. Бондаренко, М. Сидоров, Т. Морозова, І. Мендзєбровський // Вища школа. — 2009. — № 4. — С. 50—61.

72) Гагарский Р.К. Программа подготовки специалистов в IT-компаниях. // В сб. «Системное программирование». / Вып. 3, под ред.

А.Н.Терехова и Д.Ю.Булычева. СПб: Изд. СПбГУ, 2008. С. 141-156.

73) Диагностика конкурентоспособности специалиста / Н.К.Нуриев, С.Д.Старыгина // Мониторинг качества воспитания и творческого развития конкурентоспособности личности: Мат. XIII Всерос. науч.-практич. конф. – Казань: Центр инновационных технологий, 2005. – С. 265 – 271.

74) Европейская рамка ИКТ-компетенций 2.0. Часть 1. Общая европейская рамка компетенций ИКТ-специалистов для всех секторов индустрии. – Соглашение рабочей группы SEN. – Федеральное агенство по техническому регулированию и метрологии (РОССТАНДАРТ). – Группа МКС 13.340.40. – Москва, 2011. – стр. 78.

75) Европейская рамка ИКТ-компетенций 2.0. Часть 2. Руководство по использованию европейской рамки ИКТ-компетенций. – Соглашение рабочей группы SEN. – Федеральное агенство по техническому регулированию и метрологии (РОССТАНДАРТ). – Группа МКС 13.340.40. – Москва, 2011. – стр. 33.

76) Европейская рамка ИКТ-компетенций 2.0. Часть 3. Создание e-CF – соединение методологических основ и опыта экспертов. – Соглашение рабочей группы SEN. – Федеральное агенство по техническому регулированию и метрологии (РОССТАНДАРТ). – Группа МКС 13.340.40. – Москва, 2011. – стр. 28.

77) Журбенко Л.Н., Нуриев Н.К., Старыгина С.Д. Модель системы подготовки инженеров в компетентностном формате. – Educational Technology & Society. – 11 (4) 2008.

78) Зимняя, И.А. Компетентностный подход. Каково его место в системе современных подходов к проблемам образования? (теоретико-методологический аспект) / И. А. Зимняя // Высш. образование сегодня : реформы, нововведения, опыт: журнал. - 2006. - №8. - С. 20-26.

79) Луцький М., Сидоров М. Умови підготовки фахівців з програмного забезпечення// Вища школа. - 2009. - №11. - С.78 - 98.

80) Луцький М.Г., Сидоров М.О. Інформатизація процесів підготовки фахівців з програмного забезпечення. – Київ: НАУ, 2010, 57 с.

81) Морозова Т.Ю. Модель класифікації освітніх спеціальностей в IT-сфері. – Луганськ: Східноукраїнський університет ім. Володимира Даля // Інженерія програмного забезпечення. - №2, 2010. – 7 с.

82) Нуриев Н.К., Журбенко Л.Н., Старыгина С.Д. Мониторинг качества подготовки будущего инженера (бакалавра, магистра в компетентностном формате): учебное пособие /

Н.К.Нуриев, Л.Н.Журбенко, С.Д.Старыгина. – Казань: Изд-во Казан. гос. технол. ун-та, 2007. – 80 с.

83) Нуриев Н.К., Журбенко Л.Н., Фатыхов Р.Х., Старыгина С.Д. Дидактическая система подготовки конкурентоспособных специалистов в области программной инженерии в условиях технологического университета / Н.К.Нуриев, Л.Н.Журбенко, Р.Х.Фатыхов, С.Д.Старыгина, 2005 // Educational Technology & Society – 2005 (<http://ifets.ieee.org/russian/periodical/journal.html>) - V.8. - N 3. - 120 с.

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

85) Сидоров М.О., І.Б. Мендзєбровський, О.А.Орехов. Професійна практика програмної інженерії – досвід викладання. – Инженерія програмного забезпечення, №2, 2010, Київ: Національний авіаційний університет, с. 56-65.

86) Морозова Т.Ю. Модель класифікації освітніх спеціальностей в IT-сфері, Инженерія програмного забезпечення, №2, 2010, Київ: Національний авіаційний університет, с. 65-73

87) Сидоров Н.А. Инженерия программного обеспечения — дисциплина или бакалаврат? // Матеріали міжнародної науково-практичної конференції "Розробка систем програмного забезпечення: виклики часу та роль у інформаційному суспільстві". Київ, 27—28 січня 2005 р.

88) Кириленко Е.Г., Лучшева О.В. Обоснование содержания обучения в рамках методологии преподавания профессионально-ориентированной дисциплины «Групповая

динамика и коммуникация», - Инженерія програмного забезпечення, №1, 2010, Київ: Національний авіаційний університет, с. 62-71.

89) Сидоров Н.А., Мендзєбровський І.Б. Подготовка в Украине инженеров по программному обеспечению // Управление качеством инженерного образования и инновационные образовательные технологии: Сборник докладов Международной научно-метод. конференции. Москва, 28—30 октября 2008 г. — М.: МГТУ им. Н.Э. Баумана, 2008. — Ч. I. — С. 14—18.

90) Сидоров Н.А., Мендзєбровський І.Б., Морозова Т.Ю. Подготовка в Украине инженеров по программному обеспечению // Матеріали IV Международная конференция "Стратегия качества в промышленности и образовании", 30.05—06.06 2008, г. Варна, Болгария.

91) Морозова Т.Ю. Освітні та наукові IT-спеціальності у кількісному вимірі, - Инженерія програмного забезпечення, №1, 2010, Київ: Національний авіаційний університет, с. 79-88.

92) Сидоров Н.А., Петрук А.Г. M-Education на факультете компьютерных наук: опыт Национального авиационного университета // Матеріали науково-практичної конференції "Проблеми та перспективи підготовки фахівців у сфері інформаційно-комп'ютерних технологій" 2007, м. Львів.

93) Сухомлин В.А. "IT-образование: концепция, образовательные стандарты, процесс стандартизации". М.: Горячая линия - Телеком, 2005.

94) Сухомлин В.А., В.В. Сухомлин, "Концепция нового образовательного направления", Открытые системы, 2003, № 2.

95) Терехов Н.А., А.А.Терехов. Computing Curricula: Software Engineering и российское образование. // Открытые системы. 31/10/2006. – № 08.

Відомості про авторів



Сидорова Ніка Миколаївна – аспірант кафедри педагогіки та психології професійної освіти Національного авіаційного університету. Наукові інтереси – педагогіка та психологія професійної освіти

E-mail: nika.sidorova@gmail.com



Лузік Ельвіра Василівна - д.п.н., професор, завідувач кафедри педагогіки та психології професійної освіти. Наукові інтереси – теорія та методика професійної освіти; методика викладання спеціальних дисциплін.

E-mail: kppo-nau@ukr.net