

## ФОРМУВАННЯ ГОТОВНОСТІ МАЙБУТНІХ БАКАЛАВРІВ З ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДО ПРОФЕСІЙНОЇ КОМУНІКАЦІЇ

*Н.М. Сидорова*

Національний авіаційний університет, 03058, проспект Космонавта Комарова, 1,  
факс (044) 497 3141, тел.: (044) 497 4105, nika.sidorova@gmail.com

Розробка програмного забезпечення в Україні 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.

Software engineering in Ukraine is a 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.

### Вступ

Розробка програмного забезпечення в Україні є індустрією, що постійно розвивається. Отже, виникає необхідність у фахівцях різних спеціальностей. У 2006 році в Україні засновано новий напрямок навчання з підготовки інженерів програмного забезпечення «Програмна інженерія». Розроблено відповідні стандарти та розпочато викладання дисциплін [1]. Чотири роки існування напрямку 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.

**Актуальність.** Сучасний період в економічно розвинених країнах характеризується переходом від суспільства конкретних знань до інформаційно-компетентного суспільства. Каталізатором змін є глобалізаційні процеси побудови єдиного світового та європейського інформаційно-освітнього простору. Прискорити процеси переходу України до ринкових умов господарювання, формування нових інтегративно-компетентнісних відносин можливо передусім за рахунок більш ефективного використання інтелектуального й освітнього потенціалу нації. Створення програмного забезпечення в Україні є індустрією, яка постійно розвивається, що вимагає необхідності підготовки фахівців даного спрямування. Саме тому в 2006 році в Україні було засновано новий напрямок підготовки інженерів програмного забезпечення 6.050103 «Програмна інженерія» на основі відповідного галузевого стандарту [2]. Чотири роки існування напрямку 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 [3 – 5]. 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.

Метою дослідження є теоретичне обґрунтування та експериментальна перевірка організаційно-педагогічних засобів формування у майбутніх фахівців у галузі інженерії ПЗ готовності до професійної комунікації на основі інтегрованого підходу.

Об'єкт дослідження – процес професійної підготовки майбутніх фахівців у галузі інженерії програмного забезпечення у ВНЗ авіаційної галузі.

Предмет дослідження – організаційно-педагогічні умови формування готовності майбутніх фахівців з інженерії програмного забезпечення до професійної комунікації.

Гіпотеза дослідження полягає у тому, що готовність майбутніх бакалаврів з інженерії програмного забезпечення має поєднувати ознаки загального, особливого та індивідуального, де загальне буде обумовлюватися впровадженням в навчально-виховний процес дидактичної моделі інтегрованої готовності майбутніх бакалаврів з інженерії програмного забезпечення готовності заснованої на застосуванні інноваційних технологій; особливе буде обґрунтоване використання державних стандартів, відповідних ОКХ та ОПП для скоригованості змістового та процесуального компонентів педагогічного впливу. Як індивідуальне – посилення мотивації навчання через використання при формуванні готовності майбутніх бакалаврів з інженерії програмного забезпечення проектного інтегративного навчального курсу з відповідними засобами інтерактивних технологій.

Першочерговим завданням є завдання здійснити систематичний аналіз навчальної, монографічної та наукової літератури, законодавчої та нормативної бази відповідно до теми; порівняльний аналіз стандартів вищої освіти щодо підготовки майбутніх фахівців у галузі інженерії програмного забезпечення, чинних у США, РФ та Україні.

Систематичний огляд літератури – це один з методів дослідження, що останнім часом набуває все більшого поширення у сфері дослідження інженерії програмного забезпечення [6]. *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]. *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 короткий 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 [6]. 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. А *systematic mapping study* дозволяє проводити дуже точні та детальні дослідження у необхідній галузі.

### 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.

### 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.

### 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» [7].

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

- пошук. Визначення низки науково-дослідницьких робіт, в яких можуть розкритися поставлені питання/ завдання дослідження;
- включення/виключення науково-дослідницьких робіт із дослідницького процесу. Відбір літератури для проведення подальшого дослідження;
- відповідність питанням/завданням дослідження. Якщо необхідно, проведення якісного оцінювання

віднайдених науково-дослідницьких робіт.

## 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* [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 пропонуються?

### 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:

- дослідження, надруковані англійською, російською та українською мовами;
- дослідження, які містять у назві або тексті науково-дослідницької роботи терміни «освіта інженерії програмного забезпечення», «професійна компетентність інженера програмного забезпечення»;
- кандидатські та докторські роботи з навчання інженерії програмного забезпечення;
- навчання з нефіксованим часовим періодом;
- «сіра література», включаючи доповіді, надруковані незалежно від освітніх та професійних організацій.

Наступними є критерії виключення робіт з нашого дослідження:

- науково-дослідницькі роботи, що повторюються у декількох електронних бібліотеках (базах даних). У таких випадках, тільки одна робота включається до наукового дослідження;
- науково-дослідницькі роботи, що мають однакові результати дослідження. У такому випадку, до уваги береться лише одна найбільш повна робота;
- науково-дослідницькі роботи та книги, що є недоступними до завантаження;
- 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;
- 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 [8], 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, 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:

- визначити основні терміни для здійснення пошуку;
- перевірити низку визначених ключових слів у науково-дослідницьких роботах, що вже були знайдені нами;
- 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. Остаточна 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.

Таблиця 1. Процес відбору відповідних наукових досліджень

	Етап I		Етап II		Етап III		Етап IV
Автоматичний (електронний) пошук	9.711	Крок 1	630	Крок 2	113	Крок 3	40
Неавтоматичний пошук	358		145		126		53
Загальна кількість робіт	10.069		485		239		93

У процесі синтезу та аналізу знайдених науково-дослідницьких робіт ми користувалися методиками, запропонованими вченим Дуба [9]. Ми використовували методику порівняння робіт поєднану із якісним їх аналізом. Публікації та параграфи робіт, що відповідають дослідницьким завданням були ретельно досліджені згідно з поставленими питаннями та критеріями.

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

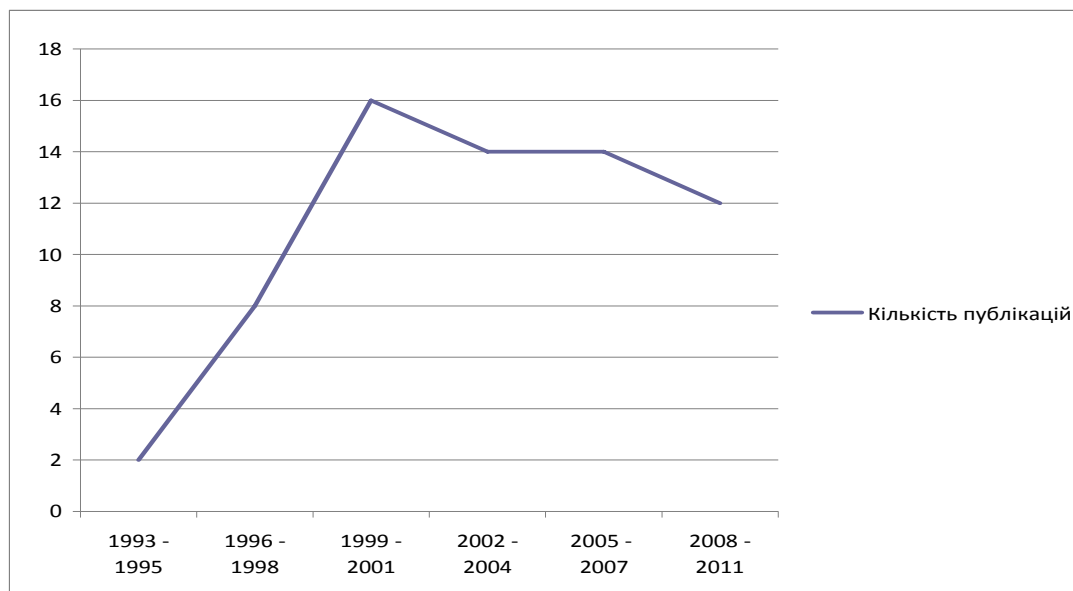


Рис. 1. Класифікація робіт відповідно до року їх публікації

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. Далі всі посилання на знайдену літературу можна деталізувати в роботі [10].

**(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.

Таблиця 2. Загальна характеристика системи викладання інженерії програмного забезпечення

	Досліджені питання	Науково-дослідницькі роботи
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 та 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.

Таблиця 3. Педагогічні питання

	Досліджені питання	Науково-дослідницькі роботи
1	Дидактичні та педагогічні питання викладання інженерії програмного забезпечення	11, 19, 33, 39, 53, 58, 59, 68, 81 - 85
2	Методики 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

**(ПДЗ)** 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 довіра 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 методики покращення якості викладання інженерії програмного забезпечення для підвищення професійної компетентності спеціалістів даної галузі.

Таблиця 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**

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.

1. *Бондаренко М., Сидоров М., Морозова Т., Мендзєбровський І.* Модель випусника бакалаврату "Програмна інженерія" (3 досвіду роботи науково-методичної підкомісії 050103) // Вища школа. – 2009. – № 4. – С. 50 – 61.
2. SE2004. Curriculum Guidelines for Undergraduate Degree programs in SE. A volume of the Computing Curricula Series. August 23, 2004. – 278 P.
3. *Сидоров М.О.* Групова динаміка і комунікації: Конспект лекцій. – К.:Видавництво НАУ «НАУ-друк», 2010. – 60 с.
4. *Кириленко Е.Г., Лучшева О.В.* Обоснование содержания обучения в рамках методологии преподавания профессионально-ориентированной дисциплины «Групповая динамика и коммуникация» // Інженерія програмного забезпечення. – 2010. – № 1. – С. 62 – 71.
5. *Сидоров М.О., Мендзєбровський І.Б., Орехов А.А.* «Професійна практика програмної інженерії» - досвід викладання. - Інженерія програмного забезпечення. – 2010. – № 2. – С. 56 – 63.
6. *Petersen K., Feldt R., Maltsson M.* Systematic Mapping Studies in Software Engineering: In 12 th International Conference on Evaluation and Assessment in Software Engineering. – 2008. – P. 378–390.
7. *Kitchenha 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. – 2009. –Vol. 51, Issue 1.
8. *Kitchenham B.* Guidelines for performing Systematic Literature Reviews in Software Engineering. – 2007. – Vol 2. 3 EBSE Technical Report.
9. *Dyba T. Dingsoyr T.* Empirical studies of agile software development: A systematic review. Information and Software Technology. – 2008. – Vol. 50. – P. 833 – 859.
10. *Сидорова Н.Н.* Навчання інженерії програмного забезпечення – системний огляд літератури // Інженерія програмного забезпечення. 2011. – № 2 (6). – С. 56 – 67.