

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Кваліфікаційна наукова  
праця на правах рукопису

**ГУРА Олександр Олександрович**

УДК: 378.091.3:004.4-051:374

**ДИСЕРТАЦІЯ**

**ПІДГОТОВКА МАЙБУТНІХ ІНЖЕНЕРІВ-ПРОГРАМІСТІВ ДО  
ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ В УМОВАХ  
НЕФОРМАЛЬНОЇ ОСВІТИ**

Спеціальність: 015 Професійна освіта (за спеціалізаціями)

Галузь знань: 01 Освіта/Педагогіка

Подається на здобуття наукового ступеня доктора філософії

Дисертація містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело \_\_\_\_\_ О.О. Гура

Науковий керівник: Іваницький Олександр Іванович, доктор педагогічних наук, професор

Запоріжжя - 2021

## АНОТАЦІЯ

**Гура О.О. Підготовка майбутніх інженерів-програмістів до тестування програмного забезпечення в умовах неформальної освіти.**  
– *Кваліфікаційна наукова праця на правах рукопису.*

Дисертація на здобуття наукового ступеня доктора філософії за спеціальністю 015 Професійна освіта (за спеціалізаціями). - Запорізький національний університет, Запоріжжя, 2021.

У дисертації досліджено проблему професійної підготовки майбутніх інженерів-програмістів до тестування програмного забезпечення в умовах неформальної освіти. Розкрито й охарактеризовано сутність і зміст професійної готовності інженерів-програмістів до тестування програмного забезпечення, визначені науково-методичні засади її формування в умовах неформальної освіти.

Здійснений аналіз сучасних вимог до професійної діяльності та професійного розвитку інженерів-програмістів, їх кваліфікаційних категорій та функціональних обов'язків, змісту професійних завдань за різними методологіями і типами тестування на різних його етапах, а також результати проведеного серед фахівців з тестування ПЗ різних ІТ-компаній емпіричного дослідження, дали підстави для виокремлення таких специфічних особливостей професійної діяльності інженера-програміста з тестування ПЗ як: 1) первинна відповідальність за якість продукту на всіх етапах його розробки, що у свою чергу вимагає здійснення аналізу та проведення максимальної кількості перевірок ПЗ; 2) метасистемний характер – вимагає від тестувальника професійних компетенцій як програміста (мати досвід програмування, знати програмний код), так і бізнес-аналітика, менеджера проекту (знати його організаційну структуру, мати організаційні та комунікативні вміння) тощо; 3) необхідність здійснення аналізу продукту з декількох позицій: з точки зору замовника програмного продукту, розробника ПЗ, а також, - користувача продуктом;

4) творчий характер - вимагає постійного пошуку шляхів перевірки, оптимізації тестового покриття, прийняття рішень в умовах невизначеності.

Професійна готовність майбутніх інженерів-програмістів з тестування ПЗ визначена системним утворенням, інтегративним особистісно-професійним явищем, що є результатом їх цілеспрямованої, спеціально організованої професійної підготовки в умовах неперервної освіти, є передумовою професійної компетентності та визначає успішність їх подальшої професійної діяльності і професійного розвитку. Вона складається з таких компонентів як: 1) ціннісно-мотиваційний (визначається їх мотивацією професійної діяльності та подальшого їх професійного розвитку і навчання, ціннісним ставленням до виконання професійних обов'язків – тестування ПЗ, спрямованістю на досягнення командного результату); 2) когнітивний (віддзеркалює систему загальних та спеціальних професійних знань); 3) операційно-діяльнісний (складається з технічних - відображають завдання/етапи професійної діяльності тестувальника ПЗ та нетехнічних умінь - відображають вимоги до їх мисленнєвої, комунікативної, регулятивної та метапізнавальної діяльності.

Результати здійсненого аналізу зарубіжного та вітчизняного досвіду професійної підготовки, майбутніх-інженерів програмістів до професійної діяльності в цілому та до тестування ПЗ зокрема, контент-аналізу її нормативно-правового забезпечення, дали підстави стверджувати про таке: 1) її характерними ознаками в зарубіжних закладах вищої освіти є: забезпечення фундаментальної та технічної підготовки у гармонійній єдності з практичним навчанням – отримання студентами ґрунтового практичного досвіду в спеціалізованих сферах ІТ, у тому числі - у сфері тестування ПЗ; метапредметне спрямування – розширення змісту завдяки суміжним галузям та сферам діяльності (менеджмент, безпека, торгівля, медіа тощо); зорієнтованість навчальних форм і методів на виробничий

процес розробки ПЗ, що вимагає проведення тематичних досліджень, локальних та загальних квазіпрофесійних проєктних заходів та ін.; забезпечення цілеспрямованого розвитку нетехнічних професійних знань та умінь студентів у процесі групових, проєктних форм, командної роботи, навчальних та виробничих практик тощо; ефективне використання ресурсу неформальної освіти: обов'язковості проходження тривалої виробничої практики на ІТ-підприємствах; 2) у вітчизняних закладах вищої освіти спостерігається: домінування теоретичної, фундаментальної підготовки над практичною, що зумовлена у тому числі застарілістю матеріально-технічного та програмного забезпечення освітнього процесу; ригідність оновлення освітньо-професійних програм, що загострює протиріччя між потребою ІТ-індустрії у кваліфікованих фахівцях та неготовністю випускників закладів вищої освіти до роботи в умовах сучасних ІТ-компаній; недостатня зорієнтованість змісту навчання на іншомовну (передусім, англomовну) підготовку студентів; недостатній рівень співпраці закладів вищої освіти та роботодавців, що у поєднанні з проблемами матеріально-технічного та програмного забезпечення освітнього процесу, ускладнюють створення необхідних умов для практичної підготовки студентів; недостатня спрямованість на підготовку майбутніх фахівців до діяльності у спеціалізованих ІТ-сферах, у тому числі – тестування ПЗ.

Грунтуючись на положеннях: діалектичного матеріалізму, прагматизму та конструктивізму як філософської методології педагогічного дослідження; системного, синергетичного, інформаційного (цифрового) підходів, що є загальнонауковими методологіями; діяльнісного, контекстного, техніко-технологічного, компетентнісного, особистісного та метакогнітивного підходів, які є спеціально методологічними засадами, а також враховуючи авторську структуру професійної готовності майбутніх інженерів-програмістів до тестування

ПЗ, створена та обґрунтована структурно-функціональна модель їх професійної підготовки в умовах неформальної освіти.

Структурно-функціональна складається з трьох блоків: 1) цільового, що презентує мету та завдання професійної підготовки на двох рівнях – формальної та неформальної освіти; 2) змістовно-технологічного, який визначає принципи (загально педагогічні та спеціальні), зміст, етапи, форми та методи підготовки, та 3) результативного, що висвітлює її критерії (ціннісний, знаннєвий та вміннєвий), показники та рівні.

Обґрунтовано, що професійна підготовка майбутніх інженерів-програмістів до тестування ПЗ в умовах неформальної освіти набирає ефективності, якщо: розроблена і реалізується відповідна модель освітнього процесу, яка забезпечує інтеграцію та синхронізацію неформальної і формальної освіти; запроваджено спеціальну теоретичну підготовку, зорієнтовану на формування орієнтовної основи діяльності з тестування ПЗ; організовано практичну підготовку на засадах менторства; забезпечено цілеспрямований розвиток нетехнічних умінь (soft skills) майбутніх фахівців.

У ході експериментальної перевірки отримано статистично значущу різницю коефіцієнтів професійної готовності майбутніх інженерів-програмістів до тестування ПЗ експериментальних груп до та після впровадження педагогічних заходів, що засвідчує про ефективність визначених та запроваджених науково-методичних засад їх підготовки в умовах неформальної освіти.

Теоретично обґрунтовані й експериментально перевірені науково-методичні засади підготовки майбутніх інженерів-програмістів до тестування ПЗ в умовах неформальної освіти впроваджено в освітню практику на основі: 1) авторського курсу «Основи тестування програмного забезпечення»; 2) практичної підготовки на засадах індивідуального та групового менторства за навчально-адаптаційною і виробничою фазами; 3) тренінгової програми розвитку нетехнічних умінь («Тренінг-дизайн

мислення», «Тренінг професійної комунікації») та мітапів, що забезпечують актуалізацію мотиваційних ресурсів майбутніх фахівців щодо їх професійного саморозвитку в сфері ІТ.

*Ключові слова:* професійна підготовка, інженер-програміст, тестування програмного забезпечення, майбутній фахівець, неформальна освіта.

## SUMMARY

**Gura O. Preparation of future software engineers for testing in the terms of non-formal education.** - Qualifying scientific work on the rights of the manuscript.

The dissertation on competition of a scientific degree of the doctor of philosophy on a specialty 015 Professional education. - Zaporizhzhya National University, Zaporizhzhya, 2021.

The dissertation investigates the problem of professional preparation of future software engineers for testing in the terms of non - formal education. The essence and content of professional readiness of software engineers for testing are revealed and characterized, scientific and methodical bases of its formation in the conditions of non - formal education are defined.

The analysis of modern requirements to professional activity and professional development of software engineers, their qualification categories and functional duties, the maintenance of professional tasks on various methodologies and types of testing at its various stages, and also results of the carried-out empirical research among software testing experts of various IT - companies, gave grounds for highlighting such specific features of the professional activity of a software engineer as: 1) primary responsibility for product quality at all stages of its development, which requires analysis and maximum number of verifications; 2) metasystemic nature – it requires professional competencies of a programmer (have programming experience, know the program code), business analyst, project manager (know its

organizational structure, have organizational and communication skills), etc.; 3) the need to analyze the product from several positions: from the point of view of the customer of the product, software developer, a user of the product; 4) creative nature - requires a constant search for ways to test, optimize test coverage, make decisions in conditions of uncertainty.

The professional readiness of future software engineers for testing is determined by systemic education, integrative personal and professional phenomenon, which is the result of their purposeful, specially organized training in continuing education, is a prerequisite for professional competence and determines the success of their further professional activity and professional development, consists of such components as: 1) value-motivational (determined by their motivation for professional activity and their further professional development and training, value attitude to the performance of professional duties - software testing, focus on achieving team results); 2) cognitive (reflects the system of general and special professional knowledge); 3) active-operational (consists of technical - reflect the tasks / stages of professional activity of the software tester, and non-technical skills - reflect the requirements for their mental, communicative, regulatory and metacognitive activities).

The results of the analysis of foreign and domestic experience of professional training of future software engineers to professional activity in general and to testing in particular, as well as content analysis of its legal support, gave grounds to state the following: 1) its special features in foreign higher education institutions are: providing fundamental and technical training in harmonious unity with practical training - students gain a solid practical experience in specialized areas of IT, including the sphere of software testing; meta-subject direction - expansion of content due to related industries and areas of activity (management, security, trade, media, etc.); orientation of educational forms and methods on the production process of software development, which requires thematic research, local and general quasi-professional project

activities, etc.; ensuring the purposeful development of non-technical professional knowledge and skills of students in group, project forms, teamwork, training and production practices etc.; effective use of non-formal education resources: mandatory long-term internships at IT enterprises; 2) in domestic institutions of higher education there is: the dominance of theoretical, fundamental training over practical, which is due in part to the obsolescence of logistics and software of the educational process; rigidity of updating educational and professional programs, which exacerbates the contradiction between the need of the IT industry for qualified specialists and the unwillingness of graduates of higher education institutions to work in modern IT companies; insufficient focus of the content of education on foreign language (especially English) training of students; insufficient level of cooperation between higher education institutions and employers, which in combination with the problems of material and technical and software of the educational process, complicate the creation of the necessary conditions for the practical training of students; insufficient focus on training future professionals to work in specialized IT areas, including software testing.

Grounding on the provisions of: dialectical materialism, pragmatism and constructivism as a philosophical methodology of pedagogical research; system, synergetic, informational (digital) approaches, which are general scientific methodologies; activity, contextual, technical-technological, competence, personal and metacognitive approaches, which are specially methodological principles, as well as taking into account the author's structure of professional readiness of future software engineers to test software, the structural-functional model of their training in non-formal education has been created.

The structural-functional model consists of three blocks: 1) target, which presents the purpose and objectives of training at two levels - formal and non-formal education; 2) content-technological, which determines the principles (general pedagogical and special), content, stages, forms and methods of

preparation, and 3) effective, which highlights its criteria (value, knowledge and skills), indicators and levels.

It is substantiated that the professional training of future software engineers for software testing in non-formal education is gaining effectiveness if: an appropriate model of the educational process is developed and implemented, which provides integration and synchronization of non-formal and formal education; a special theoretical training focused on the formation of an indicative basis for software testing activities is introduced; a practical training on the basis of mentoring is organized; purposeful development of non-technical skills (soft skills) of future specialists is provided.

During the experimental research, a statistically significant difference in the coefficients of professional readiness of future software engineers for testing of experimental groups before and after the implementation of pedagogical measures, which indicates the effectiveness of defined and implemented scientific and methodological principles of their training in the terms of non-formal education. Theoretically substantiated and experimentally tested scientific and methodological principles of preparation of future software engineers for testing in the conditions of non-formal education are introduced into educational practice on the basis of: 1) author's course «Fundamentals of software testing»; 2) practical training on the basis of individual and group mentoring at the adaptation and production phases; 3) training program for the development of non-technical skills («Thinking Design Training», «Professional Communication Training») and metaphors that provide updating of motivational resources of future professionals for their professional self-development in the field of IT.

*Key words:* professional training, software engineer, software testing, future specialist, non-formal education.

## СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА

### Наукові праці, в яких опубліковані основні наукові результати дисертації

#### *Статті в наукових виданнях, включених до міжнародних наукометричних баз Scopus та Web of Science*

1. Gura Jr. O., Gura O., Gura T., Chernikova L. Research on metacognitive skills of software testers: a problem statement. *Proceedings of the 16th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer. Volume II: Workshops. Part IV: V International Workshop on Professional Retraining and Life-Long Learning using ICT: Person-oriented Approach (3L-Person).* Kharkiv, Ukraine, October 06-10, 2020, 607-618. (*Scopus*)

#### *Статті в періодичних наукових виданнях інших держав*

2. Гура О. Аналіз зарубіжного досвіду впровадження системи дуальної підготовки майбутніх фахівців. *European Humanities Studies: State & Society.* 2017. Issue 4. С. 31-43. (*Index Copernicus*)

3. Гура О. Матриця компетенцій (skill matrix) як орієнтир підготовки майбутніх ІТ-фахівців. *European Humanities Studies: State & Society.* 2018. Issue 6. С. 181-191. (*Index Copernicus*)

#### *Статті в наукових фахових виданнях України*

4. Гура О. Основні сфери реалізації фахівців ІТ на сучасному ринку праці: аналіз професійних вимог. *Науковий вісник Миколаївського національного університету імені В. О. Сухомлинського. Педагогічні науки.* 2017. Вип. 2 (57). С. 166-170. (*Index Copernicus*)

5. Гура О. Особливості розвитку м'яких навичок студентів ІТ-спеціальностей засобами навчальних SCRUM проєктів. *ScienceRise: Pedagogical Education.* 2019. №4(31). С. 8-15. (*Index Copernicus*)

6. Гура О. Підготовка інженерів-програмістів до тестування програмного забезпечення в умовах неформальної освіти: проблеми та

шляхи реалізації. *Педагогічні науки: теорія, історія, інноваційні технології: наук. журнал* / голов. ред. А.А. Сбруюва. Суми: Вид-во СумДПУ імені А.С.Макаренка. 2020. № 7 (101). С. 55-62. (*Index Copernicus*)

### ***Статті в інших наукових виданнях***

7. Гура О. Особливості співбесіди як форми контролю якості освіти студентів ІТ спеціальностей. *Педагогічні науки та освіта*. 2016. Вип. 18-19. С. 29-34.

### ***Матеріали конференцій***

8. Гура О. Актуальні проблеми професійної підготовки фахівців ІТ-сфери. *Електронний збірник наукових праць ЗОІППО*. 2016. № 3(25). URL: <https://drive.google.com/file/d/0B6QknaQCh-IHTFpkeVpSeHhBdGM/view>

9. Гура О. Особливості основних сфер реалізації ІТ фахівця на сучасному ринку праці. *Електронний збірник наукових праць ЗОІППО*. 2017. Вип. 1(27). URL: <https://drive.google.com/file/d/0B6QknaQCh-IHWGtSWEhiYXJGTIU/view>

10. Гура О. Особливості впровадження системи дуальної підготовки майбутніх фахівців: порівняльний аналіз. *Педагогіка вищої школи: досвід і тенденції розвитку*. Матеріали III Всеукраїнської науково-практичної конференції з міжнародною участю. Запоріжжя: Запорізький Національний Університет, 2018. С. 41-42.

11. Гура О. Професійна підготовка фахівців ІТ сфери на засадах матриці компетенцій (skills matrix). *Україна в гуманітарних та соціально-економічних вимірах*. Матеріали III Всеукраїнської наукової конференції (30-31 березня 2018р. м. Дніпро). ч. II. Наук.ред. О.Ю. Висоцький. Дніпро: «Охотник», 2018. 268 с. С. 103-105.

## ЗМІСТ

ВСТУП		13
РОЗДІЛ 1.	НАУКОВО-ТЕОРЕТИЧНІ ОСНОВИ ПРОФЕСІЙНОЇ ПІДГОТОВКИ МАЙБУТНІХ ІНЖЕНЕРІВ-ПРОГРАМІСТІВ ДО ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	21
1.1.	Особливості професійної діяльності та професійного розвитку інженерів-програмістів у сучасній ІТ-галузі	21
1.2.	Сутність та зміст професійної готовності інженерів-програмістів до тестування програмного забезпечення	43
1.3.	Зарубіжний та вітчизняний досвід професійної підготовки майбутніх інженерів-програмістів до тестування програмного забезпечення	74
	Висновки до першого розділу	97
РОЗДІЛ 2.	НАУКОВО-МЕТОДИЧНІ ЗАСАДИ ПРОФЕСІЙНОЇ ПІДГОТОВКИ МАЙБУТНІХ ІНЖЕНЕРІВ-ПРОГРАМІСТІВ ДО ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ В УМОВАХ НЕФОРМАЛЬНОЇ ОСВІТИ	101
2.1.	Особливості сучасної неформальної освіти фахівців ІТ-галузі	101
2.2.	Структурно-функціональна модель професійної підготовки майбутніх інженерів-програмістів до тестування програмного забезпечення в умовах неформальної освіти	112
2.3.	Спеціально організована теоретична підготовка майбутніх інженерів-програмістів до тестування програмного забезпечення	132
2.4.	Практична підготовка майбутніх інженерів-програмістів з тестування програмного забезпечення на засадах менторства	149
2.5.	Розвиток нетехнічних умінь (soft skills) майбутніх інженерів-програмістів до тестування програмного забезпечення	171
	Висновки до другого розділу	182
РОЗДІЛ 3.	ЕКСПЕРИМЕНТАЛЬНА ПЕРЕВІРКА ЕФЕКТИВНОСТІ НАУКОВО-МЕТОДИЧНИХ ЗАСАД ПРОФЕСІЙНОЇ ПІДГОТОВКИ МАЙБУТНІХ ІНЖЕНЕРІВ-ПРОГРАМІСТІВ ДО ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ В УМОВАХ НЕФОРМАЛЬНОЇ ОСВІТИ	185
3.1.	Організація та хід педагогічного експерименту	185
3.2.	Аналіз результатів педагогічного експерименту	199
	Висновки до третього розділу	212
ВИСНОВКИ		214
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ		219
ДОДАТКИ		245

## ВСТУП

**Актуальність теми.** Стрімкий розвиток інформаційно-комунікаційних технологій, IT-індустрії України, яка значно випереджає середні темпи розвитку сегменту у світі, її фінансова значущість в масштабах економіки держави, що зумовлює залучення до неї величезної кількості молодих і креативних фахівців, призвела до появи відокремленої професійної спільноти IT-фахівців - тестувальників програмного забезпечення. Це в свою чергу, ставить нові виклики перед освітньою системою країни і зумовлює нагальну модернізацію змісту, форм і методів професійної підготовки майбутніх інженерів-програмістів в напрямках: метапредметної спрямованості, розширення змісту навчання завдяки суміжним галузям та сферам діяльності (менеджменту, маркетингу, дизайну та ін.); практичної спрямованості навчального процесу, його зорієнтованості на виробничий процес розробки та тестування програмного забезпечення; використання потенціалу інформальної та неформальної освіти, що знаходить своє відображення на нормативному рівні (Закони України «Про освіту», «Про вищу освіту», «Про фахову передвищу освіту», «Про державну підтримку розвитку індустрії програмної продукції», Експортна стратегія для сектору інформаційних технологій на 2019-2023 роки, Концепція розвитку цифрової економіки та суспільства України на 2018-2020 роки та ін.).

На сьогодні саме неформальна освіта є одним з основних механізмів, який може забезпечити стратегічну співпрацю бізнесу та системи вищої IT-освіти не тільки в контексті покращення матеріально-технічного забезпечення, але й організаційного та змістовного забезпечення підготовки висококваліфікованих IT-фахівців різних спеціалізацій, серед яких однією з найбільш актуальних та перспективних є тестування програмного забезпечення. З огляду на це, набуває нагальної потреби розробка науково-методичного забезпечення системи неформальної освіти майбутніх інженерів-програмістів, яке б створило необхідні умови для

формування їх професійної готовності тестування програмного забезпечення.

Актуальність окресленої проблеми й необхідність її вирішення, крім того, зумовлені наявними суперечностями між:

- зростаючими обсягами запиту ринку праці на фахівців ІТ-спеціальностей, визначеного рівня та неспроможністю вітчизняної освітньої системи забезпечити ці потреби повною мірою;

- швидкими темпами розвитку сучасних інформаційно-комунікаційних технологій, і, відповідно, вимог до організації та змісту професійної підготовки фахівців ІТ-сфери та ригідністю закладів вищої освіти щодо оперативної модернізації навчального процесу;

- об'єктивною потребою ІТ-галузі, системи професійної підготовки фахівців ІТ-спеціальностей у використанні всього потенціалу інформальної та неформальної освіти й недостатньою розробленістю її методології та методики.

Проблема підготовки майбутніх інженерів-програмістів, формування їх професійної готовності до тестування програмного забезпечення в психолого-педагогічній науці знайшла відображення при вирішенні широкого кола теоретичних і прикладних питань, пов'язаних з: упровадженням компетентнісного підходу в освіті в цілому (І. Бех [13], О. Божович [127], О.І. Гура [28], А. Деркач [39], В. Краєвський [72], О. Овчарук [104] та ін.) та ІТ-освіті зокрема (О. Кучерук [83], С. Літвінова [90], В. Осадчий [80], В. Седов [141], Д. Щедролосьєв [171] та ін.); цифровізації освіти та сучасними технологіями відкритої освіти (В. Биков [14], О. Спирін [14] та ін.); аналізом стану і перспектив формування системи неформальної освіти (І. Ахновська, Т. Вакалюк [128], С. Бабушко [7], А. Боярська-Хоменко [19], О. Глазунова [23], В. Горленко [26], А. Гуржій [23], В. Єлагін [44], В. Корольчук [23], О. Огієнко [106], Н. Павлик [114], О. Пархоменко [23], І. Рантюк [128], Н. Терьохіна [157], Ю. Шустак [170] та ін.); дослідженням особливостей ІТ-сфери та

професійної діяльності інженерів-програмістів (Г. Бабій [5], Е. Бейч [9], Л. Гришко [124], Н. Длугунович [40], П. Павленко [113], С. Пермінова [117], С. Попершняк [119], В. Седов [141], С. Сухоняк [154], С. Щедролосьєв [172], О. Щербатенко [173] та ін.), у тому числі - в сфері тестування програмного забезпечення (Т. Баджетт [92], Б. Бейзер [8], Г. Кравченко [71], С. Куліков [81], Г. Майерс [92], Дж. Макгрегор [93], Д. Сайкс [93], F. Ahmed, etc. [190], I. Burnstein [176], K. Beck, etc. [175], L. Crispin [181], D. Dobbs [202], J. Gregory [181], J. Glenford Myers [192], W. Lewis [202], G. Veerapillai [202], G.M. Weinberg [219] та ін.); вивченням зарубіжного та вітчизняного досвіду професійної підготовки інженерів-програмістів (О. Бабкин [6], А. Коломієць [67], С. Конюхов [70], В. Круглик [78], І. Крашеніннік [74], Г. Лебедь [85], О. Наумчук [102], С. Попершняк [119], Я. Сікора [143], Б. Якимчук [143] та ін.) та ін.

Разом з тим, результати здійсненого теоретичного аналізу свідчать про те, що не зважаючи на значну кількість наукових досліджень, присвячених вивченню різних аспектів професійної діяльності фахівців ІТ-галузі та їх професійної підготовки, питання підготовки майбутніх інженерів-програмістів до тестування програмного забезпечення в умовах неформальної освіти не отримали достатньої уваги, що і зумовило вибір теми дослідження

**Зв'язок роботи з науковими програмами, планами, темами.** Дисертаційна робота виконана відповідно до плану наукових досліджень кафедри педагогіки та психології освітньої діяльності Запорізького національного університету за темою «Психолого-педагогічні засади розвитку компетентності суб'єктів освітнього простору» (номер державної реєстрації 0116U004863), одним з виконавців якої є здобувач. Тема дисертації затверджена Науково-технічною радою Запорізького національного університету (протокол № 7 від 26.12.2019).

**Мета й завдання дослідження.** Мета дослідження полягає в науковому обґрунтуванні науково-методичних засад, що забезпечують

успішність підготовки майбутніх інженерів-програмістів до тестування програмного забезпечення в умовах неформальної освіти.

Відповідно до цієї мети поставлено такі *завдання*:

– визначити особливості професійної діяльності майбутніх інженерів-програмістів, виявити й охарактеризувати складові їх готовності до тестування програмного забезпечення;

– розкрити зарубіжний та вітчизняний досвід підготовки майбутніх інженерів-програмістів до тестування програмного забезпечення;

– розробити й обґрунтувати науково-методичні засади, що забезпечують успішність підготовки майбутніх інженерів-програмістів до тестування програмного забезпечення в умовах неформальної освіти та змодельовати цей процес;

– експериментально перевірити ефективність розроблених науково-методичних засад підготовки майбутніх інженерів-програмістів до тестування програмного забезпечення в умовах неформальної освіти.

*Об'єкт дослідження* – професійна підготовка майбутніх інженерів-програмістів у закладах вищої освіти.

*Предмет дослідження* – науково-методичні засади формування готовності майбутніх інженерів-програмістів до тестування програмного забезпечення в умовах неформальної освіти.

В основу дослідження покладено *припущення* про те, що формування готовності майбутніх інженерів-програмістів до тестування програмного забезпечення в умовах неформальної освіти набуває ефективності, якщо: розроблена і реалізується відповідна модель освітнього процесу, яка забезпечує інтеграцію та синхронізацію неформальної і формальної підготовки; реалізована спеціально організована теоретична підготовка, спрямована на формування орієнтовної основи діяльності студентів із тестування програмного забезпечення; організовано їх практичну підготовку на засадах менторства; забезпечено розвиток системи нетехнічних умінь (soft skills).

**Методи дослідження.** Для досягнення мети, вирішення завдань і перевірки гіпотези дослідження використано комплекс сучасних загальнонаукових методів:

– *теоретичних*: аналіз (історичний, порівняльний), зіставлення та узагальнення різних поглядів на досліджувану проблему, систематизація наукової та навчально-методичної літератури; моделювання – для визначення і уточнення змісту основних понять дослідження, з’ясування сутності готовності майбутніх інженерів-програмістів до тестування програмного забезпечення, обґрунтування науково-методичних засад її формування в умовах неформальної освіти, розробки моделі освітнього процесу;

– *емпіричних*: методи діагностики (бесіда, анкетування, тестування, експертне оцінювання), контент-аналіз нормативної та навчально-методичної документації (навчальних планів, освітньо-професійних програм, стандартів) – для визначення результатів підготовки та рівня сформованості готовності майбутніх інженерів-програмістів; особливостей організації підготовки в умовах неформальної освіти; педагогічний експеримент – для перевірки ефективності розроблених науково-методичних засад;

– *статистичних*: метод середніх величин, метод кореляційного аналізу, факторний аналіз, а також методи аналізу емпіричних даних за допомогою сучасних інформаційних технологій (IBM SPSS Statistics) – для опрацювання та оцінювання отриманих експериментальних даних.

**Наукова новизна одержаних результатів** полягає в тому, що:

– *вперше розроблено й апробовано* науково-методичні засади, що забезпечують ефективність формування готовності майбутніх інженерів-програмістів до тестування програмного забезпечення в умовах неформальної освіти: структурно-функціональну модель освітнього процесу, що забезпечує інтеграцію та синхронізацію неформальної і формальної підготовки на рівні цільового, змістовно-технологічного і

результативного компонентів; авторський курс «Основи тестування програмного забезпечення», зорієнтований на формування орієнтовної основи для подальшої професійної діяльності з тестування програмного забезпечення; зміст та технологію організації практичної підготовки на базі ІТ-компанії на засадах менторства; розвиток нетехнічних умінь завдяки впровадженню тренінгів розвитку мисленнєвих та метакогнітивних умінь, комунікативних та організаційних здібностей;

– *уточнено* зміст понятійно-категоріального апарату теорії професійної підготовки майбутніх інженерів-програмістів в умовах неформальної освіти до тестування програмного забезпечення («неформальна освіта», «професійна готовність до тестування програмного забезпечення», «менторство», тощо); критерії (ціннісний, знаннєвий та вміннєвий), показники й рівні сформованості професійної готовності майбутніх інженерів програмістів до тестування програмного забезпечення;

– *удосконалено* організаційно-методичні засади професійної підготовки майбутніх інженерів-програмістів шляхом синхронізації та інтеграції змісту (курс, практика, менторство, тренінги) і результатів їх навчання (достатній рівень сформованості професійної готовності) в умовах неформальної освіти;

– *набули подальшого розвитку* науково-теоретичні положення про сутність професійної готовності майбутніх інженерів-програмістів до тестування програмного забезпечення, етапи, форми і методи її формування в системі неформальної освіти.

**Практичне значення одержаних результатів** зумовлено тим, що розроблено, апробовано та впроваджено систему неформальної професійної підготовки майбутніх інженерів-програмістів до тестування програмного забезпечення, складовими якої є: структурно-функціональна модель освітнього процесу; авторський курс «Основи тестування програмного забезпечення» (робоча програма, матеріали до лекційних і

практичних занять, самостійної роботи); матеріали лекційно-консультаційних занять, які відбувались в процесі практичної підготовки (за темами: «Тестування користувацького інтерфейсу веб-додатків», «Тестування сервісного шару веб-додатків», «Автоматизація тестування веб-додатків», «Тестування мобільних додатків»); методичне забезпечення практичної підготовки на засадах менторства (програма практичної підготовки на засадах менторства, «Рефлексивний щоденник», завдання для практичної роботи, програма співбесіди); програми Тренінгу-дизайн мислення, Тренінгу професійної комунікації; комплекс методик діагностики рівня сформованості компонентів професійної готовності.

Теоретичні положення, авторські науково-методичні розробки та висновки дослідження створюють основу для вдосконалення професійної підготовки майбутніх інженерів-програмістів.

Результати дослідження *впроваджено* в освітній процес Запорізького національного університету, Національного технічного університету «Харківський політехнічний інститут», Харківського національного економічного університету імені Семена Кузнеця, Хмельницької гуманітарної педагогічної академії, а також у кадрову та рекрутингову діяльність Товариства з обмеженою відповідальністю «ЕПАМ СИСТЕМЗ».

**Особистий внесок здобувача.** Основні положення й висновки, викладені в дисертації, належать здобувачу особисто та відображені в публікаціях. У праці, написаній у співавторстві («Research on metacognitive skills of software testers: a problem statement»), автору належать: обґрунтування методології та результатів здійсненого емпіричного дослідження.

**Апробація результатів дослідження.** Основні положення та результати дисертації були оприлюднені на науково-практичних конференціях:

- *всеукраїнського рівня*: «Науково-методичні засади професійного розвитку фахівця у системі неперервної освіти» (Запоріжжя,

2016), «Неперервна освіта нового сторіччя: досягнення та перспективи» (Запоріжжя, 2017); «Педагогіка вищої школи: досвід і тенденції розвитку» (Запоріжжя, 2018); «Україна в гуманітарних та соціально-економічних вимірах» (Дніпро, 2018);

- *міжнародного рівня*: «Вища освіта України у контексті інтеграції до Європейського освітнього простору» (Київ, 2017, 2018); International Workshop on Professional Retraining and Life-Long Learning using ICT: Person-oriented Approach (3L-Person) (Kharkiv, 2020).

**Публікації.** Основні наукові положення й результати дослідження викладено в 11 друкованих працях, з них: 1 – стаття в зарубіжному науковому виданні, що індексується в SCOPUS; 2 – статті в періодичних наукових виданнях інших держав; 3 – статті в наукових фахових виданнях України, які входять до міжнародних наукометричних баз; 1 – стаття в іншому науковому виданні; 4 – матеріали конференцій.

**Структура та обсяг дисертації.** Дисертація складається зі вступу, трьох розділів, висновків, 8 додатків, списку використаних джерел (221 найменування, з них 51 – іноземною мовою). Загальний обсяг роботи становить 326 сторінок, з них основного тексту – 192 сторінки. Дисертація містить 12 таблиць і 17 рисунків.

## РОЗДІЛ 1.

# НАУКОВО-ТЕОРЕТИЧНІ ОСНОВИ ПРОФЕСІЙНОЇ ПІДГОТОВКИ ІНЖЕНЕРІВ-ПРОГРАМІСТІВ ДО ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 1.1. Особливості професійної діяльності та професійного розвитку інженерів-програмістів у сучасній ІТ-галузі

Дослідження професійної підготовки інженерів-програмістів до тестування програмного забезпечення (далі - ПЗ) доцільно розпочати з визначення загальних особливостей їх професійної діяльності, яка інтенсивно розгортається та утверджується серед інших сфер людської діяльності протягом останніх 50 років.

Насьогодні ринок ІТ-послуг в Україні називають одним з найбільш швидкозростаючих в Європі [136]. Однак, розвиток і формування сфери інформаційних технологій в нашій країні почалося в радянські роки. Розглянемо основні історичні вехи розвитку ІТ-сфери в Україні, адже вони визначають як особливості професійної діяльності вітчизняних інженерів-програмістів, так і специфіку їх професійного розвитку та фахової підготовки. Зазначимо, що історичні аспекти розвитку ІТ-сфери в Україні присвятили свої наукові дослідження І. Крилов, Б. Малиновський, О. Щербатенко та ін.

Отже, ще в п'ятдесяті роки 20 століття на території України відкривається Інститут напівпровідників НАН УРСР, в якому Є. Лашкар'ов описує транзисторний ефект і можливі сфери його застосування, що і стало основою розвитку комп'ютерної науки того часу. Пізніше на території України під керівництвом академіка С. Лебедева створена одна з перших в Європі електронних обчислювальних машин, на основі якої створені ЕОМ вже наступних поколінь, широко застосовуються у військовій промисловості (Діана 1 і Діана 2) і виробництві (Дніпро) [84].

Проголошення незалежності суттєво вплинуло на зміну більшості сфер і галузей людської діяльності в Україні, в тому числі і ринку інформаційних технологій (далі - ІТ), який став відкритим для зовнішньої конкуренції, в той час як державні науково-дослідні інститути вкрай швидко приходили в занепад через нестачу фінансування і відсутність можливості конкурувати із західними продуктами [173]. Однак, цей період можна назвати епоєю зародження сучасних ІТ-підприємств в Україні - студенти і співробітники вищезазначених інститутів організовують власні ІТ-компанії і продовжують працювати в своїй сфері, отримуючи при цьому доступ як до технічних розробок інших країн, так і опановуючи зарубіжні практики ведення бізнесу. У цей період почали свою діяльність Miratech, SoftServe, NIX solutions [173] – ІТ-компанії, які входять у рейтинг найбільших та найприбутковіших на території України.

У XXI тисячолітті технології стають більш доступними і масовими. Освітні заклади відкривають нові актуальні програми підготовки, а ринок праці наповнюється високопрофесійними технічними фахівцями. Як зазначає О. Щербатенко, не маючи довгої виробничої історії та відповідного досвіду, українські компанії все ще не здатні повноцінно конкурувати на ринку інформаційних технологій із зарубіжними, проте велика кількість перспективних кадрів і нестабільний економічний стан привертає західний бізнес і сприяє відкриттю офісів і студій в великих містах України. До числа таких компаній відносяться Oracle, Eram, Dataart, Sigma та інші [173]. Розвитку галузі також сприяла державна політика: у 2011 році були проведені парламентські слухання на тему «Створення в Україні сприятливих умов для розвитку індустрії програмного забезпечення», а нормативно-правова база на сьогодні нараховує понад 20 законів і указів президента [136].

Отриманий досвід сприяє відкриттю українських компаній. У той же час з'явилася мода на маленькі стартапи, яка зміцнила вітчизняний ІТ-ринок та наповнила його власне українськими продуктами, зорієнтованими

як на вітчизняного, так і на зарубіжного споживача, до яких відносяться GitLab, Grammarly, People.ai [136]. У той же час, ще більших масштабів досягає розвиток аутсорс-напрямів в українському ІТ: вже існуючі компанії нарощують штат і відкривають офіси по всій Україні, при цьому нові компанії продовжують заходити на ще вільний від конкуренції зарубіжних фахівців ринок.

Як свідчать дані статистики, на сьогоднішній день, розвиток ІТ-індустрії в Україні значно випереджає середні темпи розвитку сегменту у світі. Відносно молода для країни галузь, крім безпосереднього економічного ефекту, сьогодні стала і важливим елементом створення сучасного іміджу держави : провідна міжнародна організація Global Sourcing Association відзначила Україну в якості найкращої країни - постачальниці ІТ-послуг до Великої Британії, а авторитетний американський бізнес-журнал Inc. включив українські компанії до переліків таких, що розвиваються найбільш динамічно [4].

За неофіційними даними, в Україні на сьогодні - близько 4 тисяч ІТ-компаній, більшість із яких мають до 80 співробітників, однак чимало фахівців працюють у фірмах, що налічують понад 80 осіб персоналу. З них кількість компаній, активних на ринку праці – більше 2 300. Приблизно 70% з них надають ІТ-послуги широкому колу клієнтів (EPAM, GlobalLogic, Netcracker та ін.), близько 15% - працюють як Global In-house Center (GIC) для однієї материнської компанії (Wargaming.net, Ring, Samsung R&D Institute Ukraine, Oracle) та 15% створюють власний продукт (Genesis, EVO, Terrasoft).

Окрім того, значна кількість ІТ-спеціалістів співпрацюють з компаніями як фізичні особи-підприємці (далі - ФОП) і самостійно сплачують податки. Абсолютна більшість з них перебуває на спрощеній системі оподаткування і сплачує єдиний податок. Обсяг його надходжень зростав у середньому на 58,8% протягом 2013-2017 рр. і становив 3.2 млрд грн у 2017. Слід також зазначити, що єдиний податок, який сплачують

ФОП, на 100% залишається в місцевих бюджетах за місцем реєстрації, тоді як ПДФО розподіляють серед місцевих бюджетів різного рівня на 75%, а податок на прибуток на 90% надходить до центрального. Також можна сказати, що працівники ІТ-сфери сприяють наповненню місцевих бюджетів і розвитку регіонів. Так, наприклад, надходження єдиного податку від представників ІТ-галузі у м. Київ у 2017 році становили лише 28% від загальних, і ще 10 областей перевищили планку в 2%, що свідчить про більш рівномірний розподіл, ніж, наприклад, податку на прибуток [136].

Популярність й актуальність ІТ-сфери, її фінансова значущість в масштабах економіки держави і залучення значної кількості молодих і креативних фахівців призвела й до появи відокремленого суспільства, професійної спільноти ІТ, яка за рахунок неформального спілкування зміцнює і розвиває саму сферу. Відбувається поява так званих ІТ-кластерів, об'єднань програмістів або навіть компаній одного регіону для просування спільних проєктів, організації технічних або розважальних тематичних заходів та акцій благодійності. До них відносяться ІТ Dnipro Community, Kyiv IT Cluster, Kharkiv IT Cluster та ін. Так само з'являються й «інкубатори» - організації, що забезпечують підтримку і стартове фінансування молодих і перспективних стартапів, такі як Open Data Incubator, Concepter, GrowthUP. Щорічно на території України проводиться значна кількість міжнародних технічних конференцій різних спрямувань, в яких беруть участь тисячі студентів та фахівців (популярна українська конференція iForum в 2018 році зібрала понад 12000 активних слухачів з усього світу [54]).

Значне економічне і культурне зростання, що відбувається в ІТ-сфері, визначає й активний розвиток окремих ІТ-спеціалізацій, зокрема, тестування. З'являються платформи і для QA (quality assessment)-орієнтованих стартапів. Наприклад, TestUA - спільнота, яка проводить «тестатони» - хакатони для тестувальників ПЗ. Учасники ставлять перед

собою нові завдання, отримують відгуки та накопичують свій досвід. Технологічні продукти різного ступеня зрілості можуть пройти тестування своїх продуктів безкоштовно. На щорічній основі проводиться і найбільша в СНД конференція з тестування - QAFest, яка запрошує спікерів з усього світу і стала майданчиком для анонсів нових технологій і інструментів тестування.

Підводячи підсумки, можна з упевненістю сказати, що на сьогодні ІТ-індустрія України знаходиться на етапі свого інтенсивного розвитку, зміцнюючись в регіональній та загальнодержавній економіці, нарощуючи штат співробітників, сфери своєї інтеграції та впливу.

Зростання сфери ІТ в світі в цілому і в Україні зокрема призвело до розширення її функцій в суміжні галузі виробництва, що, в свою чергу, значно вплинуло як на організаційну структуру ІТ-підприємств, так і на особливості діяльності фахівців і вимоги до їх підготовки.

Крім класичної розробки комерційного програмного забезпечення для кінцевого споживача або внутрішніх потреб засобами вже існуючої промислової компанії (ІТ-компанії), з'являється і така новітня організаційна форма надання ІТ-послуг як стартап, що визначається сучасними вченими як процес реалізації абсолютно молодого проєкту за короткий термін з мінімальними капіталовкладеннями [155]. Основні характеристики цієї форми надання ІТ-послуг закладені в самому визначенні: «старт» - як початок, «ап» - як швидкий підйом, зростання.

Основними характеристиками стартапів є: наявність креативної ідеї, підприємницька ініціатива, короткий термін реалізації ідеї, мінімальні обсяги інвестицій, продовження у формі нового підприємства. Стартапи як елемент ринкової інфраструктури виконують дедалі більшу роль. Великі фірми є в середньому більш продуктивними, ніж дрібні, особливо в виробничому секторі, але деякі невеликі фірми часто перевершують великі компанії, використовуючи конкурентні переваги в високобрендових або високоінтелектуальних нішах діяльності з інтелектуальною власністю.

Важливим фактором розвитку стартапів в сучасних умовах є саме швидкий розвиток ІТ та цифровізації економіки. [63]. Незважаючи на те, що на сьогоднішній день Україна значно відстає в інноваційному розвитку та підтримки діяльності інноваційних підприємств, зокрема стартапів. Пов'язано це з незадовільним станом інноваційної інфраструктури, відсутністю дієвих механізмів підтримки інноваційного підприємництва. Проте, зростання цієї форми організації компаній все ще залишається колосальним, а тільки в 2018 році в українські стартапи було інвестовано понад 280 мільйонів доларів, що можна порівняти з сумарними показниками відповідних інвестицій за 2013-2016 роки [136].

Ще одним популярним сегментом ринку ІТ-послуг є консалтинг, що з'явився майже одночасно із самою сферою ІТ. Згідно з визначенням, сформульованим українськими вченими-економістами, консалтинг являє собою індивідуальну підприємницьку діяльність, основою якої є комунікаційний процес передачі знань від консультанта до замовника послуг. Базова передумова консалтингу - необхідність підвищення ефективності існуючих бізнес-процесів не тільки під час кризових явищ, а й у фазі зростання [129]. До основних причин стрімкого розвитку сфери консультування і зростання попиту на консалтингові послуги на світовому ринку відносять такі: - загальну тенденцію глобалізації бізнесу, що однаково сприяє попиту на консалтингові послуги і серед транснаціональних корпорацій, які захоплюють нові ринки, і серед щойно створених компаній, які сподіваються позиціонуватися і зміцнитися; - можливість використання ідей і таланту консультанта як конкурентну перевагу в ринковій боротьбі; - необхідність впровадження ІТ, які дозволяють в значній мірі підвищити продуктивність компанії [9]. Послуги консультантів можуть бути у таких формах як: аналітична діяльність (аналіз і оцінка діяльності підприємства, аналіз інвестиційних проектів і ін.); прогнозування; консультування, аудит діяльності підприємства (перевірка на відповідність певним стандартам і надання рекомендацій

щодо усунення виявлених проблем); участь в діяльності підприємства (стратегічне планування, організація управління, розробка і впровадження інформаційних систем, системна інтеграція). Результат консалтингу, ефект, який може отримати підприємство, у значній мірі залежить від самого підприємства та від кваліфікації консалтингової компанії. Тому, важливим є врахування таких стандартних вимог, що пред'являються до консалтингових компаній, як: наявність відпрацьованої технології вирішення завдань; незалежність від постачальників програмних продуктів або комп'ютерної техніки та мережевого обладнання; незалежність від об'єкта консультування; надання допомоги клієнтам у використанні їх власного досвіду для безперервного вдосконалення своєї діяльності; наявність багатьох клієнтів (з метою накопичення, аналізу, переробки та використання досвіду); наявність системи навчання клієнтів під час проєктів. Аналіз стану ринку консалтингу в Україні та тенденцій його розвитку дозволяє констатувати обсяг національного ринку консалтингу у межах 0,46% ВВП (585 млн дол.), що в цілому відповідає середньоєвропейському рівню, а динаміка ринку консалтингу в Україні є позитивною і становить 14% щорічно [129].

Одним з найбільш прибуткових і поширених сегментів надання ІТ-послуг на території України є напрям аутсорсингу, який сучасні вчені і практики ІТ-індустрії характеризують як передачу окремих дій і функцій постачальникам послуг в тому випадку, коли останні здатні гарантувати необхідні параметри якості та результативності виконання з можливістю передачі, у тому числі, й частини працівників підприємства замовника [49]. За своїми типами і видами делегованих діяльностей, аутсорсинг диференціюється на:

- 1) цільовий - описує спектр основних цілей і завдань, які досягаються за допомогою використання аутсорсингу в діяльності підприємств. Серед найбільш поширених цілей - пошук відповідних ресурсів, зниження собівартості, досягнення високого рівня гнучкості;

2) процесний - передача певних (як правило, вторинних) функцій, окремих завдань або етапів виробничо-господарської діяльності, не цікавих для підприємства з огляду на його основні цілі, постачальникам товарів і послуг;

3) стратегічний - удосконалення наявної корпоративної або ділової стратегії, окремих функціональних або операційних стратегій, або вироблення на основі аутсорсингу абсолютно нової стратегії для отримання конкурентних переваг та досягнення вищого рівня конкурентоспроможності на ринку;

4) стейкхолдерський - знаходження за допомогою аутсорсингу нового балансу інтересів між зацікавленими групами (стейкхолдерами) в діяльності підприємства через залучення постачальників, конкурентів, споживачів або інших стейкхолдерів до побудови загальних бізнес-процесів [154].

Сам по собі аутсорсинг характерний для багатьох галузей бізнесу, проте особливо яскраво він проявляється в умовах української сфери ІТ, у просторі якої найпопулярнішою формою аутсорсингу є «ніаршорінг» (від англ. «near» - поруч) – це передача процесів на аутсорсинг в сусідню країну, переважно в безпосередній географічній близькості, що допомагає частіше подорожувати і проводити особисті зустрічі з меншими витратами, а культурна сумісність знижує ймовірність непорозуміння і полегшує координацію роботи.

Державна політика планової інтеграції України в Європейський союз, зручне розташування, відсутність радикальних культурних і релігійних особливостей, поширене знання іноземних мов та наявна науково-технічна база, в поєднанні з низьким рівнем заробітних плат, робить Україну вкрай бажаною платформою для аутсорс-відносин з Європою. Це виражається в величезній кількості представлених на ринку великих ІТ-компаній, до яких відносяться EPAM Systems (понад 6500 співробітників), Global Logic Ukraine (більше 4000 співробітників), Luxsoft

(більше 4000 співробітників) та ін., а сумарна капіталізація цього ринку становить понад 3,6 мільярди доларів [136].

Ще одним перспективним напрямком реалізації ІТ-сфери є «Research and development» (далі - R&D) (від англ. «дослідження і розвиток») – це набір інноваційних дій, що здійснюються компанією для розробки нових послуг або продуктів і поліпшення існуючих [197]. R&D процеси відокремлені від більшості основних видів діяльності, виконуваних компанією. Дослідження та/або розробки, як правило, не передбачають негайного прибутку. Замість цього очікується, що це буде сприяти довгостроковій прибутковості компанії, а дослідницькі активності можуть призвести до патентів, авторських прав і товарних знаків у міру того, як відбуваються відкриття і створюються продукти. Одна з моделей R&D - це відділ, укомплектований в основному інженерами, які розробляють нові продукти - завдання, що зазвичай включає великі дослідження. У цій моделі немає конкретної мети або застосування. Замість цього дослідження проводиться заради дослідження. Інша модель включає в себе відділ, що складається з учених-промисловців або дослідників, яким доручено прикладні дослідження в технічних, наукових або промислових областях. Ця модель полегшує розробку майбутніх продуктів або покращення поточних продуктів та/або операційних процедур. Існують також бізнес-інкубатори й акселератори, в яких корпорації інвестують в стартапи і надають фінансову допомогу і рекомендації підприємцям в надії, що в результаті з'являться нові інновації, які вони зможуть використовувати в своїх інтересах. Крім того, злиття і поглинання та партнерство також є формами R&D, оскільки компанії об'єднують зусилля, щоб скористатися інституційними знаннями та досвідом одне одного.

Фактично, будучи більш стратегічними і бізнес-орієнтованими, ніж структурними, R&D процеси тісно переплітаються з іншими напрямками в ІТ-сфері в цілому і на ринку українського ІТ зокрема. Наприклад, корейська компанія Samsung Electronics розгорнула окремий офіс на

території України, зорієнтований в першу чергу на покращення і розширення своєї ІТ-інфраструктури, з кількістю працівників понад 1100 осіб [214]. За нею йдуть такі компанії, як NetCracker, Aricent, SysIQ, Wargaming, Siemens, Magento, Yandex, АВВУУ. R&D напрям також часто інтегрується з аутсорсингом та консалтингом, забезпечуючи вихід компаній на новий рівень: компанія-виконавець не просто відповідальна за виконання певного заздалегідь погодженого переліку робіт, але й фактично бере на себе обов'язки цілої філії компанії-замовника, виконуючи під ключ всі етапи, починаючи від стратегічного бізнес-планування в довірній галузі і закінчуючи наймом персоналу і виконанням безпосередньої розробки.

Також варто акцентувати увагу на тому, що на сьогодні отримує все більшу популярність RD напрям (від англ. «resource development» - розвиток ресурсів) у великих українських ІТ-компаніях, метою якого є підготовка та підвищення кваліфікації кадрів за рахунок внутрішніх ресурсів і відділів. За словами директора центру підготовки EPAM Systems, компанія співпрацює з 20 вузами і випускає 2000-3000 фахівців щорічно, більшість з яких приймають на роботу. Призначення навчальних програм EPAM - дати слухачам додаткову підготовку і проєктну адаптацію завдяки роботі в малих групах, вирішення проєктних завдань і зануренню в індустріальний процес розробки [66]. Подібні практики з впровадження спеціальних відділів підготовки кадрів мають і такі компанії як GlobalLogic Ukraine зі своєю програмою «Gl Trainee program», а також SoftServe з «IT Academy».

Вищезазначене дає змогу зробити висновок, що на поточний момент ІТ-індустрія України пройшла великий шлях еволюції, розширилася й успішно інтегрувалася в світовій економічній простір, перетворившись з вузькоспрямованої сфери розробки ПЗ в широку міжнародну інфраструктуру, яка бере участь у всіх бізнес-процесах, починаючи від

продукування, розробки ідеї і її реалізації, та закінчуючи власноручним підбором і, що більш важливо, підготовкою своїх кадрів.

Розробка великих проєктів у сфері ІТ, незважаючи на її контекст, передбачає роботу однієї або декількох команд професіоналів, які, зважаючи на різноманітність завдань, технічних засобів і кадрового складу, в свою чергу, вимагають особливого підходу щодо організації ефективного управління людськими ресурсами. У сучасному ІТ дуже рідко зустрічаються такі компанії, що в силу різних технічних або кадрових особливостей використовують концепцію проєктних команд з рівноправними членами, які виконують загальний список завдань.

Проте, на сьогоднішній день переважна більшість ІТ-компаній на своїх проєктах практикують розподіл співробітників за спеціалізаціями і вибудовують субординацію, ґрунтуючись на їх кваліфікаційному рівні. Саме спеціалізація та кваліфікаційний рівень спеціаліста багато в чому визначають особливості його діяльності, місце у внутрішній ієрархії компанії або проєкту, вимоги до його підготовки та інші особливості взаємодії всередині команди, що робить вивчення цих двох характеристик ключовим аспектом розуміння специфіки і вимог до діяльності сучасних інженерів-програмістів.

Отже, важливим для вирішення завдань дисертаційного дослідження є визначення фахової ієрархії працівників сфери ІТ, їх кваліфікаційних категорій, що презентують вимоги до фахівця кожного з її рівнів та віддзеркалюють особливості їх підготовки, а також етапи професійного розвитку.

Ідеологія рівнів субординації працівників сфери ІТ не є жорстко стандартизованою, а деякі особливості в оцінці кваліфікаційного рівня фахівців можуть варіюватися від компанії до компанії. Проте, загальні основи і концепти багато в чому повторюють модель придбання навичок, розроблену братами Дрейфус. «Дрейфусовська» система - модель

отримання студентами вмінь та навичок за допомогою формального навчання і практики, реального досвіду.

Вона складається з п'яти етапів професійного розвитку [41]:

1) новачок – працівник, досвіду якого недостатньо, щоб рухатися в правильному напрямку під час вирішення професійного завдання. Однак, він може бути ефективним при виконанні контекстно незалежних завдань в форматі алгоритмів під керівництвом більш досвідченого колеги;

2) початківець високого рівня, який починає відступати від фіксованих правил та алгоритмів. Він може спробувати вирішити завдання самостійно, але часто зустрічається з труднощами у процесі. Початківець може скористатися порадами, з огляду на свій досвід подібних ситуаціях;

3) компетентний працівник - будує правильні моделі проблемної ситуації та ефективно нею користується, здатний усувати проблеми, з якими раніше не стикався, ґрунтуючись або на вже отриманому досвіді, або спираючись на досвід своїх колег;

4) досвідчений працівник - потребує «повної картини» проблемної ситуації, адже прагне розуміти увесь концепт; постійно оцінює виконану роботу і переглядає свої підходи, щоб наступного разу бути ще більш ефективним; також може вчитися, використовуючи досвід інших;

5) експерт - основне джерело знань та інформації в будь-якій сфері; постійно знаходиться у пошуку кращих методів роботи, застосовує весь свій величезний досвід для вирішення проблеми; презентує свій досвід у формі книг, статей та проведення семінарів; у діяльності керується інтуїцією [41].

За аналогією з вищеописаною системою, професійний розвиток фахівця сфери розробки ПЗ складається теж з чотирьох ступенів – кваліфікаційних категорій, а саме: молодшого спеціаліста («Junior Specialist»), спеціаліста («Middle specialist»), старшого фахівця («Senior specialist»), керівника команди та експерта («Team Leader»).

Так само, до цього переліку можуть включатися *інтерни*, що знаходяться на випробувальному терміні і проходять фазу входження в команду розробника. Промисловість застосовує зазначену ієрархію для кращого розуміння ступеня кваліфікації співробітників, що дозволяє оптимально розподілити проєктні обов'язки, організувати систему внутрішньопроєктної субординації, а також визначити організаційні умови роботи у компанії – політику відпусток, заробітної плати та позапроєктних активностей.

На сьогоднішній день у багатьох українських ІТ-компаніях є програми стажувань для інженерів початкового рівня, які в основному мають лише теоретичні знання і зовсім не мають досвіду роботи. Таким фахівцям присвоюють рівень інтерна, а основним завданням стає отримання мінімально необхідного досвіду і практичних знань, які дозволять йому виконувати найпростіші проєктні завдання. Аналогічно етапу «новачок» з «Дрейфусовської» системи, інтерни здатні виконувати найпростіші завдання по чітко сформульованому алгоритму, при цьому вимагаючи постійного контролю більш досвідченим співробітником.

Пройшовши етап підготовки й отримавши мінімальний практичний досвід, співробітник переходить на позицію молодшого спеціаліста («Junior Specialist»), або «Початківця високого рівня». Ці фахівці починають відходити від проходження встановленим правилам і формулюють загальні принципи, ґрунтуючись на схожих ситуаціях, з якими вони зустрічалися. Основною вимогою для цієї позиції є вміння самостійно виконувати локальні та чітко сформульовані технічні завдання. Якщо в проєкті є реалізована архітектура, молодший спеціаліст повинен мати можливість реалізувати наступну частину логіки програми за вже існуючими зразками. Однак, він може час від часу робити помилки, не розуміти специфічних аспектів і повинен перебувати під постійним контролем керівника групи, обговорюючи і погоджуючи свої наступні кроки.

Спеціалісти середнього рівня («Middle specialist») мають найбільший вплив на виконання повсякденної, так званої рутинної роботи. Модель Дрейфусів описує спеціаліста середнього рівня як компетентного фахівця, який є дуже ініціативним і гнучким. Він є основою команди, виконує більшу частину роботи і вже може наставляти новачків [41]. Фахівці середнього рівня знайомі зі стандартними шаблонами і рішеннями при створенні додатків в своїй сфері, розуміють, навіщо вони потрібні, і знають, як їх застосовувати. Стандартизація рішень має велике значення в колективній розробці коду, оскільки вона дозволяє новій людині швидко зрозуміти, що до чого, і зводить до мінімуму кількість помилок. Розуміння структури типового проєкту чи функціональності робить задачу її створення з нуля досить тривіальною і дозволяє обговорити принципи правильної реалізації та відрізнити хороший код від поганого. Проте, компетентні розробники все ще не мають досвіду, який дає їм змогу будувати глобальну стратегію своєї роботи, оцінювати її у розрізі бізнесу та передбачати довгострокові результати, тому потребують консультації у більш досвідчених колег.

Старші фахівці («Senior specialists») – це розробники з глибокими спеціалізованими знаннями і значним досвідом, який дозволяє їм наставляти своїх колег. Їх позиція відповідає етапу моделі «досвідчений», що передбачає розуміння контексту вирішуваних завдань і бачення цілісної картини. Крім того, старші розробники є добре обізнаними в сфері бізнесу, яку вони підтримують. У загальному значенні, їх робота полягає в тому, щоб пропонувати найкращі варіанти вирішення проблем, засновуючи вибір на своєму широкому практичному досвіді, або здійснювати дії, що попередять її появу. Більш того, старші фахівці характеризуються розвинутими комунікативними навичками, тому що мають не тільки запропонувати правильне рішення, але і переконати свого клієнта і команду погодитися з ним. Це твердження актуально і в контексті взаємодії з командою: недостатньо мати чітке уявлення процесу, також

потрібно вміти донести це до членів команди, серед яких можуть бути співробітники з вкрай слабкими технічними знаннями і відсутністю досвіду.

Керівник команди («Team Leader») – це скоріше, відхід від командної ієрархії, ніж її наступний щабель. Зазвичай на цій позиції працюють добре знайомі з колективом проєкту старші фахівці, що володіють навичками планування і управління. Ця посада передбачає розподіл завдань між підлеглими та контроль їх подальшого виконання. Керівник працює безпосередньо з представниками бізнеса та є своєрідним посередником між замовниками і командою розробки.

Подальший шлях кар'єрного зростання та в цілому професійного розвитку, якого може досягти старший фахівець або керівник команди, також є опціональним, контекстним, і може відрізнятись серед різних компаній і спеціалізацій. Зазвичай мається на увазі, що фахівець, який став експертом в певній галузі і підтвердив свої навички управління командою, продовжує або розвиватися як управлінець, або розширює свою технічну компетентність, переходячи на одну з наступних позицій, як-то:

1) технічний експерт («Tech expert»), статус якого передбачає глибоке знання конкретної вузької сфери, наприклад, інтеграції з банківськими системами. Він має бути здатним реалізувати свій глибокий технічний досвід і максимально кваліфіковано надати консультацію щодо пов'язаної зі сферою проблеми;

2) промисловий експерт («Domain expert») – фахівець, який має вивчити бізнес клієнта, зрозуміти, як працює галузь, а також дізнатися про проблеми і вирішення, загальні для конкретної сфери. Саме такі фахівці зазвичай складають основу проєкту і надають консультації щодо глобальних бізнес-процесів та поширених промислових практик;

3) фронтмен («Frontman») – фахівець, визначальними якостями якого є здатність правильно презентувати себе і свою компанію, говорити про складні технічні речі простими словами, швидко створювати прототип і

показувати перші результати, а також «говорити на одній мові» з представниками клієнта. Таких людей часто викликають на перші зустрічі і відправляють у відрядження до клієнта. Їх завдання - швидко зрозуміти проблему і точно пояснити, як компанія зможе допомогти в її усуненні. Поєднання технічних знань і навичок презентації дозволяє компанії легко знаходити спільну мову з не-технічним бізнесом и налагоджувати нові перспективні зв'язки;

4) архітектор («Architect») – фахівець, головним завданням діяльності якого є розробка технічного дизайну складних систем ще до початку їх імплементації, що вимагає глибокого досвіду і широкого кругозору з точки зору платформ і компонентів, доступних на ринку, з яких можна синтезувати рішення.

Вищеописані етапи професійного розвитку є широко застосовними в контексті будь-якої діяльності інженера-програміста і використовуються для побудови внутрішньої ієрархії в переважній більшості сучасних ІТ-компаній. Однак, в силу різноманітності можливих сфер реалізацій фахівців у робочому процесі, багато особливостей їх розподілу і субординації також залежать від технічної спеціалізації.

В цілому, особливості діяльності інженерів-програмістів можна розглядати відповідно до двох критеріїв у контексті їх технічної спеціалізації: *за їх проектною позицією або за характером програмного забезпечення, яке вони розробляють*. Так, продукти, які розроблюються, можуть бути різними: комп'ютерні ігри, веб-сайти, офісні додатки, програмне забезпечення для смартфонів та ін. Розробка кожного з них вимагає специфічних теоретичних та практичних знань у галузі, різних мов програмування та технічних засобів. Проте, незважаючи на технічні аспекти, сам процес та методології розробки залишаються незмінними. Найбільш контрастним є порівняння проектних позицій працівників ІТ, які, в свою чергу, й зумовлюють їх основні проектні обов'язки.

Найбільш поширеною та загальновідомою спеціалізацією фахівця ІТ-сфери є розробник ПЗ (з англ. – «developer»), основною метою професійної діяльності якого є розробка кінцевого продукту чи його елементів. До основних обов'язків розробника ПЗ відносяться:

- написання коду та налагодження програм з використанням відповідних програмних технологій і апаратних засобів, баз даних і мов програмування, прийнятих в компанії;
- зв'язування модулів програми;
- розробка модифікацій програм; внесення змін за вимогами;
- розробка високорівневої програмної логіки;
- створення і супровід програмної документації.

За напрямками діяльності розробники ПЗ, у свою чергу, можуть розподілятися на спеціалістів з: клієнтської частини, серверної частини, баз даних, хмарних технологій та ін., що також визначається їх спеціалізаціями. Зазначені спеціалізації вимагають знань різних технологій та технічних галузей, однак, незважаючи на різноманіття напрямів, глобальний підхід до роботи та обов'язки у всіх є схожими.

До спеціалістів з розробки ПЗ висувають такі базові вимоги: знання з програмування (конкретної мови); знання алгоритмів та паттернів, а також загальних підходів до розробки; вміння працювати з допоміжним ПЗ; знання методологій розробки. Професійні обов'язки вимагають від працівника логічності, системності та гнучкості мислення, технічної творчості.

Кожен комерційний проект з розробки ПЗ має менеджера, який керує процесом розробки і може приймати як адміністративні, так і технічні та архітектурні рішення. Завдання менеджера (з англ. – «project manager») - розробити стратегію розробки й управління проектним процесом, зробити його максимально гнучким та стабільним. ІТ-менеджер організовує і керує щоденною роботою усієї команди, яка складається з розробників, дизайнерів, рекламистів і фахівців зі зв'язків з громадськістю. Він знає, що

може зробити програміст, адміністратор або технічний фахівець, до якого результату призведе використання певної технології, в чому зацікавлені замовники, які програмні рішення використовувати з метою більш ефективної підприємницької діяльності, як формувати команду та яке фінансування потрібно для виконання певної роботи. Якщо узагальнити вищезазначене, до обов'язків ІТ-менеджера входять: з'ясування потреб кінцевих користувачів передбачуваного продукту; визначення концепту продукту, його цілей, вимог до нього; розробка плану реалізації продукту спільно з командою розробників; планування процесу розробки; прогнозування ризиків; формування команди.

Жоден процес розробки не може обійтись без апаратної та низькорівневої програмної структури. Весь код, який пишуть розробники, повинен інсталюватися та виконуватися на сервері. Помилки повинні автоматично перетворюватися в електронний лист, який буде нотифікувати розробників (що є дуже актуальним, коли команда розподілена між різними країнами). Для вирішення таких завдань в ІТ існують спеціалісти окремого напрямку – системні адміністратори (з англ. – «system engineer»), або системні інженери. Головне завдання системного адміністратора - покращувати та модернізувати всю інформаційну інфраструктуру проекту, а також стежити за її працездатністю, реагувати на проблеми, що виникають. Також системний адміністратор за сумісництвом виконує завдання мережевого адміністратора (робота з мережами на рівні комутаторів, маршрутизаторів, віртуальних мереж). У класичному варіанті системний адміністратор займається вирішенням проблем на рівні операційної системи і прикладних програм, а мережевий - на рівні мережі та програм, що працюють з мережею. В професійні обов'язки системного адміністратора входить: налаштування апаратного і програмного забезпечення для стабільної роботи; конфігурація серверів, відмовостійких рішень, інфраструктурних елементів; установка / інсталяція серверів / сервісів, модернізація існуючих; написання

серверного ПЗ; тестування апаратного забезпечення; налаштування робочих станцій, мереж і мережевого устаткування (роутери, модеми); інформаційна безпека; організація резервного копіювання; організація віддаленого доступу. На відміну від спеціалізацій, розглянутих вище, системні адміністратори повинні вміти працювати з апаратними засобами. Для успішного виконання проєктних обов'язків потрібно: знати і вміти налаштовувати операційні системи; розуміти принципи роботи мережевого устаткування, мережевих протоколів; вміти працювати з доменними службами, поштовими службами; вміти інсталювати і налаштовувати системи збірки та безперервної інтеграції проєкту.

Ще однією спеціалізацією ІТ-спеціаліста є бізнес-аналітика. Головне завдання бізнес-аналітика (з англ. «business analyst») - виявити проблеми бізнесу замовника і знайти максимально ефективне рішення. Для цього він повинен володіти знаннями в предметній галузі. Бізнес-аналітик працює з вимогами на всіх етапах життєвого циклу розробки ПЗ і постійно виступає посередником між замовником і командою програмістів. Робота бізнес-аналітика включає такі етапи як: виявити потреби замовника, зрозуміти проблему, яку він хоче вирішити; самостійно або з допомогою команди сформулювати концепцію рішення; оформити концепцію в технічне завдання з конкретними вимогами до майбутнього продукту. Для цього використовуються різні техніки бізнес-аналізу: моделювання процесів і структур, прототипів інтерфейсу, сценаріїв використання продукту. В цей же час робиться точна оцінка трудовитрат і тривалості робіт. Для роботи бізнес-аналітика важливо: знати методології збору, аналізу і формалізації; знати предметну область, яку потрібно аналізувати; розуміти життєвий цикл ПЗ відповідно до різних методологій; знати основи програмування, тестування, алгоритмів, економіки. Взагалі можна виділити два шляхи становлення фахівця як бізнес-аналітика, які фактично дублюють ситуацію проєктного менеджера. Аналітиком може бути ІТ-фахівець, який спрямований у більшій мірі на комунікацію, спілкування, ніж написання

коду. Такий аналітик буде розуміти процес розробки, знає можливості програмного забезпечення і розуміє, що потрібно знати розробнику для якісної роботи. Однак йому необхідно окремо отримувати професійні знання в сфері розробки. З іншого боку, - це може бути спеціаліст і без ІТ-освіти, який є професіоналом в певній предметній галузі. Такий аналітик розуміє всі нюанси бізнесу і розмовляє з замовником «на одній мові». Але він має опанувати знання щодо того, що саме підлягає автоматизації, як коректно зіставити бізнес-ідеї з технічною реалізацією та які дані потрібні розробникам для роботи.

Масовість впровадження ІТ в життя сучасної людини сильно змістила акценти та вектор розвитку усієї сфери. Акцент в розробці ПЗ в більшості випадків перейшов з технологічності та інноваційності продукту в бік його зручності, стабільності, надійності і, головне, якості.

Акцентування уваги на якості програмного забезпечення призвело до появи окремої спеціалізації працівників ІТ-сфери, головною метою яких є безпосередній контроль відповідності розробленого продукту очікуванням користувачів і замовника – тестувальників ПЗ.

Тестування ПЗ (з англ. – «software testing») - це процес:

- аналізу або експлуатації програмного забезпечення з метою виявлення його дефектів [93];
- визначення відповідності об'єкта тестування заданим специфікаціям (характеристикам, функціям), який полягає в опрацюванні програмою послідовності різноманітних контрольних наборів тестів з відомими результатами [1];
- експериментального аналізу функціональності версії розроблюваного програмного продукту, що складається з тестових завдань, кожне з яких полягає в дослідженні якоїсь окремої риси випробуваної системи [92]. Завдання тестування полягає у визначенні умов, при яких виявляються дефекти системи та протоколювання цих умов (К. Канер, Дж. Фолк, Нгуен Енг Кек та ін.);

- складний процес, що передбачає комплекс різноманітних задач і дій для перевірки відсутності помилок і виявлення дефектів в програмних системах, оцінки продуктивності, контролю наявності та повноти документації, оцінки якості прийнятих проєктних рішень (I. Burnstien [176]).

Отже, тестування, як зазначають дослідники (А. Авраменко, В. Авраменко, Т. Баджетт, Г. Косенюк, В. Липаєв, Г. Майерс, Дж. Макгрегор, Д. Сайкс, К. Сандлер та ін.), і з чим ми повністю погоджуємось, є процесом керованого експериментування з програмним продуктом за допомогою тестів з метою виявлення в ньому помилок та неточностей, що допущені розробниками ПЗ. Воно пронизує весь життєвий цикл ПЗ, починаючи від аналізу вимог, проєктування і закінчуючи невизначено довгим етапом експлуатації.

У сучасній промисловій практиці визначення тестування значно розширилося: в нього було додано поняття «оптимізація бізнес-технологій», яка спрямовує розвиток ІТ відповідно до цілей бізнесу завдяки оцінці і максимізації значущості всіх етапів життєвого циклу розробки ПЗ для досягнення необхідного рівня якості, продуктивності, доступності.

Відтак, тестування є процесом, який полягає у перевірці відповідності ПЗ заявленим характеристикам та вимогам: експлуатації в різних середовищах, зрізними навантаженнями, безпеки, ергономіки та зручності користування (В. Котляров, Т. Колікова [111]).

Тестування ПЗ як окремий напрям ІТ має свою історію становлення та розвитку, починаючи з появи перших тестових систем у межах програм наукових досліджень та програм для потреб міністерств оборони [1, с. 11]. Тестування таких продуктів здійснювалось строго формалізовано із фіксацією усіх тестових процедур, даних та отриманих результатів; воно виокремлювалось у самостійний процес, що розпочинався після

завершення кодування та, як правило, виконувався тим же персоналом [111].

У період з початку 60-х років до кінця 90-х років 20 століття тестування ПЗ пройшло шлях від так званого «вичерпного» тестування (з використанням усіх шляхів у кодї, усіх можливих вхідних даних) та «доказу правильності» (тестування як діяльність з підтвердження правильності роботи ПЗ) до тестування як запобігання помилок, дефектів та появи власної його методології.

На сьогоднішній день тестування розвинулось в самостійну галузь ІТ зі своєю методологією, теоретичною базою та своїми унікальними методиками. Попит на відповідних фахівців за останні декілька років стрімко підвищився - якщо раніше в багатьох компаніях розробники зазвичай самі тестували свій продукт на предмет його працездатності, то в сучасних ІТ-компаніях, як зарубіжних, так і вітчизняних, створюються окремі відділи і команди фахівців з тестування, зі своєю внутрішньою ієрархією та субординацією.

Завдання тестування полягає у визначенні та протоколюванні умов, при яких виявляються дефекти системи. При чому тестування не передбачає виявлення конкретних дефектних ділянок програмного коду та виправлення дефектів (це є завданням діяльності розробників, що виконується за результатами тестування системи). До основних посадових обов'язків тестувальників відносяться: розробка тест-стратегії, тест-планів, тест-сценаріїв; проведення різних типів тестування згідно з планом; реєстрація та моніторинг виявлених дефектів; написання звітів про проведені тестування; взаємодія з розробниками і аналітиками.

Як один з важливих етапів розробки програмного продукту (розробка додатку – розробка коду - тестування), як зазначають методологи тестування (А. Авраменко, В. Авраменко, Т. Баджетт, С. Канер, В. Котляров, Г. Майєрс та ін.), тестування відзначається суттєвим внеском до сумарної трудомісткості між іншими етапами його

створення. Так, за результатами оцінки розподілу трудомісткості між етапами створення ПЗ їх співвідношення виглядає таким чином: 40%-20%-40%. Відтак, найбільший ефект щодо зниження трудомісткості може бути досягнутий передусім на етапах розробки і тестування; основний внесок в автоматизацію та генерацію коду потрібно здійснювати саме на цих двох етапах [111].

Таким чином, з огляду на зростання ІТ-індустрії в цілому й особливому акценті на питанні якості її продуктів зокрема, підготовка фахівців з тестування ПЗ є вкрай актуальним завданням сучасної вищої освіти.

## **1.2. Сутність та зміст професійної готовності інженерів-програмістів до тестування програмного забезпечення**

На сьогоднішній день сфера інформаційних технологій значно розвинулась та вийшла за межі класичного написання промислового коду, характерного для індустрії 2000-х років. Сучасний процес виробництва програмних продуктів може суттєво відрізнитися в залежності від низки факторів, таких як: тип продукту, що розробляється; структура і політика компанії-розробника; обрані технології розробки; особливості і мотивація замовника та ін. Процес тестування, як один з ключових етапів розробки будь-якого продукту, також залежить від вищезазначених факторів і радикально змінюється під їх впливом як в технологічному і організаційному, так і в соціальному аспектах. Виходячи з цього можна стверджувати, що вкрай актуальним завданням дослідження є визначення загальних підходів і завдань тестування, характерних для будь-якого комерційного проєкту, і на підставі цього - формулювання загальних вимог до рівня підготовки тестувальників ПЗ – змісту їх професійної готовності як результату їх успішної професійної підготовки.

Відзначимо, що загальній характеристиці різним моделям розробки ПЗ присвятили свої праці Дж. Макгрегор, Д. Сайкс [93], I. Burnstein [176], L. Crispin, J. Gregory [181], J. Glenford Myers [192], W. Lewis, D. Dobbs [202] та ін.

Передусім зазначимо, що саме обрана модель розробки ПЗ визначає особливості діяльності тестувальників ПЗ, адже саме вона є структурою, що систематизує різні види проєктної діяльності, їх взаємодію і послідовність. Вибір тієї чи іншої моделі залежить від масштабу і складності проєкту, предметної галузі, доступних ресурсів та багатьох інших чинників. Вибір моделі розробки ПЗ фундаментально впливає на безпосередньо робочий процес, визначаючи склад команди, вибір стратегії, графік роботи, необхідні ресурси та вимоги до учасників процесу.

Будучи одним з етапів розробки програмного забезпечення, тестування у першу чергу залежить від вибору тієї чи іншої методології побудови процесу в таких аспектах як: цілі тестування; доступні для нього технічні та людські ресурси; час старту і закінчення фази тестування; а також застосовувані технічні та організаційні підходи. На сьогоднішній день існує багато різновидів моделей та методологій розробки ПЗ, але в загальному випадку можна виділити дві їх категорії: монолітні та гнучкі. Розглянемо зміст цих категорій моделей для виокремлення специфічних вимог до тестувальників ПЗ.

Так, однією з класичних монолітних моделей є водоспадна («waterfall model»). Вона передбачає одноразове виконання кожної з фаз проєкту, які, в свою чергу, чітко слідують одна за одною (рис. 1.1). Дуже спрощено можна сказати, що у межах цієї моделі в будь-який момент часу команда рухається лише з попередньої в наступну фазу, ігноруючи загальне бачення проєкту та ні на крок не відходячи від попередньо створеного плану.



Рис. 1.1. Водоспадна модель розробки ПЗ

Водоспадна модель часто інтуїтивно застосовується при виконанні відносно простих завдань, де комунікація між членами команди зведена до мінімуму, а обсяг роботи та залежності заздалегідь відомі та залишаються незмінними. Тестування в зазначеній моделі є незалежним і повністю відокремленим процесом. Особливістю в цьому випадку є факт закінченості етапів аналізу вимог, проектування й основного обсягу розробки, що в свою чергу знімає значну кількість завдань і проблем, з якими потенційно може зіткнутися фахівець з тестування, таких як: з'ясування і зміна вимог до продукту, зміни технологій, необхідність в масштабованості документації та технічних засобів.

Ітераційно-інкрементальна модель є фундаментальною основою сучасного підходу до розробки ПЗ. Ключовою особливістю даної моделі є розподілення проєкту на відносно невеликі проміжки (ітерації), кожен з яких в загальному випадку може включати в себе всі класичні стадії, властиві водоспадній моделі (рис. 1.2).

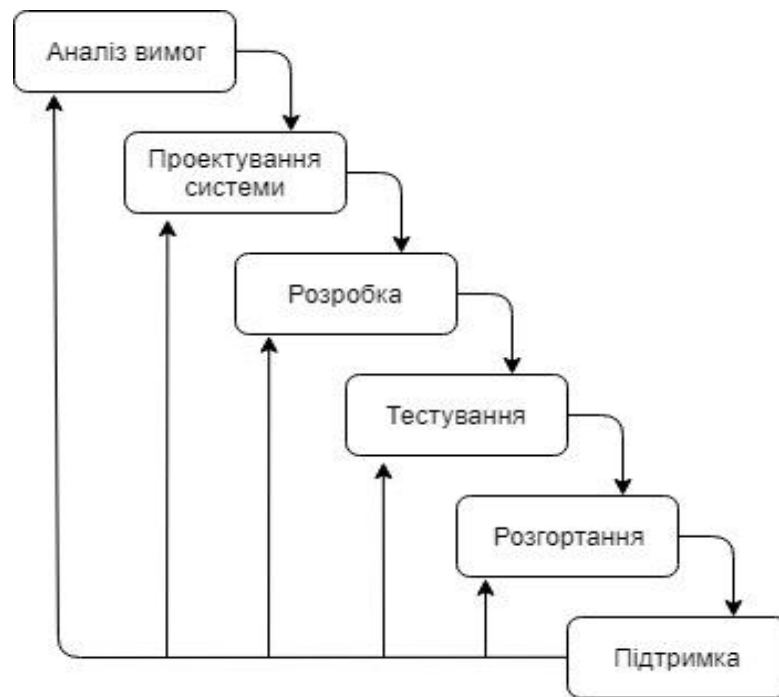


Рис. 1.2. Ітераційно-інкрементальна модель розробки ПЗ

Кожна ітерація додає новий функціонал до вже існуючого базису і може бути розцінена як окремий міні-проект, зі своїми власними етапами, характерними класичній водоспадній методології.

Виходячи з назви моделі, їй властива певна подвійність: з точки зору життєвого циклу модель є ітераційною, так як передбачає багаторазове повторення одних і тих же стадій; з точки зору розвитку продукту (збільшення його корисних функцій) модель є інкрементальною. Довжина ітерацій може змінюватися в залежності від багатьох факторів, однак сам принцип багаторазового повторення дозволяє гарантувати, що і тестування, і демонстрація продукту кінцевому замовнику (з отриманням зворотного зв'язку) буде активно застосовуватися з самого початку і протягом усього часу розробки проєкту.

Фаза тестування повністю відповідає загальному контексту цієї методології: з одного боку воно є відокремленим і незалежним у межах однієї ітерації, з іншого - в глобальному перебігу процесу, що складається з декількох таких ітерацій, продукт еволюціонує і отримує нові функції, не втрачаючи при цьому більшість старих. Повторюваність деяких етапів

тестування в рамках декількох ітерацій вже вимагає від проєктної команди застосовувати особливі підходи до ведення тестової документації і застосовувати практики, спрямовані на максимальне використання продуктів своєї роботи і оптимізації процесу. Також, контролю якості продукту, створеного в ітеративній моделі, властиве «регресійне» тестування – тип тестування, метою якого є перевірка вже існуючого функціоналу після завершення фази нового інкременту.

Найбільш популярними і затребуваними в умовах сучасної індустрії інформаційних технологій і комерційної розробки ПЗ є «гнучкі» методології, основними перевагами яких є баланс між формалізацією і контролем процесу з одного боку, і готовністю до швидкої зміни вимог, умов роботи, складу команди та інших організаційних чинників, при цьому без ризику руйнування структури проєкту та глобального плану розробки. На відміну від класичних монолітних моделей, гнучкі методології основними пріоритетами в управлінні проєктом визначають людей та їх взаємодію для створення продукту на основі співробітництва з замовником. Їх «гнучкість» забезпечується постійною відкритістю та готовністю до змін, на противагу бюрократичному дотриманню планів та постійних узгоджень монолітних моделей. Подібні технології націлені на подолання очікуваної неповноти вимог та їх постійних змін, пропонуючи гнучкий ітеративно-інкрементальний підхід в результаті постійної взаємодії команд, які самоорганізуються і складаються з фахівців різного профілю.

Однією з найбільш поширених та популярних гнучких методологій розробки у сучасній індустрії є Scrum (з англ. «сутичка»). Методологія дає змогу команді обирати завдання для виконання, враховуючи бізнес пріоритети та технічні можливості, а також вирішувати, як їх ефективно реалізувати. Це дає змогу створити умови, в яких команда працює із задоволенням і максимально продуктивно. Scrum фокусується на постійному визначенні пріоритетних завдань, ґрунтуючись на цілях

бізнесу, що збільшує корисність і прибутковість проекту на його ранніх стадіях. Оскільки за ініціації проекту його прибутковість визначити майже неможливо, Scrum пропонує концентрувати увагу на якості розроблення та до кінця кожної ітерації мати проміжний продукт, який можна використовувати, навіть і з мінімальними можливостями.

Першим кроком реалізації будь-якого проекту є створення бачення кінцевого продукту його власником (рис. 1.3).

Після цього власник продукту зустрічається зі скрам-майстром (фахівцем, який контролює та фасилітує процес) та командою, для того, щоб надати їм інформацію про те, яким має бути кінцевий продукт у форматі «user-story» (з англ. «сценарій користувача») - алгоритмів використання кінцевого продукту цільовим користувачем. Також на цьому етапі розраховуються трудовитрати для того, щоб зорієнтувати власника щодо термінів та необхідних ресурсів для виконання проекту.

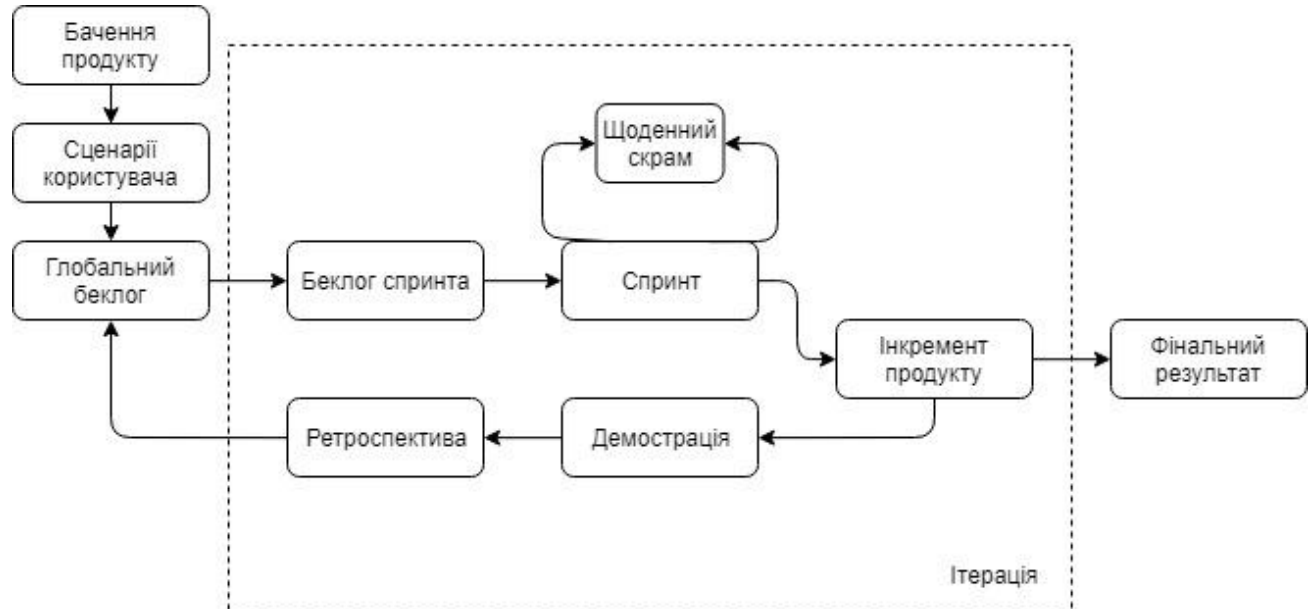


Рис. 1.3. Гнучка методологія розробки ПЗ «Scrum»

Після опрацювання бачення продукту представниками замовника та скрам-майстром, всі члени проекту сортують, пріорітизують та згруповують отримані «user-story» для розуміння того, які елементи є

обов'язковими до реалізації, а які можуть бути відкладені чи змінені у майбутньому. У результаті цього процесу створюється «backlog» (з англ., «резерв») - список формалізованих та структурованих вимог та побажань щодо продукту, які впорядковані за ступенем важливості.

Основою Scrum є «sprint» (з англ., «спринт») - фіксований термін, в процесі якого виконується робота над продуктом. По закінченню спринту повинна бути отримана нова робоча версія продукту.

Наступним етапом після формування резерву є його розподіл між спринтами. При цьому слід раціонально розподілити всі елементи списку, щоб поставлені задачі можна було реально реалізувати у стислий проміжок часу. Перед початком кожного спринту проводиться «sprint planning» (з англ. «планування спринту») - нарада, на якій проводиться оцінка вмісту глобального резерву і формування «sprint backlog» (з англ. «резерв спринту») - окремого списку, який містить завдання, що повинні бути виконані в поточному спринті. Кожен спринт повинен мати мету, яка є мотивуючим фактором і досягається за допомогою виконання завдань з «sprint backlog». Кожен день проводиться коротка нарада «daily scrum» (з англ., «щоденний скрам»), на якій кожен член команди відповідає на питання «що я зробив вчора?», «що я планую зробити сьогодні?», «які перешкоди на своїй роботі я зустрів?». Завдання «daily scrum» - визначення статусу і прогресу роботи над завданнями, раннє виявлення перешкод, що виникли, вироблення рішень щодо зміни стратегії, необхідних для досягнення цілей.

Подібні ітерації (планування, розробка, демонстрація, ретроспектива) проводяться до тих пір, поки резерв завдань не буде повністю вичерпаним, а продукт буде відповідати вимогам замовника.

Слідуючи за загальною логікою ітеративного підходу, що домінує у всіх сучасних моделях розробки ПЗ, життєвий цикл тестування також виражається замкнутою послідовністю дій.

Важливо розуміти, що довжина такої ітерації  $i$ , відповідно, ступінь детальності кожної стадії може варіюватися в найширшому діапазоні, проте загальна структурна логіка залишається незмінною: від стадії аналізу та уточнення вимог до стадії аналізу та фіксації дефектів (рис. 1.4).

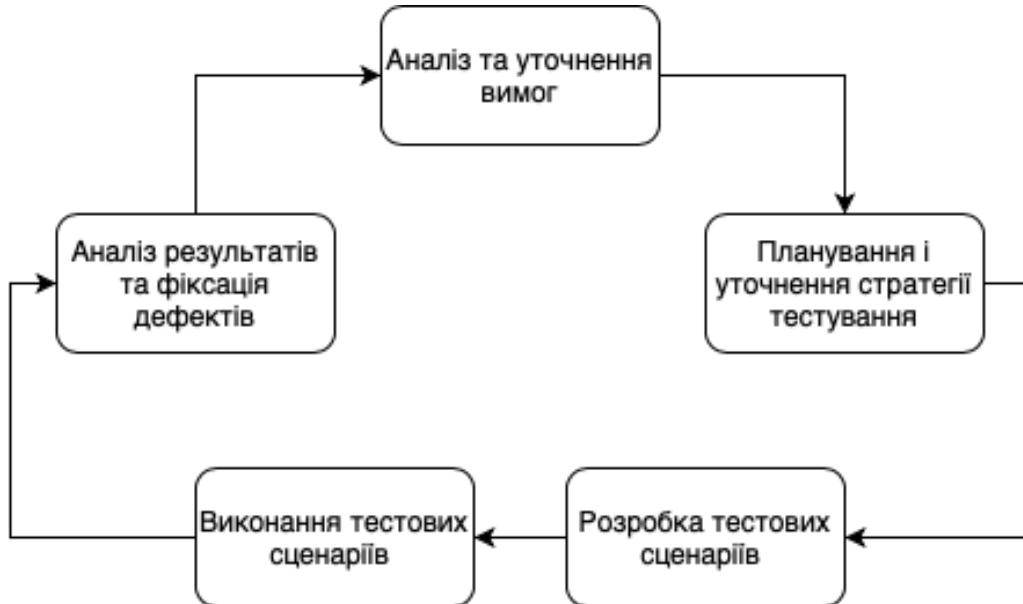


Рис. 1.4. Життєвий цикл тестування в ітеративних моделях розробки ПЗ

Так, стадія 1 («Аналіз і уточнення вимог») дозволяє актуалізувати і затвердити з замовником вимоги, що висуваються до продукту або окремої його частини, яка потрапляє під тестування в даній ітерації, що формує однозначну відповідність очікувань замовника з тестовими сценаріями.

Стадія 2 («Планування та уточнення стратегії тестування») потрібна для визначення термінів і умов проведення тестування, уточнення та оновлення його стратегії і аналізу необхідних для його виконання ресурсів.

Стадія 3 («Розробка тестових сценаріїв») полягає в створенні нових або актуалізації вже існуючих сценаріїв тестування і формування необхідних і достатніх наборів згідно зі стратегією тестування.

Стадія 4 («Виконання тестових сценаріїв») передбачає безпосередню взаємодію з тестовим додатком відповідно до раніше розроблених сценаріїв.

Стадія 5 («Аналіз і фіксація дефектів») потрібна для підведення підсумків тестування, оформлення знайдених дефектів в зручній формі і збору статистики тестування з подальшим її аналізом.

Подібний цикл тестування в своєму базовому алгоритмі не змінюється протягом всієї роботи на проєкті і повторюється від ітерації до ітерації. Однак, наповнення даних етапів може значно варіюватися. В теорії тестування існує абстрактне поняття «надлишкового тестування» - перевірки всіх компонентів програми з усіма можливими комбінаціями всіх можливих вхідних даних у всіх можливих умовах виконання. По суті, тільки воно може повністю гарантувати повну відсутність дефектів в продукті і його відповідність очікуванням замовника. Однак, з огляду на бізнес-орієнтованість сучасної індустрії розробки, процеси і підходи до тестування ПЗ значно відрізняються для різних продуктів, а їх наповнення залежить від низки факторів, таких як:

- технології, що використовуються для розробки продукту;
- терміни розробки - розмір і ефективність команди тестування;
- особливості архітектури продукту;
- особливості бізнес-логіки і функціоналу продукту;
- пріоритети та побажання замовника.

Відсутність фінансової, а в багатьох випадках, і фізичної можливості протестувати всі сценарії використання всіх компонентів продукту у різноманітних умовах, спонукало до створення і класифікації різних підходів і принципів тестування, що дозволяють застосовувати максимально ефективні техніки і підходи для кожної типової ситуації на проєкті, знаходячи баланс між загальним обсягом тестування і вірогідністю пропустити важливий критичний дефект.

До критеріїв, на підставі яких можна побудувати класифікацію, відносяться:

- цілі і завдання тестування;
- доступ до коду та архітектури додатку;

- дискретизація системної логіки (поділ на рівні);
- принцип кінцевої взаємодії з додатком;
- можливість залучення користувачів до тестування [1; 111; 176; 181; 202].

Щодо глобальних цілей і завдань, тестування можна розділити на два типи - *функціональне* та *нефункціональне*.

Функціональне тестування перевіряє, що кожна функція програмного додатка працює відповідно до вимог специфікації. Функціональне тестування показує «Що робить система». Мета цього тестування - перевірити, чи є система функціонально досконалою.

Нефункціональні типи тестування пов'язані з нефункціональними вимогами і допомагають оцінити готовність системи у відповідності з різними критеріями, які не охоплюються функціональним тестуванням. До основних типів нефункціонального тестування відносять:

- інсталяційне тестування - спрямоване на виявлення дефектів, що впливають на перебіг стадії інсталяції програмного продукту;
- тестування зручності використання - спрямоване на дослідження того, наскільки кінцевому користувачеві зрозуміло, як працювати з продуктом;
- тестування безпеки - передбачає перевірку здатності додатків протистояти зловмисним спробам отримання доступу до даних або функцій, права на доступ до яких у зловмисника немає;
- тестування інтернаціоналізації - спрямоване на перевірку готовності продукту до роботи з використанням різних мов і з урахуванням різних національних і культурних особливостей;
- тестування локалізації - передбачає перевірку коректності та якості адаптації продукту до використання на тій чи іншій мові з урахуванням національних і культурних особливостей;
- тестування продуктивності - дослідження показників швидкості реакції програми на зовнішні впливи при різній за характером і інтенсивності навантаження.

Далеко не завжди у команди тестування може бути постійна можливість безпосередньо взаємодіяти з продуктом зважаючи на різні фактори, серед яких: ранні етапи розробки, відсутність середовища виконання або функціонально необхідних ресурсів, політика безпеки проєкту. Отже, за критерієм доступу до коду та архітектури додатку, тестування може бути *статичним* та *динамічним*.

Статичне тестування – це тестування без запуску коду на виконання. У межах цього підходу під процес тестування можуть потрапити:

- документи (вимоги, опис архітектури додатку, схеми баз даних та ін.);
- графічні прототипи (ескізи призначеного для користувача інтерфейсу, медіа-матеріали);
- код програми (аудит програмного коду на виявлення стилістичних або архітектурних помилок без безпосереднього запуску);
- параметри (налаштування) середовища виконання програми;
- заздалегідь підготовлені тестові дані.

*Динамічне тестування* - тестування з запуском коду на виконання. Запускатися на виконання може як код всієї програми, так і його окремі частини або модулі. В даному контексті виділяються такі рівні тестування як:

1) модульне тестування - перевірка поведінки певного розділу коду, зазвичай на функціональному рівні. Ці типи тестів зазвичай пишуться розробниками в процесі роботи над кодом, щоб гарантувати, що конкретна функція працює належним чином. Само по собі модульне тестування не може перевірити функціональність частини ПЗ, а скоріше використовується для забезпечення того, щоб модулі програмного забезпечення працювали незалежно один від одного;

2) інтеграційне тестування - перевірка інтерфейсів між компонентами на відповідність дизайну ПЗ. Воно спрямоване на виявлення дефектів у взаємодії між інтегрованими компонентами (модулями);

3) системне тестування, яке на відміну від попередніх рівнів, тестує повністю інтегровану систему як єдине ціле.

Крім того, з позиції можливості доступу до коду, закритим ресурсам та архітектури додатку, зазвичай виділяють [8; 89; 92; 93]:

- тестування чорного ящика - це високий рівень тестування, який фокусується на поведінці програмного забезпечення. Він передбачає тестування з точки зору зовнішнього або кінцевого користувача, коли тестувальник використовує ті ж інтерфейси, точки входу і знання про функціонал і принцип роботи. Тестування чорного ящика може застосовуватися практично на всіх рівнях тестування програмного забезпечення: модульне, інтеграційне системне;

- тестування білого ящика - це метод тестування, який передбачає розуміння внутрішнього функціонування системи. У цьому методі тестування засноване на покритті операторів коду, гілок, шляхів або умов і інших нюансів, невідомих кінцевим користувачам. Метод тестування білого ящика передбачає, що шлях логіки в модулі або програмі є відомим, а також існують внутрішні точки входу, що дозволяють тестувальнику взаємодіяти з додатком з використанням додаткових функцій, недоступних в призначеній для користувача версії;

- тестування сірого ящика - проміжний варіант між двома попередніми. Зазвичай передбачає знання тестувальника про низькорівневу архітектуру продукту, нюанси і принципи його роботи, а також можливості взаємодіяти з окремими його модулями, однак не передбачає низькорівневого доступу до коду або окремих його елементів, припускаючи використання функціоналу таким же чином, яким їм буде користуватися кінцевий користувач.

Щодо *критерію можливості підключення кінцевих користувачів до тестування*, виділяють наступні типи тестування [1; 8; 89; 92; 93]:

- альфа-тестування - це внутрішні перевірки, що виконуються внутрішньої командою розробників або тестувальників, рідше - самим

замовником. Його основна мета - виявлення помилок програмного забезпечення на ранніх етапах розробки. На етапі альфа-тестування поведінка програмного забезпечення перевіряється шляхом імітації дій кінцевих користувачів;

- бета-тестування можна назвати попереднім тестуванням. Його проводять серед обмеженої кількості спеціально відібраних кінцевих користувачів, які називаються бета-тестерами, до офіційної поставки продукту на масовий ринок. Основна мета бета-тестування - перевірити сумісність програмного забезпечення з різними конфігураціями програмного і апаратного забезпечення, типами мережевого підключення, а також отримати відгуки користувачів про зручність використання і функціональності програмного забезпечення;

- гамма-тестування - це заключний етап процесу тестування, проведеного перед випуском програмного забезпечення. Воно гарантує, що продукт готовий до випуску на ринок у відповідності з усіма зазначеними вимогами. Гамма-тестування фокусується на безпеці і функціональності ПЗ. Але це не включає ніяких внутрішніх заходів щодо забезпечення якості. Під час гамма-тестування програмне забезпечення не зазнає ніяких змін, якщо виявлена помилка не має високого пріоритету та значущості.

Перед початком проекту або чергової його ітерації, спеціаліст, відповідальний за якість продукту, вибирає комбінацію з типів та підходів до тестування, описаних вище, у відповідності до наявних ресурсів, термінів закінчення фази тестування та побажань замовника, та, відповідно, створює детальну стратегію.

Список обов'язків та основних завдань, що виконуються фахівцем з тестування ПЗ, у першу чергу залежить від його кваліфікаційного рівня. Як вже було зазначено у попередньому підрозділі дисертаційної роботи, саме кваліфікаційний рівень визначає знаходження фахівця в ланцюжку делегування завдань і можливості виконувати складну творчу роботу.

Особливість сфери тестування полягає в тому, що, крім високорівневого планування, управлінських завдань і вибору технологій і підходів, основний список завдань, що виконуються фахівцем протягом більшої частини часу перебігу проєктів, не змінюється.

Проте, в контексті теми дисертаційного дослідження, предметом вивчення якого є підготовка майбутніх інженерів-програмістів до тестування ПЗ акцент буде зроблений на фахівців з кваліфікаційним рівнем «Junior».

Незважаючи на істотні як технічні, так і організаційні відмінності в підходах, що детерміновані особливостями цільового програмного продукту і організації команди, що його розробляє, можна виділити наступні глобальні завдання, характерні для діяльності фахівця з тестування з кваліфікаційним рівнем «Junior» в контексті будь-якої моделі процесу тестування:

- аналіз і уточнення вимог до продукту;
- естімація і планування;
- створення і підтримка тестової документації;
- проходження тестів і пошук дефектів.

Аналіз вимог є першим і, багато в чому, ключовим завданням, з яким стикається фахівець з тестування ПЗ в своїй роботі. Незважаючи на те, що глобальна документація, бачення кінцевого продукту і загальний список вимог зазвичай готується ще до початку роботи над безпосередньо розробкою і тестуванням проєкту, а також незважаючи на опціональну, але досить часту наявність у складі команди окремих фахівців з комунікації з клієнтом і аналізом вимог - бізнес-аналітиків, в умовах сучасних проєктів, які працюють за гнучкими методологіями, бачення бізнесу і технічна база проєкту може радикально змінюватися. Виходячи з базового визначення, тестування – це процес виявлення відповідності підсумкового продукту представленим до нього вимогам замовника. Отже, наявність максимально детальних, уточнених та актуальних вимог та їх повне розуміння

тестувальником є ключовим фактором успішного тестування, а, значить, з процесом аналізу та уточнення вимог стикається кожен фахівець з тестування незалежно від його кваліфікаційного рівня та особливостей проєкту.

Процес аналізу вимог відбувається у більшості випадків за таким алгоритмом:

1) на початку чергового інкременту проєкту замовник або бізнес-аналітик представляє команді список вимог до певного функціоналу продукту;

2) представники команди розробки проводять аналіз вимог на предмет їх логічної і технічної здійсненності. В цей же час представники команди тестування проводять аналіз вимог на предмет їх повноти, несуперечності, конкретності, завершеності і можливості перевірки;

3) перелік запитань і уточнень обговорюється з замовником і затверджується для подальшого прийняття в роботу.

Відтак, для успішного виконання завдань з аналізу і уточнення вимог, фахівець з тестування повинен знати:

- методології та особливості процесу розробки, орієнтуватися в доменній області майбутнього продукту і особливості проєкту розробки;

- типові підходи до написання і аналізу вимог, а також розуміти критерії їх якості та ознаки наявності в них проблем.

З точки зору практичних умінь, фахівець з тестування повинен вміти:

- аналізувати і уточнювати вимоги;
- комунікувати, взаємодіяти із замовником: визначати вимоги, з'ясовувати очікувану поведінку програмного продукту, а також, оперуючи своїм досвідом, пропонувати замовнику альтернативи.

Щодо особистісних якостей фахівця, то процес аналізу вимог передбачає відповідальне, ціннісне до себе ставлення: він запобігає величезній кількості потенційних проблем і втрат часу і грошей команди розробки, який враховує важливість даного етапу, а також уважність,

концентрацію і налагодження довірчих відносин з замовником та командою розробки .

Естімація як оцінка кількості часу на виконання завдання (трудовитрат) і планування є ще одним важливим підготовчим етапом на початку кожної ітерації проєкту. Незважаючи на поширену практику побудови ієрархічної структури в команді тестування, яка передбачає наявність більш досвідчених колег, даний етап все одно є обов'язковий навіть для молодих фахівців: в сучасних гнучких методологіях часто зустрічаються практики, коли всі учасники команди займаються спільним плануванням незалежно від спеціалізації та кваліфікаційного рівня, що передбачає спільну командну відповідальність за свої естімації.

Безпосередньо процес планування наступної ітерації відбувається за таким алгоритмом, а саме:

- 1) замовник пріорітизує список функціоналу проєкту відповідно до свого бачення бізнесу;
- 2) проєктні менеджери, лідери команд та найбільш кваліфіковані фахівці проєкту проводять декомпозицію функціоналу на список незалежних і атомарних завдань, що включають до себе етапи аналізу вимог, розробки і тестування;
- 3) всі учасники команд розробки та тестування на спеціальних регулярних нарадах у форматі дискусії оцінюють завдання на предмет їх складності та необхідного для виконання часу;
- 4) формується список оцінених завдань, який надалі буде взятий у роботу на наступну ітерацію.

Для успішної та продуктивної участі в цьому процесі фахівець з тестування повинен знати теоретичні основи естімації та тайм-менеджменту, вміти оцінювати і планувати свою діяльність на найближчі короткі проміжки часу, бути здатним аргументувати і доносити свої думки членам команди, відстоювати свою позицію. З точки зору особистісних якостей, тестувальник повинен відповідально ставитися до цього етапу,

мати високу професійну вмотивованість щодо прогнозування результату діяльності.

Створення та підтримка тестової документації є основоположною активністю, яка займає більшу частину часу фахівців з тестування. Це твердження пояснюється особливостями розробки програмного забезпечення в умовах гнучких ітераційних методологій розробки: постійні ітерації представляють собою зміни в коді і структурі продукту, які можуть викликати помилки в раніше протестованих місцях, що в свою чергу вимагає регулярного повторного («регресійного») тестування вже готових частин функціоналу. Більш того, в умовах регулярної зміни вимог, постійних переробок та оновлень вже готового функціоналу, а також розподілених, міжнародних команд розробки, дуже важливим фактором успішності тестування є повна формалізація процесу і докладний опис кожного його етапу для того, щоб в будь-який момент часу і на будь-якій стадії з'явилася можливість коригування раніше написаних сценаріїв тестування і відстеження будь-яких змін в функціоналі продукту.

Основоположним документом, на якому базується все тестування проєкту є план тестування або «тест-план». План тестування - це докладний документ, в якому описуються стратегія тестування, цілі, графік, оцінка, результати і ресурси, необхідні для виконання всього циклу тестування програмного продукту. План тестування допомагає визначити зусилля, необхідні для перевірки якості тестової програми та є алгоритмом проведення дій з тестування у вигляді певного процесу, який ретельно відстежується і контролюється менеджером тестування. Даний документ зазвичай складається найбільш досвідченими кваліфікованими фахівцями і узгоджується із замовником ще до початку роботи над проєктом. Проте, розуміння логіки його побудови і вміння працювати згідно з описаною в ньому методикою є одним з ключових факторів успіху в роботі молодого фахівця-тестувальника.

Ключовим і процесоутворювальним документом у тестуванні є «тестовий сценарій» або «тест-кейс» - це набір дій, які виконуються для перевірки певної функції або функціональності програмного додатка. Тестовий сценарій містить у собі етапи тестування, тестові дані, попередні умови та іншу інформацію, необхідну для перевірки відповідної вимоги. Тест-кейс включає до себе певні змінні або умови, за допомогою яких інженер з тестування може порівняти очікувані і фактичні результати, щоб визначити, чи працює програмний продукт відповідно до вимог замовника. Основною передумовою створення тест-кейсів і підтримки їх в актуальному стані є формалізація і збереження напрацювань процесу тестування для подальшого їх аналізу і застосування в наступних ітераціях.

Класичний процес покриття функціоналу тестовими сценаріями складається з наступних етапів:

- 1) аналіз та актуалізація вимог щодо розглянутого функціоналу;
- 2) написання короткого неформального списку необхідних і достатніх перевірок;
- 3) оцінка функціоналу, технічних і людських ресурсів, на підставі якої визначається підхід до вибору техніки написання тестів, оптимізація їх кількості та наповнення;
- 4) написання та оформлення тест-кейсів за прийнятим на проєкті шаблоном;
- 5) проходження контролю, експертної оцінки (рев'ю) лідера команди тестування або її членів, виправлення виявлених недоліків.

Результатом проведеної роботи є набір тестів, актуальний до моменту, поки в описаному функціоналі не відбуваються будь-які зміни вимог або технологій його реалізації.

В результаті проходження описаних вище сценаріїв і тестування продукту часто знаходяться невідповідності між роботою програми та очікуваною від неї поведінкою. В такому випадку оформляється ще один тип документа - «bug report» (з англ. «звіт про дефект»). Цей документ

містить в собі всю необхідну інформацію про знайдену помилку: умови її перебування (версія програми, дата і час, конфігурація), кроки, що призводять до помилки, очікуваний результат (з обов'язковим посиланням на відповідну вимогу) та опис поточної некоректної поведінки. Після закінчення чергової ітерації тестування, зазначені документи передаються команді розробки, пріорітизуються і в подальшому виправляються.

Фінальним документом, що знаменує про закінчення ітерації тестування, є звіт, який містить в собі: перелік пройдених тестових сценаріїв, в якому відзначені успішні та неуспішні; перелік знайдених і оформлених дефектів; загальна статистика, зібрана в процесі тестування та інша інформація, необхідна команді для прийняття рішення про готовність функціоналу до роботи.

Робота з тестовою документацією і проходження тестів є найбільш розповсюдженими завданнями, з якими стикається фахівець з тестування поза контекстом проєкту та методології. При цьому, багато в чому саме вони вимагають обов'язкової наявності специфічних теоретичних знань, до яких відносяться: теоретичні основи функціонального та нефункціонального тестування; структура і правила роботи з тестовою документацією; знання технік тест-дизайну, правил написання та оптимізації тестових сценаріїв, а також знання принципів роботи допоміжного програмного забезпечення для тестування.

Крім того, у цьому контексті необхідними також є і загальні знання сфери інформаційних технологій, такі як: основи алгоритмів; основи промислового програмування; процеси та методології розробки; основи архітектури комерційних додатків; основи мережевих технологій і адміністрування; операційні системи; засоби віртуалізації та ін.

З практичної точки зору спеціаліст з тестування ПЗ повинен бути здатний: аналізувати, створювати і підтримувати тестову документацію; взаємодіяти з додатком на призначеному для користувача і

розробницькому рівні; проходити тестові сценарії і коректно описувати дефекти.

Описані вище завдання вимагають від фахівця уважності, високого рівня розвитку критичного мислення, здатності до аналізу і дослідження, вміння стисло, проте змістовно викладати свої думки.

Таким чином, незалежно від технічних та організаційних умов проєкту, складу команди та особливостей середовища, можна виділити основні завдання, що постають перед фахівцем з тестування ПЗ та ті професійні знання, а також технічні та нетехнічні вміння, що забезпечують успішність їх вирішення. Що і презентовано нами у табл. 1.1.

Зазначимо, що багато вчених-дослідників та досвідчених практиків, що займаються вивченням особливостей тестування ПЗ як особливого виду діяльності сучасних програмістів зазначають, що попри наявність багатьох підходів до розв'язання завдань тестування ПЗ, ефективність тестування програмних продуктів є процесом найвищої міри творчості, який не зводиться до слідування або створення чітких та строгих процедур [1].

*Таблиця 1.1*

**Завдання професійної діяльності інженера-програміста з тестування ПЗ, необхідні професійні знання та вміння для їх вирішення**

Професійні завдання	Професійні знання	Професійні вміння	
		Технічні вміння	Нетехнічні вміння (soft skills)
Аналіз та уточнення вимог	знання предметної області; розуміння етапів, процесів і методологій розробки ПЗ; знання теоретичних аспектів роботи з вимогами	аналізувати вимоги на предмет їх адекватності, здійсненості та тестованості застосовувати й аргументувати вибір технік тест-дизайну створювати, оформлювати та реалізовувати тестові сценарії	саморегуляції; аналітичності, гнучкості мислення; комунікувати із замовниками, комунікувати з членами команди та обґрунтовувати/відстоювати свою точку зору
Естимація та планування	розуміння етапів, процесів і методологій розробки ПЗ; знання технік естимації та планування	застосовувати техніки естимації, оцінювати трудовитрати завдань з тестування	саморегуляції, тайм-менеджменту; комунікувати з членами команди та обґрунтовувати/відстоювати свою точку зору; проєктування та

			планування; прогнозування; рефлексивні вміння (оцінювати власну діяльність);
Створення та підтримка тестової документації	знання основ: алгоритмів, промислового програмування, процесів та методології розробки; архітектури комерційних додатків; мережевих технологій і адміністрування; операційних систем, засобів віртуалізації	аналізувати, створювати і підтримувати тестову документацію, оформлювати та аналізувати звіти про дефекти і тестування	саморегуляції (уважність, витривалість); комунікувати з членами команди та обґрунтовувати/відстоювати свою точку зору; презентувати результати роботи
Проходження тестів, пошук дефектів	знання фундаментальних основ тестування: типи тестування, підходи до тестування, інструменти тестування; знання технік тест-дизайну; знання принципів роботи допоміжного ПЗ для тестування	взаємодіяти з додатком на призначеному для користувача і розробницькому рівні, проходити тестові сценарії і коректно описувати дефекти	саморегуляції; комунікувати з членами команди та обґрунтовувати/відстоювати свою точку зору; презентувати результати роботи

Діяльність тестувальників дає змогу зробити процес розробки ПЗ прозорим і керованим для всіх учасників проєкту. Розробникам вона дає впевненість у правильному розумінні завдань, які ставить перед ними замовник; менеджерам – розуміння еволюції проєкту, проблемних місць в процесі розробки, а також інформацію для прийняття оперативних рішень щодо готовності проєкту або його версії до продуктивної експлуатації, продажу.

Для визначення специфічних особливостей професійної діяльності інженерів-програмістів-тестувальників ПЗ протягом грудня 2019 р. - січня 2020 р. нами було проведено опитування серед українських фахівців у сфері тестування ПЗ. У дослідженні взяли участь 76 фахівців, які працюють у провідних ІТ-компаніях (EPAM Systems, Plarium, Global Logic) різних регіонів України, віком від 21 до 45 років (42,1% жінок та 57,8% чоловіків), зі стажем професійної діяльності у сфері ІТ від 4 місяців до 12 років, різних професійних рівнів (Junior -19,7%, Middle – 32,1%, Senior – 27,6%, Team Lead – 19,7%).

За результатами проведеного опитування, специфічними особливостями діяльності фахівця з тестування ПЗ в сучасних ІТ-компаніях були названі (за частотою висловлювань):

1) відповідальність за якість продукту на всіх етапах його розробки, він є «інженером по якості», що у свою чергу вимагає здійснення аналізу та проведення максимальної кількості перевірок ПЗ аби переконатися, що продукт працює правильно;

2) метасистемний характер, оскільки вимагає від тестувальника професійних компетенцій як програміста (мати досвід програмування, знати програмний код), так і бізнес-аналітика, менеджера проєкту (знати його організаційну структуру, мати організаційні та комунікативні вміння);

3) необхідність здійснення аналізу продукту з декількох позицій: з точки зору замовника програмного продукту, крім того, - розробника ПЗ, а також, що є особливо важливим, - з точки зору користувача продуктом;

4) творчий характер, адже вимагає постійного пошуку шляхів перевірки, оптимізації тестового покриття.

Підсумовуючи усе вищезазначене, необхідно зробити акцент на тому, що основними факторами, які забезпечують продуктивність та ефективність професійної діяльності фахівців з тестування ПЗ рівня «Junior» в умовах ІТ-проєкту, є: 1) загальні знання галузі інформаційних технологій, методологій, підходів та інструментів розробки програмного забезпечення; 2) спеціалізовані теоретичні знання тестування програмного забезпечення, основних його типів і видів, принципів роботи з тестовою документацією і особливостей проходження тестових сценаріїв; 3) розвинені технічні вміння: аналізувати вимоги на предмет їх адекватності, здійсненності та тестованості; аналізувати, створювати і підтримувати тестову документацію, взаємодіяти з додатком на призначеному для користувача і розробницькому рівні, проходити тестові сценарії і коректно описувати дефекти 4) розвинені нетехнічні вміння.

Отже, здійснений аналіз особливостей професійної діяльності тестувальників ПЗ дає змогу визначити основні орієнтири професійної підготовки майбутніх фахівців, здобуття ними фахової освіти, що відповідно до сучасної нормативно-правової бази (Закон України «Про вищу освіту», Рекомендації Ради Європи щодо ключових компетентностей для освіти впродовж життя та ін.), а також положень вітчизняної та зарубіжної психолого-педагогічної науки, визначаються через категорію «готовність до професійної діяльності» як передумови професійної компетентності фахівця.

Готовність особистості до професійної діяльності або професійна готовність як особливе педагогічне явище має величезний досвід свого наукового осмислення.

В Академічному тлумачному словнику української мови готовність визначається як певний стан, що виражає закінченість, кінцевий результат якої-небудь дії, етапу [2]. Згідно зі словником-довідником з професійної педагогіки, готовність (від англ. *readiness*) є цілісним утворенням, що характеризує емотивно-когнітивну та вольову мобілізаційність суб'єкта в момент його включення в діяльність певної спрямованості [149]. Готовність до діяльності не є природженою, а формується в результаті певного досвіду суб'єкта діяльності та визначається передусім позитивним ставленням до неї, усвідомленості мотивів та потреб у неї, об'єктивації її предмету та способів взаємодії з ним.

Виходячи з того, що готовність до діяльності може виявлятися у ситуативній, нетривалій (визначається окремими чинникам, конкретними обставинами діяльності, що вимагають тимчасового стану) та загальній, тривалій формах, професійна готовність особистості, на нашу думку, є тривалою за своєю суттю, складноорганізованою та є цілеспрямовано сформованою задля успішності виконання діяльності у певній галузі.

За результатами здійсненого нами аналізу наукової літератури з проблеми професійної готовності особистості виявлено декілька підходів щодо розуміння її сутності.

По-перше, - *особистісний підхід*, в якому вона визначається сукупністю індивідуально-особистісних якостей людини, особливим утворенням з різноманітних властивостей (інтелектуальних, емоційно-вольових, моральних тощо) та ставлень особистості (Л. Божович [127], Д. Леонт'єв [87] та ін.).

По-друге, - *діяльнісний підхід*, що акцентує увагу на готовності як функціональному стані, що забезпечується різними професійними вміннями особистості, у тому числі такими вміннями як: мобілізуватися, активізувати власні фізичні та психічні ресурси для здійснення діяльності в певних умовах, специфічні професійні вміння за сферою діяльності, тощо (Б. Бодров [15], Н. Левітов [86] та ін.).

По-третє, - *особистісно-діяльнісний* (інтегративний) підхід, в якому професійна готовність розглядається як єдність як особистісних, так і функціональних, діяльнісних характеристик (О.І. Гура [28], А. Деркач [39], та ін.). Саме особистісно-діяльнісний (інтегративний) підхід дає змогу визначити цілісний зміст такого явища як професійна готовність особистості і саме цей підхід був покладений нами в основу розуміння сутності професійної готовності майбутніх інженерів-програмістів до тестування ПЗ.

Відтак, *професійна готовність майбутніх інженерів-програмістів з тестування ПЗ*, на нашу думку, є складноорганізованим утворенням, інтегративним особистісно-професійним явищем, що є результатом їх цілеспрямованої, спеціально організованої професійної підготовки в умовах неперервної освіти, передумовою професійної компетентності та визначає успішність їх подальшої професійної діяльності та професійного розвитку.

Дослідження змісту професійної готовності майбутніх тестувальників ПЗ вимагає уточнення структури професійної компетентності та професійної готовності майбутніх інженерів-програмістів, що є базовими, загальними. Вони, у свою чергу, вивчалась у працях таких науковців як Л. Гришко, В. Круглік, Д. Мустафіна, В. Осадчий, П. Павленко, В. Седов, Д. Щедролосьєв та ін.

Визначення сучасних вимог до професійної компетентності інженерів-програмістів доцільно, на нашу думку, розпочати із твердження видатного радянського програміста та математика, який є одним з творців «шкільної інформатики», піонером у галузі теорії й автоматизації програмування та став методологом сучасної ІТ-науки - академіка Андрія Єршова: «Програміст має мати здатність першокласного математика до абстракції та логічного мислення у поєднанні з едісонівським талантом створювати все, що завгодно з нулів та одиниць. Він має поєднувати акуратність бухгалтера з проникливістю розвідника, фантазію автора детективних романів з тверезою практичністю економіста. Крім того, - він має мати вкус до колективної роботи, розуміти інтереси користувача та багато чого іншого» [47].

Як зазначав Д. Щедролосьєв [171], професійна компетентність інженерів-програмістів включає до себе: професійні знання (з технологій програмування, ведення проєктів, організації роботи в команді, стандартів якості, визначати архітектуру програми, управління ризиками та ін.) та професійні вміння (застосовувати і комбінувати різні прийоми програмування та алгоритми; модифікувати програму; бачити задачу на різних рівнях деталізації та ін.), а також такі якості як: наполегливість, уважність, відповідальність, працездатність, ініціативність, вміння працювати в колективі, комунікабельність, лідерство, гнучкість та стратегічність мислення, критичність, адекватна самооцінка тощо [171].

Професійна або фахова компетентність інженера-програміста, за твердженнями В. Седова, є системою здібностей, якостей, цінностей,

мотиваційних установок, знань і практичних навичок для здійснення професійної діяльності за фахом інженер-програміст, а також дослідника та викладача вищої школи (на освітньому рівні магістра), котрі забезпечують високий рівень професійної підготовки і свідомості [141]. За своєю структурою, на думку автора, вона включає декілька компонентів: 1) ціннісно-мотиваційний (мотивація професійної діяльності, підвищення її престижності, кар'єрного зростання); 2) рефлексивний (усвідомлене сприйняття себе суб'єктом навчально-виховного процесу з метою вибору стилю взаємодії, управління, спілкування; усвідомлення відповідальності за результат діяльності; здатність до усвідомлення та аналізу власних помилок; самооцінювання професійного рівня та визначення програми особистісного професійного розвитку); 3) змістовний (комплекс спеціальних (з предмета) знань, необхідних для продуктивної діяльності; аргументоване обґрунтування власних думок щодо вирішення професійних ситуацій); 4) операційно-технологічний (сукупність вмінь та навичок, необхідних для практичного вирішення завдань у процесі професійної діяльності, зокрема прийняття рішень у повсякденних і екстремальних умовах професійної діяльності, вибір технологій програмування, контроль за перебігом діяльності, оцінка діяльності учасників проєкту, адекватна самооцінка значущості своєї участі у спільній роботі; корекція власної поведінки) та 5) особистісний (сукупність професійно важливих особистісних якостей фахівця, важливих для виконання професійної діяльності) [141].

Проблема визначення змісту професійної компетентності ІТ-фахівців, як зазначають більшість науковців (О. Кучерук, І. Медзєбровський, С. Попершняк, В. Седов та ін.) полягає у швидкій зміні технологій програмування, швидкому старінні професійних знань, адже нові підходи та технології програмування з'являються не рідше ніж раз на десять місяців; принципово нові концепції та рішення – кожні три роки; зміна апаратних платформ та операційних систем – кожні п'ять років.

Тому дуже важливим є постійне оновлення професійних знань фахівців ІТ-сфери та формування їх нових умінь, а прагнення до професійного розвитку, самоорганізації та вміння знаходити нестандартні рішення, відкритість новому досвіду стають системоутворювальними складовими їх професійної компетентності. Саме вони є певними каталізаторами таких професійних умінь сучасних інженерів-програмістів як-то: вміння вирішувати завдання розробки, вибору та перетворення алгоритмів та математичних моделей; вміння використовувати сучасні ІТ та комп'ютерну техніку для побудови та обслуговування ІК систем у різних галузях науки та виробництва та ін. [83].

На думку В. Кругліка [78], структура професійної компетентності майбутніх інженерів-програмістів містить такі складові-компетентності як: 1) з програмування (здатність розробляти програмні продукти, будувати алгоритми та обирати раціональні, розробляти комплексні інформаційні рішення, користуватися сучасними парадигмами програмування, розробляти та реалізовувати процеси життєвого циклу інформаційних систем, ПЗ тощо); 2) цифрова (володіння сучасними методами ефективного доступу до інформації, її збору, систематизації та збереження; самостійно здобувати нові знання та вміння та ін.); 3) математична (здатність демонструвати знання фундаментальних і суміжних спеціальних дисциплін для вирішення прикладних завдань); 4) інженерна (уміння розробляти моделі різних типів залежно від завдання проєктування ПЗ, застосовувати методи і механізми оцінки та аналізу функціонування засобів і систем, розробки проєктної та програмної документації); 5) комунікативна (уміння спілкуватися, включаючи усну та письмову комунікацію українською та іноземними мовами); 6) особистісно-професійна (уміння враховувати вплив факторів зовнішнього середовища на результативність діяльності; розширювати і поглиблювати своє наукове світосприйняття), що забезпечується лабільністю мислення, здатністю до самонавчання та дивергентним мисленням; 7) управлінська (уміння

працювати у групах у розробці складних програмних комплексах, організувати власну діяльність та ефективно керувати часом).

Особливий акцент у питанні професійної компетентності сучасного інженера-програміста у науковій літературі робиться на його мисленнєвих якостях (Л. Гришко, М. Смульсон та ін.), а саме:

- комплексність мислення (здатність вільно переходити від опису задачі у загальних поняттях до нижчого рівня), вміння мисленнєвого проєктування; вміння узагальнювати типові ситуації; здатність аналізувати власні помилки [16];

- гнучкість та стратегічність мислення; творчі властивості мислення; уважність, що виявляється в умінні не припускати помилок; логічний характер мислення; оперативність мислення; вміння приймати рішення в умовах обмеженого часу; критичність мислення (М. Смульсон);

- розвинене абстрактне мислення, аналітичність, здатність до візуалізації [174].

Що стосується фахівців з тестування ПЗ, то зазначимо, що у науковій літературі проблема їх професійної компетентності є малодослідженою, окремі її питання розглядалися у працях А. Авраменко, В. Авраменко, Т. Баджетта, Г. Косенюка, В. Котлярова, В. Липаєва, Г. Майерса, Дж. Макгрегора, Н. Налютина, Д. Сайкса, К. Сандлера С. Синицина, I. Burnstein, G. Weinberg та ін. Проте більшість з учених наполягають на творчому характері професійної діяльності тестувальників ПЗ, на їх роботі в умовах невизначеності, постійної ітерації, а відтак, наголошують на важливості не тільки їх технічних, але і нетехнічних умінь та якостей [144].

Серед основних професійних вимог до тестувальника в сучасних ІТ-компаніях висувають такі як: 1) відповідальне ставлення до контролю якості виконаної роботи; 2) грамотне документування знайдених багів, їх опис та надання вичерпної інформації для того аби розробники в майбутньому могли їх пофіксити і уникнути в подальшому; 3) постійне навчання та розвиток (за даними <https://sqa.lviv.ua/vymohy-do-testuvalnyka-seredni-zarplaty-v-ukrayini-ta->

[sviti; https://melni.me/profesiya-testuvalnyk-programnogo-zabezpechennya-yak-yiyi-osvoyity-ta-shvydko-znajty-robotu/](https://melni.me/profesiya-testuvalnyk-programnogo-zabezpechennya-yak-yiyi-osvoyity-ta-shvydko-znajty-robotu/)).

Крім того, головними нетехнічними вимогами роботодавців до фахівців з тестування ПЗ (за результатами аналізу здійснених вакансій за даними: <https://melni.me/profesiya-testuvalnyk-programnogo-zabezpechennya-yak-yiyi-osvoyity-ta-shvydko-znajty-robotu/>) є такі: 1) відповідальність – повний контроль успіху виконуваного завдання, в тому числі не тільки своєї частини, а й фінального командного результату; 2) комунікативність - уміння чітко формулювати думки, намагатися бути ввічливим і уникати потенційно небезпечних та деструктивних тем, вміння аргументувати власну позицію, презентувати її; неконфліктність; 3) стресостійкість - здатність протистояти ескалаціям (наприклад, явній незадоволеності роботою з боку клієнта); здатність витримувати надмірно високе навантаження та відновлюватися після коротких перерв; вміння контролювати власну продуктивність; зосередженість на перевірці продукту та прискіпливість до деталей, уміння перечитувати та перевіряти написаний тест більше двох разів; 4) наставництво – уміння передавати свої знання та досвід іншим членам команди, направляти розвиток молодих спеціалістів у корисний компанії напрям; 5) делегація - можливість делегувати більшість видів діяльності підлеглим без ризику погіршення якості фінального результату; здатність правильно контролювати делеговані завдання; 6) планування і селф-менеджмент - здатність пріоритизувати та вчасно виконувати свої поточні завдання; відсутність необхідності в постійному контролі з боку керівництва; 7) командна робота – здатність вести ефективну і продуктивну діяльність у кооперації з колегами, досягати спільних цілей та оптимально використовувати спільні ресурси; здатність підтримувати позитивну атмосферу в команді.

Вкрай важливими професійними якостями фахівців також зазначаються якості мисленнєвої діяльності: аналітичність, оперативність та гнучкість мислення; нестандартність, нешаблонність мислення; дослідницькі вміння [195].

Ґрунтуючись на положеннях особистісно-діяльнісного підходу до розуміння сутності професійної готовності особистості, а також на підставі Рекомендацій Ради Європи щодо ключових компетентностей для освіти впродовж життя [133], що забезпечують відповідний рамковий інструментарій, структура готовності майбутніх фахівців з тестування ПЗ як передумови їх професійної компетентності має трикомпонентну будову, а саме:

- а) знання, що складаються з фактів, понять, ідей і теорій, що вже існують та допомагають покращити розуміння певної сфери діяльності;
- б) уміння та навички як здатність здійснювати діяльність та використовувати наявні знання для досягнення результатів;
- в) ставлення як певна налаштованість та спосіб мислення, що визначають дії та реакцію на ідеї, людей та ситуації.

Отже, спираючись на результати вищезазначеного аналізу особливостей діяльності тестувальників ПЗ, основних її завдань та етапів (див. табл. 1.1), готовність майбутніх фахівців з тестування ПЗ за своїм змістом, на нашу думку, складається з таких компонентів як:

1) *ціннісно-мотиваційний*, що визначається мотивацією професійної діяльності майбутніх фахівців з тестування ПЗ та подальшого їх професійного розвитку і навчання, ціннісним ставленням до виконання професійних обов'язків, спрямованістю на досягнення результату;

2) *когнітивний*, який віддзеркалює систему їх загальних та спеціальних професійних знань, у тому числі: методології розробки ПЗ; фундаментальних основ тестування, технік тест-дизайну, принципів роботи допоміжного ПЗ для тестування; основ естімації та планування; основ алгоритмів, промислового програмування, процесів та методології розробки; архітектури комерційних додатків; мережевих технологій і адміністрування; операційних систем, засобів віртуалізації;

3) *операційно-діяльнісний*, що складається з двох груп професійних умінь:

а) *технічні вміння*, що відображають завдання/етапи професійної діяльності тестувальника ПЗ, а саме: аналізувати вимоги на предмет їх

адекватності, здійсненності та тестованості; аналізувати, створювати і підтримувати тестову документацію, взаємодіяти з додатком на призначеному для користувача і розробницькому рівні, проходити тестові сценарії і коректно описувати дефекти тощо;

б) *нетехнічні вміння* (soft skills), що відображають вимоги до їх:

- мисленнєвої діяльності – *мисленнєві вміння* (аналітичність, оперативність та гнучкість мислення, нестандартно мислити, самостійність мислення);

- спілкування - *комунікативні вміння* (вміння працювати у команді, підтримувати позитивну атмосферу, домовлятися, комунікувати з членами команди та з замовником, обґрунтовувати і відстоювати свою позицію, вміння індивідуального підходу, неконфліктність);

- регуляції власної поведінки та діяльності – *регулятивні вміння* (самоконтролю власної діяльності, самоконтролю власних емоцій, активізації працездатності, вміння керувати часом, бути зосередженим та уважним);

- метапізнавальної діяльності – *метакогнітивні вміння* (вміння проблематизації, проєктувальні вміння, планування, прийняття рішення в умовах невизначеності), включаючи рефлексивні вміння (аналізувати, оцінювати власну діяльність, знаходити помилки у неї та коригувати власну мисленнєву діяльність).

Зазначені складові професійної готовності майбутніх інженерів-програмістів до тестування ПЗ, презентовані нами на рис. 1.5., з одного боку, є рівнозначними та тільки у своїй єдності забезпечують цілісне утворення – професійну готовність та у подальшому, в умовах реальної професійної діяльності – професійну компетентність тестувальників ПЗ. А з іншого боку, вони мають свою логіку розгортання відповідно до положень діяльнісного /системодіяльнісного підходу (В. Шадриков): ціннісно-мотиваційний компонент є системоутворювальним та зумовлює усі інші, а когнітивний забезпечує створення орієнтовної основи дії – реалізації операційно-діяльнісного компоненту, що забезпечує перехід від теорії до практики.

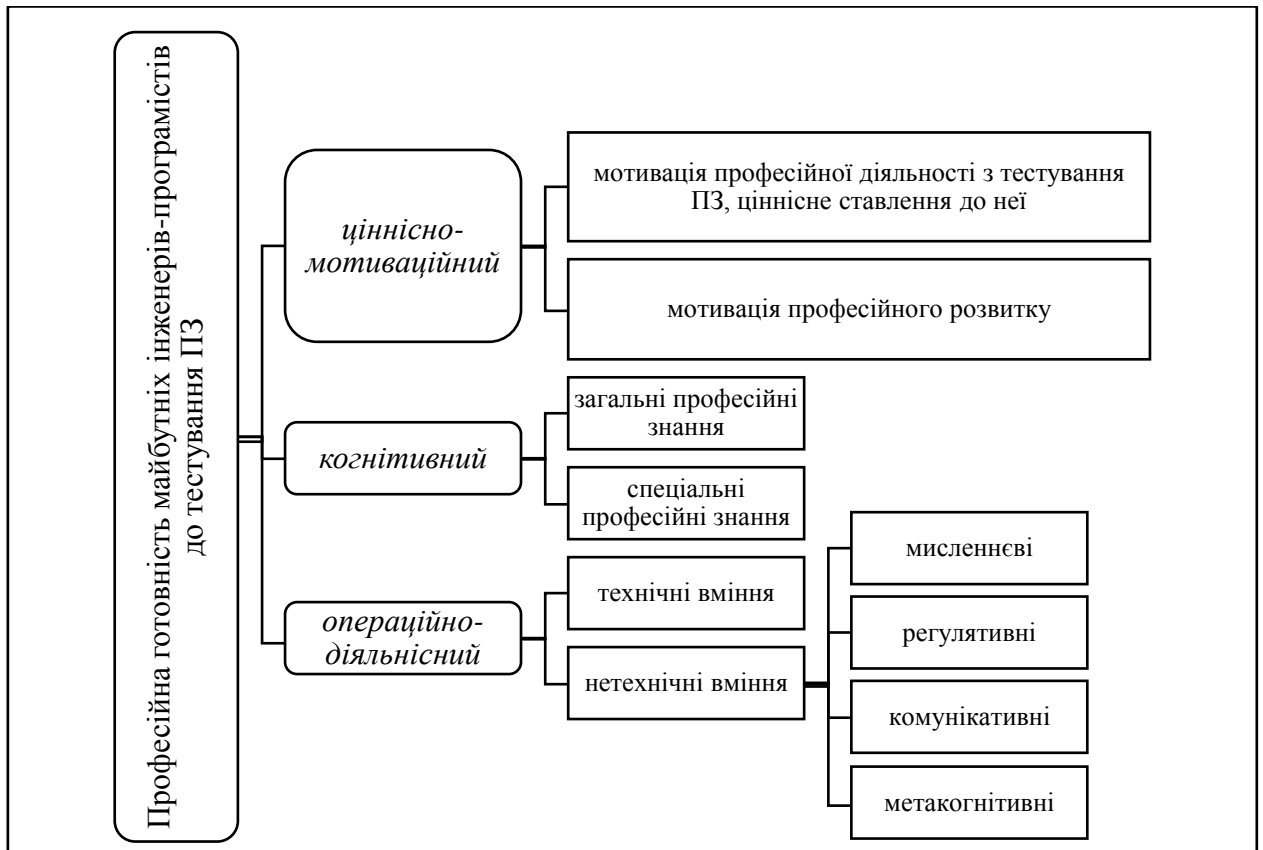


Рис. 1.5. Структурна організація професійної готовності майбутніх інженерів-програмістів до тестування ПЗ

Таким чином, професійна підготовка інженерів-програмістів до тестування ПЗ має забезпечити формування їх цілісної професійної готовності в єдності ціннісно-мотиваційного, когнітивного та операційно-діяльнісного компонентів.

### 1.3. Зарубіжний та вітчизняний досвід професійної підготовки майбутніх інженерів-програмістів до тестування програмного забезпечення

Стрімке зростання популярності і затребуваності інформаційних технологій як у виробничій і бізнес, так і в освітній сфері, що спостерігається в Україні, характерне і для всіх інших світу.

Будучи головними центрами зосередження найбільш успішних світових ІТ-компаній, Європа та Америка є законодавцями моди в контексті розвитку ІТ, підходів до побудови процесів розробки, підтримки і просування ПЗ.

Розвиток і зміцнення ринку ІТ закономірно позначається на затребуваності фахівців відповідного профілю, що, в свою чергу, стимулює зарубіжні університети відкривати нові спеціальності, додавати дисципліни та актуалізувати навчальні програми, щоб відповідати сучасному ринку і забезпечувати його новими кадрами.

ІТ-галузь в Україні, в свою чергу, є досить молодою та сформованою в першу чергу під впливом зарубіжних тенденцій: переважна частина українських ІТ-роботодавців є аутсорс-філіями Європейських і Американських компаній, що, в свою чергу, позначається на загальних підходах до організації робочого процесу, менеджменту, виборі технологій, складах команд і методологій розробки програмних продуктів. Схожа ситуація простежується, зокрема, й із статусом тестування: як в зарубіжних країнах, так і в Україні, тестування ПЗ не виділяється офіційними класифікаторами в окрему професію, але є однією з обов'язкових компетенцій інженера-програміста [103].

Зорієнтованість сучасної української ІТ-сфери на західні вимоги, наслідуваність усіх підходів і технологій, а також однаковий правовий статус професії інженера-програміста в цілому та тестувальника ПЗ зокрема, дозволяють зробити висновок про схожі вимоги до процесу підготовки фахівців в галузі, що, в свою чергу, зумовлює актуальність вивчення зарубіжного досвіду підготовки інженерів-програмістів до тестування.

При аналізі зарубіжного досвіду професійної підготовки фахівців ІТ-сфери в закладах вищої освіти, на наш погляд, перш за все доцільно акцентувати увагу на тому, що починаючи з кінця 1960-х

років започатковано і в подальшому реалізується міжнародний проєкт «Computing Curricula», спрямований на систематизацію і видання всесвітнього досвіду викладання програмної інженерії в університетах і коледжах. Згідно з цим документом, який по своїй структурі має рекомендації щодо розробки та змісту навчальних планів, переліку дисциплін, змісту програм навчання, форм та методів викладання, в подальшому розроблюється і зміст професійної підготовки ІТ-фахівців в більшості зарубіжних закладів освіти. Він є певним стандартом, що періодично оновлюється і визначає вектор подальшого розвитку ІТ-освіти.

Так, наприклад «Software Engineering 2004: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering» в контексті нашого предмету дослідження, передбачає наявність в змісті підготовки обов'язкового курсу «SE221. Тестування програмного забезпечення», який має розкрити всі аспекти тестування, а також інші аспекти верифікації і атестації, враховуючи визначення вимог, що тестуються, рецензування і якості продукту» [132].

Причому в рекомендаціях до курсу зазначено, що він має забезпечити розвиток практичних навичок і надати студенту досвід (бажано шляхом його участі в розробці реальних програм) [132].

Заслуговують на увагу і загальні рекомендації до розробки навчальних планів і практичної підготовки, зокрема:

- студенти мають отримувати навички, що виходять за рамки предмету (рекомендація 8);
- у навчанні мають використовуватись сучасні засоби, навіть якщо вони не є метою навчання (рекомендація 12);

- навчальний план має базуватись на реальному практичному досвіді (рекомендація 14);
- навчання програмній інженерії має вийти за межі лекційного формату, тому слід звернути увагу на важливість навчання різноманітним підходам до викладання й отримання знань (рекомендація 17) та ін. [132].

Більш нова ревізія даного документа, «Software Engineering 2014: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering», показує, наскільки змінилася та еволюціонувала індустрія розробки ПЗ і, відповідно, підходи до підготовки молодих фахівців. Самі автори роботи стверджують, що нове видання враховує появу альтернативних моделей і процесів в життєвому циклі розробки ПЗ, в тому числі, орієнтованих на гнучкість, а також робить акцент на аналізі вимог до програмного продукту, забезпечення якості та безпеки, оскільки ці теми викликають все більший інтерес роботодавців [132].

У рекомендаціях з розробки навчальних планів додалася готовність до постійного розширювання різноманітності сфер застосування програмної інженерії (нові галузі бізнесу, медіа тощо) [132], а також увага до всіх другорядних галузей знань, що застосовуються в індустрії, таких як менеджмент, тестування, безпека та ін. Більш того, акцент ставиться і на знання «доменів» - цільових галузей застосування кінцевих програмних продуктів, таких як торгівля, логістика, медіа, охорона здоров'я. Вивчення такого матеріалу дає студентам безпосередні знання в предметній галузі, які вони можуть застосувати до завдань програмної інженерії, збільшуючи свою ефективність за рахунок розуміння проблем бізнесу та тенденцій індустрії [132].

Крім технічних знань, помітний акцент робиться і на розвитку нетехнічних умінь молодих спеціалістів, таких як критичне мислення, оцінне сприйняття, ефективна комунікація. Вони, на думку авторів, є

обов'язковими майже для всіх сфер реалізації молодих спеціалістів та видів діяльності, з якими вони зіткнуться на робочому місці [132].

У рекомендаціях щодо вибору форм і методів навчання ключовий акцент робиться на реальному виробничому процесі розробки як основі побудови теоретичних та практичних знань студентів. Серед класичних форм і методів навчання в програмі повинні також бути присутніми [132]:

- тематичні дослідження: знайомство з реальними системами і дослідження комерційних проєктів. Студенти повинні вміти критикувати ці приклади, шукати закономірності і причинно-наслідкові зв'язки, використовувати кращі практики;
- проєктні заходи: імітації типових виробничих проєктів, що повинні включати групову роботу, презентації, формальні огляди, контроль якості;
- заключний проєкт: командний проєкт, що дозволяє максимально наблизитися до індустріального процесу розробки і застосувати на практиці весь обсяг отриманих за курс знань;
- виробнича практика: залучення студентів до роботи на реальних комерційних проєктах під керівництвом кваліфікованих фахівців заради отримання релевантного практичного досвіду. Участь студентів в проєктах з відкритим вихідним кодом - ще один можливий підхід до надання такого досвіду.

Таким чином, на прикладі аналізу двох документів стає очевидним вектор розвитку як самої ІТ-індустрії, так і підходів до підготовки відповідних фахівців: орієнтованість на клієнта, якість, командну роботу, процеси, розуміння доменної галузі та практичне застосування технічних знань.

Одними з амбасадорів розвитку ІТ у світі є Сполучені Штати Америки, які стали батьківщиною таких ІТ-гігантів як Google, Microsoft, Facebook і Amazon. В цілому по країні фахівців ІТ-сфери готують понад 220 закладів вищої освіти, значна частина яких входить до світових академічних рейтингів: Стенфордський університет, Массачусетський

технологічний університет та Каліфорнійський університет в Берклі займають перші місця. Підготовка у цих закладах здійснюється за такими основними спеціальностями як: комп'ютерні науки (Computer science), обчислювальна техніка (Computer engineering), інформаційні технології (Information Technology), інформаційні системи (Information Systems) і програмна інженерія (Software Engineering). Програми підготовки пропонуються на рівнях асоціату (Associate's Degree in Software Engineering), бакалаврату (Bachelor's Degree in Software Engineering) і магістратури (Master's Degree in Software Engineering) і є подібними за змістом, а навчальні плани поєднують загальнонаукову - перші два роки (гуманітарні, математичні та природничі курси) та професійну підготовку (обов'язкові курси, курси за вибором а також вільні курси) та включають спільне навчання, практику, дослідницьку роботу і валеологію [77].

Так, С. Конюхов, здійснюючи аналіз закордонних досліджень підготовки майбутніх інженерів програмістів до об'єктно-орієнтованого програмування, акцентує увагу на тому, що у закладах освіти США досить поширеним є ігрові та проєктні методи підготовки. Зокрема, ігрові методи мотивують і підтримують інтерес здобувачів освіти до навчання, дозволяють наочно продемонструвати окремі концепції. Наприклад, дослідниками університетів Вінстон-Сейлем та Боуї розроблені на основі ігор спеціальні програмні засоби, які використовуються в освітньому процесі. Використання проєктного підходу дає можливість змоделювати реальний процес промислової розробки програмних проєктів, де студенти приймають участь як професійні програмісти [70].

Широкого розповсюдження на рівні з традиційними формами і методами підготовки набули навчальні інформаційні on-line ресурси (Complete List) для самостійної роботи та самовдосконалення. Один з перших таких відкритих ресурсів - Open Course Ware (MIT OCW) - було створено у 2001 році за ініціативи Массачусетського технологічного інституту, в якому інтернет-користувачі мали вільний доступ до

розроблених навчальних матеріалів. Ця ініціатива в подальшому отримала поширення в більшості університетів Америки та Європи [67].

Одним з найпопулярніших і затребуваних навчальних закладів США є Каліфорнійський Політехнічний Університет. Інженерний факультет університету готує бакалаврів за освітніми програмами «Комп'ютерна Інженерія», «Комп'ютерні науки» та «Програмна інженерія»[177]. Освітня програма підготовки бакалаврів за спеціальністю «Software Engineering» (Програмна інженерія) включає в себе велику кількість як загальноосвітніх і гуманітарних, так і професійних дисциплін, в тому числі за вибором, і ставить собі за основну мету різнобічний розвиток студента в контексті різноманіття спеціалізацій і напрямів їх подальшої діяльності, про що свідчить наявність таких обов'язкових дисциплін як: розробка ігор (CSC 371. Game Design), розробка мобільних додатків (CSC 436. Mobile Application Development), комп'ютерна творчість (CSC 311. Computational Art.)[177]. Навчальний план не містить окремих дисциплін з тестування ПЗ. Однак, тестування виділяють як обов'язковий етап розробки і вивчають в контексті кожної дисципліни, в якій воно може бути присутнім згідно з сучасними процесами та методологіями. Відповідні модулі з тестування наявні в програмах таких курсів як «фундаментальні основи комп'ютерних наук» (CSC 101. Fundamentals of Computer Science) – вивчення тестування як одної з основних стадій розробки ПЗ; «структури даних» (CSC 202. Data Structures) – тестування в контексті аналізу структур даних та алгоритмів вивчення їх продуктивності і вузьких місць; «розробка ігор» (CSC 371. Game Design) – особливості тестування у контексті розробки комп'ютерних ігор, аналіз специфічних практик та інструментів. Щодо формування проєктного досвіду та розвитку нетехнічних умінь та навичок студентів, навчальна програма включає в себе дослідницький персональний проєкт (CSC 497. Research Senior Project), основною метою якого є закріплення теоретичних знань на практиці та створення власного програмного продукту під наглядом викладача, та командний проєкт (CSC

406. Senior Project), основною метою якого є формування у студентів навичок командної роботи завдяки розробці та розгортанню масштабного комерційного проєкту з його подальшою підтримкою.

Схожої політики щодо підготовки студентів-програмістів дотримується ще один представник найбільш рейтингових Американських навчальних закладів – Мічіганський технічний університет (Michigan Technical University). Навчальна програма підготовки бакалавра за спеціальністю «програмна інженерія» (Software Engineering SSEN) містить перелік обов’язкових дисциплін, завдяки яким надається базове теоретичне та практичне підґрунтя щодо основних практик та технологій розробки ПЗ: «введення до програмування» (CS 1121 - Introduction to Programming) - теми включають проєктування, програмування, документацію, налагодження та тестування програм; системне програмування (CS3411 – Systems Programming) - курс знайомить із технологіями проєктування ПЗ, використовує UML для вимог та специфікації проєкту, а також досліджує функціонування, тестування та документацію в контексті великого командного проєкту [204]. Незважаючи на те, що курси з програмування також містять окремі модулі з тестування, у навчальній програмі присутній і окремий обов’язковий курс «Забезпечення якості програмного продукту» (CS 3712 - Software Quality Assurance) [204], що детально і послідовно досліджує практики забезпечення якості на всіх етапах процесу розробки програмного продукту, а модулі курсу включають такі проблемні теми як: визначення вимог, аналіз та документація, тестування та управління якістю. Окрім теоретичного аспекту підготовки, програма передбачає і практичну складову: командний проєкт (CS3141 Team Software project), що ставить за мету виконання командою студентів навчального проєкту за всіма стандартами індустрії. Більш того, університет пропонує і неформальний тип навчання - спеціальну програму «The Enterprise Program», яка передбачає проходження студентами виробничої практики та стажування у реальних ІТ-компаніях. Програма розроблена за

підтримки великої кількості підприємств-партнерів та пропонує різні професійні напрями, а студенти проходять повний цикл працевлаштування, включаючи складання резюме, співбесіду та стажування. Практика повністю узгоджена з розкладом лекцій і може бути зарахована замість дисциплін за вибором. За результатами її проходження багато студентів залишаються в командах і отримують повноцінне місце в компанії [204].

Зазначені підходи до підготовки майбутніх інженерів-програмістів до тестування ПЗ практикує й один з найпопулярніших технічних навчальних закладів Австралії - Університет Дікін, розташований в Мельбурні. Освітня програма з підготовки «Bachelor of Software Engineering» (Бакалавр програмної інженерії), аналогічно до американських прикладів, не містить обов'язкових дисциплін з тестування ПЗ, проте містить модулі з тестування у програмах дисциплін: «Вступ до програмування» (SIT102 - Introduction to Programming), «Об'єктно орієнтоване програмування» (SIT232 - Object-Oriented Development) та ін. [185]. Різноманіття спеціалізацій та різнобічний розвиток студентів забезпечують дисципліни за вибором, в числі яких - SIT707 - Software Quality and Testing («Якість програмного забезпечення та тестування»), яка розглядає сферу тестування ПЗ, включаючи роль тестування протягом життєвого циклу, процеси та стандарти тестування, а також інструменти та технології для тестування та відстеження дефектів [Deakin University. SIT707 - Software Quality and Testing. Unit details]. Щодо розвитку практичних навичок студентів, програма також передбачає організацію освітніх проєктів завдяки спеціальному курсу (SIT374 - Team Project - Project Management and Practices), що надає можливість студентам взяти участь у реальному, галузевому ІТ-проєкті та продемонструвати знання та навички, які вони здобули за час навчання. Присутня у курсі і практика на підприємстві (SEP499 - Professional Engineering Practice), яка дає змогу

студентам брати участь в інтегрованому навчанні в організації-партнері (або організаціях) терміном мінімум 30 робочих днів (225 годин). [185].

Схожі освітні практики підготовки інженерів-програмістів присутні й у Європейських високореєтингових навчальних закладах. Програма підготовки бакалавра «SOFTWARE DEVELOPMENT AND ENTREPRENEURSHIP» (Розробка програмного забезпечення і підприємництво) Університету прикладних наук Естонії, розташованого у місті Таллінн, складається з 6 глобальних модулів: «Підприємництво» (Entrepreneurship module, 15 ECTS CREDIT), орієнтований на створення бізнес-ідеї, розробку та процес впровадження, а також дослідження теми розвитку та управління підприємництвом; «Базові дисципліни» (Basic courses module, 22 ECTS CREDIT), такі як «Естонська мова», «Психологія та саморозвиток»; «Управління IT-проєктами» (IT project management module, 11 ECTS CREDIT), що включає такі дисципліни, як «Управління програмними продуктами та послугами», «Бухгалтерський облік та бюджетування»; «Математики та методології досліджень» (Mathematics and research methodology module, 30 ECTS CREDIT); «Розробка програмного забезпечення» (Software development and entrepreneurship specialisation module 62 ECTS CREDIT); фінальні іспити та інтернатура (Final thesis and internship 40 ECTS CREDIT) [188].

Варто зазначити, що модуль з розробки програмного забезпечення має окремий обов'язковий до вивчення курс з тестування ПЗ (SE-018 - Software testing) на 4 кредити ECTS. Згідно з програмою, курс уводить в термінологію, пов'язану зі сферою тестування. Студенти вчаться планувати тестування на проєктах різних типів та за моделями життєвого циклу ПЗ, писати тести за допомогою різних технік проєктування [188]. Щодо проєктної роботи та стажування, всі ці освітні практики присутні у останньому з модулів: для отримання диплому бакалавра студенти зобов'язані успішно завершити командний проєкт з розробки, в ході якого розвиваються навички комунікації та командної роботи, а також завершити

практику на підприємстві. Практика та дипломна робота виконуються одночасно і вони тісно пов'язані між собою - під час спеціальної практики студент збирає / впроваджує / створює матеріал, що стає основою все самої дипломної роботи.

Європейський підхід до підготовки інженерів-програмістів є актуальним і для одного з найбільш затребуваних навчальних закладів Швеції – Королівського технологічного інституту у місті Стокгольм, який готує бакалаврів за спеціальністю «Інформаційні і комунікаційні технології» [199.] (Bachelor's Programme in Information and Communication Technology). При наявності класичних базових курсів таких дисциплін, як «Програмування» (ID1018 Programming) чи «Операційні системи» (ID1206 Operating Systems), особливістю цієї програми є велика кількість різнопрофільних дисциплін за вибором, які, починаючи вже з другого курсу навчання, складають значну частину розкладу. До них відносяться не тільки специфічні технічні дисципліни, такі як: «Мережеве програмування» (ID1212 Network Programming), «Контроль якості програмного забезпечення» (MG2033 Quality Control) чи «Розробка мобільних додатків» (ID2216 Developing Mobile Applications), але й соціальні та управлінські дисципліни: «Менеджмент проєктів» (ME2016 Project Management: Leadership and Control), «Управління персоналом та організаціями» (ME2163 Leading People and Organizations in Different Contexts) [199]. Дипломний проєкт є заключною частиною навчання, передбачає синтез та практичне застосування раніше набутих знань, при цьому знайомлячи студента з основами командної роботи та рольової моделі сучасної індустрії розробки ПЗ.

Незважаючи на очевидні історичні, культурні та соціальні особливості, загальний підхід до складання програм підготовки інженерів-програмістів є незмінним навіть для країн східної Азії. Наньянський технологічний університет, розташований в Сінгапурі, пропонує підготовку бакалаврів за спеціальностями «Комп'ютерна інженерія»

(Computer Engineering) та «Комп'ютерні науки» (Computer Science) [207]. Варто зазначити, що окрім класичних програм підготовки, університет також пропонує дві гібридні програми подвійних ступенів: «Бізнес та комп'ютерна інженерія» та «Бізнес та комп'ютерні науки». Програми подвійного диплому об'єднують дві дисципліни, розширюючи тим самим спектр навичок і сам класичний курс спеціальними модулями бізнес-спрямування. Базовий курс з підготовки, аналогічно західним практикам, пропонує як такі загальні технічні дисципліни як-то: «Програмна інженерія» (CZ2106 Software Engineering), «Створення та аналіз алгоритмів» (CZ2101 Algorithm Design and Analysis), «Комп'ютерні мережі» (CZ2108 Computing Networks) та ін., так і спеціалізовані курси за вибором: «Криптографія та мережева безпека» (CE/CZ4024 Network Security), «Віртуальна реальність» (CE/CZ4001 Virtual and Augmented Reality), «Технології блокчейн» (CE/CZ4153 Blockchain Technology) [207]. Варто зазначити, що тестування ПЗ не представлено обов'язковою або додатковою дисципліною, проте, теоретичні навички щодо нього формуються завдяки окремим модулям загальних курсів, таких як «Програмна інженерія» (CZ2106 Software Engineering), а практичні – у процесі виконання мультидисциплінарного командного проєкту (CZ2179 Multidisciplinary Design Project), у ході якого студенти проходять повний цикл розробки програмного продукту, знаходячись у ролі одного з членів команди. Практика на підприємстві (CZ3179 Professional Internship) є обов'язковою для кожного студента і проходить протягом цілого семестру, під час якого молодий фахівець має змогу працювати над комерційними проєктами у командах-партнерах університету з подальшою перспективою повного працевлаштування.

Стрімке зростання сфери ІТ в цілому і розробки ПЗ зокрема стався і на території країн післярадянського простору на початку 21 століття, що спричинений як приходом на ринок філій великих іноземних ІТ-компаній, так і появою вітчизняних, побудованих за зразком перших. Ринок ІТ,

створений під впливом західного досвіду, методологій і процесів, перейняв і вимоги до фахівців, які, в свою чергу, і визначили стандарти професійної підготовки відповідних спеціальностей.

Підготовка фахівців ІТ-сфери в Республіці Білорусь здійснюється в межах 20 спеціальностей в 16 закладах вищої освіти. Інженерів-програмістів готують в межах спеціальності 40.01.01. Програмне забезпечення інформаційних технологій [105]. Зміст підготовки розроблюється на основі єдиних освітніх стандартів вищої освіти I та II ступенями, які визначають характеристику професійної діяльності (сфери, об'єкти, види і завдання діяльності, вимоги до професійних компетентностей). Результати здійсненого порівняльного аналізу освітніх стандартів ІТ спеціальностей Білорусії з міжнародними стандартами підготовки ІТ-фахівців [169] засвідчили про те, що підготовка інженерів-програмістів найбільш зорієнтована на міжнародну практику. Зокрема, відповідними є специфікація обсягу знань та перелік необхідних для фахівця компетентностей, які включають і компетентності з тестування ПЗ: володіння сучасними методами, мовами, технологіями та інструментальними засобами проєктування і розробки програмних продуктів; володіння принципами та основними навичками, прийомами, методами настройки, адаптації й супроводу програмних засобів; аналіз й обґрунтування вибору технічних, програмних засобів і систем для автоматизованої підтримки процесів професійної діяльності; розробка програмних засобів і системне забезпечення автоматизованої підтримки рішення задач професійної діяльності; здійснення контролю ефективності використання розрахункових засобів й інформаційних систем в професійній діяльності; здійснення тестування програмної продукції та програмних засобів, що використовуються у відповідності до технічних вимог та ін. [105].

Білоруський національний технічний університет, розташований в місті Мінськ, є другим у рейтингу кращих університетів країни. Кафедра

«Програмне забезпечення інформаційних систем і технологій» здійснює підготовку фахівців за профілем освіти «Техніка і технології» напряму освіти «Обчислювальна техніка за такими спеціальностями» «1-40 01 01 Програмне забезпечення інформаційних технологій», «1 40 05 01 04 Інформаційні системи і технології в обробці і представленні інформації», «1-40 05 01-01 Інформаційні системи і технології в проектуванні і виробництві» [10].

У межах програми підготовки бакалаврів за спеціальністю «Програмне забезпечення інформаційних технологій» вивчаються фундаментальні основи апаратного і програмного забезпечення комп'ютерів, технології створення програмних систем і обчислювальних комплексів, організації баз даних, питання захисту інформації, розробки локальних і мережових додатків. Серед дисциплін, що вивчаються присутні як загальноосвітні та інженерні дисципліни, так і ряд спеціальних таких, як «Основи алгоритмізації та програмування», «Об'єктно-орієнтоване програмування», «Технології програмування», «Організація і функціонування ЕОМ», «Операційні системи», «Комп'ютерні мережі », що базуються на вивченні мов програмування C ++, C #, Asp, Visual Basic, HTML, PHP, технологій .NET і Java. У низці курсів («Системи комп'ютерної графіки», «3-D моделювання інженерних конструкцій») викладаються прийоми роботи з комп'ютерною графікою, отримання і обробки зображень системами автоматизованого проектування. Вивчаються курси ( «Бази даних», «Бази знань і підтримка прийняття рішення в САПР») в галузі проектування баз даних і додатків баз даних, технологій передачі і зберігання даних, систем управління базами даних. Серед форм підготовки представлена в тому числі і командна робота, що дозволяє студентам вивчити рольову модель сучасних процесів розробки і розвинути такі нетехнічні навички, як комунікативність і критичне мислення [10].

Особливістю напряму підготовки є можливість вибору спеціалізації «1 40 01 01 05 Управління якістю і тестування програмного забезпечення», яка визначає наявність в навчальній програмі цілої низки курсів, присвячених проблемі тестування ПЗ: «Модульне тестування», «Тестування продуктивності», «Тестування і налагодження програмного забезпечення». Навчальні матеріали останнього, що включають в себе такі теми і модулі як «Тестування вимог» і «Автоматизоване тестування», дозволяють зробити висновок про наявність поглибленого вивчення змісту тестування і розгляду його не як додатку процесу розробки, а як окремої наукової галузі [11].

На дев'ятому семестрі навчання у процесі написання дипломної роботи студенти проходять переддипломну практику. Протягом чотирьох тижнів студенти беруть участь у виробничому процесі підприємств-партнерів, серед яких є як державні («БПС-Сбербанк», «Белкомкомплекспроект»), так і філії приватних міжнародних ІТ-компаній (Eram Systems). Студенти отримують досвід командної взаємодії, реальних виробничих процесів та одночасно вивчають вимоги актуальних роботодавців, які, в свою чергу, запрошують майбутніх випускників на стажування і подальшу роботу.

На концептуальному рівні подібний підхід підготовки інженерів-програмістів характерний і для освітніх закладів Російської Федерації. Згідно з офіційним федеральним державним освітнім стандартом вищої освіти за напрямом підготовки «09.00.00 Інформатика та обчислювальна техніка» відбувається підготовка бакалаврів спеціальностей як-то: «09.03.01 Інформатика та обчислювальна техніка», «09.03.02 Інформаційні системи і технології», «09.03.03 Прикладна інформатика», «09.03.04 Програмна інженерія», що здійснюється в 64 університетах країни [120].

Порівняльний аналіз вимог до підготовки фахівців з ІТ в вишах Росії та за кордоном дозволяє зробити висновок, що, незважаючи на деякі відмінні риси, характерні російськими програмами підготовки інженерів-

програмістів (підвищена увага до фундаментальної підготовки, наявність в ФГОС серед обов'язкових дисциплін з електротехніки і схемотехніки, а також відсутність розділу «Соціальні та професійні питання», в рамках якого передбачено розгляд питань професійної та етичної відповідальності, інтелектуальної власності, соціального контексту інформатики), можна констатувати, що основні вимоги до підготовки бакалаврів в галузі інформатики та обчислювальної техніки і програмної інженерії, встановлені ФГОС ВПО в Російській Федерації та рекомендаціями АСМ IEEE Computer Society практично збігаються [58].

Одним з найбільш рейтингових і популярних технічних університетів країни є МГТУ імені Баумана, в якому готують інженерів-програмістів за спеціальностями «Бізнес-інформатика», «Інформатика та обчислювальна техніка», «Інформаційні системи і технології», «Прикладна інформатика» та «Програмна інженерія» [96].

Навчальний план підготовки бакалаврів за спеціальністю «Програмна інженерія», аналогічно західним програмам, містить ряд обов'язкових спеціалізованих дисциплін, що розширюють сфери професійної самореалізації молодих фахівців, такі як «Бездротові технології передачі даних», «Хмарні технології», «Розробка мобільного програмного забезпечення». Проблема тестування ПЗ не виноситься в окрему дисципліну, але розглядається як модуль в таких базових курсах, як «Основи програмної інженерії», «Об'єктно орієнтоване програмування», «Високорівневе програмування» [95].

Для формування у молодих фахівців бачення сфери ІТ та розуміння контексту їх майбутньої професійної діяльності, в межах курсу так само проводиться кілька блоків практики: «навчально-технологічна», що за своєю суттю є навчальним проектом, участь в якому стимулює розвиток нетехнічних навичок студентів, і «виробнича», що дозволяє четвертокурсникам займатися професійною діяльністю на підприємствах-

партнерах протягом чотирьох тижнів. Досвід, отриманий студентами протягом практики, стає основою написання дипломної роботи.

Здійснений аналіз досвіду підготовки інженерів-програмістів в зарубіжних закладах вищої освіти, розглянутих вище, дозволяє виділити наступні тенденції в складанні навчальних планів та організації освітнього процесу.

По-перше, - наявність міцного теоретичного технічного базису: незалежно від особливостей спеціалізації і набору відповідних дисциплін, перші два роки навчання молодих фахівців проходить у вивченні загальних технічних дисциплін, таких як вища математика, фізика та основи електроніки, що дозволяє сформувати теоретичне підґрунтя для вивчення спеціалізованих дисциплін.

По-друге, - акцентування уваги на отриманні різноманітного практичного досвіду в спеціалізованих сферах: в силу різноманітності сфер реалізації інженерів-програмістів, освітні програми включають в себе технічні галузі, суміжні з безпосередньою розробкою, такі як «Тестування програмного забезпечення», «Мережева інженерія», «Інформаційна безпека».

По-третє, - розвиток нетехнічних знань і умінь студентів: з огляду на підвищення ролі м'яких умінь на ринку праці, освітні програми включають до себе застосування різноманітних форм і методів, спрямованих на розвиток відповідних навичок. До них відносяться: групова і командна робота, організація навчальних та виробничих практик, захист індивідуальних та групових дипломних проєктів.

По-четверте, - застосування неформальних форм освіти та організації практики: обов'язковим критерієм проходження програми підготовки для переважної більшості вишів є проходження виробничої практики в підприємствах-партнерах, які працюють в предметній галузі та залучених до створення програми практики. Позааудиторна робота і взаємодія з

діючими фахівцями в реальному виробничому середовищі дозволяє сформувати у студентів практичний досвід і реальне бачення IT-індустрії.

В Україні підготовку інженерів-програмістів, за даними єдиної державної електронної бази з питань освіти (ЄДБО) станом на 2020 рік, здійснюють в межах галузі знань 12 «інформаційні технології» спеціальності 121 «Інженерія програмного забезпечення» 57 закладів вищої освіти (<https://vstup.edbo.gov.ua/>) (табл. 1.2).

Таблиця 1.2

Перелік вітчизняних закладів вищої освіти, які здійснюють підготовку фахівців спеціальності 121 «Інженерія програмного забезпечення»

№	Назва закладу вищої освіти
1	Вінницький національний технічний університет
2	Державний вищий навчальний заклад «Донецький національний технічний університет»
3	Державний вищий навчальний заклад «Прикарпатський національний університет імені Василя Стефаника»
4	Державний вищий навчальний заклад «Ужгородський національний університет»
5	Державний вищий навчальний заклад «Український державний хіміко-технологічний університет»
6	Державний заклад «Луганський національний університет імені Тараса Шевченка»
7	Державний університет «Житомирська політехніка»
8	Державний університет інфраструктури та технологій
9	Державний університет телекомунікацій
10	Дніпровський державний технічний університет
11	Дніпровський національний університет залізничного транспорту імені академіка В. Лазаряна
12	Дніпровський національний університет імені Олеса Гончара
13	Запорізький національний університет
14	Івано-Франківський національний технічний університет нафти і газу
15	Київський національний торговельно-економічний університет
16	Київський національний університет імені Тараса Шевченка
17	Криворізький національний університет
18	Луцький національний технічний університет
19	Львівський національний університет імені Івана Франка
20	Національна металургійна академія України
21	Національний авіаційний університет
22	Національний аерокосмічний університет ім. М.Є. Жуковського «Харківський авіаційний інститут»
23	Національний педагогічний університет імені М.П. Драгоманова
24	Національний технічний університет «Дніпровська політехніка»
25	Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»
26	Національний технічний університет «Харківський політехнічний інститут»

27	Національний транспортний університет
28	Національний університет біоресурсів і природокористування України
29	Національний університет водного господарства та природокористування
30	Національний університет «Запорізька політехніка»
31	Національний університет «Києво-могилянська академія»
32	Національний університет кораблебудування імені адмірала Макарова
33	Національний університет «Львівська політехніка»
34	Національний університет «Одеська юридична академія»
35	Новокаховський гуманітарний інститут вищого навчального закладу «Відкритий міжнародний університет розвитку людини «Україна»
36	Одеська національна академія зв'язку ім. О.С. Попова
37	Одеський національний політехнічний університет
38	Приватний вищий навчальний заклад «Європейський університет»
39	Приватний вищий навчальний заклад «Міжнародний економіко-гуманітарний університет імені академіка Степана Дем'янчука»
40	Рівненський державний гуманітарний університет
41	Східноукраїнський національний університет імені Володимира Даля
42	Тернопільський національний економічний університет
43	Тернопільський національний технічний університет імені Івана Пулюя
44	Університет митної справи та фінансів
45	Харківський національний автомобільно-дорожній університет
46	Харківський національний економічний університет імені Семена Кузнеця
47	Харківський національний університет радіоелектроніки
48	Херсонська філія Національного університету кораблебудування імені адмірала Макарова
49	Херсонський державний університет
50	Херсонський національний технічний університет
51	Хмельницький національний університет
52	Черкаський державний технологічний університет
53	Черкаський навчально-науковий інститут державного вищого навчального закладу «Університет банківської справи»
54	Черкаський національний університет імені Богдана Хмельницького
55	Чернівецький національний університет імені Юрія Федьковича
56	Чернігівський національний технологічний університет
57	Чорноморський національний університет імені Петра Могили

Після закінчення терміну підготовки випускники, згідно з Національним класифікатором професій ДК 003:2010, в подальшому можуть працювати за такими професіями як: 3114 Технік обчислювального (інформаційно-обчислювального) центру; 3121 Технік із системного адміністрування; 3121 Технік-програміст; 3121 Фахівець з інформаційних технологій; 3121 Фахівець з розробки та тестування програмного забезпечення; 3121 Фахівець з розроблення комп'ютерних програм; 3491 Лаборант наукового підрозділу (інші сфери (галузі) наукових досліджень).

Зміст підготовки розроблюється у відповідності до затвердженого стандарту вищої освіти для першого (бакалаврського) рівня (наказ МОН України від 29.10.2018 №1166), який і визначає основний перелік компетентностей випускника. Зокрема, це інтегральна компетентність, загальні компетентності, спеціальні (фахові, предметні) компетентності. Результати здійсненого нами аналізу переліку компетентностей, дають підстави зробити висновок про те, що він, на жаль, не зорієнтований, не зважаючи на всю актуальність і затребуваність ринком праці, на підготовку майбутніх фахівців до тестування програмного продукту. В той же час нормативний зміст підготовки здобувачів вищої освіти, сформульований у термінах результатів навчання передбачає «ПР14 Застосовувати на практиці інструментальні програмні засоби доменного аналізу, проектування, тестування, візуалізації, вимірювань та документування програмного забезпечення» (Додаток А).

Відповідно до чинного законодавства, зміст стандартів вищої освіти реалізується на рівні освітньо-професійних програм професійної підготовки, в яких заклади вищої освіти, в межах своєї автономії, можуть і мають право враховувати нагальні потреби суспільства, рекомендації стейкхолдерів та ін., а відтак - мати свої відмінності.

Враховуючи вищезазначене, нами здійснений вибірковий контент-аналіз змісту нормативно-методичної документації (освітньо-професійна програма та/або навчальний план) підготовки фахівців спеціальності 121 «Інженерія програмного забезпечення» у деяких провідних вітчизняних закладах вищої освіти (вибір визначений наявною можливістю відкритого доступу до публічної інформації на сайті університету) (табл. 1.3).

За результатами здійсненого контент-аналізу встановлено, що зміст освітнього процесу переважної більшості закладів вищої освіти (10 з 13) не зорієнтований на підготовку майбутніх інженерів-програмістів до тестування ПЗ. Це підтверджується відсутністю в нормативно-методичній документації переліку відповідних компетентностей.

Тільки в освітньо-професійних програмах та навчальних планах трьох з тринадцяти закладах вищої освіти в тій чи іншій мірі акцентовано увагу на підготовці майбутніх інженерів-програмістів тестування.

Так, в блоці спеціальних фахових компетентностей освітньо-професійної програми Національного університету водного господарства та природокористування (2017) серед практичних навичок в предметній області визначено: «здатність до розробки та проведення тестування для кожного програмного модуля проекту, далі кваліфіковане тестування всього програмного комплексу відповідно до існуючих або сформульованих вимог; подальша інсталяція програмного продукту та його обслуговування».

*Таблиця 1.3*

**Контент-аналіз змісту нормативно-методичної документації (освітньо-професійна програма та/або навчальний план) підготовки фахівців спеціальності 121 «Інженерія програмного забезпечення»**

<b>№</b>	<b>Назва закладу вищої освіти</b>	<b>Документ, що вивчався (освітньо-професійна програма та/або навчальний план підготовки), рік</b>	<b>Електронна адреса документу (режим доступу 10.04.2020)</b>
1	Вінницький національний технічний університет	освітньо-професійна програма (2018)	<a href="https://vntu.edu.ua/docs/2018/fi/tki/per7.pdf">https://vntu.edu.ua/docs/2018/fi/tki/per7.pdf</a>
2	Державний вищий навчальний заклад «Прикарпатський національний університет імені Василя Стефаника»	навчальний план підготовки (2019)	<a href="https://mif.pnu.edu.ua/">https://mif.pnu.edu.ua/</a>
3	Державний університет інфраструктури та технологій	навчальний план підготовки (2019)	<a href="http://duit.edu.ua/vstupnyku/perelik-spetsialnostei/">http://duit.edu.ua/vstupnyku/perelik-spetsialnostei/</a>
4	Дніпровський національний університет імені Олеся Гончара	освітньо-професійна програма (2019)	<a href="http://www.dnu.dp.ua/view/osvitni_programy">http://www.dnu.dp.ua/view/osvitni_programy</a>
5	Запорізький національний університет	навчальний план підготовки (2019)	<a href="https://www.znu.edu.ua/opp2020/bak/math/op_121_programna_nzhener_ya_bak.pdf">https://www.znu.edu.ua/opp2020/bak/math/op_121_programna_nzhener_ya_bak.pdf</a>
6	Київський національний університет імені Тараса Шевченка	освітньо-професійна програма і навчальний план підготовки (2019)	<a href="https://iss.csc.knu.ua/uk/study/bachelor-degree.html">https://iss.csc.knu.ua/uk/study/bachelor-degree.html</a>
7	Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»	навчальний план підготовки (2019)	<a href="http://pk.kpi.ua/specialities/s-121/">http://pk.kpi.ua/specialities/s-121/</a>
8	Національний транспортний університет	навчальний план підготовки (2017)	<a href="http://vstup.ntu.edu.ua/navchplan/FTIT-PR.pdf">http://vstup.ntu.edu.ua/navchplan/FTIT-PR.pdf</a>
9	Національний університет водного господарства та природокористування	освітньо-професійна програма (2017)	<a href="http://ep3.nuwm.edu.ua/12084/1/%D0%9E%D0%9FBD%29.pdf">http://ep3.nuwm.edu.ua/12084/1/%D0%9E%D0%9FBD%29.pdf</a>
10	Одеський національний політехнічний університет	навчальний план підготовки (2018)	<a href="https://opu.ua/education/programs">https://opu.ua/education/programs</a>
11	Харківський національний університет	навчальний план	<a href="https://nure.ua/abituriyentam/sp">https://nure.ua/abituriyentam/sp</a>

	радіоелектроніки	підготовки (2019 )	etsialnosti-ta-spetsializatsiyi
12	Херсонський національний технічний університет	освітньо-професійна програма (2019 )	<a href="http://kntu.net.ua/index.php/ukr">http://kntu.net.ua/index.php/ukr</a>
13	Чорноморський національний університет імені Петра Могили	освітньо-професійна програма і навчальний план підготовки (2019)	<a href="https://chmnu.edu.ua/navchalno-metodichne-zabespechennya-2019-r-vstupu/">https://chmnu.edu.ua/navchalno-metodichne-zabespechennya-2019-r-vstupu/</a>

Програма Херсонського національного технічного університету (2019) передбачає освітню кваліфікацію «бакалавр з інженерії програмного забезпечення, фахівець з розробки і тестування програмного забезпечення» і відповідно включає такі компетентності як: «ФК23. Здатність виявлення та аналізу вимог, оцінювати вимоги за критеріями узгодженості з потребами, тестованості, здійсненності архітектурного проекту програмного забезпечення, функціонування та супроводу програмних засобів. ФК29. Здатність визначати готовність програмних засобів до тестування, розробляти та налагоджувати процедури тестування для програмних модулів. ФК30. Здатність до проведення кваліфікаційного тестування програмних засобів відповідно до кваліфікаційних вимог, тестування програмних засобів і систем та забезпечення готовності системи до впровадження». В той же час, цикл професійної підготовки включає тільки одну дисципліну «НПП14 Якість і тестування програмного забезпечення», що на наш погляд, є недостатнім.

Більш насиченим є зміст навчання майбутніх фахівців у Харківському національному університеті радіоелектроніки, навчальний план (2019) якого включає в блоці «Дисципліни професійної та практичної підготовки» такі курси, як: «Якість програмного забезпечення та тестування» обсягом 4 кредити у 7 семестрі, форма контролю іспит та вибірковий блок дисциплін 1, який містить: «Автоматизація тестування» обсягом 4 кредити у 7 семестрі форма контролю залік; «Управління тестуванням» обсягом 5 кредитів в 6 семестрі форма контролю залік та «Тестування ігрових застосувань» обсягом 5 кредитів у 8 семестрі форма контролю іспит.

В інших закладах вищої освіти підготовка до тестування здійснюється в межах однієї дисципліни «Якість програмного забезпечення та тестування» обсягом від 3 до 5 кредитів у 5-6 семестрі (форма контролю залік або іспит).

Вивчення наукових робіт дослідників сучасного стану підготовки майбутніх інженерів-програмістів у вітчизняних закладах вищої освіти

(В. Круглик [78], Г. Лебедь [85], О. Наумчук [102], В. Осадчий [109], Я. Сікора [143], Д. Щедролосьєв [172] та ін.), аналіз нормативно-методичного забезпечення освітнього процесу (освітньо-професійних програм, навчальних планів, положень про організацію освітнього процесу, практичну підготовку та ін.), вивчення думки фахівців-практиків дають підстави акцентувати увагу на певних особливостях сучасної вітчизняної системи ІТ-освіти, а саме:

- традиційне домінування фундаментальних аспектів підготовки над практичними як на рівні конкретної дисципліни, так і в цілому змісту навчання. Це, в свою чергу, у поєднанні з низькими темпами оновлення освітньо-професійних програм, зумовлює його ригідність. В той же час, сучасний ринок вимагає гнучкої, системи освіти, здатної реагувати на нагальні потреби ІТ-компаній і підприємств, забезпечувати відповідність рівня підготовки міжнародним стандартам та індивідуальним потребам здобувачів освіти;

- недостатня зорієнтованість змісту навчання на іншомовну підготовку майбутнього фахівця. На сьогодні ІТ сфера є відкритою міжнародною системою, основним засобом спілкування є англійська мова. Відтак, формування компетентності з іноземної мови є вкрай актуальним і нагальним для успішної подальшої професійної самореалізації;

- неналежний рівень організації співпраці закладів вищої освіти і роботодавців, що у поєднанні з проблемами матеріально-технічного та програмного забезпечення освітнього процесу, ускладнюють створення необхідних умов для практичної підготовки студентів. Квазіпрофесійна діяльність майбутнього ІТ-фахівця, як під час практики на підприємствах, так і в лабораторних умовах закладу вищої освіти, має займати одне з ключових місць в системі професійної підготовки, оскільки забезпечує можливість працювати з конкретними задачами, що відповідають вимогам часу. Інакше студенти нехтують теоретичним навчанням на користь прикладної діяльності: на сьогодні більшість студентів 2-3 курсів працює в ІТ-компаніях, віддаючи навчання неналежну увагу. Тому нагальною вимогою часу є залучення до практичної підготовки фахівців-практиків та запровадження принципів неформальної та дуальної форм навчання;

- низькі темпи модернізації освітнього процесу в напрямку приведення його у відповідність до міжнародної практики: створення в умовах кооперації з провідними ІТ-компаніями навчально-виробничих лабораторій; запровадження проєктних і колективних форм та методів професійної підготовки тощо.

Вищезазначене підтверджують і результати опитувань та моніторингу якості професійної підготовки інженерів-програмістів, проведених співтовариством програмістів DOU (<https://dou.ua/>). В опитуванні приймали участь 3081 респондент з досвідом роботи в ІТ-сфері та навчання у вітчизняному закладі вищої освіти. Так, за 10 бальною шкалою, актуальність отриманих знань фахівці оцінили в 4,1 бали; корисність для використання в ІТ-сфері - 5,4 бали; практичний досвід використання матеріалу - 5,2 бали; технологічні можливості для навчання - 5,7; корисність для професійної діяльності - 5,2 бали. Неактуальність знань відображена і в оцінках роботи викладачів: якщо зрозумілість пояснень і доступність матеріалу оцінені в середньому в 6,1 бали, то практичний досвід застосування та вміння пояснити як застосовуються знання на практиці - в 5,2.

Отже, усе вищезазначене актуалізує гостру потребу у пошуку шляхів підвищення якості професійної підготовки майбутніх-інженерів-програмістів у цілому та до тестування ПЗ зокрема у вітчизняних закладах вищої освіти, що забезпечується, у тому числі, їх неформальною освітою.

### **Висновки до першого розділу**

Узагальнення науково-теоретичних основ дослідження підготовки майбутніх інженерів-програмістів до тестування ПЗ дає змогу зробити такі висновки за розділом.

1. Здійснений аналіз сучасних вимог до професійної діяльності та професійного розвитку інженерів-програмістів, їх кваліфікаційних категорій та функціональних обов'язків, а також проведений аналізу змісту професійних завдань за різними методологіями і типами тестування на різних його етапах, та, крім того, - результати здійсненого емпіричного дослідження серед фахівців з тестування ПЗ різних ІТ-компаній, дали підстави для визначення таких

специфічних особливостей професійної діяльності інженера-програміста з тестування ПЗ як: 1) первинна відповідальність за якість продукту на всіх етапах його розробки, що у свою чергу вимагає здійснення аналізу та проведення максимальної кількості перевірок ПЗ; 2) метасистемний характер – вона вимагає від тестувальника професійних компетенцій як програміста (мати досвід програмування, знати програмний код), так і бізнес-аналітика, менеджера проєкту (знати його організаційну структуру, мати організаційні та комунікативні вміння); 3) необхідність здійснення аналізу продукту з декількох позицій: з точки зору замовника програмного продукту, розробника ПЗ, а також, - користувача продуктом; 4) творчий характер - вимагає постійного пошуку шляхів перевірки, оптимізації тестового покриття, прийняття рішень в умовах невизначеності.

2. Грунтуючись на положеннях особистісно-діяльнісного підходу до розуміння сутності готовності особистості до професійної діяльності, *професійна готовність майбутніх інженерів-програмістів з тестування ПЗ* визначена системним утворенням, інтегративним особистісно-професійним явищем, що є результатом їх цілеспрямованої, спеціально організованої професійної підготовки в умовах неперервної освіти, є передумовою професійної компетентності та визначає успішність їх подальшої професійної діяльності та професійного розвитку.

Спираючись на результати здійсненого аналізу завдань та етапів діяльності тестувальників ПЗ в ІТ-компаніях, а також на підставі Рекомендацій Ради Європи щодо ключових компетентностей для освіти впродовж життя, що забезпечують відповідний рамковий інструментарій, структура професійної готовності майбутніх фахівців з тестування ПЗ складається з таких компонентів як:

1) *ціннісно-мотиваційний*, що визначається їх мотивацією професійної діяльності та подальшого їх професійного розвитку і навчання, ціннісним ставленням до виконання професійних обов'язків – тестування ПЗ, спрямованістю на досягнення командного результату;

2) *когнітивний*, який віддзеркалює систему їх загальних та спеціальних професійних знань, у тому числі: методології розробки ПЗ; фундаментальних

основ тестування, технік тест-дизайну, принципів роботи допоміжного ПЗ для тестування; основ естімації та планування; основ алгоритмів, промислового програмування, процесів та методології розробки; архітектури комерційних додатків; мережевих технологій і адміністрування; операційних систем, засобів віртуалізації;

3) *операційно-діяльнісний*, що складається з двох груп професійних умінь:

а) *технічні вміння*, що відображають завдання/етапи професійної діяльності тестувальника ПЗ, а саме: аналізувати вимоги на предмет їх адекватності, здійсненності та тестованості; аналізувати, створювати і підтримувати тестову документацію, взаємодіяти з додатком на призначеному для користувача і розробницькому рівні, проходити тестові сценарії і коректно описувати дефекти тощо;

б) *нетехнічні вміння* (soft skills), що відображають вимоги до їх: мисленнєвої діяльності – *мисленнєві вміння* (аналітичність, оперативність та гнучкість мислення, нестандартно мислити, самостійність мислення); спілкування - *комунікативні вміння* (вміння працювати у команді, підтримувати позитивну атмосферу, домовлятися, комунікувати з членами команди та з замовником, обґрунтовувати і відстоювати свою позицію, вміння індивідуального підходу, неконфліктність); регуляції власної поведінки та діяльності – *регулятивні вміння* (самоконтролю власної діяльності, самоконтролю власних емоцій, активізації працездатності, вміння керувати часом, бути зосередженим та уважним, вміння організувати себе та інших); метапізнавальної діяльності – *метакогнітивні вміння* (вміння проблематизації, проєктувальні вміння, планування, прийняття рішення в умовах невизначеності), включаючи рефлексивні вміння (аналізувати, оцінювати власну діяльність, знаходити помилки у неї та коригувати власну мисленнєву діяльність).

3. Здійснення аналізу зарубіжного та вітчизняного досвіду професійної підготовки (включаючи контент-аналіз її нормативно-правового забезпечення) майбутніх-інженерів програмістів до професійної діяльності в цілому та до тестування ПЗ зокрема дає підстави стверджувати про таке:

1) її характерними ознаками в зарубіжних закладах вищої освіти є: забезпечення фундаментальної та технічної підготовки у гармонійній єдності з практичним навчанням – отримання студентами ґрунтовного практичного досвіду в спеціалізованих сферах ІТ, у тому числі - у сфері тестування ПЗ; метапредметне спрямування – розширення змісту завдяки суміжним галузям та сферам діяльності (менеджмент, безпека, торгівля, медіа тощо); зорієнтованість навчальних форм і методів на виробничий процес розробки ПЗ, що вимагає проведення тематичних досліджень, локальних та загальних квазіпрофесійних проєктних заходів та ін.; забезпечення цілеспрямованого розвитку нетехнічних професійних знань та умінь студентів у процесі групових, проєктних форм, командної роботи, навчальних та виробничих практик тощо; ефективне використання ресурсу неформальної освіти: обов'язковості проходження тривалої виробничої практики на ІТ-підприємствах;

2) у вітчизняних закладах вищої освіти спостерігається: домінування теоретичної, фундаментальної підготовки над практичною, що зумовлена у тому числі застарілістю матеріально-технічного та програмного забезпечення освітнього процесу; ригідність оновлення освітньо-професійних програм, що загострює протиріччя між потребою ІТ-індустрії у кваліфікованих фахівцях та неготовністю випускників закладів вищої освіти до роботи в умовах сучасних ІТ-компаній; недостатня зорієнтованість змісту навчання на іншомовну (передусім, англomовну) підготовку студентів; недостатній рівень співпраці закладів вищої освіти та роботодавців, що у поєднанні з проблемами матеріально-технічного та програмного забезпечення освітнього процесу, ускладнюють створення необхідних умов для практичної підготовки студентів; недостатня спрямованість на підготовку майбутніх фахівців до діяльності у спеціалізованих ІТ-сферах, у тому числі – тестування ПЗ.

Вищезазначене актуалізує гостру потребу у пошуку шляхів підвищення якості професійної підготовки майбутніх-інженерів-програмістів у цілому та до тестування ПЗ у тому числі завдяки ресурсам неформальної освіти.

Основні положення розділу висвітлені у наукових працях [29; 30; 34; 37].

## РОЗДІЛ 2.

# НАУКОВО-МЕТОДИЧНІ ЗАСАДИ ПРОФЕСІЙНОЇ ПІДГОТОВКИ МАЙБУТНІХ ІНЖЕНЕРІВ-ПРОГРАМІСТІВ ДО ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ В УМОВАХ НЕФОРМАЛЬНОЇ ОСВІТИ

### 2.1. Особливості сучасної неформальної освіти фахівців ІТ-галузі

Результати вивчення зарубіжного та вітчизняного досвіду професійної підготовки засвідчили, що незважаючи на відмінності систем фахової освіти майбутніх інженерів-програмістів, є загальне усвідомлення необхідності максимальної наближеності змісту навчання до сучасної ІТ-діяльності. Одним із шляхів, що забезпечує зазначене твердження, поряд зі збільшенням практичної складової освітнього процесу (навчальних і виробничих практик на базі ІТ-компаній), створення сучасних ІТ-лабораторій, залучення провідних фахівців ІТ-сфері тощо, є урізноманітнення форм професійної підготовки, зокрема використання дидактичного потенціалу неформальної освіти, яка на сьогодні стала невід'ємною складовою системи професійного становлення майбутніх фахівців.

Здійснений аналіз наукових джерел дає підстави констатувати, що розвиток неформальної освіти є пріоритетним напрямом функціонування більшості країн світу. На концептуальному рівні цей процес безпосередньо пов'язується зі становленням освітньої системи дорослих Північної Америки, зокрема США та Канади, яких вважають законодавцями у цій сфері. Дослідники американського досвіду виділяють навіть етапи генези неформальної освіти: структурно-організаційний етап (1924-1945 рр.) – перетворення її на самостійну галузь; реформістський етап (1946-1966 рр.) – інтенсивне формування законодавчо-правової бази; інституціональний етап (1967-1990 рр.) – концептуальне оформлення неформальної освіти як

складової системи неперервної освіти; конструктивний етап (1991-2000 рр.) – розширення соціальних цілей неформальної освіти дорослих; упроваджувальний етап (2001 р. - початок XXI століття) – взаємодоповнення формальної та неформальної освіти дорослих, пошук механізмів валідізації (або валідації [115]) результатів неформальної освіти дорослих, вертикальна і горизонтальна диверсифікація [43, с. 9].

Як показує досвід, функціонування неформальної освіти забезпечувалось і забезпечується значною кількістю провайдерів:

- місцеві (локальні) організації (освітні заклади, агенції, центри зайнятості, громадські центри та інші), які, зазвичай, отримують грантову підтримку;
- інститути неперервної освіти, вищі народні школи, університети третього віку;
- некомерційні організації, які неформальну освіту розглядають як передумовою розбудови демократичного громадянського суспільства (входять громадські і релігійні організації, асоціації та ін.);
- приватні організації, які надають освітні послуги (тренінгові, мовні, консалтингові центри), підприємства і компанії, що підтримують корпоративну освіту і навчання на робочому місці та ін.

Координуючі функції зазвичай виконують державні органи. На прикладі США - це: Міністерство освіти, Департамент професійної освіти і освіти дорослих, Міністерство праці, Департамент з питань зайнятості і професійної підготовки, Служба зайнятості США та ін. [106, с. 81].

Широкого поширення неформальна освіта отримала і в країнах Європи. Зокрема в Скандинавії вона зорієнтована на потреби конкретних людей і розвивається залежно від ініціатив громад. Механізмом реалізації є децентралізована діяльність навчальних гуртків. В Європейському Союзі вона характеризується відродженням традицій громадської освіти; перенесенням ідеї особистісного самовдосконалення у нові контексти,

зокрема на робочі місця; активному функціонуванні освітніх центрів, відділення неперервної освіти та ін. У більшості пострадянських країнах неформальна освіта ще залишається додатковою несертифікованою формою, що не має потенціалу. Науковці пояснюють це відсутністю довіри до діяльності громадських об'єднань, а також нестачею досвіду делегування повноважень з боку влади [114].

Відтак, на сьогодні неформальна освіта розуміється світовою спільнотою як:

- навчання, що є добровільним і цілеспрямованим, має місце в різних умовах і ситуаціях, для яких підготовка необов'язково є єдиним або основним видом діяльності (Рада Європи);

- інституціолізоване, цілеспрямоване і сплановане особистістю або організацією навчання, що забезпечує надання освітніх послуг (МСКО 2011 (ЮНЕСКО));

- навчання, що відбувається в навчальних закладах та громадських організаціях, а також під час індивідуальних занять з репетитором або тренером (Меморандум безперервного навчання Європейського союзу); в центрах освіти дорослих, у лекторіях, по телебаченню, на різних курсах інтенсивного навчання (Національний інститут стратегічних досліджень України) [3, с. 70].

Слід акцентувати увагу і на тому, що провідною тенденцією сучасного етапу розвитку неформальної освіти є всесвітній рух за визнання, сертифікацію й валідізацію її результатів, проголошений і підтриманий міжнародними освітніми організаціями багатьох країнах світу, про що свідчить низка прийнятих останнім часом у цій сфері документів (Рекомендації Ради Європи з валідізації неформальної та інформальної освіти (2012), Рекомендації ЮНЕСКО з визнання, сертифікації та акредитації результатів неформального та інформального навчання (2012), «Європейські Рекомендації з визнання та валідізації результатів неформальної та інформальної освіти» (2015), План дій

ЮНЕСКО з розвитку освіти після 2015 року «Забезпечення справедливої освіти і навчання протягом усього життя для всіх до 2030 року» [70] та ін.) [112].

Сертифікація є складовою формального навчання, що визнає інституційне набуття освіти (дипломи та свідоцтва), в той час як валідація спрямована на офіційне визнання результатів підготовки (отриманих знань, вмінь, навичок, компетенцій) незалежно від часу і місця їх набуття. Її зміст полягає в тому, «щоб зробити навчання видимим» [157]. Освітні системи країн Євросоюзу на сьогодні вже мають спеціальні норми, що забезпечують визнання результатів якості підготовки в неформальній освіті.

Так, наприклад, у Фінляндії та Австралії це передбачено законодавством, і відбувається за результатами так званої демонстрації вмінь, внаслідок чого присвоюється кваліфікація (Фінляндія) та оцінювання результатів навчання, його корелювання з офіційною кваліфікацією, на яку вона претендує (Австралія). У Швеції визнання відбувається шляхом проведення системного моніторингу на базі науково обґрунтованих методик оцінювання навчальної діяльності, який здійснює Всешведська рада неформальної освіти дорослих, і який спрямований на оцінку діяльності провайдера неформальної освіти, чисельність учасників, кількість навчальних годин (кредитів) та їх тематичний зміст, джерела фінансування та ін. Якість неформальної освіти оцінюють за певними індикаторами, найпоширенішими серед яких є: освітні індикатори Ради з освіти Європейського Союзу, Міжнародного Консультативного форуму з освіти для усіх, освітні індикатори ЮНЕСКО, ОЕСР та ін. [46].

У той же час, результати аналізу свідчать, що 26 із 35 країн-членів Євросоюзу застосовують однакові стандарти для визнання результатів неформальної освіти, що найменше в одному із секторів системи освіти, зокрема мова йде про професійну освіту, де використовуються ті ж самі стандарти і для валідації результатів формального і неформального

навчання. У 12 країнах стандарти підтвердження результатів однакові у всіх освітніх секторах. Їх перевірка і підтвердження здійснюється за допомогою національних рамок кваліфікацій, а також зарахуванням кредитів, що є традиційним для закладів вищої освіти. Це зумовлено розповсюдженням усталеної системи використання ECTS [7].

Для України, як і для більшості держав, питання становлення і розвитку системи неформальної освіти і визнання її результатів є вкрай актуальним. Це зумовлено, насамперед, відставанням змісту професійної підготовки від вимог професійного середовища, які постійно оновлюються, розвитком нових інформаційно-комунікаційних технологій, що відкривають нові можливості для навчання, зростанням безробіття, неузгодженістю нормативно-методичної бази. Це призводить до того, що неформальна освіта розвивається як інноваційна альтернатива система, яка базується на особистісних потребах суб'єкта навчання, його власній мотивації. Вона є задекларованою, але ще не легітимізованою в Україні платформою для взаємодії освітніх закладів, громадських організацій та інших суб'єктів соціально-виробничої сфери.

Результати дослідження динаміки популярності неформальної освіти в Україні, статистичні дані з «Аналітичного звіту дослідження у сфері неформальної освіти дорослих у пілотних областях України: Львівська та Полтавська» (2010) та річного звіту проекту «Інша Освіта» (2017), що реалізується в Києві, Краматорську та Івано-Франківську в рамках проекту «Діалог заради змін» за партнерства зі спілкою MitOst e.V. та фінансової підтримки Міністерством закордонних справ Німеччини засвідчили, що за період останніх 12 років сегмент неформальної освіти збільшився з 2,4 % до 44 % респондентів [45].

Слід акцентувати увагу та тому, що це стало можливим і в наслідок активної модернізації та реформування вітчизняного освітнього простору, в якому на сьогодні розпочались значні позитивні зрушення в напрямку використання можливостей системи неформальної освіти на всіх її рівнях.

Так, зокрема, відповідно до ст. 8 Закону України «Про освіту» (2017) держава визнала право громадян на навчання впродовж життя шляхом формальної, неформальної та інформальної освіти. Неформальна освіта визначається як «освіта, що здобувається, як правило, за освітніми програмами та не передбачає присудження визнаних державою освітніх кваліфікацій за рівнями освіти, але може завершуватися присвоєнням професійних та/або присудженням часткових освітніх кваліфікацій. Результати навчання, здобуті шляхом неформальної освіти, визнаються в системі формальної освіти в порядку, визначеному законодавством».

Незважаючи на те, що неформальна освіта не знайшла своє відображення в Законі України «Про вищу освіту», в ст. 49. «Форми здобуття освіти в закладах вищої освіти», заклад вищої освіти отримав право використовувати інші форми здобуття вищої освіти і поєднувати форми здобуття вищої освіти відповідно до положення про організацію освітнього процесу в закладі вищої освіти, а також встановлювати вимоги до поєднання форм здобуття вищої освіти.

Вже починаючи з 2019 року більшість закладів вищої освіти України в межах своєї автономії унормувала механізми і процедури визнання результатів неформального навчання, отриманих здобувачами вищої освіти в процесі професійної підготовки відповідними локальними нормативними актами (положеннями (Додаток Б), порядками) (табл. 2.1).

Здійснений нами аналіз зазначених локальних нормативних актів провідних університетів дає підстави констатувати таке.

По-перше, - на сьогодні всі заклади вищої освіти почали визнавати в процесі професійної підготовки результати навчання, які здобувачі вищої освіти отримали в умовах неформальної освіти.

По-друге, - механізми і процедури їх визнання і зарахування в цілому ідентичні і передбачають експертне оцінювання (оцінювання експертною комісією), мета якого полягає у валідації змісту неформальної підготовки зі змістом освітньо-професійних програм відповідно до спеціальності.

Зокрема, акцент робиться на кількості кредитів підготовки, переліку здобутих компетенцій, а також їх відповідності освітнім стандартам.

Таблиця 2.1

**Заклади вищої освіти, що вноормували визнання результатів неформального навчання, отриманих здобувачами вищої освіти**

№	Заклад вищої освіти	Документ, що регламентує процедури визнання результатів неформальної освіти	Рік
1	Вінницький національний університет	Положення про порядок визнання результатів навчання, отриманих у неформальній освіті	2019
2	ДВНЗ «Прикарпатський національний університет імені Василя Стефаника»	Положення про порядок зарахування результатів неформальної освіти у ДВНЗ «Прикарпатський національний університет імені Василя Стефаника»	2019
3	Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»	Положення про визнання в КПІ ім. Ігоря Сікорського результатів навчання, набутих у неформальній/інформальній освіті	2020
4	Центральноукраїнський національний технічний університет	Положення про порядок визнання результатів навчання, отриманих у неформальній та/або інформальній освіті	2019
5	Дніпропетровський державний університет внутрішніх справ	Положення про порядок визнання результатів навчання, отриманих у неформальній освіті	2019
6	Запорізький національний університет	Положення Запорізького національного університету про порядок визнання результатів навчання, отриманих в неформальній	2020
7	Криворізький національний університет	Положення про порядок визнання в Криворізькому національному університеті результатів навчання, отриманих в умовах неформальної освіти	2020
8	Західноукраїнський національний університет	Положення про неформальну та інформальну освіту здобувачів вищої освіти в Західноукраїнському національному університеті	2020
9	Національний університет кораблебудування імені адмірала Макарова	Положення про визнання результатів навчання, здобутих у неформальній та/або інформальній освіті в Національному університеті кораблебудування імені адмірала Макарова	2020
10	Чернівецький національний університет імені Юрія Федьковича	Положення про взаємодію формальної та неформальної освіти, визнання результатів навчання (здобутих шляхом неформальної та/або інформальної освіти, у системі формальної освіти) у	2019

		Чернівецькому національному університеті імені Юрія Федьковича	
11	Сумський державний університет	Положення про перезарахування результатів навчання здобувачів вищої освіти Сумського державного університету, отриманих у неформальній освіті	2020
12	Харківський національний університет імені В. Н. Каразіна	Порядок визнання результатів навчання, отриманих у неформальній освіті, в Харківському національному університеті імені В. Н. Каразіна	2020

По-третє, - формалізація результатів підготовки в системі неформальної освіти відбувається тільки на рівні змісту формальної освіти, що передбачає або перезарахування навчальних дисциплін, або доповнення в межах варіативної частини підготовки (дисциплін за вибором). Тобто результати враховуються і визначаються в додатку диплому про вищу освіту в межах переліку компетентностей, які отримав здобувач вищої освіти.

По-четверте, - значна увага приділяється освітній базі, яка забезпечувала навчання: в яких умовах воно відбувалось, відповідно до яких програм, що передбачав зміст підготовки, наскільки він є актуальним для тієї чи іншої спеціальності.

І в цьому контексті використання потенціалу неформальної освіти в процесі професійної підготовки саме майбутніх інженерів-програмістів є не тільки вкрай актуальним, а й продуктивним, оскільки на сьогодні у виробничій ІТ-сфері, системі вищої та професійної освіти складаються усі необхідні для цього умови.

Зокрема, науковці (І. Крашеніннік, В. Осадчий [73] та ін.) звертають увагу на значну кількість навчальних центрів, які пропонують слухачам достатньо великий вибір ІТ-курсів різних рівнів (Київська компанія GoIT (<http://goit.ua/>): «Quality Assurance», «HTML/CSS + JavaScript» (підготовка фахівців з frontend-розробки), «GoFrontend Advanced» (підвищення кваліфікації з frontend-розробки), «GoiOS» та ін); Львівська ІТ Школа –

LITS (<http://lits.ua/>) пропонує понад 30 курсів з підготовки фахівців для ІТ-галузі у Львові, Харкові, Одесі й Івано-Франківську; Українська ІТ-фабрика UNIT (Ukrainian National IT Factory) започатковує трирічну програму підготовки програмістів на безкоштовних засадах у режимі офлайн за спеціальностями: менеджер ІТ-проектів, архітектор програмного забезпечення, архітектор мереж, адміністратор мережі, веб-розробник, розробник мобільних програм, фахівець з ІТ-безпеки, розробник відеоігор та ін. [74].

З причини підвищеного попиту щодо можливості отримати високооплачуване робоче місце у великій компанії, значне поширення набули перш за все комерційні проекти, спрямовані на надання послуг вузькоспеціалізованого неформального навчання по найбільш популярним ІТ-спеціальностям. Список найбільш популярних закладів неформальної ІТ-освіти на комерційній основі очолюють:

- «Комп'ютерна Академія ШАГ», яка представлена більш ніж в десяти містах України і випускає щорічно більше 4500 випускників за такими напрямками як: «Розробка ПО», «Мережі і безпеку», «Комп'ютерна графіка та дизайн», «Розробка та просування web- проектів »[69];

- «Main Academy», яка щорічно навчає більш ніж 3400 студентів за спеціальностями QA, Java, Front-End, Android, PHP, Python [203];

- «CyberBionic Systematics», яка співпрацює з великими українськими ІТ-компаніями і готує понад 3000 студентів на рік [82].

З іншого боку, з кожним роком все більше великих ІТ-компаній створює на своїй виробничій базі спеціалізовані центри неформальної освіти, основною метою яких є підготовка та підвищення кваліфікації власних кадрів, а також активний рекрутинг, перш за все серед здобувачів вищої освіти, і залучення нових співробітників за рахунок курсів на безкоштовній основі. До найбільш розвинених і популярних центрів неформальної освіти на базі діючих ІТ підприємств відносяться:

- EPAM RD Lab - центр підготовки компанії EPAM Systems, який співпрацює більш ніж з 20 університетами України та випускає понад 2000 студентів щорічно, більшість з яких залишаються в компанії на проектних посадах;

- SoftServe University та SoftServe Academy - відділи підготовки фахівців на базі компанії SoftServe Ukraine, призначені для випускників університетів і вже практикуючих фахівців відповідно. Вони пропонують більше 38 напрямів навчання на 79 курсах. За рік IT Academy випускає понад 1000 студентів [22].

Згідно з опитуваннями, проведеними профільними виданнями [138], в 2016 році в Україні представлено понад 70 великих IT-шкіл і навчальних центрів, котрі навчили понад 43 тисяч студентів в 24 містах України. Очевидним є так само і зростання індустрії неформальної підготовки інженерів-програмістів: згідно з відповідною статистикою 2015 року спостерігається більш, ніж 30% зростання числа її випускників [137].

Вищезазначене, на тлі активізації процесів, пов'язаних з впровадженням принципів дуальної освіти, призводить до того, що на сьогодні значна кількість здобувачів вищої освіти, що проходять професійну підготовку за IT-спеціальностями в закладах вищої освіти, вже починаючи з другого курсу є, з одного боку, суб'єктами системи неформальної освіти на базі навчальних центрів провідних IT компаній, з іншого боку, - їх співробітниками.

Це зумовлює і модернізацію системи формального навчання, зокрема коригування змісту підготовки в напрямку приведення його до сучасних потреб ринку, розширення спектру професійних спеціалізацій, збільшення обсягу і варіативності освітнього процесу, індивідуалізації підготовки, активної взаємодії зі стейкхолдерами, врегулювання питань практичної підготовки на засадах дуальної підготовки та ін.

Зазначимо, що неформальна освіта:

- має такі характерні властивості, що відрізняють її від формальної освіти, а саме: рівність і демократичність; діалогічність; доброзичливість; практичність змісту; активність життєвої позиції; особиста відповідальність; свобода вибору; відсутність нав'язування певних думок та способів діяльності; реактивний характер (Н. Павлик та ін.);

- забезпечує такі етапи внутрішніх перетворень особистості як: прийняття відповідальності за власні цілі, дії та їх майбутні результати; емоційне переживання щодо визначених цілей, дій та їх результатів; оперативну реалізацію опанованих умінь; усвідомлену оцінку отриманих результатів власної активності (С. Зінченко [51]).

Отже, *під професійною підготовкою майбутніх інженерів-програмістів в системі неформальної освіти* ми розуміємо невід'ємну складову цілісної системи їх освіти, що відбувається в умовах опанування певних освітніх програм різних провайдерів освітніх послуг (у тому числі – самих ІТ-компаній), не передбачає присудження визнаних державою освітніх кваліфікацій за рівнями освіти, проте забезпечує здобуття кваліфікаційного статусу в ІТ-компанії (на рівнях молодшого спеціаліста («Junior Specialist»), спеціаліста («Middle specialist»), старшого фахівця («Senior specialists»), керівника команди й експерта («Team Leader»), спрямована на створення умов для реалізації їх особистісного потенціалу, та визначає успішність їх професійної діяльності і професійного розвитку.

Таким чином, здійснений аналіз системи неформальної освіти в цілому та інженерів-програмістів зокрема, дає підстави зробити такі висновки.

По-перше, - неформальна освіта є невід'ємною і сформованою складовою міжнародної системи неперервної освіти, яка на сьогодні має відпрацьовані і формалізовані процедури сертифікації та валідації у більшості країн світу.

По-друге, - сучасний стан нормативно-методичної бази вітчизняної системи професійної підготовки враховує міжнародні здобутки і

зорієнтований на продуктивне використання потенціалу неформальної підготовки.

По-третє, - на сьогодні в Україні достатньо швидко формується організаційно-методична структура суб'єктів освітніх послуг, що забезпечує необхідні умови для додаткового неформального навчання.

По-четверте, - одним з перспективних і найбільш продуктивних напрямів розвитку системи неформальної освіти є створення і функціонування центрів підготовки на базі провідних ІТ-компаній, які активно співпрацюють з закладами вищої освіти.

По-п'яте, - потребує нагального вирішення питання науково-методичного узгодження і забезпечення професійної підготовки майбутніх інженерів-програмістів в цілому та до тестування ПЗ зокрема в умовах реалізації формальної і неформальної освіти.

## **2.2. Структурно-функціональна модель професійної підготовки майбутніх інженерів-програмістів до тестування програмного забезпечення в умовах неформальної освіти**

Здійснений аналіз особливостей професійної діяльності інженерів-програмістів з тестування ПЗ на сучасному етапі розвитку ІТ-індустрії, змісту професійної готовності майбутніх фахівців до тестування ПЗ, що є результатом їх професійної підготовки, а також зарубіжного та вітчизняного досвіду їх фахової формальної та неформальної освіти, створює підґрунтя для побудови моделі професійної підготовки майбутніх інженерів-програмістів до тестування ПЗ в умовах неформальної освіти як необхідного етапу її наукового обґрунтування. Саме модель як уявна або матеріально-реалізована система, котра відображає або відтворює об'єкт дослідження і здатна змінювати його так, що її вивчення дає нову інформацію стосовно цього об'єкта [42, с. 516]; як штучна система елементів, що фіксує найсуттєвіші властивості, сторони, зв'язки об'єкта,

який вивчається [24, с. 120], є важливою проміжною ланкою між висунутими теоретичними положеннями та їх реальною перевіркою в реальній педагогічній дійсності, освітньому процесі.

Перед тим, як презентувати, описати модель професійної підготовки майбутніх інженерів-програмістів до тестування ПЗ в умовах неформальної освіти, охарактеризуємо методологічні засади, на підставі яких вона розробляється.

Насамперед зазначимо, що методологія, погоджуючись з С. Гончаренко [24], визначає: основний метод пізнання як сукупність пізнавальних способів, методів, прийомів, що застосовуються у науковому дослідженні; передумови і принципи організації діяльності щодо вирішення наукової проблеми.

Визначаючи методологічні основи дослідження та власне моделювання професійної підготовки майбутніх інженерів-програмістів до тестування ПЗ в умовах неформальної освіти ми виходимо з розуміння рівневої структури методологічного знання [24], а саме: перший рівень – філософські знання (філософська методологія); другий рівень – загальнонаукова методологія; третій рівень – конкретно-наукова методологія; четвертий рівень – методика і техніка дослідження.

Отже, на рівні *філософської методології* базовими положеннями є твердження:

- *діалектичного матеріалізму* про: первинність об'єктивного, матеріального світу перед суб'єктивним, внутрішнім; полідетермінізм процесів розвитку, взаємозумовленість та взаємозв'язок психічного та соціального; багатодермінованість професійного розвитку та освітньої діяльності людини, їх зумовленість зовнішніми (соціальними) та внутрішніми (нейрофізіологічними, психологічними, особистісними) чинниками;

- *прагматизму* (Дж. Дьюї та ін.) про: безперечну практичну спрямованість освіти, її зорієнтованість на задоволення життєвих потреб

людини; пріоритетність практичних методів навчання; втілення освітніх результатів у практичну діяльність людини; навчання як соціально-інтерактивний, максимально індивідуалізований процес рівноправної взаємодії суб'єктів освітнього простору;

- *конструктивізму* про: пізнання як активну побудову суб'єктом моделі світу всупереч його механічного відображення, пріоритету конструювання, моделювання, творчості як головних видів діяльності людини.

На рівні *загальнонаукової методології* базовими є такі положення:

- *системного підходу* (В. Краєвський та ін.) про: цілісність, системність явища професійної готовності майбутніх інженерів-програмістів до тестування ПЗ, єдність його структурних компонентів; інтеграцію різних сфер знання щодо дослідження сутності явища професійної готовності; єдність результатів формальної та неформальної освіти інженерів-програмістів; необхідність врахування зовнішніх та внутрішніх умов професійної підготовки майбутніх фахівців у системі неформальної освіти, визначення форм, методів та засобів навчання відповідно до визначених організаційно-педагогічних умов;

- *синергетичного підходу* (І. Пригожин та ін.) про: нелінійність, біфуркаційність процесу професійної підготовки та професійного розвитку майбутніх інженерів-програмістів, що спричиняються у першу чергу внутрішніми, психологічними чинниками – мотивацією їх професійного навчання та професійного розвитку, здатністю до самоорганізації, саморегуляції, рефлексії, ціннісним ставленням до тестування ПЗ;

- *інформаційного (цифрового) підходу* (В. Мартін та ін.) про: першочерговість виявлення та аналізу найхарактерніших інформаційних аспектів щодо підготовки розвитку майбутніх інженерів-програмістів до тестування ПЗ; первинність інформації як головного чинника природних та суспільних процесів; врахування найновітніших знань в інформаційному просторі; важливість опанування майбутніми тестувальниками ПЗ

метакогнітивними вміннями у процесі їх неформальної освіти, цілеспрямованого розвитку критичності мислення; врахування вимог до працівників інформаційних служб у контексті визначення професійної готовності майбутніх тестувальників ПЗ.

На рівні *конкретно-наукової, власне педагогічної, методології* базовими є положення:

- *діяльнісного підходу* (П. Гальперін, О. Леонт'єв, І. Лернер та ін.) про: єдність психіки та діяльності, навчання та розвитку, структури предметної та мисленнєвої діяльності; відповідності структури професійної готовності майбутніх тестувальників ПЗ структурі їх професійної діяльності; вибудовування освітнього процесу на підставі етапів та механізмів формування розумових дій; пріоритету групових форм і методів як спільної навчально-професійної діяльності майбутніх фахівців під керівництвом викладача-досвідченого фахівця;

- *контекстного підходу* (А. Вербицький, В. Шадриков та ін.) про: 1) забезпечення руху розумової та практичної діяльності майбутніх фахівців у напрямку від навчальної діяльності за академічним типом до квазіпрофесійної та навчально-професійної діяльності в умовах реальних виробничих умовах; 2) послідовне моделювання у навчальній діяльності студентів цілісного змісту, форм та умов реальної професійної діяльності фахівців; 3) проблемність змісту освіти та його розгортання в освітньому процесі; 4) провідну роль спільної діяльності, міжособистісної взаємодії та діалогічного спілкування суб'єктів освітнього процесу; 5) відкритість (не монопідхідності), застосування різних педагогічних технологій, що пропонуються різними теоріями та підходами; 6) педагогічно обґрунтоване та доцільне поєднання інноваційних та традиційних педагогічних технологій; 7) природню метапредметність навчання;

- *техніко-технологічного підходу* (В. Круглік [88], В. Осадчий [109], Д. Щедролосьєв [171] та ін.) про: автоматизацію професійної підготовки майбутніх інженерів-програмістів, використання інноваційних ІКТ

технологій, застосування сучасного професійного інструментарію; важливість поєднання різних освітніх форм і методів; значний акцент на самостійній роботі майбутніх фахівців; необхідність організації персонального освітнього середовища;

- *компетентнісного підходу* (О.І. Гура [28], А. Деркач [39] та ін.) про: цілеспрямований, спеціально організований розвиток компетентностей особистості в цілому та професійної компетентності зокрема; пріоритетність формування та розвитку професійної компетентності фахівців з тестування ПЗ в умовах системи освіти в цілому та неформальної освіти зокрема; єдність, взаємозумовленість професійної компетентності як актуального, реального та професійної готовності як потенційного; про важливість формування ключових, надпредметних компетентностей; зміщення акцентів з формування певного набору професійних знань на формування вміння використовувати знання для вирішення практичних професійних проблем, на розвитку таких якостей як відповідальність, самоорганізованість, рефлексивність, тощо;

- *особистісного підходу* (І. Бех, М. Ноулз та ін.) про: добровільність навчальної діяльності; рівність навчальних взаємодій; створення розвивального освітнього середовища, що зорієнтоване на задоволення потреб та інтересів майбутніх фахівців, включаючи надання можливостей для формування індивідуальної освітньої траєкторії в умовах неформальної освіти; побудову освітнього процесу на засадах взаємної поваги і партнерства; спрямування навчання на реалізацію майбутніми фахівцями їх особистісних функцій: мотиваційної, рефлексивної, смислоутворювальної, критичного ставлення, самореалізувальної, творчо-перетворювальної;

- *метакогнітивного підходу* (А. Карпов [60; 62] та ін.) про: важливість врахування та цілеспрямованого розвитку метакогніцій майбутніх фахівців - складних, багаторівневих та системно-організованих психічних утворень, що мають власне процесуальну та автономно

розгорнуту форму, спрямовані на організацію, регуляцію та координацію їх первинних когнітивних процесів та забезпечують загальну регуляцію їх майбутньої професійної діяльності [60], - таких як: цілепокладання, антиципація; прийняття рішення; планування; програмування; самоконтроль та рефлексія.

На рівні методики і техніки дослідження базовими методологічними положеннями є реалізація вимог до наукового методу дослідження як системи підходів і способів, які відповідають предмету і завданням педагогічної науки, а також її вимогам до проведення дослідження в аспекті необхідності пошуку та вибудовування його стратегії, організації та проведення, способів методологічного аналізу, інтерпретації результатів [24].

Отже, зазначені методологічні підходи стали підґрунтям для моделювання професійної підготовки майбутніх інженерів-програмістів до тестування ПЗ в умовах неформальної освіти. Причому створення зазначеної моделі, на нашу думку, має ґрунтуватися на реалізації таких вимог до прагматичних моделей, адже саме вони відображають поки не існуюче у педагогічній практиці, але бажане, ідеальне та здійснене у майбутньому [24]. Відтак, розроблена нами модель як прагматична має характеризуватися: інгерентністю – погодженістю з середовищем, в якій вона буде функціонувати; простотою – вона має фіксувати головні властивості професійної підготовки; адекватністю – вона має бути точною, повною та істинною, відтворювати реальні характеристики, функції досліджуваного процесу.

Розроблена модель є структурно-функціональною, що означає те, що вона відображає структуру професійної підготовки майбутніх інженерів-програмістів до тестування ПЗ в умовах неформальної освіти, її основні складові, їх взаємозв'язок та взаємозалежність; крім того, - вона побудована відповідно до структури діяльності і презентує функцію кожної з її складових-блоків у досягненні загального результату.

Таким чином, створена нами структурно-функціональна модель складається з трьох блоків, а саме: 1) цільового, 2) змістово-технологічного та 3) результативного (рис. 2.1). *Цільовий блок* презентує мету та завдання професійної підготовки майбутніх інженерів-програмістів до тестування ПЗ на двох рівнях – рівні формальної освіти та неформальної освіти.

Важливим є акцентування уваги на тому, що професійна підготовка майбутніх інженерів-програмістів в цілісній системі освіти має бути інтегрованою та синхронізованою, тобто зміст, форми і методи підготовки у формальній та неформальній освіті мають бути: поєднаними, нерозривно пов'язаними, взаємоузгодженими, упорядкованими, спрямованими на подолання існуючих невідповідностей між ними.

Адже, як свідчать результати здійсненого нами аналізу існуючого досвіду, поданого у попередньому розділі, фахова освіта майбутніх інженерів-програмістів у системі вищої - формальної - освіти недостатньо спрямована на їх підготовку до тестування ПЗ. Створена структурно-функціональна модель, таким чином, презентує узгодженість, синхронізованість двох видів освіти в цілому та всіх їх складових зокрема, що відображено як взаємозалежність цільового, змістово-технологічного та результативного її блоків. Зазначене забезпечує гнучкість, оперативність, практико зорієнтованість та відкритість цілісної підготовки майбутніх інженерів-програмістів до тестування ПЗ.

Відтак, *метою* професійної підготовки майбутніх інженерів-програмістів до тестування ПЗ в умовах неформальної освіти є формування їх готовності до професійної діяльності у єдності трьох компонентів: 1) *ціннісно-мотиваційного*, що визначається мотивацією їх професійної діяльності з тестування ПЗ, а також мотивацією їх подальшого професійного розвитку і навчання, ціннісним ставленням до виконання професійних обов'язків, до тестування ПЗ, спрямованістю на досягнення результату;

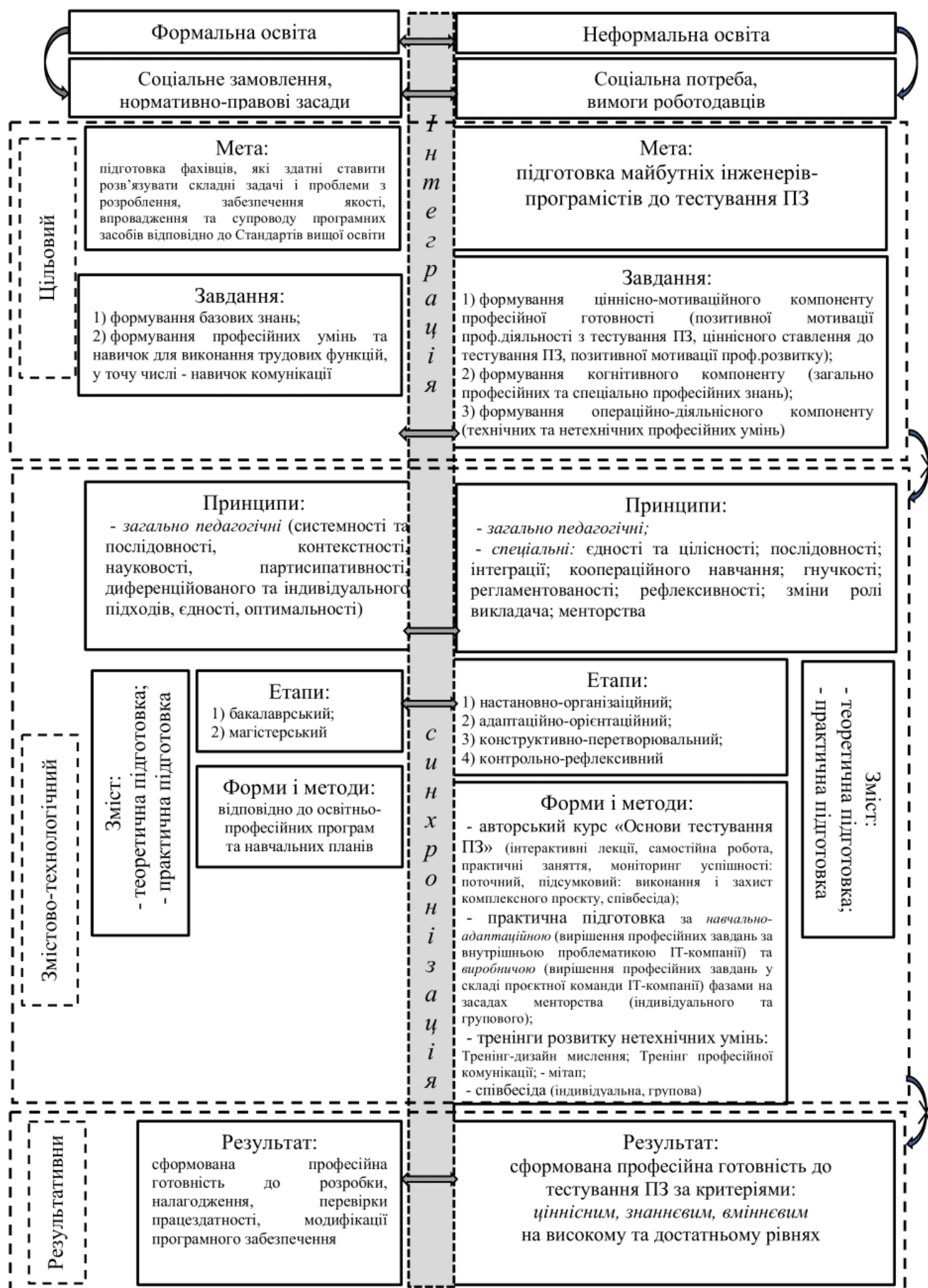


Рис. 2.1. Структурно-функціональна модель професійної підготовки майбутніх інженерів-програмістів умовах неформальної освіти

2) *когнітивного* як системи їх загальних та спеціальних професійних знань; 3) *операційно-діяльнісного*, який визначає сформованість їх технічних та нетехнічних умінь. Зазначені компоненти і зумовлюють *основні завдання* професійної підготовки майбутніх інженерів-програмістів до тестування ПЗ в умовах неформальної освіти щодо їх цілеспрямованого формування та розвитку.

*Змістово-технологічний блок* структурно-функціональної моделі презентує принципи, організаційно-педагогічні умови, зміст, форми і методи професійної підготовки майбутніх інженерів-програмістів до тестування ПЗ в умовах неформальної освіти.

Принципи професійної підготовки майбутніх інженерів-програмістів до тестування ПЗ в умовах неформальної освіти мають рівневу структуру – це: 1) *загальнопедагогічні принципи*, що визначають загальні ідеї, покладені в організацію та здійснення освіти в цілому, фахової вищої ІТ-освіти та неформальної освіти ІТ-освіти зокрема, та 2) *спеціальні принципи*, реалізація яких забезпечує успішність саме професійної підготовки майбутніх тестувальників ПЗ в системі неформальної освіти.

Так, до *загально педагогічних принципів*, на нашу думку, відносяться такі базові принципи як-то: партисипативності (співучасті, залученості, активності та свідомості здобувачів освіти, пріоритетності їх самостійного прийняття рішень); системності та послідовності; контекстності; науковості змісту та форм і методів навчання; диференційованого та індивідуального підходів; єдності навчання, розвитку та виховання; органічної єдності теоретичної та практичної підготовки; оптимального поєднання загальних, групових та індивідуальних форм роботи.

До *спеціальних принципів* нами віднесені такі як:

- принцип *єдності та цілісності* формування та розвитку усіх складових професійної готовності майбутніх інженерів-програмістів до тестування ПЗ;

- принцип *послідовності* формування та розвитку усіх компонентів професійної готовності, який передбачає поетапну підготовку майбутніх тестувальників ПЗ, починаючи з формування та розвитку їх ціннісного ставлення до професійної діяльності, до тестування в цілому, забезпечення створення інформаційної основи дій як опанування ними спеціальних професійних знань та надалі – формування та розвиток технічних та нетехнічних професійних умінь, їх інтеграція в особистий професійний досвід;

- принцип *інтеграції*, який забезпечує доцільне поєднання різних форм і методів активності (інтелектуальної, соціальної), індивідуальної та групової;

- принцип *коопераційного навчання* – пріоритету проєктної роботи – участі майбутніх тестувальників ПЗ у квазі (навчальних) та реальних ІТ-проєктах;

- принцип *гнучкості* змісту, освітніх форма і методів;

- принцип *пріоритетної значущості індивідуального досвіду* професійної діяльності здобувачів освіти, що зумовлює досягнення навчальних результатів, є головним ресурсом навчання;

- принцип *регламентованості* – наявності регламенту різних форм і методів, що забезпечує їх організованість, а також загальна регламентованість соціальними і груповими нормами;

- принцип *рефлексивності* – пріоритету опанування здобувачами механізмів актуалізації рефлексії, домінування рефлексивних методів, синхронізації взаємодіяльності засобами рефлексії;

- принцип *зміни ролі викладача-наставника* з того, хто навчає, контролює й оцінює на того, хто досліджує та комунікує;

- принцип *менторства* – постійного професійного супроводу майбутніх тестувальників ПЗ, що здійснюється у різних формах.

Успішність професійної підготовки майбутніх інженерів-програмістів до тестування ПЗ в умовах неформальної освіти

забезпечується сукупністю науково-методичних засад – базових положень, що є результатом цілеспрямованого відбору, конструювання та застосування елементів змісту педагогічного процесу, методів та організаційних форм для забезпечення успішного досягнення освітньої мети [24]. А саме: запроваджена спеціально організована теоретична підготовка майбутніх інженерів-програмістів, зорієнтована на формування орієнтовної основи їх діяльності з тестування ПЗ; організована практична підготовка майбутніх інженерів-програмістів до тестування ПЗ на засадах менторства; забезпечений цілеспрямований розвиток нетехнічних вмінь (soft skills) майбутніх фахівців.

Виходячи із зазначених принципів, а також визначених особливостей професійної діяльності з тестування ПЗ в ІТ-компаніях та її вимог до фахівці та структури готовності до неї, зміст професійної підготовки майбутніх інженерів-програмістів до тестування ПЗ в умовах неформальної освіти передбачає органічну єдність теоретичної та практичної складових.

Щодо форм та методів підготовки, які також презентовані у змістово-технологічному блоці структурно-функціональної моделі, то потрібно зазначити про таке.

Виходячи з того, що у загально педагогічному контексті формою є зовнішня сторона організації освітнього процесу, що пов'язана із кількістю, часом та місцем, а також порядком його здійснення [Псих и пед., с. 190]. Відтак, за кількістю майбутніх інженерів-програмістів, успішність їх професійної підготовки до тестування ПЗ в умовах неформальної освіти забезпечується використанням різних форм: індивідуальної, командної та колективної (фронтальної).

За місцем проведення така підготовка відбувається як у закладі вищої освіти (за межами формальної освітньої програми), так і (у більшій мірі) на базі ІТ-компанії, як очно, так і в умовах дистанційного (опосередкованого) навчання та спілкування.

Організаційними формами професійної підготовки майбутніх інженерів-програмістів до тестування ПЗ в умовах неформальної освіти, що визначаються домінуючим спрямуванням на: 1) теоретичний аспект підготовки; 2) практичний аспект підготовки та 3) контроль знань та умінь [125, с. 192], є інтерактивні, практико зорієнтовані форми. А саме: 1) авторський курс «Основи тестування програмного забезпечення»; 2) практична підготовка на засадах менторства, що передбачає у тому числі роботу в проєктній команді ІТ-компанії; 3) тренінги розвитку нетехнічних умінь «Тренінг-дизайн мислення» та «Тренінг професійної комунікації».

Методи професійної підготовки майбутніх інженерів-програмістів до тестування ПЗ в умовах неформальної освіти відповідають таким загальним ознаками - є:

- *проблемними*: їх зміст є проблемним, забезпечує усвідомлення майбутніми тестувальниками шляхів вирішення професійних проблем у самостійному пошуку;

- *проєктними* – відтворюють діяльність проєктної команди;

- *ситуативними* – побудовані на конкретних професійних проблемних ситуаціях, що вимагають актуалізації професійних знань та технічних і нетехнічних умінь майбутніх фахівців;

- *діалогічними* – передбачають паритетні, партнерські стосунки з іншими учасниками проєктної команди;

- *рефлексивними* – забезпечують розвиток рефлексивних умінь, передбачають рух мисленнєвої діяльності майбутніх тестувальників за етапами рефлексивної діяльності (рух по рефлексивному колу);

- *розвивальними* – забезпечують цілісний особистісний та професійний розвиток майбутніх інженерів-програмістів, розвиток їх ціннісних орієнтацій, метакогнітивних, організаційно-вольових та комунікативних якостей, мотивації особистісного та професійного самовдосконалення.

Зазначені форми і методи ґрунтуються на спільній діяльності учасників, їх активній взаємодії, спільному вирішенні проєктних завдань, індивідуальній та груповій рефлексії, зануренні в атмосферу реальної професійної комунікації.

Зміст, форми і методи професійної підготовки майбутніх інженерів-програмістів до тестування ПЗ в умовах неформальної освіти розгортаються, відповідно до створеної моделі, крізь декілька етапів.

Зазначимо, що у наукових дослідженнях (Н. Павлик [115], О. Шапочкіна [168] та ін.) обґрунтовані такі загальні етапи навчально-пізнавальної діяльності особистості в умовах неформальної освіти як: 1) інтуїтивний, що розпочинається з появи мотиву пізнавальної діяльності, стійкого інтересу до певної нової діяльності; 2) нормативний, який характеризує процес пошуку методів, засобів, прийомів організації пізнавальної діяльності; 3) активний – усвідомлений вибір особистістю різних форм організації неформальної освіти, застосуванням сформованих компетенцій у професійній діяльності та життєдіяльності в цілому; 4) креативний – висвітлює пошук, прагнення особистості до подальшого особистісного та професійного розвитку, удосконалення набутих компетенцій та вихід на новий рівень.

Основними етапами формування готовності майбутніх інженерів-програмістів до тестування ПЗ в умовах неформальної освіти, на нашу думку, є такі.

Перший – *настановно-організаційний*, що розгортається з моменту усвідомлення майбутніми інженерами-програмістами потреби в опануванні діяльністю з тестування ПЗ, появи стійкого інтересу до зазначеної ІТ-сфери та входження них в нову для них відповідну освітню програму з її певними організаційними викликами. Відтак, на цьому етапі відбувається рух з інтуїтивної, підготовчої з психологічної точки зору стадії (Н. Павлик, О. Шапочкіна та ін.) - потребнісної, до суб'єктної, усвідомленої – настановної, стадії наміру.

Другий етап – *адаптаційно-орієнтаційний*, передбачає опанування майбутніми інженерами-програмістами орієнтувальної (інформаційної) основи їх діяльності з тестування ПЗ, тобто актуалізацію загальних професійних знань та оволодіння спеціальними професійними знаннями з тестування ПЗ.

Третій етап – *конструктивно-перетворювальний*, спрямований у першу чергу на формування та розвиток технічних і нетехнічних умінь майбутніх інженерів-програмістів, формування досвіду їх діяльності у складі команди реального ІТ-проєкта, що забезпечує формування операційно-діяльнісного компоненту їх професійної готовності до тестування ПЗ.

Четвертий етап – *контрольно-рефлексивний*, забезпечує: узагальнення, зміцнення та поглиблення когнітивного й операційно-діяльнісного компонентів професійної готовності майбутніх тестувальників ПЗ; коригування їх освітньої траєкторії за результатами оцінювання та самооцінювання здобутих освітніх результатів; пролонгацію їх освітньої та самоосвітньої діяльності і, таким чином, вихід на новий етап професійного розвитку в умовах неформальної освіти.

На кожному з етапів професійної підготовки майбутніх інженерів-програмістів до тестування ПЗ в умовах неформальної освіти відбувається *моніторинг її якості (поточний та підсумковий контроль)*, що здійснюється у різних формах: тестування, виконання практичних завдань, самооцінка та рефлексивний аналіз здобувачами власної діяльності, які є традиційними у системі формальної ІТ-освіти. Однак, особливою, провідною формою контролю якості такої підготовки в ІТ-компаніях є *співбесіда (інтерв'ю)*. Вона використовується в абсолютній більшості великих ІТ-компаній, представлених на українському ринку, таких як EPAM Systems, GlobalLogic, NIX Solutions, DataArt та ін. Адже співбесіда є одним з найбільш поширених методів відбору і оцінки персоналу [Гончаров; Управление персоналом]; вона дає змогу: 1) здійснити оцінку

професійної компетентності претендента - його професійних знань та вмінь, професійних, індивідуально-психологічних та психофізіологічних якостей; 2) визначити, наскільки претендент здатний до подальшого професійного розвитку; 3) встановити достовірність інформації, що надається кандидатом про свої професійні та особистісні якості, досвід роботи тощо. Взагалі співбесіди за своєю структурою розподіляються на:

- 1) жорстку (структуровану): здійснюється за раніше підготовленим жорстким планом, в якому визначені теми для розмови та запитання здобувачеві; застосовується для оцінки професійних знань та умінь;
- 2) вільну (неструктуровану): не має чітко визначеного плану (намічаються лише загальні теми спілкування), інтерв'юер за своєю роллю є фасилітатором. Для його проведення зазвичай намічаються тільки основні теми;
- 3) комбіновану: здійснюється за раніше підготовленим планом, що надає можливість оцінити професійні знання та вміння здобувача у межах чітко заданих запитань, та передбачає вільну бесіду, що забезпечує виявлення професійних та особистісних якостей претендента. За формою організації співбесіди в ІТ-компанії розрізняють на такі види як:
  - 1) індивідуальна - один інтерв'юер працює окремо з кожним новим здобувачем, що дозволяє встановити найбільш тісний контакт; однак її недоліками є ризик суб'єктивної оцінки претендента;
  - 2) групова - зі здобувачем спілкуються кілька інтерв'юерів, ролі яких чітко розподілені; вона забезпечує надійну оцінку професійних знань, умінь та якостей.

Крім того, за поставленими цілями в ІТ-компаніях застосовують:

- 1) попередню («відсівальну») співбесіду, завданнями якої є визначення та відбір з усіх кандидатів тих, які будуть запрошені для продовження переговорів. Така співбесіда може бути організована різними способами, проте частіше всього – у формі завдань по електронній пошті, на основі яких здійснюється первинний відбір;
- 2) відбіркову співбесіду – як одиничну, так і серійну, групову, її метою й є безпосередня оцінка професійної компетентності претендента.

Самий процес співбесіди в ІТ-компаніях, як правило, розподіляється на декілька логічних блоків питань. Кожна частина має за мету виділити та оцінити окрему складову професійної компетентності кандидата. До основних блоків співбесіди відносяться:

1) біографія та загальні питання. Перше, що хочуть почути інтерв'юери – загальні факти з біографії, дані про освіту, цілі, особистісні характеристики та професійні вміння кандидата. Основна мета цього блоку – виявити, наскільки кандидат здатний охарактеризувати себе як особистість, відокремити найбільш значущі моменти зі своєї біографії. Саме на цьому етапі найбільш яскраво визначаються його індивідуально-психологічні і психофізіологічні якості;

2) технічна частина: теорія. Так як ІТ-компанія зацікавлена наймати найкращих фахівців, незважаючи на те, що більша частина їх знань і досвіду може і не знадобитися для поставлених завдань, їй важливо визначити реальний рівень технічних знань кандидата. В цьому блоці співбесіди зазвичай піднімаються питання, що стосуються синтаксису та семантики мов програмування, базових знань методології і загального процесу розробки. Відповіді на ці питання очікуються як від розробників з багаторічним стажем, так і від новачків у цій галузі;

3) технічна частина: практика. Практична частина співбесіди зазвичай полягає в написанні невеликої абстрактної програми, коду, документації або інших продуктів, що відносяться до конкретної сфери ІТ, на яку зорієнтована посада. Ці завдання спрямовані не стільки на кінцевий результат, скільки на розуміння підходу здобувача до вирішення завдань. Людина, що проходить співбесіду, має продемонструвати хід своїх думок і показати знання предметної галузі та вміння шукати рішення проблем на практичному прикладі;

4) досвід роботи, попередні проекти. Цей блок співбесіди дає змогу здобувачу більш докладно поділитися своїм досвідом роботи, ситуаціями, з якими він стикався на проектах та способами їх вирішення. У разі відсутності

реального досвіду роботи в ІТ-компаніях важливо розуміти, що для інтерв'юера в пріоритеті знаходиться не кількість різноманітних ситуацій, технологій та проєктів, а їх специфіка та залученість в них самого здобувача;

5) іноземна мова. В реаліях сучасного ІТ, спрямованого на всебічну роботу із зарубіжними співробітниками та клієнтами, рівень володіння іноземними мовами (в основному, англійською) є вкрай важливим й, іноді, вирішальним фактором відбору. На цьому етапі інтерв'юера цікавить не скільки академічний рівень іноземної мови, скільки вміння здобувача продовжувати діалог та спілкуватися із професійних питань. Саме тому перехід на спілкування іноземною мовою може відбутися раптово і, зазвичай, в контексті розмови про попередній досвід роботи або практичні навички;

б) мотивація, перспективи. В цьому блоці питань співбесіди інтерв'юер намагається оцінити вмотивованість потенційного кандидата, його бажання вирішувати професійні завдання і реальний інтерес в отриманні даної посади. На цьому етапі здобувачу важливо показати свою зацікавленість в новому місці роботи, подальшому розвитку, придбанню нового досвіду;

7) питання до інтерв'юерів. Наприкінці співбесіди інтерв'юер зазвичай пропонує кандидату самому задати питання і дізнатися необхідні для себе відомості. Насправді, форма, контекст і зміст питань дозволяють оцінити професіоналізм та мотивацію самого здобувача так само, як і питання у попередніх блоках.

Вкрай важливо відмітити, що незважаючи на перспективу отримання нової роботи, результати співбесіди визначаються на загальному професійному розвитку фахівця ІТ-сфери, адже вона є певною формою підвищення його кваліфікації. Підготовка до співбесіди вимагає: поглиблення, оновлення та систематизації професійних знань, здобуття нових знань в суміжних галузях; активізацію та розвиток як технічних, так і нетехнічних умінь – актуалізацію метакогнітивних та метарегулятивних механізмів.

Для студентів ІТ-спеціальностей розуміння самого процесу співбесіди і готовність до нього є головним чинником отримання першої роботи і

становлення себе як IT-фахівця. Зазвичай основна причина відмов у прийомі на роботу молодих спеціалістів полягає у невдало пройдених співбесідах, що, в свою чергу, викликано відсутністю навичок правильного ведення діалогу з інтерв'юером, вміння зацентувати увагу на своїх технічних знаннях та професійної вмотивованості. Варто також враховувати, що для багатьох студентів-випускників проходження свого першого інтерв'ю – надзвичайно травматична, стресова ситуація і в багатьох випадках схвильованість кандидата і незвичність такої форми контролю для нього дуже негативно позначається на якості відповідей.

Відтак, співбесіда є важливою формою не тільки контролю якості, вона є *особливою наскрізною* формою професійної підготовки майбутніх інженерів-програмістів до тестування ПЗ, що виявляється на всіх її етапах та актуалізується у процесі реалізації авторського курсу «Основи тестування ПЗ», практичної підготовки та тренінгів розвитку нетехнічних умінь як:

- 1) моделювання ситуацій професійних співбесід з їх подальшим аналізом, у тому числі – помилок та шляхів отримання найкращого результату;
- 2) форма проміжного (модульного) та підсумкового контролю;
- 3) окремий метод діагностування професійної готовності майбутніх фахівців з тестування ПЗ.

*Результативний блок* структурно-функціональної моделі презентує критерії, показники та рівні професійної підготовки майбутніх інженерів-програмістів до тестування ПЗ в умовах неформальної освіти – сформованості їх відповідної професійної готовності.

На підставі того, що під критерієм у сучасній педагогічній науці розуміється сукупність ознак, на основі яких складається оцінка умов, процесу та результату освітньої діяльності, що відповідають її цілям [42, с. 434], а показники - цими ознаками, які є, у свою чергу, конкретними характеристиками, що підлягають вимірюванню, діагностиці. А також ґрунтуючись на структурній організації професійної готовності майбутніх

інженерів-програмістів, критеріями їх успішної підготовки до тестування ПЗ є такі:

1) *ціннісний*, що відображає їх: ставлення до тестування ПЗ, до виконання професійних обов'язків; мотивацію професійної діяльності з тестування ПЗ, а також мотивацію подальшого професійного розвитку і навчання, загальною спрямованістю на досягнення результату;

2) *знаннєвий*, показниками якого є наукові, усвідомлені (відрефлексовані), системні та дієві знання про: методології розробки ПЗ; фундаментальні основи тестування, техніки тест-дизайну, принципи роботи допоміжного ПЗ для тестування; основи естимації та планування; основи алгоритмів, промислового програмування, процесів та методології розробки; архітектури комерційних додатків; мережеві технологій і адміністрування; операційні системи, засоби віртуалізації тощо;

3) *вміннєвий*, що віддзеркалює сформованість усіх груп професійних умінь тестувальників ПЗ – як технічних, так і нетехнічних, показниками якої є їх:

- стійкість та повторюваність дій (виявляються у збереженні дій та їх відтворюваності у нових ситуаціях);

- ефективність дій – результативність кожної групи технічних та нетехнічних умінь, що виявляється у досягненні визначеного професійними завданнями (етапами проєктної роботи) результатом.

Рівнями професійної підготовки майбутніх інженерів-програмістів до тестування ПЗ у неформальній освіті є такі три, а саме:

1) *початковий*, який характеризується: нестійкою, амбівалентною мотивацією здобувачів освіти до професійної діяльності з тестування ПЗ, до подальшого професійного розвитку, а також переважно позитивним ставленням до тестування ПЗ, виконання професійних обов'язків; несистемними та слабо відрефлексованими (як загальними, так і спеціальними) професійними знаннями; нездатністю відтворити професійні

технічні вміння у нових, нетипових ситуаціях, їх недостатньою ефективністю; недостатньою сформованістю нетехнічних умінь усіх груп;

2) *достатній*, що виявляється в: стійкій позитивній мотивації здобувачів освіти до професійної діяльності з тестування ПЗ, до подальшого професійного розвитку, а також ціннісним ставленням до тестування ПЗ, виконання професійних обов'язків; усвідомленості, системності та дієвості професійних знань (як загальних, так і спеціальних); недостатній стійкості, повторюваності та ефективності окремих технічних та нетехнічних умінь;

3) *високий*, який визначається: стійкою внутрішньою мотивацією майбутніх тестувальників до професійної діяльності з тестування ПЗ, до подальшого професійного розвитку, а також ціннісним ставленням до тестування ПЗ, виконання професійних обов'язків; усвідомленістю, системністю та дієвістю загальних та спеціальних професійних знань як орієнтаційної основи їх професійної діяльності; сформованістю усіх груп технічних та нетехнічних умінь, їх стійкістю, повторюваністю та ефективністю.

Для вимірювання рівнів сформованості професійної готовності майбутніх інженерів-програмістів до тестування ПЗ в умовах неформальної освіти важливим є визначення методологічного підходу до їх оцінювання.

У цьому контексті важливо зазначити, що в умовах неформальної освіти оцінювання якості досягнутих результатів визначається науковцями як одна з невирішених актуальних проблем. Так, у світовій практиці неформальної освіти існують різні моделі оцінки, що визначають специфіку валідації її результатів. Серед таких моделей виокремлюються такі (Н. Павлик [115], О. Сандецька [140] та ін.):

1) модель об'єктивної оцінки (або технологічна) - передбачає оцінювання досягнення освітніх цілей незалежно від обставин, оцінка відображає знання здобувачів освіти за певними стандартами;

2) модель суб'єктивної оцінки – в більшій мірі базується на самооцінці здобувачами неформальної освіти отриманих результатів;

3) змішана модель, що передбачає як самооцінку здобувачем неформальної освіти отриманих результатів, так й об'єктивну оцінку результатів. Вона здійснюється за рівнями моделі суб'єктної оцінки Д. Кіркпатріка, яка переважно застосовується у тренінговій роботі. Зазначена модель передбачає: самооцінку емоційного стану (задоволеність від навчання); самооцінку навченості за результатами виконання навчальних завдань; зовнішню оцінку – підтвердження достовірності рівня навченості третіми особами (роботодавцями, колегами тощо); рівень результатів (непряме оцінювання без кількісного відображення в конкретних показниках).

Як свідчать результати багатьох досліджень у сфері забезпечення якості формальної та неформальної освіти, саме у неформальній освіті оцінка результатів за змішаною моделлю є найбільш достовірною. Отже, вона має здійснюватися як на рівні самооцінки здобувачів як замовників освітніх послуг, так і на рівні зовнішньої оцінки, у тому числі – третьою стороною. Це вимагає визначення особливих діагностичних процедур, що ґрунтуються на суб'єктивних та об'єктивних методах.

Отже, підсумовуючи усе вищезазначене, наступним етапом дисертаційного дослідження, відповідно до його нормативних вимог, є екстраполяція розробленої моделі, що потребує детального розкриття презентованих науково-методичних засад.

### **2.3. Спеціально організована теоретична підготовка майбутніх інженерів-програмістів до тестування програмного забезпечення**

Підготовка інженерів-програмістів до тестування ПЗ є особливим процесом, що вимагає альтернативного підходу як до структури, так і до його організації. Враховуючи результати здійсненого аналізу сутності професійної готовості інженера-програміста до тестування ПЗ, які були презентовані на попередніх етапах дисертаційного дослідження, основними орієнтирами при розробці змісту освітнього процесу в умовах

неформальної освіти, що спричинено недостатнім її забезпеченням у формальній освіті, нами визначено такі:

- *на ціннісному рівні*: забезпечення сформованості позитивної мотивації діяльності із тестування ПЗ, формування ціннісного ставлення до професійної діяльності, позитивної мотивації професійного розвитку, що в цілому складають ціннісно-мотиваційний компонент;

- *на теоретичному рівні*: забезпечення опанування як загальними знаннями в галузі інформаційних технологій, методологій, підходів та інструментів розробки ПЗ, так і спеціалізованими знаннями з тестування ПЗ, основних його типів і видів, принципів роботи з тестовою документацією і проходження тестових сценаріїв, що у своїй цілісності є когнітивним компонентом;

- *на практичному рівні*: забезпечення сформованості технічних і нетехнічних професійних умінь, що у своїй єдності складають операційно-діяльнісний компонент професійної готовності майбутніх тестувальників ПЗ.

Досягнення зазначеного відбувається за допомогою форм і методів навчання, презентованих в розробленій структурно-функціональній моделі. Для теоретичної підготовки, й відповідно, формування у першу чергу когнітивної складової готовності майбутніх інженерів-програмістів до тестування ПЗ розроблено *авторський курс «Основи тестування програмного забезпечення»*. Він розрахований на 3 кредити (90) годин, з яких: 20 годин – лекційні заняття, 18 годин – практичні заняття, 52 години – самостійна робота (Додаток В). Реалізація курсу передбачена в 5/7 семестрі формального навчання (третій/четвертий курс). Це зумовлено тим, що студенти вже мають необхідний теоретичний базис для успішного засвоєння матеріалу, оскільки вивчали такі дисципліни як: «Основи алгоритмів»; «Дискретна математика»; «Об'єктно-орієнтоване програмування»; «Вступ до тестування»; «Основи управління проектами»; «Архітектура і проектування» та ін. Як свідчать результати аналізу

освітніх програм підготовки вітчизняних закладів вищої освіти, зазначені дисципліни входять до блоку обов'язкових.

Суттєву роль відіграє й те, що студенти попередньо мали змогу підвищити свій рівень знання іноземної мови (дисципліна «Англійська мова за професійним спрямуванням»), що дозволяє передбачити в межах курсу в переліку матеріалів для самостійного вивчення документацію, міжнародні шаблони, а також спеціалізовані профільні видання іноземними мовами.

Крім того, розробка авторського курсу ґрунтувалась на концепції І. Лернера [88] щодо способів та рівнів засвоєння змісту освіти. Її головними положення є такі:

1) першим, базовим компонентом засвоєння змісту освіти є знання, при чому уміння як досвід здійснення вже відомих способів діяльності є другим компонентом [88]. Перший компонент складається зі знань про світ та знань про способи діяльності, передбачає рух від - сприймання, розуміння та запам'ятовування знань, а також застосування знань у знаковій ситуації, до - застосування знань у новій ситуації;

2) єдиним механізмом формування умінь є неодноразове: повторюване відтворення відомих способів дій за допомогою різноманітних вправ, технік, методів;

3) формування вмінь має здійснюватися поетапно: від виконання способів діяльності за зразком до їх виконання у новій ситуації, що забезпечує формування досвіду здійснення відомих способів діяльності та досвіду творчої діяльності [88];

4) у процесі формування вмінь мають бути задіяні різні дидактичні методи: - інформаційно-рецептивний, що реалізується у формі лекції, доповіді, екскурсії, тощо завдяки різним засобам (візуальним, текстовим, практичним) та формам пред'явлення (говоріння і слухання, показ та споглядання, презентація предмету та способів дії з ним) з метою усвідомленого сприймання та запам'ятання інформації; - репродуктивний -

передбачає відтворення знань та способів розумової і практичної діяльності за допомогою різних завдань, вправ, практичних та лабораторних робіт, що мають забезпечувати опонування алгоритму дій, поелементно (стосується знань) та поопераційно (стосується вмінь); - проблемний як такий метод оволодіння досвідом творчої діяльності, що забезпечує вирішення проблемних ситуацій; - евристичний - спрямований власне на знаходження нового за допомогою різноманітних евристик – технік відкриттів; - дослідницький, який ґрунтується на вирішенні проблемних ситуацій завдяки пошуковій діяльності, проходженню етапів дослідження як-то: а) спостереження та вивчення фактів, явищ; б) визначення незрозумілих явищ та постановка проблем; висунення гіпотез та побудови плану дослідження; в) здійснення плану дослідження; г) формулювання рішення та його обґрунтування; д) перевірка рішення та здійснення практичних висновків [88, с. 104];

5) освітній результат може бути досягненим через організацію навчання на трьох базових рівнях: - рівень усвідомленого сприйняття та запам'ятовування інформації; - рівень відтворення способу діяльності за зразком; - рівень творчого використання знань та вмінь у незвичній, нестандартній ситуації [88, с. 177];

б) оволодіння творчою діяльністю можливе лише при включенні суб'єктів навчання у вирішення спеціально розробленої системи значущих для них проблемних ситуацій та завдань [88, с. 179];

Отже, у нашому дослідженні розроблений курс є тим формувальним заходом, що забезпечує засвоєння майбутніми інженерами-програмістами теоретичного компоненту змісту їх підготовки до тестування ПЗ – спеціальних професійних знань та актуалізацію (з коригуванням при необхідності) загальних професійних знань. Проте і сам він будується за вищезазначеними етапами формування професійних умінь та досвіду їх реалізації студентами у стандартних і нестандартних умовах, що вимагають творчої діяльності.

Відтак, метою курсу є формування у студентів системи спеціальних теоретичних, організаційних і технічних знань в сфері розробки та тестування ПЗ. Відповідно, основними завданнями визначено такі:

- розкриття сутності тестування ПЗ, а також технічних і організаційних аспектів його впровадження в процес розробки;
- сприяння професійному самовизначенню студентів-програмістів завдяки визначенню організаційної та процесуальної ролі тестувальника ПЗ в команді розробки;
- формування стійких знань щодо теорії тестування ПЗ, його структурних складових, виробничих практик, звітної документації.

В результаті вивчення курсу студент повинен знати:

- зміст базових понять дисципліни: тестування, контроль якості, вимоги, методологія розробки, планування, естімація, тест-план, тест-стратегія, тест-кейс, чек-лист, дефект, звіт, автоматизація, фреймворк;
- теоретичні основи тестування ПЗ, його структурну класифікацію;
- поняття методології розробки ПЗ, їх класифікацію та найбільш популярних представників;
- сутність тест-дизайну, технік написання та проходження тестових сценаріїв;
- сутність, структуру, правила оформлення і контекст використання основних документів тестування;
- організаційні аспекти планування процесу тестування і естімації робочих завдань;
- поняття автоматизації, технічні та структурні особливості її впровадження в процес тестування.

Вміти:

- розкривати сутність основних понять і концептуальних положень дисципліни;
- аналізувати і тестувати вимоги до ПЗ;
- застосовувати й аргументувати вибір технік тест-дизайну;

- створювати, оформлювати та реалізовувати тестові сценарії;
- застосовувати техніки естимації і оцінювати трудовитрати завдань з тестування;
- створювати, оформлювати та аналізувати звіти про дефекти і тестування;
- аргументувати вибір інструментів для автоматизації тестування та вміти користуватися ними для вирішення професійних завдань.

Результати вивчення теорії питання, практичний досвід роботи в ІТ-сфері, дає підстави констатувати, що тестування, з одного боку, є похідною від загального процесу розробки і лише одним з його етапів, з іншого, - самостійною галуззю знань зі своїм теоретичним базисом, термінологією, процесами і документацією. Відтак, викладання цього предмету вимагає певної логіки: паралельне розкриття як специфічних тем, що безпосередньо стосуються сфери тестування ПЗ, так і глобального процесу розробки, а також їх взаємозв'язку. Враховуючи це, в основу курсу покладено алгоритм, що реалізується за такими етапами (рис. 2.2), а саме:



Рис. 2.2. Етапи реалізації авторського курсу «Основи тестування ПЗ»

- по-перше, - *вступ у предметну галузь*. На цьому етапі відбувається формулювання загальної проблеми тестування, розкриття передумов його виникнення, постановка основних цілей і завдань, характеристика особливостей діяльності в сучасних умовах стрімкого розвитку інформаційно-комунікаційної сфери. Результатом етапу є формування у студентів базового понятійного апарату, усвідомлення чинників актуальності та соціального попиту;

- по-друге, - *визначення місця в глобальному процесі*. Цей етап спрямований на усвідомлення студентами місця і ролі тестування серед інших етапів і практик процесу розробки ПЗ;

- по-третє, - *сутність професійної діяльності*, що передбачає розкриття особливостей професійної діяльності тестувальника в виробничих умовах, визначення її характеристики основних алгоритмів, методів роботи, а також інструментарію, що використовується;

- по-четверте, - *умови, організація і глобальний контекст діяльності*. Цей етап забезпечує формування у студентів уявлення про організацію роботи всередині команди тестування, зокрема: визначення робочих завдань, їх естімації, умов виконання/невиконання, планування діяльності, а також подальшої оцінки роботи;

-по-п'яте, - *результати діяльності* - етап, що спрямований на аналіз результатів діяльності з тестування, визначення структури звітності та алгоритмів поведінки за умов різних виробничих ситуацій, ефективних технік оцінювання і аналізу артефактів тестування;

- по-шосте, - *можливості вдосконалення та оптимізації процесу*. Цей етап є останнім, що передбачає вивчення виробничих технічних та організаційних практик, які підвищують результативність форм і методів тестування в певних виробничих умовах, зокрема, автоматизації як найбільш поширеної.

Контент курсу, згідно запропонованої логіки, розділено на шість відповідних модулів:

- Модуль 1. «Вступ до тестування програмного забезпечення»;
- Модуль 2. «Тестування в контексті сучасних методологій розробки»;
- Модуль 3. «Тест-дизайн»;
- Модуль 4. «Планування і естімація»;
- Модуль 5. «Дефекти і звітність»;
- Модуль 6. «Автоматизація тестування».

Кожен з цих модулів, у свою чергу включає один або декілька тематичних блоків, які структурно складаються з наступних елементів (рис. 2.3):

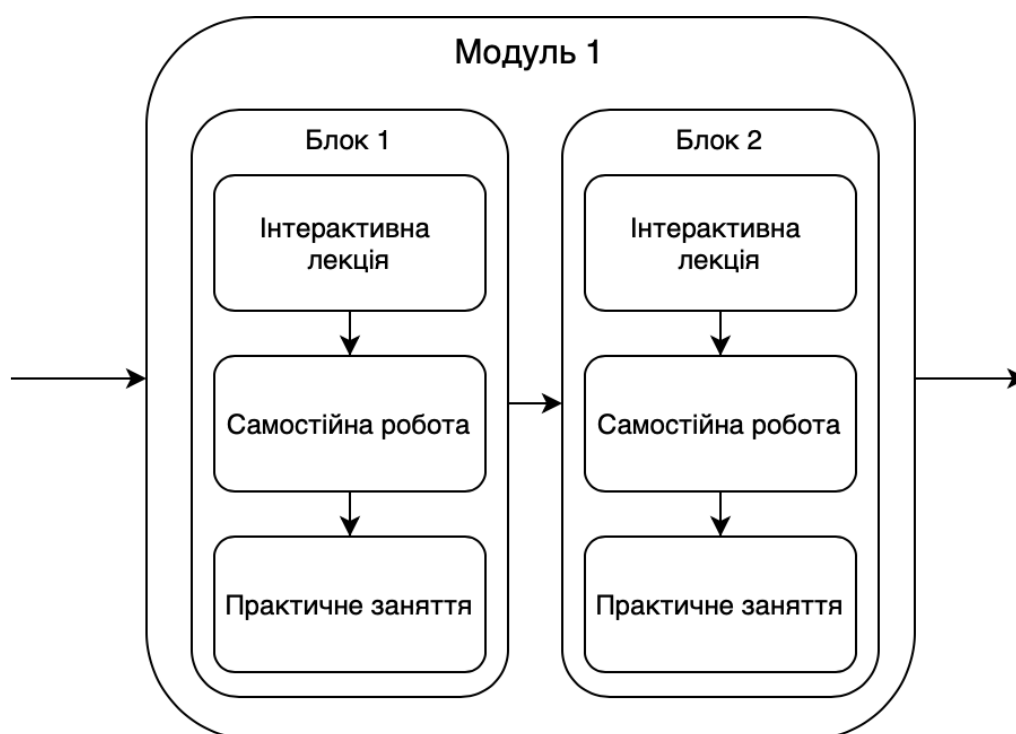


Рис. 2.3. Структура модулів авторського курсу «Основи тестування ПЗ»

1) *інтерактивна лекція*. Вибір цієї форми (і методу) роботи зумовлений декількома чинниками.

По-перше, - майбутні фахівці вже мають певне загальне уявлення про предмет тестування і його теоретичний базис, оскільки, як свідчать навчальні плани підготовки, на попередніх етапах вони прослухали курс «Якість програмного забезпечення та тестування» (3-5 кредитів залежно від закладу вищої освіти).

По-друге, - вона забезпечує безпосередній контакт з аудиторією й необхідні умови для відкритого діалогу [68; 94]. У процесі інтерактивної лекції викладач-досвідчений практик, з одного боку, може здійснювати постійний моніторинг рівня засвоєння матеріалу, шляхом постановки проблемних питань як всім учасникам спілкування разом, так і кожному окремо, з іншого, - відповідати на запитання і більш детально зупинитись на проблемних питаннях, які потребують додаткового роз'яснення, підвищуючи ступінь усвідомлення і засвоєння навчального матеріалу [100]. Інтерактивна лекція передбачає кардинальну зміну традиційної поведінки викладача та студента, що до сих пір домінує у вітчизняних закладах вищої освіти; вона надає студентам стати активними діячами, а викладач не виступає у ролі джерела інформації, а стає організатором дискусії, фасилітатором, консультантом тощо.

По-третє, - вона: забезпечує емоційність подання навчального матеріалу; дає змогу змінювати різні види діяльності студентів, включати методи проблемного, дослідницького навчання, активізувати пізнавальну діяльність;

По-четверте, - вона забезпечує створення комфортних умов навчання в активній взаємодії усіх її учасників, а також, що для нас особливо значущим, - створює умови для формування позитивної внутрішньої мотивації навчально-професійної, майбутньої професійної діяльності студентів та мотивації їх професійного розвитку (М. Огренич [107] та ін.);

2) *самостійна робота*. Враховуючи значний обсяг теоретичного матеріалу, його постійне оновлення, значна увага у програмі курсу приділяється організації самостійної роботи студентів. В її основу покладено дві основні складові: 1) ознайомлення з додатковою літературою, яка розширює варіативність підходів до розкриття теми, що розглядається, та 2) виконання практичного завдання, яке спрямоване на апробацію і закріплення отриманих знань, формування технічних професійних умінь;

3) *практичне заняття*. Ця форма роботи спрямована на коригування інформаційної/орієнтовної основи дій студента шляхом демонстрації, обговорення (обґрунтування) результатів самостійної практичної роботи і внесення, за необхідності, коректив в її результати.

Важливим компонентом програми є моніторинг продуктивності навчальної діяльності студентів, який відбувається за принципом накопичувальної бально-рейтингової системи, прийнятої у вітчизняних закладах вищої освіти. Загальна кількість балів, яку студенти можуть отримати в ході вивчення дисципліни, складає 100, зокрема: 60 балів – за результати поточного контролю (виконання практичних завдань), 40 балів нараховується за результатами обов'язкового підсумкового контролю, який передбачає виконання і захист комплексного проєкту, а також діагностику рівня теоретичних знань шляхом співбесіди з експертною комісією.

Відтак, *перший модуль* курсу - «Вступ до тестування програмного забезпечення» - розрахований на формування у майбутніх фахівців та актуалізацію понятійного апарату з теорії тестування ПЗ. Він передбачає одну вступну лекцію, спрямовану на знайомство групи з викладачем та студентів між собою, а також розкриття наступних базових питань:

- історія розвитку індустрії комерційної розробки ПЗ, поява тестування як окремої галузі;
- визначення понять якості, контролю якості, тестування;
- визначення мети тестування, його основних завдань;
- поточний стан галузі, основні тенденції та потенційні шляхи розвитку;
- професійний портрет сучасного тестувальника ПЗ, ринок праці, рівень заробітних плат, перспективи розвитку кар'єри;
- огляд основних вимог до фахівця з тестування в флагманських компаніях з розробки ПЗ.

Враховуючи інтродуктивну та абстрактну спрямованість, модуль не передбачає проведення практичних занять, проте, забезпечує акцентування

уваги на самостійній роботі студентів, зокрема вивченні рекомендованої літератури (професійних видань з теорії тестування, міжнародної і вітчизняної статистики сучасної сфери інформаційних технологій, індустріальних звітів та ін.).

Основною проблематикою *другого модуля* - «Тестування в контексті сучасних методологій розробки» - є визначення місця етапу тестування в загальному процесі розробки ПЗ, а також умов і передумов початку тестування та його завершення. Він складається з двох блоків. Перший - присвячений методології побудови процесів розробки ПЗ в ІТ-компаніях. Він передбачає лекцію за темою «Тестування в контексті сучасних методологій розробки», яка розгортає такі питання:

- поняття методології процесу розробки ПЗ;
- історія розвитку методологій і форм організації процесу розробки;
- класифікація методологій розробки, особливості груп;
- монолітні методології розробки, їх реалізація, особливості побудови процесу тестування;
- гнучкі методології розробки, їх види, особливості побудови процесу тестування;
- детальний аналіз методології «Scrum»: процеси, проєктні ролі, формальні наради та заходи.

Самостійна робота передбачає: 1) опрацювання додаткової літератури і 2) виконання практичного завдання: за вибраною сферою виробництва, процес якої може бути представлений в форматі методології Scrum: визначити та опрацювати проєктні ролі, описати глобальні процеси, акцентуючи увагу на тестуванні і контролі якості та розробити схему ітерацій роботи команди, наприкінці аналізуючи й оцінюючи її ефективність, зробити відповідні висновки.

Обговорення і захист роботи відбувається на практичному занятті за темою «Візуалізація сучасних методологій розробки програмного забезпечення і місця тестування в ньому». Інтерпретація складних

методологій розробки на більш звичних і зрозумілих для студентів предметних сферах дозволяє провести аналогії між викладеним лекційним матеріалом і власним життєвим досвідом, формуючи у них розуміння описаних процесів і ролі кожного члена команди.

Другий блок модуля спрямований на розкриття вимог до ПЗ, що є концептуальною основою всього процесу тестування. Зміст лекції за темою «Робота з вимогами», яка входить до його складу, передбачає розкриття таких питань як:

- поняття «вимоги до програмного забезпечення»;
- види і типи вимог;
- джерела отримання вимог;
- процес розробки, аналізу і тестування вимог;
- критерії якості вимог;
- тестування вимог до ПЗ.

Інтерактивна складова лекції також передбачає наведення прикладів як абстрактних, так і реальних вимог до ПЗ з подальшим аналізом і виявленням в них таких ознак як: вторинність, неоднозначність, нездійсненність, двозначність, неперевірюваність та ін.

Як самостійна робота, студентам запропоновано навести приклади коректних і некоректних вимог до ПЗ відповідно до розглянутого на лекції матеріалу з подальшою їх презентацією та обговоренням на практичному занятті за темою «Робота з вимогами».

*Третій модуль* - «Тест дизайн» - є «наріжним каменем» всього курсу, оскільки передбачає розкриття і вивчення сутності професійної діяльності фахівця з тестування ПЗ - підтримку, написання і роботу з тестовими сценаріями. Аналогічно до попереднього модуля, він також складається з двох логічних блоків, що включають в себе по одному лекційному і практичному заняттю, а також завдання для самостійної роботи.

Перший блок - «Види і типи тестування» - спрямований на ознайомлення студентів з теоретичними аспектами тест-дизайну. В процесі лекційного заняття розглядаються такі питання як-то:

- еволюція і причини формалізації процесу тестування;
- форми і методи тестування, класифікація видів і типів;
- функціональне тестування, відповідність видів функціонального тестування функціональним вимогам;
- нефункціональне тестування, відповідність нефункціональним вимогам, види нефункціонального тестування.

Самостійна робота передбачає: 1) ознайомлення з додатковою літературою, що включає розширені відомості про класифікацію тестування і його типи, та 2) виконання практичного завдання: вибрати комерційний веб-додаток та описати сценарії його тестування відповідно до кожного з типів, які було розглянуто під час лекції. Результати роботи передбачають подальше обговорення і захист на практичному занятті.

Другий блок модулю забезпечує акцентування уваги майбутніх фахівців на методиці створення тестових сценаріїв як технічної документації, що використовується в сучасних методологіях розробки ПЗ. Лекція з теми «Техніки тест дизайну» розкриває такі питання як:

- поняття «чек-лист», техніки та переваги складання чек-листів;
- поняття «тест кейс», елементи і обов'язкові поля тест-кейса;
- особливості написання тест-кейсів у виробничому процесі розробки;
- поняття «тест дизайн», огляд технік тест-дизайну, їх переваг та умов застосування.

Для самостійної роботи студентам пропонується: 1) ознайомитись зі змістом профільної літератури, що містить розширену теоретичну викладку технік тест-дизайну і особливостей їх застосування в професійній практиці, а також 2) підготувати та захистити у межах практичного заняття набір тест-кейсів, які описують сценарії тестування функціональності комерційного додатку за умови використання оптимальних технік тест-

дизайну і обов'язкового дотримання вимог до оформлення тест-кейсів згідно з індустріальними стандартами.

Знання особливостей та умов написання тестових сценаріїв надає можливість безперешкодного вивчення теми наступного – *четвертого модуля* - «Планування і естімація». Він зорієнтований на розкриття питань попередньої оцінки процесу тестування, визначення працезатрат, створення відповідних документів.

Перший блок модулю розкриває особливості підготовки формальних документів, що описують і попередньо визначають процес тестування. Так, лекційне заняття на тему «Планування. Тест-план і тест-стратегія» розглядає в форматі відкритої дискусії наступні питання:

- причини і необхідність попереднього планування тестування;
- фактори, що впливають на планування етапу тестування;
- тест-план: структура, основні розділи, правила і особливості створення;
- тест-стратегія: структура, правила та особливості створення.

Самостійна робота традиційно передбачає перелік рекомендованої літератури, а також практичне завдання: розробку тест-плану і тест-стратегії для обраного веб-додатку. При подальшому обговоренні результатів на практичному занятті акцентується увага на відповідності структури планів міжнародним стандартам оформлення, а також аргументації прийнятих стратегічних рішень.

Наступний тематичний блок розкриває особливості планування поточних завдань у межах уже активного процесу тестування, в контексті гнучких методологій розробки. Лекційне заняття на тему «Техніки естімації» передбачає обговорення таких питань, як:

- визначення понять «планування» і «естімація» в контексті сучасних методологій розробки ПЗ;
- особливості естімації та планування завдань з тестування;
- техніки планування та естімації.

Самостійна робота студентів полягає в декомпозиції завдань курсу та проведенні їх планування у форматі групової роботи. Обговорення результатів відбувається у межах практичного завдання.

*П'ятий модуль* - «Дефекти і звітність» - складається з одного тематичного блоку і передбачає вивчення документів, артефактів та поетапних результатів процесу тестування.

Лекційний матеріал зорієнтований на розкриття таких питань, як:

- артефакти процесу тестування;
- поняття дефекту, види і особливості дефектів;
- звіт про дефект: правила оформлення і життєвий цикл;
- звіти про тестування, системи менеджменту тестових сценаріїв;
- поняття метрики, види метрик, основні метрики тестування.

Запропоноване завдання для самостійної роботи студентам полягає в наступному: а) підготувати набір тест кейсів; б) перенести їх в систему тест менеджменту, запропоновану викладачем; в) пройти тести; г) сформувати звіт про тестування зі створенням звітів як мінімум для одного функціонального та нефункціонального дефекту, а також калькуляцією і наданням метрик. Результати роботи обговорюються з групою у межах практичного заняття.

*Шостий модуль* курсу повністю присвячений проблемі оптимізації процесу тестування в межах сучасних методологій розробки, шляхом автоматизації тестових сценаріїв програмними засобами. Модуль складається з двох блоків, перший з яких спрямований на вивчення глобальних цілей, підходів і доцільності впровадження автоматизації в процес тестування на проєкті. Лекція на тему «Введення в автоматизацію тестування. Сторонні інструменти для автоматизації» зосереджена на обговоренні таких проблемних питань як:

- поняття автоматизації;
- цілі і завдання автоматизації;
- класифікація підходів до автоматизації тестування;

- переваги і недоліки використання готових інструментів для автоматизації, їх класифікація та огляд.

Самостійна робота передбачає: 1) ознайомлення з документацією по використанню одного з найпоширеніших інструментів для тестування веб-сервісів - «Postman», який є базовим прикладом інструмента для автоматизації. Він поєднує в собі клієнт для відправки і подальшого аналізу веб-запитів, інфраструктуру для зручної їх систематизації і зберігання, а також функціональність з автоматичної відправки запитів і верифікації результатів; 2) написання, за власним вибором, набору тестових сценаріїв для веб-сервісу з подальшою їх автоматизацією за допомогою використання вищезазначеного інструменту. Результати роботи обговорюються на практичному занятті.

Другий блок модуля розкриває інший підхід до автоматизації, що передбачає розробку власного рішення для автоматизації процесу тестування на основі використання актуальних об'єктно-орієнтованих мов програмування. Особливістю цього блоку є проведення попереднього опитування студентів щодо вибору об'єктно-орієнтованої мови програмування, яка складе синтаксичну основу при підготовці матеріалів лекції на тему «Архітектура фреймворка автоматизації тестування».

В процесі лекції розглядаються наступні питання:

- передумови і вигода самостійної розробки інструментів автоматизації тестування;
- поняття фреймворка для автоматизації ПЗ;
- особливості вибору мови програмування для розробки фреймворка автоматизації;
- вимоги до архітектури фреймворка автоматизації;
- структура фреймворка і його логічні складові;
- популярні бібліотеки та інструменти для реалізації фреймворка;
- залежність структури фреймворка від типу тестової програми.

В межах самостійної роботи студентам заплановано розробити своє рішення для автоматизації сценаріїв тестування веб сервісу, використаного в

завданні з попереднього блоку. Право вибору потрібної мови програмування надається студенту, проте використання допоміжних інструментів і бібліотек обмежується викладачем. Також необхідно здійснити порівняльний аналіз двох підходів до автоматизації тестування - з використанням сторонніх або самостійно розроблених технічних рішень. Аргументація рішень, висновки і результати роботи обговорюються в межах практичного заняття.

За результатами вивчення курсу здійснюється: 1) *підсумковий контроль*, який передбачає виконання і захист комплексного проєкту, а також 2) діагностика рівня теоретичних знань з курсу шляхом *співбесіди*. До складу експертної комісії, окрім викладача курсу, залучаються фахівці-практики, які будуть працювати зі студентами на наступному етапі роботи.

Виконання студентом комплексного практичного завдання, забезпечує систематизацію, узагальнення всіх знань, отриманих у процесі вивчення матеріалів авторського курсу, і полягає передбачає таке:

- вибір публічно доступного веб-додатку;
- написання тест-плану, що включає в себе тест-стратегію та відповідає міжнародним форматам оформлення;
- написання набору приймальних тестових сценаріїв, який покриває основні функціональності обраної програми;
- автоматизація написаних сценаріїв, використовуючи один з розглянутих в лекціях підходів;
- виконання автоматизованих сценаріїв і формування звіту, включаючи можливі дефекти.

Основною метою співбесіди є моніторинг рівня засвоєння студентами матеріалу курсу та обговорення результатів виконання практичного завдання. В процесі обговорення експертна комісія приділяє підвищену увагу таким аспектам як:

- знання теоретичних основ процесу тестування;
- поглиблене знання теоретичних аспектів, отриманих шляхом самостійного вивчення рекомендованої літератури;

- використання практик і підходів в процесі виконання підсумкового завдання;

- здатність студента аргументувати вибір практик, підходів і технологій в контексті планування, опису та реалізації процесу тестування обраними їм засобами.

Результати підсумкового контролю в поєднанні з балами, отриманими в процесі поточної роботи, формують підсумковий бал за весь курс, на підставі чого формується рейтинг студента, а також готується розгорнутий відгук, в якому зазначаються як позитивні досягнення, так і сфери, що потребують додаткової роботи.

Таким чином, розроблений курс «Основи тестування програмного забезпечення», що реалізується у системі неформальної освіти майбутніх інженерів-програмістів забезпечує:

- по-перше, - формування їх орієнтовної основи діяльності – спеціальних професійних знань (що складають когнітивний компонент готовності);

- по-друге, - розгортання процесу формування їх технічних і нетехнічних умінь (складових операційно-діяльнісного компоненту готовності), що має своє логічне продовження у наступних формах і методах підготовки;

- по-третє, - занурення, входження майбутніх тестувальників у ціннісний простір їх майбутньої діяльності, активізацію їх мотивації професійної діяльності та професійного розвитку.

#### **2.4. Практична підготовка майбутніх інженерів-програмістів з тестування програмного забезпечення на засадах менторства**

Здійснений аналіз фахових наукових та інформаційних джерел, а також власний професійний досвід роботи в ІТ-компанії дає змогу констатувати, що на сьогодні як на рівні освітньої системи країни в

цілому, так і конкретних закладів вищої та професійної освіти, роботодавців, є усвідомлення проблеми невідповідності/відставання змісту підготовки, яку одержують майбутні фахівці IT-сфери, та практичних навичок, компетентностей, які вимагає від них сучасний ринок праці. Це значною мірою, зумовлено недостатнім, відповідно до логіки та методології професійної підготовки, обсягом базової складової навчального процесу - практичної діяльності майбутніх фахівців в навчальних та реальних умовах, спрямованої на формування операційно-діяльнісного компоненту їх готовності. Основними шляхами вирішення цієї проблеми є активізація роботи в напрямку організації більш тісної співпраці закладів вищої освіти і підприємств та організацій, зокрема: розширення баз практики, створення наукових парків, навчальних центрів, впровадження елементів дуального навчання («Концепція підготовки фахівців за дуальною формою здобуття освіти», схвалена розпорядженням Кабінету Міністрів України від 19 вересня 2018 р. № 660-р), тощо.

На наш погляд, в умовах модернізації нормативно-методичної бази, найбільш продуктивною системою підготовки майбутніх програмістів до тестування ПЗ є саме використання потенціалу неформальної освіти, що передбачає створення умов для суміщення теоретичної підготовки в закладі вищої освіти з практичною діяльністю на базі конкретного підприємства. Слід акцентувати увагу на тому, що на сьогодні основні механізми її реалізації відпрацьовані на неформальному практичному рівні, підтвердженням чого є значна кількість студентів, які починаючи з 3 курсу паралельно з навчанням працюють на підприємствах і організаціях IT-сфери. Процесу безпосередньо професійної діяльності, як показує практика і досвід рекрутингу, передує процес адаптації та їх підготовки до роботи. Тому актуальним є питання синхронізації дидактичних завдань формальної та неформальної освіти. Результати вивчення світового і вітчизняного досвіду свідчать, що одним з найпоширенішим і ключовим

методом вирішення цього питання є менторство, яке забезпечує професійний розвиток персоналу багатьох компаній, незалежно від їх розмірів і сфери діяльності. За даними науковців, програми формального і неформального менторства на сьогодні реалізуються у більшості зарубіжних університетів та організацій. А результати проведених досліджень щодо їх доцільності та ефективності показали, що учасники цих програм отримують значний позитивний професійний досвід у порівнянні з тими, хто цього не робить (Chao Georgia T Postlethwaite B.E., Schaffer R.H., Wallack M.) [179]. Менторство також допомагає ІТ-командам підвищувати залученість й утримувати своїх найкращих фахівців, бути в курсі змін в технологіях й очікуваннях молодих співробітників.

Здійснений аналіз наукових джерел засвідчив про значний інтерес вчених до питань менторства і його організації, з одного боку, а з іншого, - про різноманітність і різноплановість підходів до розуміння сутності цього феномену.

Так, менторство на сьогодні перекладається і трактується по-різному і, відповідно, має різне значення як:

- наставник, вихователь [150, с. 791], [108, с. 337];

- модель передачі досвіду, в якій ментор, як фахівець, що відбувся в своїй сфері, виступає в якості наставника, радника, що створює умови для розвитку, зростання і підтримки менш досвідчених колег: здійснює аналіз і оцінку перспективи розвитку, надає допомогу в постановці завдань і пошуку шляхів їх вирішення, демонструє позитивний приклад в реалізації завдань, контролює їх виконання, надає емоційну і психологічну підтримку, здійснює оцінку та рекомендації для подальшої діяльності [166];

- довгострокова співпраця та навчання більш досвідченою людиною менш досвідченої, її підтримка і заохочення в напрямку розвитку особистісного потенціалу, удосконалення професійних якостей, навичок, покращення результатів діяльності [20];

- надання спеціальної педагогічної підтримки учням, чиї потреби не можуть бути задоволені в межах традиційної системи навчання [17];

- ефективний і популярний спосіб передачі професійних знань і умінь в різних галузях діяльності від більш досвідченого підприємця або спеціаліста (ментора) менш досвідченому [145];

- навчальне партнерство між двома людьми різного рівня підготовки і досвіду та потенціалу, з метою досягнення нового розуміння та особистісного зростання; створення синергії між двома людьми у навчальному союзі [198];

- особливе партнерство двох людей, засноване на діяльності, яка спрямована на передачу знань та вмінь, компетенцій від одного працівника іншому й базується на спільній цілі та чікуваннях, взаємній довірі та повазі, взаємного професійного росту [211];

- певний організований тип відносин між людьми з різним рівнем досвіду [135] тощо.

Слід зауважити, що для вітчизняного освітнього простору традиційним було і залишається використання терміну «наставництво» як форми виховання і професійної підготовки молоді на виробництві передовими, досвідченими працівниками, майстрами [150, с. 865]. Навіть сьогодні в Україні одним з основних завдань державної кадрової політики визначено впровадження системи наставництва (Стратегія державної кадрової політики на 2012-2020 роки, затверджена Наказом від 01.02.2012 р. №45/2012) [161, с. 32]. Міністерством соціальної політики України розроблені і затверджені відповідні методичні рекомендації щодо її запровадження (Наказ № 1611 від 11.10.2017 р.).

Враховуючи вищезазначене, а також недосконалість перекладу, деякі автори (Я. Бельмаз, М. Кліщ, У. Рощектаева, В. Тонгуш, В. Январев та ін.) свідомо ототожнюють поняття «менторство» з «наставництвом» і розкривають його сутність як:

- індивідуальне або колективне шефство досвідчених працівників над початківцями, або форма професійної підготовки та адаптації молодих співробітників в організації, що передбачає передачу наставником досвіду і прищеплення культури праці і корпоративних цінностей [163];

- одна з форм розвитку прикладних професійних компетенцій людини на робочому місці, коли досвідчений і кваліфікований співробітник організації передає своєму підопічному необхідні для ефективно професійної діяльності знання, вміння та навички [20];

- один з компонентів неперервного професійного розвитку, що забезпечує підтримку і допомогу молодому фахівцю [13];

- масштабне поширення, різноманітне за видами, формами і методами організації навчання, ефективно за результатами реалізації та професійним зростанням» [98] та ін.

Значна частина науковців (М. Кліщ, В. Январєв [65] та ін.) небезпідставно вважають, що саме «наставництво» є традиційним базовим феноменом, який відображає парну взаємодію двох працівників в рамках однієї організації з метою управління знаннями: передача знань, професійних навичок, культури. А інші поняття, які використовуються в сучасному менеджменті, як то: «супервізія», «менторство», «консультування», «коучинг» та ін. є різними його моделями (формами) (за М. Кліщ, В. Январєв [65]) (табл. 2.2).

Таблиця 2.2

### Моделі (форми) наставництва

Метод	Описання	Цільова група	Результати і соціальні ефекти
Класичне наставництво, менторство	Передача старшою за віком і більш досвідченою людиною своїх знань про те, як виконувати те чи інше завдання	Всі категорії співробітників; прийняті нові або переведені на посаду співробітники	Навчання співробітника, адаптація; контроль поточних результатів роботи; поліпшення комунікації; збереження і передача знань в організації
Супервізія	Співпраця двох професіоналів для критичного аналізу	Менеджери середньої ланки; прийняті нові	Відстеження роботи співробітників; відстеження прогресу;

	власної роботи	співробітники	визначення потреб у навчанні і розвитку; вирішення актуальних питань
Buddying	Підтримка колегою та/або керівником, заснована на принципі повної рівності	Всі категорії співробітників	Навчання співробітника; адаптація; оцінка ефективності змін; передача інформації між підрозділами; командування
Shadowing	Тимчасове прикріплення до керівника для включеного спостереження за особливостями і прийомами роботи	Студенти; молоді фахівці; кандидати, зацікавлені в прийомі на роботу/переведенні	Навчання співробітника; адаптація; перекваліфікація; професійна мотивація; збагачення праці

У сучасній управлінській практиці менторство зазвичай розглядається з двох сторін: 1) з позиції працівника організації, який має бажання і можливість переймати від когось, більш досвідченого фахівця, знання та досвід професійної діяльності, отримувати підтримку і допомогу в роботі (Д. Купер, Т. Міллер та ін.); та 2) з позиції ментора, як впливового, самодостатнього, досвідченого члена організації, який підтримує та всіма силами допомагає просуватися працівнику в його професійній кар'єрі (К. Мінсмойер, Дж. Томпсон та ін.). В тому чи іншому випадку менторству надають такі основні характеристики, а саме - воно:

- використовується як допоміжна форма професійного становлення та розвитку;
- спрямовується на вдосконалення професійних компетенцій та управління кар'єрою;
- сприяє вирішенню особистісних проблем, які виникають в професійному середовищі;
- забезпечує досягнення як організаційних, так і індивідуальних цілей та ін.;
- допомагає виявити творчий потенціал особистості [17; 118; 191];

На думку експертів (за К. Райорден) менторство буває трьох видів:

- допомога досвідченого колеги, коли воно обмежується простими спільними діями;

- академічне менторство, здійснюється шляхом реалізації формальних і неформальних менторських програм, спрямованих на поглиблення знання з предметів, формування в учнів впевненості, підвищення самооцінки, а також демонстрації рольової моделі для засвоєння цінностей;

- професійне менторство – дає змогу студенту отримати реальний досвід опанування певної професії або змогу зануритися в певну сферу діяльності, полегшує професійне самовизначення [17].

Як показує практика, менторство реалізується як в індивідуальних, так і групових формах [186]; як за формальними (офіційно), так і за неформальними (неофіційно) програмами [179]; як очно так і дистанційно [217] та ін.

Воно, зазвичай, відрізняється за чотирма вимірами (Vaugh & FagensonEland):

- інтенсивністю (неформальне наставництво більш інтенсивне, оскільки обидві сторони є внутрішньо мотивованими у стосунках, а сфера стосунків є необмеженою);
- видимістю (формальні стосунки більш помітні ніж неформальні, оскільки визначаються, формулюються і нормативно регулюються);
- фокусом (більшість офіційних програм зорієнтовані та обмежені організаційними цілями, відповідно основна увага приділяється розвитку співробітника, а не розвитку особистості, як в неформальних умовах);
- тривалістю (офіційні програми мають терміни початку та закінчення взаємодії, на відміну від них, неофіційне менторство є необмеженим за тривалістю, частотою або контентом зустрічей [179, с. 317].

Але не залежно від свого виду і форми, менторство реалізується за такими основними етапами як-то: 1) аналіз та оцінка перспектив розвитку

підопічного; 2) постановка завдань та визначення шляхів/програм їх досягнення; 3) розробка практичних кейсів для їх досягнення; 4) психологічна допомога; 5) знайомство з різними контрагентами; 6) контроль практичної діяльності; 7) оцінка діяльності та рекомендації для подальшої роботи [145].

На думку багатьох науковців (J. Freeman, Kirsten M. Poulsen, М. Кліщ, В. Радченко, В. Январєв О. та ін.), ключову роль в ефективності менторства як методу навчання та розвитку, відіграє особистість ментора, який повинен мати не тільки необхідні знання, вміння та навички з особливостей професійної діяльності, володіти механізмами їх передачі до підопічного, а й певний набір особистісних якостей. Це зумовлено тим, що ментор для підопічного одночасно має бути в трьох ролях: вчителя, який забезпечує формування необхідних професійних знань, вмінь та навичок; керівника в стилі «коучинг», що забезпечує зворотній зв'язок в оцінці ефективності професійної діяльності; власне ментора, який забезпечує умови для професійного становлення та розвитку в організації, формування програми кар'єрного зростання молодого фахівця [65, с. 7].

Зокрема, фахівці наголошують на таких основних якостях ментора як: терпіння, почуття такту, володіння прийомами комунікації, відповідальність, цілеспрямованість, чуйність і висока самоорганізація [126]; емоційна зрілість, толерантність, вміння слухати, ділитися, піклуватися, заохочувати та приймати помилки підопічного [191].

Деякі науковці [191] додатково акцентують увагу і на таких особистісних аспектах ментора як:

- вік - найкращі ментори - це ті, хто знаходяться на піку своєї кар'єри, але не пенсіонери, які відпочивають на досягнутому;
- стать - хлопці частіше шукають як наставників чоловіків, а дівчата - жіночі моделі професійної реалізації;
- стиль викладання в ідеалі має бути сумісним з стилем навчання.

До того ж, як показує практика менторства в ІТ-сфері, наставник може бути не один, а декілька. Наприклад, по-перше, - з питань технологій, який надає консультації і допомогу в усвідомленні та використанні тих чи інших їх типів та видів, та по-друге, - той, який може бути експертом з питань організації та управління та ін. [187].

Таким чином, враховуючи вищезазначене, в своїй роботі під менторством ми розуміємо *особливу технологію професійної підготовки майбутніх інженерів-програмістів в системі неформальної освіти, що забезпечує їх неперервний професійний розвиток в цілому та формування готовності до тестування ПЗ зокрема, реалізується в навчально-виробничих умовах ІТ-компанії та здійснюється під керівництвом її досвідчених співробітників.*

Відповідно до презентованої у п. 2.2. авторської структурно-функціональної моделі, практична підготовка майбутніх інженерів-програмістів до тестування ПЗ на засадах менторства реалізується згідно розробленої програми (Додаток Г) протягом п'яти місяців (6 семестр) у дві фази рис. 2.4.

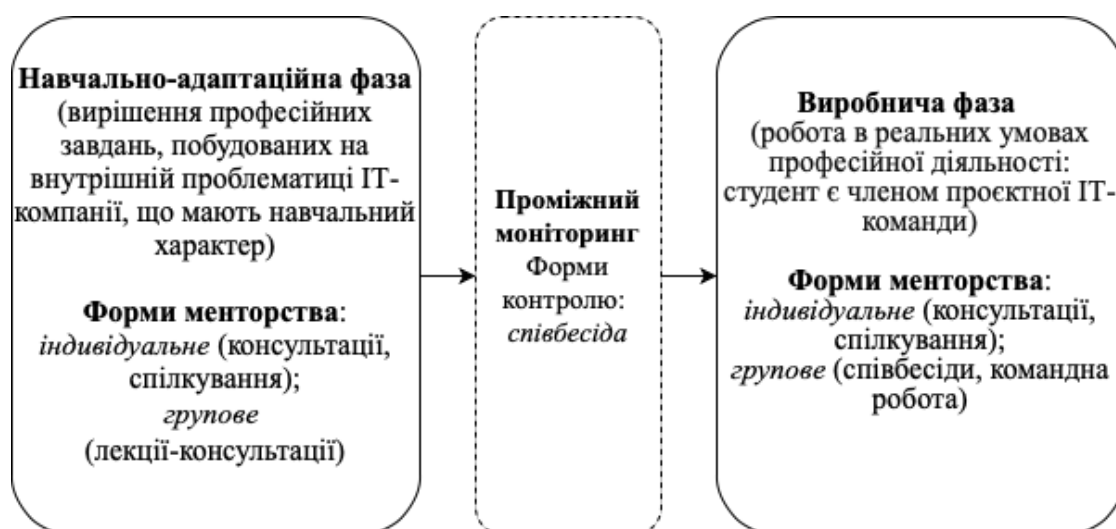


Рис. 2.4. Фази практичної підготовки студентів в ІТ-компанії на засадах менторства

Так, *перша - навчально-адаптаційна фаза* - передбачає практичну діяльність студентів протягом двох місяців, змістом якої є вирішення професійних завдань, побудованих на внутрішній проблематиці ІТ-компанії, які більшою мірою мають навчальний характер.

Проблематика зазначеної діяльності в основному охоплює такі сфери як:

- Тестування користувацького інтерфейсу веб-додатків;
- Тестування сервісного шару веб-додатків;
- Автоматизація тестування веб-додатків;
- Тестування мобільних додатків.

Відтак, студенти отримують такі завдання, послідовне виконання яких стає змістом їх практичної діяльності.

*Завдання 1.* Адміністрування і робота з системами управління проєктами. Метою завдання є формування у студентів практичних навичок зі створення інфраструктури, адміністрування і роботи з системами управління проєктами та документообігом, які активно використовуються в більшості сучасних комерційних проєктів. У межах завдання студенту пропонується створити повну інфраструктуру управління проєктом «Менторство» в системі Atlassian JIRA, що включає в себе безпосередній проєкт, команду розробки (студента та його ментора), документацію проєкту, список завдань (відповідно до програми менторства), планування ітерацій (відповідно до розкладу практичних занять) і дошки відстеження активних завдань. В подальшому ця інфраструктура буде використовуватися як засіб формальної комунікації ментора і студента, а також відстеження прогресу виконуваних завдань.

*Завдання 2.* Тестування сервісного шару веб-додатків. Метою завдання є формування практичних навичок з тестування сервісного рівня веб-додатків з урахуванням їх технічних особливостей і механізмів взаємодії з зовнішніми системами і передбачає покриття публічного веб-сервісу прогнозу погоди Weatherbit.io тестовими сценаріями, з подальшим

їх проходженням засобами веб-клієнта Postman і формуванням звіту. Увесь процес роботи, включаючи створення і планування робочого завдання, складання сценаріїв, їх проходження і звітності, повинен бути відображений в інфраструктурі Jira, створеній й завданні 1.

*Завдання 3.* Тестування користувацького інтерфейсу веб-додатків. Метою завдання є формування практичних навичок тестування користувальницького інтерфейсу веб-додатків з урахуванням його технічних і експлуатаційних особливостей. Сутність завдання полягає в створенні чек-листа і, надалі, оформленні повного набору тестових сценаріїв для функціональності «кошик» інтернет-магазину rozetka.ua з подальшим їх проходженням і формуванням звіту. Увесь процес роботи над завданням, включаючи створення і планування робочого завдання, складання сценаріїв, їх проходження і звітності, повинен бути відображений в інфраструктурі Jira, створеній у завданні 1.

*Завдання 4.* Тестування мобільних додатків, метою якого є формування практичних навичок з тестування мобільних додатків з урахуванням їх технічних особливостей та відмінностей в цільових платформах. У межах цього завдання студентам пропонується підготувати три набори тестових сценаріїв для функціональності «кошик» мобільного інтернет-магазину Rozetka з урахуванням особливостей кожної з трьох платформ: мобільний додаток Rozetka для платформи Android, аналогічний додаток для платформи iOS і мобільна версія веб-сайту, доступна для браузера цільового пристрою. Після складання описаних вище наборів тестових сценаріїв потрібно пройти кожен з наборів за рахунок емуляції або використання відповідних пристроїв з подальшим формуванням звітів. Увесь процес роботи, включаючи створення і планування кожного з трьох робочих завдань, складання сценаріїв, їх проходження і звітності, повинен бути відображений в інфраструктурі Jira, створеній у завданні 1.

*Завдання 5.* Автоматизація тестування сервісного шару веб-додатків. Метою завдання є формування практичних навичок створення

інфраструктури для автоматизації сервісного шару веб-додатків завдяки використанню самостійно розробленого фреймворка з використанням об'єктно-орієнтованого програмування. В межах цього завдання студенту потрібно створити інфраструктуру для автоматизації веб-сервісу прогнозу погоди Weatherbit.io завдяки використанню об'єктно-орієнтованої мови Java (версії не нижче 1.8), бібліотеки для тестування JUnit 5 і веб-клієнта ОКHttp з подальшою автоматизацією тестових сценаріїв, написаних в межах завдання 2. Увесь процес роботи, включаючи створення і планування робочого завдання, розробку автоматизованих тестів і звітність, має бути відображений в інфраструктурі Jira, створеної в завданні 1. Відповідність автоматизованих сценаріїв мануальним, розробленим в завданні 2, має бути відображена зв'язком завдань в інфраструктурі Jira.

*Завдання 6.* Автоматизація тестування інтерфейсу веб-додатків. Метою завдання є формування практичних навичок створення інфраструктури для автоматизації призначеного для користувача інтерфейсу веб-додатків завдяки використанню самостійно розробленого фреймворка із застосуванням об'єктно-орієнтованого програмування. У межах завдання студенту потрібно створити інфраструктуру для автоматизації функціональності «кошик» інтернет-магазину rozetka.ua завдяки використанню об'єктно-орієнтованої мови Java (версії не нижче 1.8), бібліотеки для тестування JUnit 5 та інструменту для роботи з браузером Selenium WebDriver з подальшою автоматизацією тестових сценаріїв, написаних у завданні 3. Увесь процес роботи, включаючи створення і планування робочого завдання, розробку автоматизованих тестів і звітність, має бути відображений в інфраструктурі Jira, створеній в завданні 1. Відповідність автоматизованих сценаріїв мануальним, розробленим в завданні 3, має бути відображена зв'язком завдань в інфраструктурі Jira.

*Завдання 7.* Тестування продуктивності веб-додатків. Метою даного завдання є формування практичних навичок тестування продуктивності веб-додатків. Студентам пропонується створити і розгорнути на локальній машині примітивний веб-додаток засобами об'єктно-орієнтованої мови програмування Java та інструменту SpringBoot, яке складається з одного REST сервісу, а також розробити сценарії та плани навантаження, на підставі чого написати алгоритми для інструменту тестування продуктивності Apache JMeter з подальшим їх запуском і формуванням звіту. Увесь процес роботи, включаючи створення і планування робочого завдання, складання сценаріїв навантаження, їх запуску і формування звітності, повинен бути відображений в інфраструктурі Jira, створеній у завданні 1.

*Завдання 8.* Адміністрування і робота з системами безперервної інтеграції та безперервного розгортання. Метою завдання є формування практичних навичок зі створення, адміністрування та роботи з системами безперервної інтеграції і безперервного розгортання. У межах завдання студенту пропонується розгорнути систему Jenkins на локальному комп'ютері, провести її попередню настройку та оптимізацію. Для тестового додатка на технології SpringBoot, розробленого у завданні 7, пропонується написати один автоматизований тест засобами інструменту JUnit, після чого створити інфраструктуру його розгортки засобами інструменту Jenkins, з попередньою установкою та налаштуванням останнього. Увесь процес роботи, включаючи створення і планування робочого завдання, розробки автоматичних тестів та розгортання інструмента Jenkins, повинен бути відображений в інфраструктурі Jira, створеній у завданні 1.

Цілком природно, що на цьому етапі менторство вже має професійний характер і спрямоване на допомогу студенту зануритися в певну сферу ІТ-діяльності – тестування ПЗ в умовах конкретного

підприємства, що має свої характерні правила, норми поведінки, форми і методи роботи, механізми та процедури взаємодії та ін.

Враховуючи зарубіжний досвід професійної підготовки майбутніх інженерів-програмістів, на першому етапі менторства студенти мають не одного, а декілька менторів, а самий процес взаємодії (менторство) має як індивідуальний, так і груповий характер.

*Індивідуальне менторство* з боку ІТ-компанії забезпечує Ментор, який закріплюється за кожним студентом. Воно розпочинається з першого дня практичної підготовки – роботи майбутнього фахівця в ІТ-компанії і передбачає спільну роботу протягом всього її періоду за такими основними напрямками як:

- професійний розвиток, спрямований на: визначення перспектив професійного удосконалення, аналіз та оцінку рівня професійної готовності майбутнього фахівця до роботи, постановку завдань та розробку шляхів/ практичних кейсів для їх досягнення, коригування (спрямування згідно до потреб) змісту освіти і самоосвіти та ін.;

- адаптацію студента до умов роботи в ІТ-компанії: знайомство з організацією, її структурою, співробітниками, правилами та нормами поведінки, корпоративними цінностями; психологічна підтримка у формі бесіди як в очному форматі, так і дистанційно;

- моніторинг ефективності діяльності в професійній сфері: контроль якості результатів практичної діяльності протягом першого етапу; підготовка відгуку на роботу; оцінка в цілому діяльності за перший етап та рекомендації (завдання) для подальшої роботи.

Індивідуальне менторство реалізується системно, як в формальних, так і неформальних умовах за нестандартизованим, гнучким графіком. Частота спілкування, взаємодії ментора та студента більшою мірою залежить від ініціативи останнього та відбувається за принципом постановки певної проблеми, що виникла в процесі професійної діяльності

в ІТ-компанії, яка є актуальною і потребує нагального вирішення. Робота над проблематикою відбувається відповідно до загальної схеми рефлексивного процесу (рис. 2.5) (за О.І. Гурою [28]), що покладена в основу «Рефлексивного щоденника» (Додаток Д), який заповнює студент.

Перший крок – це формулювання студентом мети діяльності, основних завдань, які він мав вирішити як в цілому в процесі практичної діяльності на в ІТ-компанії, так і відповідно до конкретного завдання. І головне – шляхів їх виконання (методології, засобів, методів та ін.). Основний акцент робиться на відповідності й узгодженості всіх змістових та процесуальних компонентів діяльності, зокрема: мети та результату, змісту та дій. Як показує практика, значна кількість ускладнень з’являється саме на цьому етапі, оскільки підготовка в напряму усвідомлення структури власної діяльності (мотиву, мети, завдань, форм методів, засобів, результатів та ін.), її проєктування та організації у закладах вищої освіти в більшій мірі не здійснюється.

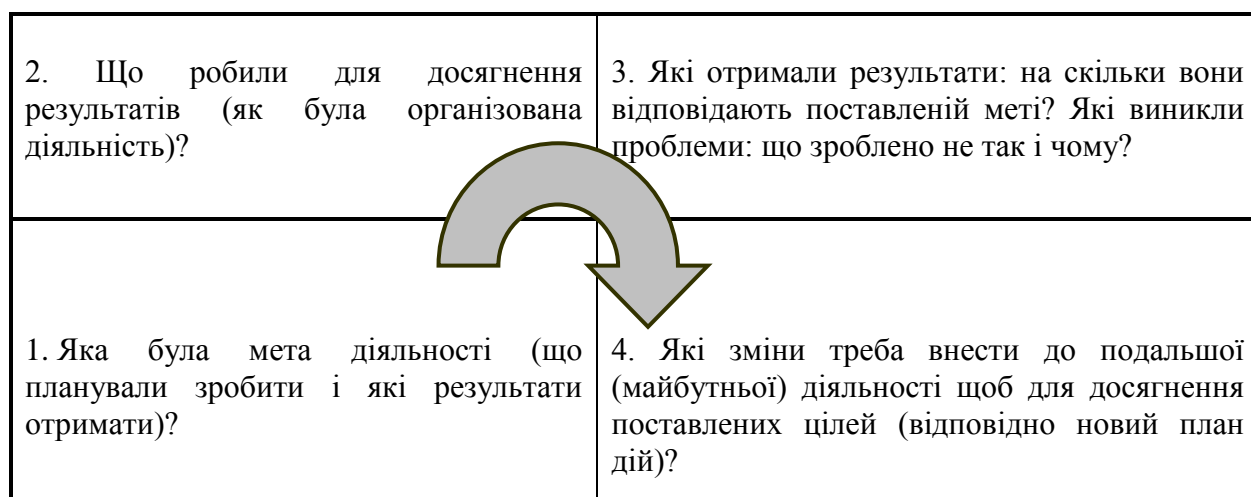


Рис. 2.5. Загальна схема рефлексивного процесу

Другий крок – це проговорення і фіксація, а відповідно й усвідомлення тих дій, які виконував студент в процесі практичної діяльності. Акцент робиться на їх відповідності як прийнятому, на

попередньому етапі плану дій, так і їх узгодженості між собою, а також із використаними засобами, методами.

Третій крок передбачає визначення, усвідомлення проблемних зон – встановлення: що зроблено не так і чому, які помилки допущені в процесі як планування, так і безпосередньо реалізації.

Четвертий крок – передбачає внесення коректив в подальшу діяльність з метою врахування помилок і недопущення їх в майбутньому.

В той же час на цьому етапі практичної підготовки Ментор, зі свого боку, має: забезпечувати постійну комунікацію зі студентом; мотивувати і стимулювати його до діяльності з тестування ПЗ та професійного саморозвитку; дотримуватись програми підготовки; допомагати студентові знайти ефективний інструментарій, методи і засоби діяльності; накопичувати і зберігати результати спільної роботи; психологічно підтримувати, і головне, - допомагати в рефлексії процесу навчання, в аналізі успіхів і невдач [135, с. 14].

Відтак, взаємодія на основі рефлексивного аналізу забезпечує, по-перше, - усвідомлення студентами причин і факторів, що зумовлюють ускладнення в професійній діяльності; по-друге, - корегування траєкторії їх особистісного і професійного розвитку; по-третє, - отримання певного досвіду в плануванні та аналізі власної діяльності; по-четверте, - підґрунтя для аналізу рівня готовності студента до професійної діяльності. В подальшому, результати цієї роботи враховуються як у моніторингу ефективності підготовки, так і для коригування її змісту. Враховуючи, те, що розвиток кожної особистості має індивідуальний характер, основними завданнями менторства й є забезпечення зони «найближчого розвитку» студента на індивідуальному та груповому рівнях у контексті продуктивної реалізації загальної програми підготовки - формування складових готовності майбутніх інженерів-програмістів до тестування ПЗ.

Отже, на цьому етапі практичної підготовки ментором здійснюється:

- моніторинг та підтримка та/або розвиток мотивації до професійної діяльності в цілому та тестування ПЗ зокрема, так і конкретно до навчання в системі неформальної освіти (наприклад, використання фінансового заохочення - так званої «стипендії» від ІТ-компанії, що на сьогодні є розповсюдженим в ІТ-сфері; залучення до неформальних заходів: корпоративних свят, спортивних змагань тощо);

- моніторинг рівня розвитку професійних знань та вмінь і коригування змісту діяльності (введення додаткових завдань та/або додаткових консультацій за тією та іншою тематикою; демонстрація позитивного досвіду та ін.);

- моніторинг рівня соціальної адаптації в професійному середовищі, підтримка розвитку нетехнічних умінь.

Враховуючи навчально-професійну спрямованість першого етапу практичної підготовки майбутніх інженерів-програмістів до тестування ПЗ, її програма передбачає і реалізацію *групового менторства*. Його забезпечують Куратори проєктних груп, які є визнаними фахівцями з проблематики тієї чи іншої професійної сфери, за якою сформовано перелік завдань, що виконують майбутні фахівці з тестування на першому етапі навчально-практичної діяльності.

Метою групового менторства є надання поточних консультацій з професійної тематики, обговорення стратегій виконання поставлених завдань. Вони відбуваються у формальних умовах з періодичністю не менше одного разу на тиждень. Формою проведення є лекція-консультація, яка на думку науковців є найбільш продуктивною при вивченні тематики з чітко визначеною прикладною (практичною) спрямованістю [68, с. 70]. Мета лекції-консультації полягає у коригуванні (доповненні і уточненні) навчального матеріалу, аналізі проблемних питань, які виникають під час самостійного навчання. Вона може відбуватись за різними сценаріями. Зокрема: виклад матеріалу – відповідь на питання студентів; запитання – відповіді - дискусія; відповіді на запитання (які підготовлені до заняття) –

відповіді на додаткові питання – обмін думками тощо. Але в тому чи іншому випадку не менше 50% часу має приділятися для відповідей на запитання студентів, що забезпечує в певній мірі індивідуалізацію процесу навчання з урахуванням розуміння матеріалу кожним студентом [94].

У контексті нашої роботи, лекція-консультація передбачає наступний сценарій. В першій частині заняття робиться невеликий за обсягом і часом узагальнений огляд матеріалу з проблематики, за якою працюють студенти, у тому числі акцент робиться на сформульованих завданнях і на тих результатах, які мають бути отримані; аналізуються труднощі, які можуть виникати в процесі роботи. Друга частина враховує ті професійні ускладнення, які ментор (ментори) фіксує на рівні індивідуальної роботи кожного і зорієнтована на висвітлення можливих шляхів їх вирішення. Третя частина роботи передбачає відповіді на питання студентів. Завершується заняття загальним обговоренням за схемою рефлексії.

Загальна тематика та початковий зміст занять формується враховуючи, з одного боку, основні завдання, які будуть вирішувати студенти в процесі практичної роботи, з іншого, - показники рівня засвоєння студентами матеріалу за результатами вивчення авторського курсу «Основи тестування ПЗ», презентованого на попередніх етапах нашої роботи (результати проміжного контролю).

В цілому лекція-консультація забезпечує умови для поглибленого вивчення теоретичних та практичних питань визначеної тематики, усвідомлення змісту навчального матеріалу, напрацювання практичних рекомендацій до його використання, підвищення рівня мотивації до професійної діяльності та ефективності самостійної роботи студентів.

Однією із складових практичної підготовки студентів, логічним завершенням першого етапу роботи є проміжний моніторинг її якості, який проходить у формі співбесіди, основними завданнями якої є:

- по-перше, - визначення теоретичного та практичного рівня підготовки студентів до роботи з тестування ПЗ;

- по-друге, - визначення й актуалізація як на рівні конкретного студента, так і ментора сфер подальшого особистісного та професійного розвитку;

- по-третє, - розподіл і закріплення студентів за конкретними командами і реальними проєктами, що реалізує компанія на ринку ІТ послуг;

- по-четверте, - присвоєння першого ІТ-титулу «Junior» і визначення розміру стипендії, яку буде отримувати студент в процесі подальшої діяльності.

До складу комісії, що проводить співбесіду входять:

- Ментор, який надає членам комісії загальну характеристику (щодо організації та реалізації практичної діяльності студента, рівня мотивації до роботи та саморозвитку, результати адаптації до умов діяльності в організації та ін.);

- Куратори групи, які забезпечували змістовну підготовку відповідно сфер діяльності; лідери («Team Leads») робочих проєктних команд, за якими планується розподіл і закріплення студентів.

Отже, перша фаза практичної підготовки майбутніх інженерів-програмістів до тестування ПЗ завершується отриманням даних щодо їх професійної готовності до діяльності в умовах конкретної ІТ-компанії – результатів проміжного моніторингу, що, у більшій мірі, визначають зміст наступного етапу їх практичної підготовки.

*Друга фаза* практичної підготовки студентів на засадах менторства – виробнича. Вона реалізується згідно розробленої програми в реальних умовах професійної діяльності: студент стає членом проєктної команди, яка забезпечує реалізацію актуального виробничого процесу.

На цьому етапі роботи передбачено менторство в індивідуальній формі, але в двох варіаціях: 1) продовження індивідуальних консультацій

закріпленим за студентом Ментором у вищеописаний спосіб; 2) проектно-спрямоване професійне консультування (консультації за проблематикою проекту), що здійснює лідер команди.

Процес менторства, організований в зв'язку з вступом в команду нового співробітника, називається «onboarding» (від англійського «on board» - сходження на борт корабля) і проводиться незалежно від проектної позиції новачка та його кваліфікаційного рівня. Основною метою «онбордінга» є оптимізація і структуризація процесу інтеграції нового члена команди в особливості організації діяльності на конкретному проекті, що включає такі предметні питання як:

- предметна сфера і бізнес-галузь;
- юридичні особливості роботи на проекті;
- специфічні технології або технічні і архітектурні рішення;
- особливості побудови процесу розробки, роботи з документацією, звітності;
- особливості взаємодії з представниками замовника і третіми особами.

Зазначений процес складається з наступних етапів:

1) визначення умов, тривалості і програми менторства, а також відповідальних за процес людей. На даному етапі лідер команди проводить оцінку наявних людських і технічних ресурсів, після чого складає програму менторства, що передбачає компроміс між повнотою викладеного матеріалу і впливом трудовитрат на продуктивність ментора в контексті його прямих проектних обов'язків;

2) входження в проект - на даному етапі відбувається ознайомлення ментором новачка з інформацією про особливості проекту в формі бесіди, лекції або складеної документації, що дозволяє новачку отримати структуровані знання про проект та його процеси;

3) робота під контролем ментора. Після ознайомлення зі структурою та особливостями проекту, новачок починає безпосередню роботу в

проектній команді і виконання виробничих завдань, при цьому постійно консультуючись з ментором щодо кожного з етапів їх виконання та проходячи перевірку результатів роботи перед безпосередньою їх демонстрацією замовнику або іншим членам команди.

Як було нами зазначено вище, значну роль в ефективності практичної підготовки майбутніх інженерів-програмістів до тестування ПЗ відіграє особистість Ментора. Тому одним із суттєвих питань стає процедура його відбору.

Сам по собі процес менторства (менторингу) може відрізнятися в ІТ-компаніях через відмінності в їх організаційній культурі, політиці рекрутингу та спеціалізації. Проте, з огляду на поширення практики внутрішньої підготовки персоналу та стажувань, в переважній більшості великих сучасних ІТ-компаній існують окремі відділи менторингу, основна мета діяльності яких – сприяти процесу передачі знань всередині організації, надавати співробітникам можливості для розширення світогляду, розвитку необхідних професійних умінь та компетенцій, які сприяють успішному професійному розвитку, кар'єрному зростанню. У цьому контексті вкрай актуальним постає питання пошуку і відбору менторів, що є ключовим для якості майбутньої підготовки молодих фахівців. Незважаючи на відмінність підходів і процесів, вимоги до ментора в переважній більшості випадків є загальними. Серед них:

- 1) високий рівень кваліфікації. Це фактично об'єктивний показник досвідченості, основний критерій готовності фахівця до ролі ментора. Кваліфікаційний рівень «Senior» передбачає наявність значного досвіду розробки ПЗ в цілому та в конкретній сфері діяльності зокрема, а також ґрунтовні знання предметної сфери, високий рівень комунікативних умінь, що й робить його оптимальною кандидатурою на статус ментора. Більш того, на сьогодні поширеною практикою в сучасних ІТ-компаніях є визначення досвіду менторства обов'язковим критерієм для отримання вищого кваліфікаційного рівня. Це, в свою чергу, стимулює досвідчених

кандидатів з кваліфікацією «Middle» також брати участь у підготовці молодих колег, тим самим створюючи цілий ланцюжок та інфраструктуру менторингу;

2) релевантний професійний досвід в цільових технічних сферах. Менторинг вимагає наявності високого рівня технічних умінь і значного професійного досвіду у конкретній ІТ-компанії, що дозволяє ментору не тільки спрямувати молодого фахівця у вирішенні його професійних завдань, а й мати можливість надати аналогії, альтернативні шляхи і порівняльну характеристику інших можливих підходів у вирішенні проблеми;

3) високий рівень розвитку нетехнічних умінь. Менторство як процес підготовки не тільки в технічному, а й в більш широкому професійному контексті, вимагає наявності у Ментора високого рівня розвитку нетехнічних умінь, релевантних для його проєктної позиції. Це дозволяє йому консультувати свого молодого колегу в тому числі і в питаннях, що стосуються комунікації з членами команди, презентації своєї роботи, планування завдань і тайм-менеджменту. Більш того, ментор повинен мати високий рівень розвитку регулятивних та комунікативних умінь, бути здатним налагодити контакт зі своїм молодим колегою (субордінантом), підтримувати його, створювати позитивний психологічний клімат;

4) проходження тематичних курсів і тренінгів. Навіть маючи значний досвід, а також розвинені технічні та нетехнічні вміння, фахівцям ІТ-індустрії може не вистачати психолого-педагогічних та управлінських знань для побудови оптимального й ефективного процесу підготовки молодих фахівців. Для вирішення цієї проблеми в більшості сучасних великих ІТ-компаній проводяться спеціалізовані курси і тренінги, що охоплюють психолого-педагогічні, організаційні та соціальні аспекти менторства.

Таким чином, підставами для визначення ментора серед співробітників ІТ-компанії є такі: відповідність зазначеним критеріям,

проходження інтерв'ю, спрямованого на визначення рівня мотивації до професійної діяльності, підтвердження релевантності технічних та нетехнічних умінь.

## **2.5. Розвиток нетехнічних умінь (soft skills) майбутніх інженерів-програмістів до тестування програмного забезпечення**

Здійснений у попередньому розділі аналіз особливостей професійної діяльності інженерів-програмістів з тестування ПЗ, визначена структура професійної готовності майбутніх фахівців до її успішної реалізації, а також вивчений досвід існуючої професійної підготовки дають підстави стверджувати, що цілеспрямований розвиток нетехнічних умінь є одним з найважливіших напрямів неформальної ІТ-освіти.

Говорячи про найбільш затребувані нетехнічні вміння фахівців з тестування ПЗ, сучасні роботодавці, практики та науковці досягли єдності у таких найбільш значущих з них як: комунікативність як вміння чітко формулювати думки, намагатися бути ввічливим і уникати потенційно небезпечних та деструктивних тем; відповідальність - повний контроль успіху виконуваного завдання, в тому числі не тільки своєї частини, а й фінального командного результату; вміння працювати у команді та вміння вирішувати складні полісистемні проблеми. Адже саме у них відображається основний зміст професійної діяльності тестувальника ПЗ – знаходити дефекти та презентувати їх членам команди для подальшого усунення.

У нормативних документах країн-членів Європейського Союзу (The European Qualifications Framework, European Skills, Competences, Qualifications and Occupations, 2019; ), а також США (IFTF, AI Forces Shaping Work and Learning in 2030, 2018), такі нетехнічні, непрофесійні (non-professional) вміння отримали назву «essential skills» або «soft skills».

Зазначені документи дають змогу ввести єдине розуміння та уніфіковану класифікацію soft skills для світового економічного простору.

На сьогодні нетехнічні, непрофесійні уміння (soft skills) визначаються комплексом неспеціалізованих, надпрофесійних характеристик фахівця, які відповідають за успішність його професійної діяльності, її високу продуктивність і, на відміну від спеціалізованих, технічних (hard skills) умінь та навичок, не пов'язані з конкретною сферою застосування.

Так, якщо hard skills – це вузькопрофесійні технічні уміння та навички, що пов'язані з діяльністю фахівців у сфері формалізованих технологій, є стійкими, вимірними, ототожнюються з конкретними завданнями [40], то вони входять в перелік вимог, що містяться в посадових інструкціях. То soft skills є такими, що їх важко відстежити, продіагностувати, особливо формалізованими в ІТ-сфері способами.

До категорії «soft skills» дослідники включають:

1) окремі якості та атрибути особистості Cobb, 2015; Goleman, 2000; Т. Yarkova, I. Cherkasova [222]), такі як самоорганізованість, саморегуляція, вміння публічно виступати;

2) уміння та навички міжособистісного спілкування (L. Raitskaya, E. Tikhonova [210]) або навички групової роботи, співробітництва, командної роботи в проєктному офісі (G. Schraw, R. Dennison [212]);

3) уміння та навички, зорієнтовані на людські стосунки, а також характеристики, які складають емоційний інтелект, здатність особистості вирішувати поведінкові та когнітивні завдання (Peterson, Van Fleet, 2004);

4) когнітивні та методологічні уміння і навички (Cinque, 2016), які відповідають за успішність вирішення проблем та виявляються у: складних формах мислення, здатності оцінювати та використовувати знання та інформацію (Matteson, 2016); критичності мислення (IFTF, AI Forces Shaping Work and Learning in 2030, 2018); здатності творчо вирішувати складні проблемні завдання; вмінні приймати рішення в ситуаціях нестачі

часу (Yarkova, Cherkasova, 2016; IFTF, AI Forces Shaping Work and Learning in 2030, 2018).

Визначаючи технічні та нетехнічні вміння фахівців ІТ-сфери доцільно визначити такі критерії їх порівняння - їх: загальне підґрунтя; орієнтовна основа; терміни критичного значення для професійної діяльності; темпи розвитку; кількість ресурсовитрат для досягнення високого рівня розвитку; зворотність. Відтак, якщо:

1) технічні вміння: ґрунтуються на професійному досвіді, професійних (технічних) знаннях; критичні для професійної діяльності у короткотерміновій перспективі; можуть швидко розвиватися з меншими зусиллями (при дотриманні базових критеріїв – мотивації і здатності до навчання) та гарантованим результатом; практично не мають зворотного напрямку розвитку (можуть лише частково не відповідати терміновим, нагальним змінам сфери діяльності, їх руйнування, деструкція відбувається досить повільно), то

2) нетехнічні вміння: базуються на особистісних цінностях та якостях, зафіксованих у дитинстві моделях поведінки; критичні для професійної діяльності в довгостроковій перспективі; їх розвиток відбувається повільно зі значними психологічними та організаційно-управлінськими зусиллями, а досягнення високого рівня не є гарантованим внаслідок складності, глибинності особистісних процесів; можуть значно руйнуватися в наслідок специфічних психологічних умов діяльності.

Важливо акцентувати увагу на тому, що у процесі професійного розвитку фахівців ІТ-сфери від рівня «Junior» до рівня «Team Leader» вага нетехнічних умінь у цілісній професійній компетентності значно зростає. Як свідчать результати досліджень [40], для успішної діяльності інженера-програміста рівня «Junior» їх співвідношення дорівнює 4:1, тобто значення технічних умінь в чотири рази більше значення нетехнічних умінь; для «Middle specialist» - 3:2, для «Senior specialists» - 2:2, то для фахівця рівня «Team Leader» вже 1:4.

При чому, результати досліджень ваги різних нетехнічних умінь у професійній компетентності та професійній кар'єрі IT-фахівців засвідчують також і про наявність її динаміки. Так, для успішної діяльності інженера-програміста рівня «Junior» передусім значення мають нетехнічні вміння так званої групи особистісної ефективності (почуття відповідальності, аналітичність мислення, креативність, стресостійкість, вміння презентувати свою роботу та ін.), для «Middle specialist» та «Senior specialists» - комунікативні та регулятивні вміння (вміння керувати часом, керувати власними емоціями, поведінкою; вміння працювати у команді, здійснювати ефективну комунікацію, розв'язувати конфлікти, вміння проведення переговорів, переконувати), для «Team Leader» - комунікативні (лідерські) (вміння: сформувати та згуртувати команду, сформувати систему комунікації в команді, мотивувати учасників команди та ін.) та так звані стратегічні або метакогнітивні вміння (вміння: формулювати проблему, планувати, прийняти рішення, працювати в умовах ризику, здійснювати рефлексивний аналіз власної діяльності, діяльності команди та ін.).

Отже, виходячи із структурної організації професійної готовності майбутніх інженерів-програмістів до тестування ПЗ (див. рис. 1.5) нетехнічні вміння диференційовані нами на 4 групи за ознакою типу діяльності, яку вони забезпечують відповідно до існуючих вимог: 1) *мисленнєві вміння*: аналітичність, оперативність та гнучкість мислення, нестандартно мислити, самостійність мислення; 2) *комунікативні вміння*: вміння працювати у команді, підтримувати позитивну атмосферу, домовлятися, комунікувати з членами команди та з замовником, обґрунтовувати і відстоювати свою позицію, вміння індивідуального підходу, неконфліктність; 3) *регулятивні вміння*: самоконтролю власної діяльності, самоконтролю власних емоцій, активізації працездатності, вміння керувати часом, бути зосередженим та уважним; 4) *метакогнітивні вміння* (вміння проблематизації, проєктувальні вміння, планування,

програмування, прийняття рішення в умовах невизначеності), включаючи рефлексивні вміння (аналізувати, оцінювати власну діяльність, знаходити помилки у неї та коригувати власну мисленнєву діяльність).

Останню групу нетехнічних умінь складають саме метакогнітивні вміння або метакомпетентності - здатність працювати над власними компетентностями, коригувати та переносити їх з однієї сфери діяльності в іншу, з неформального на формальне навчання (Cinque, 2016). Як зазначають (Haselberger, Oberhuemer, 2016), вони є динамічним поєднанням когнітивних та метакогнітивних умінь, міжособистісних, інтелектуальних та практичних навичок, що є системоутворювальними, метапредметними.

Метакогнітивні вміння фахівця є засвоєними способами метакогнітивної регуляції його професійної діяльності та, ґрунтуючись на положеннях діяльнісного та метакогнітивного підходу (А. Карпов [59], М. Холодна [165] та ін.), містять:

1) вміння з цілепокладання (розчленувати проблемну ситуацію на відоме та невідоме, формулювати мету, переводити мету у завдання, об'єктивізувати результат, визначити критерії та показники результату);

2) вміння антиципації (передбачати варіанти результату проблемної ситуації, способи вирішення проблеми, а також варіанти розвитку умов, визначати ймовірні ризики, створювати мисленнєві моделі);

3) вміння з прийняття рішення (формулювати гіпотези, альтернативи, обирати метод прийняття рішення, аналізувати тип та види прийняття рішення, працювати з помилками та ефектами прийняття рішення);

4) вміння планувати (складати план діяльності, співвідносити план діяльності з метою та моделлю);

5) програмувальні вміння (складати програму діяльності, використовувати методи управління часом);

б) рефлексивні вміння або вміння метакогнітивного самоконтролю (аналізувати результат діяльності, застосовувати механізми рефлексії, визначати шляхи коригування діяльності, сповільнювати та зупиняти діяльність, здійснювати цілісний рефлексивний аналіз власної діяльності).

Серед компетентностей фахівців сфери ІТ, які відносяться до метакогнітивної групи нетехнічних умінь найбільш важливими вважаються:

- вміння працювати зі складними проблемами, в умовах невизначеності і двозначності, уміння планувати свою діяльність та корегувати її хід [174];

- уміння бачити задачу одночасно на різних рівнях деталізації, уміння сформулювати вимоги й оцінити можливості, уміння приймати рішення в умовах обмеженого часу, стратегічність мислення, здатність аналізувати власні помилки та ін. [171];

- уміння організовувати власну діяльність та ефективно управляти часом, уміння враховувати вплив факторів зовнішнього середовища на результативність професійної діяльності [78] .

Саме метакогнітивні вміння забезпечують моніторинг та управління когнітивних, емоційних і регулятивних процесів фахівця в цілому та ІТ-сфери зокрема, що забезпечують ефективність його діяльності [60].

І якщо проблема розвитку комунікативних умінь інженерів-програмістів у процесі їх професійної підготовки протягом останніх років досить активно обговорюється у науковій та ІТ-спільнотах, то проблема розвитку метакогнітивних умінь інженерів-програмістів в цілому та тестувальників ПЗ зокрема як на науковому, так і на практичному рівні до сих пір залишається неактуалізованою, мало усвідомленою.

З метою визначення основних нетехнічних умінь, що забезпечують успішність професійної діяльності тестувальників ПЗ, а також особливостей усвідомлення інженерами-програмістами з тестування ПЗ важливих для їх роботи та професійного розвитку метакогнітивних умінь,

було проведено опитування за участю 76 українських фахівців у сфері тестування ПЗ провідних ІТ-компаній (EPAM Systems, Plarium, Global Logic) різних професійних рівнів.

За результатами проведеного дослідження визначено, що:

- по-перше, - головними нетехнічними професійними якостями фахівця з тестування ПЗ учасниками опитування були названі:

- 1) комунікативні: комунікабельність, контактність, вміння домовлятися, вміння стати на позицію іншого, вміння шукати і знаходити компроміси;
- 2) відповідальність;
- 3) самоорганізованість, вміння долати стрес, вміння роботи рутинну роботу;
- 4) вміння вирішувати проблеми, вміння приймати рішення;
- 5) уважність, вміння бачити дрібниці, скрупульозність;
- 6) вміння працювати у команді;
- 7) логічність, аналітичність, критичність, креативність мислення, здатність ясно формулювати думки;
- 8) цілепокладання та вміння планувати діяльність;
- 9) організаторські здібності;
- 10) вміння презентувати результати роботи;
- 11) прагнення до саморозвитку;
- 12) володіння іноземною мовою;

- по-друге, - серед найважливіших нетехнічних професійних умінь та якостей, які відносяться до метакогнітивних, опитаними відзначені: вміння вирішувати проблеми (26%), вміння приймати рішення та програмувати свою діяльність за часовими параметрами (18,5%), вміння цілепокладання (11%) та вміння планувати діяльність (9%), а також уміння висувати гіпотези та оцінювати ризики (7,4%);

- по- третє, - 90,8% опитаних відзначили, що підготовка у закладах вищої освіти зовсім не забезпечує формування нетехнічних умінь у майбутніх фахівців, що зумовлює необхідність, гостру потребу у додатковому навчанні у межах неформальної освіти - проходження спеціальних тренінгів, курсів як в самих ІТ-компаніях, так і за їх межами (на базі різноманітних тренінгових центрів, проходження дистанційних курсів тощо).

Отже, з метою підготовки майбутніх інженерів-програмістів до тестування ПЗ в цілому та формування в них нетехнічних умінь була розроблена спеціальна тренінгова програма, що включала 2 тренінги (Додаток Е).

Тренінги проводились у процесі практичної підготовки студентів ІТ-спеціальностей на базі ІТ-компанії послідовно. Кожний тренінг був розрахований на 16 годин та передбачав проведення занять обсягом 4 години один раз на тиждень. Таким чином увесь період практичної підготовки майбутніх тестувальників ПЗ був занурений у тренінговий процес.

За своєю організаційною структурою кожен тренінг складався з трьох основних частин: 1) вступної, 2) основної та 3) завершальної.

Тренінгова робота будувалася відповідно до вимог групового тренінгу у контексті навчання дорослих в цілому (О. Пашко та ін.), а також методів розвитку мисленнєвої і метакогнітивної діяльності (А. Карпов та ін.), емоційної регуляції та комунікативних умінь (Н. Лук'янчук, О. Сидоренко, І. Слободянюк та ін.).

Вступна частина тренінгів була спрямована на: ознайомлення учасників із загальною програмою, метою та завданнями; відпрацювання загальних правил тренінгової роботи; з'ясування очікувань, а також знайомство учасників. Важливим на цьому етапі є створення єдиного тренінгового простору, адже за своїм графіком тренінгові заняття є пролонгованими у часі. Серед основних методів і технік, що забезпечували вирішення завдань цієї частини тренінгової роботи були такі:

- ігрові вправи на знайомство: презентація себе та групи;
- групова дискусія, що передбачає спільне обговорення певного суперечливого питання;
- асоціативний зв'язок (для вироблення правил групової роботи);
- незавершені речення та ін.

Основна частина тренінгів спрямована на: проблематизацію, входження учасників у тематичний простір; актуалізацію важливої інформації, оцінку наявних та опанування учасниками новими знаннями щодо певної категорії нетехнічних умінь фахівців ІТ, їх сутності, механізмів розвитку; оволодіння методами активізації мисленневих, метакогнітивних, регулятивних та комунікативних умінь; самодіагностику рівня розвитку нетехнічних умінь.

Ця частина тренінгової роботи передбачала логічне поєднання міні-лекцій, групову роботу, виконання учасниками індивідуальних завдань, а також вправи на зняття напруги (вправи-криголами), зміни видів активності та вправи для об'єднання в групи. Ефективними методами розвитку нетехнічних умінь майбутніх інженерів-програмістів на цьому етапі тренінгової роботи є:

- кейс-метод (дослідження випадку), що забезпечив проживання учасниками в змодельованих умовах реальних професійних ситуацій;
- рольові ігри та інсценізації, що забезпечили, у першу чергу, розвиток комунікативних умінь, практикування нових стратегій працювати у команді, домовлятися, комунікувати з членами команди та з замовником, обґрунтовувати і відстоювати свою позицію;
- створення ментальних карт;
- груповий аналіз інтерактивного відео;
- метаплан (робота з картками-повідомленнями);
- створення проекту свого професійного розвитку;
- група проєктна робота;
- візуалізації свого мислення;
- панельні дискусії;
- рефлексивні вправи;
- методи керування часом;
- психотехнічні вправи на розвиток мислення, його системності, критичності, гнучкості;

- психотехнічні вправи на розвиток уваги, зосередженості, відновлення розумової працездатності;

- поведінковий метод – відпрацювання різних професійних ситуацій з програванням та аналізом власної поведінки, що забезпечує розвиток самоконтролю власної діяльності, власних емоцій, активізації працездатності;

Завершальна частина тренінгів передбачала узагальнення отриманих результатів, підведення підсумків; активізацію рефлексивної діяльності учасників; змістове завершення тренінгової роботи, оцінку її результативності.

Отже, *перший тренінг*, а саме *Тренінг-дизайн мислення*, був спрямований, у першу чергу, на формування та розвиток у майбутніх інженерів-програмістів таких груп нетехнічних умінь як мисленнєві та метакогнітивні.

Важливим завданням тренінгу було здійснення оцінки учасниками власних метакогнітивних стратегій на різних етапах вирішення проблемних ситуацій в професійній діяльності тестувальників ПЗ, а саме:

- фіксації ускладнень та формулювання проблеми;
- оцінки різноманітних аспектів свого пошуку (варіантів рішень, альтернатив, стратегій, помилкових дій);
- вибір подальшого напрямку вирішення проблемної ситуації;
- аргументації вибору;
- рефлексивної оцінки власної діяльності.

Зазначені стратегії є етапами професійного мислення інженера-програміста в цілому та тестувальника ПЗ зокрема (В. Спірідонов, Ю. Кукушкіна, М. Холодна та ін.).

Для розвитку мисленнєвих та метакогнітивних умінь у майбутніх тестувальників ПЗ у процесі тренінгової роботи передусім були реалізовані *евристичні* методи в індивідуальній та груповій формах: 1) цілепокладання (формулювання мети власної діяльності/мислєдіяльності, вербалізація

мети та завдань, формулювання тактичної та стратегічної мети); 2) антиципації (складання прогнозу на робочий день, на день роботи у команді проєкту, здійснення перспективної рефлексії власної діяльності/мислєдїяльності); 3) прийняття рїшення (генерування ідей для формулювання гіпотез, знаходження помилок); планування та програмування (складання простого плану дій для окремої операції технічної; складання плану на день; розробка плану саморозвитку; перетворення плану у програму діяльності); 4) самоконтролю: співставлення мети та результату, визначення протиріч, фіксація помилок та їх корегування тощо; 5) рефлексії (заповнення таблиці рефлексивних запитань, схематизація власної мисленнєвої діяльності та ін.).

Другий тренінг – *Тренінг професійної комунікації*, був спрямований у першу чергу на формування та розвиток комунікативних та регулятивних умінь майбутніх тестувальників ПЗ, а саме: 1) вміння працювати у команді, підтримувати позитивну атмосферу, домовлятися, комунікувати з членами команди та з замовником, обґрунтовувати і відстоювати свою позицію, вміння індивідуального підходу, неконфліктність; 2) самоконтролю власної діяльності, самоконтролю власних емоцій, активізації працездатності, вміння керувати часом, бути зосередженим та уважним.

Для цього були використані такі методи: оцінки особливостей власної невербальної комунікації й опанування механізмів встановлення невербального та вербального контакту (техніки миттєвого встановлення контакту, пошуку відмінностей партнерів, запам'ятовування тощо); удосконалення умінь емпатії (техніка емпатійного слухання) та децентрації (техніка зміни комунікативної та поведінкової позиції); активізації умінь координації та взаємодії на психомоторному рівні (командна робота у вправах з м'ячом, прищипками тощо); визначення стратегій вирішення конфліктних ситуацій (техніка суперечок за присутності свідка та ін.); розвитку презентаційних умінь, у тому числі – аргументації своєї точки

зору; а також техніки тайм-менеджменту (методи: вибудовування пріоритетів; зміни домінанти; визначення та долання викрадачів часу; внутрішнього таймеру; структурування робочого дня, доби та ін.).

І перший і другий тренінг завершуються такою організаційною формою як ІТ-мітап (з англ. meetup - неформальна зустріч, що проводиться для людей, зацікавлених в одній проблемі, і котрі знають різні підходи до її вирішення) – зустріч фахівців для обговорення проблем, що відбувається у формі доповідей спікерів з інтерактивним обговоренням. Спікерами мітапів є як самі студенти, так і запрошені фахівці ІТ-компанії, у тому числі - тестувальники ПЗ, найчастіше кваліфікаційних рівнів «Junior» та «Middle specialist». Це забезпечує близькість проблем, що обговорюється, актуальність досвіду їх вирішення, більшу зрозумілість та прийняття студентами, а також емоційність та можливість відчувати себе на рівних. Важливим є створення особливої психологічної комфортної атмосфери ІТ-мітапу, що передбачає особистісну та професійну підтримку і безоцінне ставлення один до одного. Важливо, що зазначені заходи поєднують інформативну, інтерактивну та емоційну складові професійної комунікації в ІТ-спільноті та є тематично спрямованими, контекстними.

Таким чином, проведення зазначеної тренінгової роботи як комплексу двох тренінгів забезпечить розвиток нетехнічних умінь майбутніх інженерів-програмістів, що складають операційно-діяльнісний компонент їх професійної готовності до тестування ПЗ.

### **Висновки до другого розділу**

Здійснений аналіз науково-методичних засад професійної підготовки інженерів-програмістів в умовах неформальної освіти дає підстави зробити такі висновки за розділом.

1. На підставі визначених особливостей сучасної системи неформальної освіти в цілому та інженерів-програмістів зокрема, *під*

*професійною підготовкою майбутніх інженерів-програмістів в системі неформальної освіти* визначено невід’ємну складову цілісної системи їх освіти, що відбувається в умовах опанування певних освітніх програм різних провайдерів освітніх послуг (у тому числі – самих ІТ-компаній), не передбачає присудження визнаних державою освітніх кваліфікацій за рівнями освіти, проте забезпечує здобуття кваліфікаційного статусу в ІТ-компанії (на рівнях молодшого спеціаліста («Junior Specialist»), спеціаліста («Middle specialist»), старшого фахівця («Senior specialists»), керівника команди й експерта («Team Leader»), спрямована на створення умов для реалізації їх особистісного потенціалу, та визначає успішність їх професійної діяльності і професійного розвитку.

2. Грунтуючись на положеннях: діалектичного матеріалізму, прагматизму та конструктивізму як філософської методології педагогічного дослідження; системного, синергетичного, інформаційного (цифрового) підходів, що є загальнонауковими методологіями; діяльнісного, контекстного, техніко-технологічного, компетентнісного, особистісного та метакогнітивного підходів, які є спеціально методологічними засадами, а також враховуючи авторську структуру професійної готовності майбутніх інженерів-програмістів до тестування ПЗ створена та обґрунтована структурно-функціональна модель, що складається з трьох блоків відповідно до етапів діяльності та функцій кожного з них: 1) цільового, що презентує мету та завдання професійної підготовки на двох рівнях – формальної та неформальної освіти; 2) змістовно-технологічного, який визначає принципи (загальнопедагогічні та спеціальні), зміст, етапи, форми і методи підготовки, та 3) результативного, що висвітлює її критерії, показники та рівні.

Професійна підготовка майбутніх інженерів-програмістів до тестування ПЗ в умовах неформальної освіти набирає ефективності, якщо: розроблена і реалізується відповідна модель освітнього процесу, яка забезпечує інтеграцію та синхронізацію неформальної і формальної освіти;

запроваджено спеціальну теоретичну підготовку, зорієнтовану на формування орієнтовної основи діяльності з тестування ПЗ; організовано практичну підготовку на засадах менторства; забезпечено цілеспрямований розвиток нетехнічних (м'яких) вмінь (soft skills) майбутніх фахівців.

Реалізація зазначених науково-методичних засад забезпечується комплексом педагогічних заходів, а саме: 1) авторський курс «Основи тестування програмного забезпечення», спрямований на актуалізацію загальних професійних знань, оволодіння студентами спеціальними професійними знаннями з тестування ПЗ, а також базовими технічними вміннями у процесі квазіпрофесійної діяльності; 2) практична підготовка на засадах індивідуального та групового менторства, що передбачає: вирішення студентами навчально-професійних завдань, побудованих на внутрішній проблематиці ІТ-компанії на навчально-адаптаційній фазі; та їх роботу в реальних умовах професійної діяльності - проєктної ІТ-команди - на виробничій фазі; 3) тренінги розвитку нетехнічних умінь («Тренінг-дизайн мислення», «Тренінг професійної комунікації») та мітапи, що забезпечують актуалізацію мотиваційних ресурсів майбутніх фахівців щодо їх професійного саморозвитку в ІТ-сфері.

Визначені та теоретично обґрунтовані науково-методичні засади професійної підготовки майбутніх інженерів-програмістів до тестування ПЗ в умовах неформальної освіти, відповідно до вимог педагогічного експерименту, вимагають здійснення відповідного експериментального дослідження – їх апробації в умовах освітнього процесу.

Основні положення розділу висвітлені у наукових працях [30; 31;33; 35;37; 193].

### РОЗДІЛ 3.

## ЕКСПЕРИМЕНТАЛЬНА ПЕРЕВІРКА ЕФЕКТИВНОСТІ НАУКОВО-МЕТОДИЧНИХ ЗАСАД ПРОФЕСІЙНОЇ ПІДГОТОВКИ МАЙБУТНІХ ІНЖЕНЕРІВ-ПРОГРАМІСТІВ ДО ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ В УМОВАХ НЕФОРМАЛЬНОЇ ОСВІТИ

### 3.1. Організація та хід педагогічного експерименту

З метою перевірки ефективності розроблених науково-методичних засад і структурно-функціональної моделі формування готовності майбутніх інженерів-програмістів до тестування ПЗ в умовах неформальної освіти, підтвердження або спростування гіпотези, покладеної в основу дисертаційного дослідження, нами було проведено педагогічний експеримент. Він здійснювався протягом 2018-2020 років на базі Харківського відділення EPAM Systems.

EPAM Systems (Effective Programming for America) - американська ІТ-компанія, виробник ПЗ на замовлення та консалтингу, резидент Білоруського парку високих технологій. Штаб-квартира компанії розташована в м. Ньютаун (США, штат Пенсільванія), а її філії та офіси представлені в 25 країнах світу, серед яких: Великобританія, Німеччина, Україна, Польща, Швеція, Швейцарія, Болгарія, Білорусія та ін. (<https://www.epam.com/>). Основними напрямками діяльності компанії є такі:

- ІТ-консалтинг;
- розробка ПЗ;
- інтеграція додатків;
- порування і міграція додатків;
- тестування програмного забезпечення;
- створення виділених центрів розробки;
- розробка цифрових стратегій.

Станом на початок 2020 року, EPAM Systems є лідером в рейтингу найбільших компаній України, яка працевлаштовує більше 8000 тисяч фахівців, у тому числі 7500 технічних спеціалістів, й відкриває в середньому понад 400 вакансій на місяць. Серед більш, ніж 300 проєктів, які знаходяться в активній розробці, є такі предметні сфери як фінанси, медіа, охорона здоров'я, енергетика, автомобілебудування, страхування.

Враховуючи різновекторність і складність розвитку складових готовності майбутніх інженерів-програмістів до тестування ПЗ, що вимагає довготривалості та інтенсивності педагогічного впливу, експеримент, відповідно до існуючих вимог (С. Гончаренко [24] та ін.), було реалізовано за лонгітудною формою, в три етапи:

1) констатувальний – спрямований на встановлення первинного рівня професійної готовності майбутніх фахівців;

2) формувальний, який передбачав організацію підготовки згідно розробленої структурно-функціональної моделі і реалізацію всіх педагогічних заходів;

3) контрольний, метою якого було проведення вторинного діагностування рівня сформованості професійної готовності майбутніх інженерів-програмістів до тестування ПЗ після реалізованих формувальних заходів, аналіз отриманих результатів та порівняння їх з первинними даними.

Перед проведенням констатувального етапу було проведено *пілотне дослідження*, метою якого було з'ясування актуальності обраного предмету дисертаційної роботи, апробація діагностичного інструментарію та вибір експериментальних груп.

Пілотне дослідження було здійснено протягом 2016-2018 років серед фахівців-тестувальників ПЗ, які працюють в ІТ-компаніях та майбутніх інженерів-програмістів. У ньому взяли участь 280 студентів 2-4 курсів, які навчаються за освітніми програмами галузі знань 12 – Інформаційні технології, спеціальностей 121 - Інженерія програмного забезпечення; 122

- Комп'ютерні науки та інформаційні технології; 123 - Комп'ютерна інженерія; 124 - Системний аналіз, різних закладів вищої освіти України (Харківського національного економічного університету імені Семена Кузнеця, Запорізького національного університету, Національного технічного університету «ХПІ», Харківського національного університету радіоелектроніки та ін.).

Також у пілотному дослідженні взяли участь 65 фахівців у сфері тестування ПЗ, які працюють у провідних ІТ-компаніях, віком від 22 до 48 років (27,7% жінок та 72,3% чоловіків), зі стажем професійної діяльності у сфері ІТ від 6 місяців до 13 років, різних професійних рівнів (Junior – 15,4%, Middle – 23%, Senior – 33,9%, Team Lead – 27,7%).

За результатами пілотного дослідження отримані такі дані.

По-перше, - більшість студентів ІТ-спеціальностей хоча і знають про таку спеціалізацію діяльності ІТ-фахівців як тестування ПЗ, проте не усвідомлюють її особливостей. Так, 94,2% опитаних не змогли визначити специфіку діяльності фахівця з тестування ПЗ в ІТ-компанії, що відрізняє її від інших ІТ-спеціалізацій; 20% студентів вважають, що тестування не є діяльністю окремого фахівця, але є етапом роботи програміста.

По-друге, - студентами ІТ-спеціальностей (96,8%) зазначено про недостатність їх професійної підготовки у виші, про актуальну потребу у додатковій освіті. Причому більшістю опитаних (90,3%) на перше місце серед суб'єктів /надавачів таких освітніх послуг обрані саме ІТ-компанії, їх спеціалізовані програми, курси, тренінги; на друге місце - курси в різних ІТ-школах та ІТ-академіях (75,4%); на третє місце - тренінги та курси в різноманітних тренінгових компаніях, агенціях (20,5%); і на четвертому місці – курси в закладах вищої освіти (13,2%). Такі результати, у свою чергу, ще раз підкреслюють недостатню відповідність як змісту, так і форм і методів ІТ-освіти у її формальному просторі.

По-третє, - результати опитування фахівців свідчать про гостру необхідність у цілеспрямованій професійній підготовці тестувальників ПЗ.

Так, 96,9% вважають вкрай недостатньою існуючу професійну підготовку для успішної діяльності в ІТ-компанії у закладах вищої освіти. Причому 73,8% з них зазначили про суттєву невідповідність програм підготовки у вищій реальній професійній діяльності тестувальника ПЗ, про стартовий характер вищої освіти, що має компенсуватися тривалою самостійною підготовкою до такої діяльності, а також навчанням безпосередньо в ІТ-компанії.

Крім того, 92,3% опитаних фахівців відзначили, що підготовка у вищій зовсім не забезпечує формування soft skills у майбутніх фахівців, що зумовлює необхідність додаткового навчання, проходження спеціальних тренінгів, курсів як в самій ІТ-компанії, так і за її межами (тренінгові центри, дистанційні курси тощо).

Для виявлення основних метакогнітивних стратегій (домінувальних груп метакогнітивних умінь), які частіше всього використовують фахівці з тестування ПЗ у власній професійній діяльності був проведений факторний аналіз 12 метакогнітивних умінь за методикою Шкала самооцінки метакогнітивної поведінки ЛаКоста.

Для визначення кількості факторів використовувався критерій Кеттела. Міра адекватності вибірки Кайзера-Мейера-Олкіна (КМО) склала 0,618, що свідчить про задовільну адекватність факторної моделі матриці кореляцій даному набору змінних. Значущість тесту сферичності Бартлетта складає 0,000, що свідчить про існування кореляційних зв'язків між змінними вихідного масиву та можливість їх групування на основі тісноти кореляції. Було виділено чотири відносно незалежні фактори, які пояснюють 71,3% варіацій значень за шкалами (табл. 3.1). Даний показник вважається достатнім для психологічних досліджень.

Найбільший вплив на метакогнітивну поведінку фахівців з тестування ПЗ мають уміння, що увійшли до першого фактору, який пояснює 23,83% загальної дисперсії і має факторне навантаження 2,860. До нього увійшли:

1) подолання суб'єктивних обмежень (усвідомлення можливостей вирішення складних завдань та наполегливий свідомий пошук рішення) (0,904); осмислення досягнень (співвіднесення суб'єктивних досягнень з об'єктивним зворотним зв'язком) (0,893); стратегічне планування (цілеспрямоване планування, моніторинг та оцінка діяльності) (0,699); формулювання запитань (свідоме формулювання запитань, що звернені до пробілів у тій чи іншій сфері знання) (0,644). Зазначені уміння є найбільш затребуваними для групи фахівців високого рівня професійних досягнень («senior» – «team lead»).

До другого фактору (19,26% загальної дисперсії, факторне навантаження 2,311) увійшли: 1) моделювання (побудова мисленнєвих репрезентацій досвіду) та схематизація (побудова схем явищ та процесів) (0,802); 2) визначення термінології (формулювання точних визначень первинно розмитих, багатозначних або погано зрозумілих термінів) (0,690); перефразування та резюмування отриманої інформації (переосмислення ідей, що приходять) (0,639); свідоме прийняття рішення (прогнозування ефекту та наслідків кожного варіанта вибору) (0,580).

Третій фактор пояснює 15,75% загальної дисперсії і має факторне навантаження 1,890. До нього увійшли такі показники: 1) визначення когнітивної поведінки (визначення використаних когнітивних стратегій та їх значущості для вирішення завдання) (0,775); диференційна оцінка (рефлексивна оцінка власних дій за різними критеріями) (0,760). Найменший вплив на метакогнітивну поведінку фахівців з тестування ПЗ мають уміння, що входять до четвертого фактору (пояснює 12,43% загальної дисперсії і має факторне навантаження 1,492) - ведення щоденника (письмова фіксація власних думок) (0,845) та рольові ігри (програвання позиції партнера по спілкуванню/замовника, мисленнєвий діалог з ним).

Таблиця 3.1

**Перевернута матриця компонентів**  
(метод виокремлення факторів – метод головних компонент,  
метод обертання – варімакс з нормалізацією Кайзера)

	Компонент			
	1	2	3	4
подолання суб'єктивних обмежень (усвідомлення можливостей вирішення складних завдань та наполегливий свідомий пошук рішення)	<b>,904</b>	-,113	,262	-,043
осмислення досягнень (співвіднесення суб'єктивних досягнень з об'єктивним зворотним зв'язком)	<b>,893</b>	,171	,143	-,093
стратегічне планування (цілеспрямоване планування, моніторинг та оцінка діяльності)	<b>,699</b>	,280	-,117	,332
формулювання запитань (свідоме формулювання запитань, що звернені до пробілів у тій чи іншій сфері знання)	<b>,644</b>	,508	,078	,252
моделювання (побудова мисленнєвих репрезентацій досвіду) та схематизація (побудова схем явищ та процесів)	-,083	<b>,802</b>	-,109	,172
визначення термінології (формулювання точних визначень первинно розмитих, багатозначних або погано зрозумілих термінів)	,313	<b>,690</b>	,170	,137
перепарафразування та резюмування отриманої інформації (переосмислення ідей, що приходять)	,158	<b>,639</b>	-,453	,031
свідоме прийняття рішення (прогнозування ефекту та наслідків кожного варіанта вибору)	,260	<b>,580</b>	,479	-,291
визначення когнітивної поведінки (визначення використаних когнітивних стратегій та їх значущості для вирішення завдання)	,219	,067	<b>,775</b>	,002
диференційна оцінка (рефлексивна оцінка власних дій за різними критеріями)	,022	-,099	<b>,760</b>	,133
ведення щоденника (письмова фіксація власних думок)	-,071	,230	-,070	<b>,845</b>
рольові ігри (прогривання позиції партнера по спілкуванню/замовника, мисленнєвий діалог з ним)	,301	-,029	,350	<b>,665</b>
<b>Факторне навантаження</b>	<b>2,860</b>	<b>2,311</b>	<b>1,890</b>	<b>1,492</b>
<b>Процент загальної дисперсії</b>	<b>23,832</b>	<b>19,255</b>	<b>15,748</b>	<b>12,433</b>

Відтак, за результатами факторного аналізу встановлено, що найбільший вплив на метакогнітивну поведінку фахівців з тестування ПЗ та їх успішний професійний розвиток мають уміння: долати суб'єктивні обмеження, осмислення досягнень, стратегічного планування, формулювання запитань. Найменший вплив мають такі метакогнітивні вміння як ведення щоденника (письмова фіксація власних думок) та

рольові ігри (програвання позиції партнера по спілкуванню/замовника, мисленнєвий діалог з ним).

Серед заходів з удосконалення якості професійної підготовки у закладах вищої освіти опитаними фахівцями були запропоновані такі: стажування майбутніх тестувальників в ІТ-компаніях, на реальних проєктах (92,3%); включення до освітніх програм підготовки інженерів-програмістів курсів з тестування ПЗ (64,6%); запровадження дуальної форми освіти (63,1%); викладання фахових курсів виключно фахівцями-практиками ІТ-компаній (58,5%); вирішення на лабораторних заняттях реальних професійних ситуацій, завдань (запровадження контекстного підходу) (43,1%); відмова від застарілих методологій (32,3%).

Отже, за результатами пілотного дослідження отримані дані, що засвідчили про актуальність проблеми дисертаційної роботи, надали підстави для вибору експериментальних та контрольних груп, а також експертної групи фахівців для оцінки сформованості компонентів професійної готовності майбутніх інженерів-програмістів до тестування ПЗ.

Здійснене пілотне дослідження також дало змогу визначити склад експериментальних та контрольних групи. Адже його окремим заходом було проведення тестування серед умотивованих до тестування ПЗ студентів 3 курсів спеціальності 121 - Інженерія програмного забезпечення різних закладів вищої освіти щодо сформованості їх знань з індустрії розробки ПЗ. Тестування проводилось англійською мовою, що у свою чергу забезпечило відбір студентів з базовими знаннями і вміннями іншомовної мовленнєвої компетентності. Такі студенти стали учасниками експериментальних груп.

Так, *експериментальні групи* склали студенти 3 курсів спеціальності 121 - Інженерія програмного забезпечення, які проходили підготовку з тестування ПЗ в EPAM Systems: Eg1 – студенти Національного технічного університету «Харківський політехнічний інститут» та Харківського

національного університету радіоелектроніки у складі 16 осіб; Ег2 – студенти Національного технічного університету «Харківський політехнічний інститут» та Харківського національного економічного університету імені Семена Кузнеця у складі 18 осіб.

*Контрольні* групи склали студенти Національного технічного університету «Харківський політехнічний інститут» (Кг1, 20 осіб) та Харківського національного економічного університету імені Семена Кузнеця (Кг2, 20 осіб), які не проходили з підготовку тестування ПЗ в умовах неформальної освіти.

Реалізація *констатувального* етапу педагогічного експерименту передбачала проведення діагностичного дослідження – визначення первинного рівня професійної готовності майбутніх інженерів-програмістів до тестування ПЗ за всіма її компонентами, що, у свою чергу, вимагало пошуку адекватного та надійного дослідницького інструментарію.

У зв'язку з відсутністю вже розробленого діагностичного комплексу, важливим завданням стало його створення та апробація.

Так, для діагностування *ціннісно-мотиваційного* компоненту професійної готовності майбутніх інженерів-програмістів до тестування ПЗ було обрано:

1) *Методику «Мій шлях широкий» (визначення психологічної готовності до трудової діяльності)*, авт. Л. Матвеева, І. Вибойщик, Д. Мякушкін [123], що використовується для визначення мотиваційної складової професійної готовності випускників закладів професійної/вищої освіти (до конкретної сфери трудової діяльності) та складається з 18 суджень, які опитувані мають оцінити стосовно себе (обрати один з варіантів: «так», «швидше так», «швидше ні» або «ні»). Методика дає змогу визначити ступінь вираженості: впевненості майбутніх фахівців у їх професійному виборі; спрямованості на здійснення певної професійної

діяльності; цілеспрямованості, наполегливості та відповідальності за прийняте рішення (Додаток 3);

2) *Методика вивчення кар'єрних орієнтацій «Якоря кар'єри»*, авт. Е. Шейн (адаптація В. Чикер, В. Вінокурової) [162], яка складається з 41 твердження з восьми кар'єрних орієнтацій (якорів):

- професійна компетентність: спрямованість на професійний розвиток, постійне професійне вдосконалення, досягнення високих результатів у діяльності, прийняття компетентних рішень;

- менеджмент: орієнтація особистості на інтеграцію зусиль інших людей, на взяття відповідальності за кінцевий результат, на комунікацію, забезпечення групової єдності;

- автономія: орієнтація на вивільнення від організаційних правил, обмежень, приписів; домінування потреби у самостійному вирішенні проблем, збереженні свободи будь якою ціною;

- стабільність роботи: орієнтація на задоволення потреби у безпеці та стабільності, збереженні місця роботи на тривалий час; модель поведінки «людини організації», відповідальність за управління кар'єрою особистість перекладає на роботодавця;

- стабільність місця проживання: орієнтація на збереження місця проживання, життя без зміни власного житла навіть за умови ризику втратити професійний та кар'єрний розвиток;

- служіння: орієнтація на роботу з людьми, надання ним допомоги, захисту прав споживачів, клієнтів;

- виклик: орієнтація на жорстку конкуренцію, перемогу над іншими, висування викликів;

- інтеграція стилів життя: збалансована орієнтованість діяльності на різні кар'єрні цінності;

- підприємництво: орієнтованість на власне фінансове благополуччя, отримання індивідуального прибутку.

За результатами діагностування визначається провідна кар'єрна орієнтація опитуваного (домінування інтеграцій стилів життя) або робиться висновок про те, що кар'єра не займає провідного місця у його житті (Додаток 3).

Для визначення рівня сформованості *когнітивного компоненту* був створений *авторський Тест оцінки знань з тестування ПЗ*, що складався з двох частин: 1) 20 запитань, що стосуються загальних професійних знань та 2) 20 запитань, відповіді на які висвітлюються рівень сформованості спеціальних професійних знань студентів. У тесті за своєю формою наявні як закриті запитання (15 з 20 з кожної частини), так і відкриті запитання (5 з 20). Кожна правильна відповідь на закриті запитання оцінюється в 1 бал, за відкриті запитання – 2 бали. Отже, максимальна кількість балів за пройдений тест дорівнює 50 балів (Додаток 3).

Вимірювання рівня сформованості *операційно-діяльнісного компоненту* готовності відбувалося завдяки:

1) *експертній оцінці сформованості технічних умінь*, що здійснювалась за результатами виконання студентами завдання на тестування ПЗ (Додаток 3). До складу команди експертів увійшли 3 досвідчені тестувальники проєктних команд. Оцінка виконаного завдання здійснювались експертами за 100-бальною шкалою за допомогою матриці експертної оцінки, показники якої відповідали показникам сформованості технічних умінь і складовим матриці професійних компетенцій ІТ-фахівців у сфері тестування ПЗ, що є внутрішнім документом ІТ-компанії.

*Завдання* передбачало: розробку тест-плану для обраного веб-додатку, підготовку до нього наборів тест-кейсів, що описують сценарії тестування функціональності додатку з умовою використання оптимальних технік тест-дизайну, та автоматизацію набору приймальних тестів (смоук-тестів) завдяки власноруч розробленому фреймворку. Причому студенти мали врахувати такі вимоги:

- обраний сервіс повинен бути побудованим за підходом REST або SOAP, знаходитися у відкритому доступі та мати документацію (наприклад, <https://www.weatherbit.io>); обраний сервіс повинен бути аналогічним фінальному завданню авторського курсу;

- стратегія тестування, використані техніки тест-дизайну, типи тестування та покриття як мануальними, так і автоматизованими тестами повинні відповідати написаному тест-плану;

- набір тест-кейсів повинен бути ієрархічно структурованим та збереженим у системі тест-менеджменту за вибором;

- вибір інструментів автоматизації повинен бути описаним і аргументованим у тест-плані;

- архітектура тестового фреймворку повинна бути доступною для розширення та подальшого комерційного використання;

2) *методикам діагностики нетехнічних умінь*, а саме:

- *Методика дослідження комунікативних та організаторських схильностей (КОС-2)*, авт. Б. Федоришин, що спрямована на визначення рівня розвиненості:

а) *комунікативних здібностей* особистості: вмінь легко й швидко встановлювати ділові й товариські контакти з іншими; прагнення до розширення сфери спілкування, до участі у групових, командних формах взаємодії;

б) *організаторських здібностей* особистості: вмінь впливати на людей з метою успішного вирішення ними певних завдань та досягнення результатів спільної діяльності; вмінь оперативно розібратися в ситуативній взаємодії людей та направляти цю взаємодію в конструктивну форму співпраці; прагнення до прояву ініціативи, до виконання певних обов'язків у групі. Опитувальник складається з 40 тверджень, які оцінюються опитуваним як такі, що відповідають його характеристиці та такі, що не відповідають (Додаток 3);

- *Шкала самооцінки метакогнітивної поведінки ЛаКоста* [62]. Ця методика є короткою анкетною, досить простою у використанні та інтерпретації. Випробовувані повинні оцінити когнітивні стратегії, які вони використовують. Методика рекомендована саме для використання як індикатор рівня сформованості метакогнітивних стратегій у ситуації цілеспрямованого навчання та для діагностики профілю метакогнітивних стратегій у фахівців, які використовують їх у професійній діяльності. Адже саме зазначені у методиці метакогнітивні стратегії віддзеркалюють базові механізми актуалізації метакогнітивних умінь (об'єктивізація, схематизація, нормування, вербалізація, рефлексія). Важливим є й те, що ця методика має дидактичний характер і стимулює мисленнєву, рефлексивну діяльність опитуваного, проблематизує ситуацію, настановлює на міркування про існуючі метакогнітивні механізми/стратегії. Шкала складається з 12 тверджень, які визначають ступінь генералізації метакогнітивних умінь – частоту їх використання (Додаток 3).

Отже, зазначений комплекс діагностичних методик надав змогу вимірити рівень професійної готовності майбутніх інженерів-програмістів до тестування ПЗ за всіма її компонентами.

*Формувальний етап* педагогічного експерименту здійснювався протягом 2018-2020 років у декілька підетапів.

Перший - *настановно-організаційний* підетап - передбачав входження студентів 3 курсів до формувального процесу, що відбувалося у формі їх участі в: 1) організаційному занятті у формі інтерактивної лекції, яка проводилась у закладі вищої освіти та 2) мітапі для тестувальників ПЗ, що проводився в ІТ-компанії. Завданням цих двох заходів було: первинне занурення майбутніх фахівців у сферу тестування ПЗ, посилення інтересу до нього; актуалізація мотиваційних ресурсів щодо професійного саморозвитку в ІТ-сфері; м'яке входження у нову для них освітню програму з її певними організаційними викликами – додаткове навантаження, зміна ролі (не тільки студент, але і молодий фахівець – член

проектної команди, який несе відповідальність за командний результат), потреба у виконанні завдань, самоорганізації та саморегуляції, тощо.

Другий – *адаптаційно-орієнтаційний підетап* передбачав опанування майбутніми інженерами-програмістами орієнтувальної (інформаційної) основи їх діяльності з тестування ПЗ, тобто актуалізацію загальних професійних знань, оволодіння спеціальними професійними знаннями з тестування ПЗ, а також базовими технічними вміннями у процесі квазіпрофесійної діяльності у межах авторського курсу «Основи тестування ПЗ». Авторський курс, як вже було зазначено раніше, розрахований на 3 кредити (90) годин, з яких: 20 годин – лекційні заняття, 18 годин – практичні заняття, 52 години – самостійна робота. На відміну від більшості курсів, які викладаються у закладі вищої освіти, викладач курсу є досвідченим практиком (кваліфікаційний рівень «Team Leader»), який працює в ІТ-компанії та передусім презентує майбутнім фахівцям її цінності та норми, професійне мислення та професійну поведінку.

Третій, *конструктивно-перетворювальний підетап* формувального етапу експерименту, був спрямований, у першу чергу, на формування та розвиток технічних і нетехнічних умінь майбутніх інженерів-програмістів, формування досвіду їх діяльності в ІТ-компанії, досвіду роботи у складі команди реального ІТ-проєкта, що забезпечує формування операційно-діяльнісного компоненту їх професійної готовності до тестування ПЗ.

Цей етап тривав протягом 5 місяців та передбачав розгортання таких педагогічних заходів на базі ІТ-компанії як:

1) практична підготовка майбутніх інженерів-програмістів до тестування ПЗ на засадах менторства. Вона також мала свою внутрішню структуру та, як нами було вже описано у попередньому розділі, мала:

а) *навчально-адаптаційну фазу* тривалістю перших двох місяців роботи, змістом якої було послідовне вирішення студентом професійних завдань навчального характеру, побудованих на внутрішній проблематиці ІТ-компанії, починаючи з адміністрування і роботи з системами

управління проєктами та завершуючи адмініструванням і роботою з системами безперервної інтеграції та безперервного розгортання.

Цю фазу забезпечувала реалізація менторства в індивідуальній формі (індивідуальна робота ментора-фахівця ІТ-компанії з одним студентом – консультування за нестандартизованим, гнучким графіком за схемою рефлексивного процесу) та груповій формі (щотижневі лекції-консультації). Завершення навчально-адаптаційної фази відбувається у формі співбесіди зі студентом, що проводилась групою інтерв'юерів - експертною комісією з фахівців ІТ-компанії у складі: ментора, куратора-викладача групи, лідерів («Team Leads») робочих проєктних команд, за якими планується подальший розподіл та закріплення студентів;

б) *виробничу фазу* тривалістю 3 наступних місяців практичної підготовки, що передбачала діяльність студентів з тестування ПЗ у реальній проєктній команді, яка забезпечує реалізацію актуального виробничого процесу ІТ-компанії. Проходження студентом цієї фази практичної підготовки забезпечується індивідуальним менторством у двох варіаціях: - продовження індивідуальних консультацій уже закріпленим за студентом ментором; - проєктно-спрямоване професійне консультування, що здійснює лідер команди, у тому числі за технологією «onboarding».

2) тренінгова програма розвитку нетехнічних умінь, компонентами якої були Тренінг-дизайн мислення (16 год.) та Тренінг професійної комунікації (16 год.), що передбачали проведення занять обсягом 4 години один раз на тиждень. Відтак, тренінгова робота здійснювалась протягом усієї навчально-адаптаційної фази навчальної підготовки майбутніх тестувальників ПЗ.

Четвертий – *контрольно-рефлексивний* підетап формувального етапу педагогічного експерименту забезпечував: узагальнення, зміцнення та поглиблення усіх складових готовності студентів до тестування ПЗ; презентацію досягнутих освітніх результатів; пролонгацію їх освітньої та самоосвітньої діяльності і, таким чином, вихід на новий етап свого

професійного розвитку в умовах неформальної освіти та своєї навчально-професійної діяльності (у тому числі у контексті отримання запрошення на роботу в ІТ-компанії).

Цей етап передбачав проведення:

- по-перше, - *рефлексивної сесії*, на якій обговорювалися результати підготовки студентів за участю менторів з подальшим виданням сертифікатів та,

- по-друге, - *заключного мітану* як неформального спілкування і презентації особистих досягнень студентів.

*Контрольний етап* педагогічного експерименту передбачав здійснення повторного діагностування професійної готовності майбутніх інженерів-програмістів до тестування ПЗ за тим самим діагностичним інструментарієм.

### **3.2. Аналіз результатів педагогічного експерименту**

З метою підтвердження або спростування гіпотези дисертаційного дослідження щодо ефективності впливу визначених науково-методичних засад на формування готовності майбутніх інженерів-програмістів до тестування ПЗ в умовах неформальної освіти важливим етапом є порівняння результатів її первинного (на констатувальному етапі педагогічного експерименту) та вторинного діагностування (на його контрольному етапі).

Розглянемо у порівняльному форматі отримані результати діагностування професійної готовності майбутніх інженерів-програмістів до тестування ПЗ за всіма її компонентами.

Так, ціннісно-мотиваційний компонент діагностувався за допомогою методики «*Мій шлях широкий*» (визначення психологічної готовності до трудової діяльності), авт. Л. Матвєєва, І. Вибойщик, Д. Мякушкін, та

*Методики вивчення кар'єрних орієнтацій «Якоря кар'єри», авт. Е. Шейн (адаптація В. Чикер, В. Вінокурової).*

За результатами первинного діагностування (констатувальний зріз) у студентів експериментальних та контрольних груп були отримані схожі результати, що підтвердили рівнозначність вибірок. Проте за результатами вторинного вимірювання (контрольний зріз) отримані результати значно різняться (табл. 3.2).

*Таблиця 3.2*

Динаміка показників сформованості ціннісно-мотиваційного компоненту професійної готовності студентів експериментальних та контрольних груп до та після впровадження формувальних заходів, у %

Рівень	Ег1		Ег2		Кг1		Кг2	
	конст.	контр.	конст.	контр.	конст.	контр.	конст.	контр.
п	25	0	38,8	0	30	35	40	30
д	62,5	50	38,8	33,3	50	45	40	40
в	12,5	50	22,2	66,7	20	20	20	30

Так, якщо на констатувальному етапі у більшості учасників експериментальних та контрольних груп був виявлений початковий (Ег1 – 25%, Ег2 – 38,8%; Кг1 – 30% та Кг2 – 40%) та достатній (Ег1 – 62,5%, Ег2 – 38,8%, Кг1 – 50%, Кг2 – 40%) рівні сформованості мотивації професійної діяльності з тестування ПЗ та професійного розвитку в цілому: невпевненість у виборі сфери діяльності; невизначене ставлення до тестування як виду діяльності; недостатня спрямованість на досягнення результату, на самостійне вирішення завдань; нестійке ставлення до необхідності взяти відповідальність за результати власної діяльності. То за результатами вторинного діагностування виявлена позитивна динаміка мотиваційної сфери учасників експериментальних груп: в Ег1 та Ег2 початкового рівня сформованості мотивації не виявлено; високий рівень ознаки виявлений у 50% учасників Ег1 та 66,7% Ег2.

Динаміка рівнів сформованості ціннісно-мотиваційного компоненту професійної готовності студентів експериментальних та контрольних груп до та після впровадження формувальних заходів презентована на рис. 3.1.

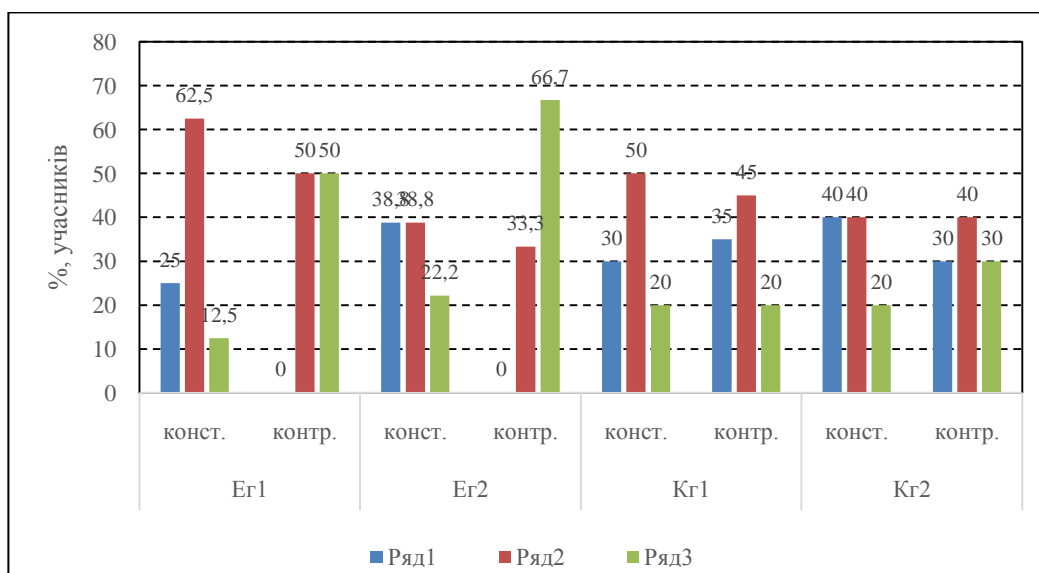


Рис. 3.1. Динаміка рівнів сформованості ціннісно-мотиваційного компоненту професійної готовності майбутніх інженерів-програмістів до тестування ПЗ (ряд 1 – початковий рівень, 2 – достатній, 3 - високий)

За результатами запровадження авторського Тесту оцінку знань з тестування ПЗ, за допомогою якого здійснювалось вимірювання когнітивного компоненту професійної готовності майбутніх тестувальників були отримані такі дані. Результати констатувального вимірювання засвідчили про домінування початкового рівня сформованості спеціальних професійних знань студентів як експериментальних, так і контрольних груп, незважаючи на наявність окремих тем та модулів в освітніх програмах підготовки у закладах вищої освіти (табл. 3.3).

Таблиця 3.3

Динаміка показників сформованості когнітивного компоненту професійної готовності студентів експериментальних та контрольних груп до та після впровадження формувальних заходів, у %

Рівень	Eг1		Eг2		Kг1		Kг2	
	конст.	контр.	конст.	контр.	конст.	контр.	конст.	контр.
п	87,5	0	77,8	0	85	60	75	65
д	12,5	25	22,2	11	15	40	25	35
в	0	75	0	89	0	0	0	0

За результатами вторинного діагностування отримана значна позитивна динаміка показників когнітивного компоненту професійної готовності учасників експериментальних груп. Так, як в Ег1, так і в Ег2 початкового рівня не виявлено, а значно переважає високий рівень сформованості зазначеної ознаки (75% учасників в Ег1 та 89% - в Ег2). В контрольних групах статистично значущої різниці в показниках когнітивного компоненту професійної готовності за контрольним етапом не виявлено (рис. 3.2): більшість студентів мають початковий рівень сформованості професійних знань.

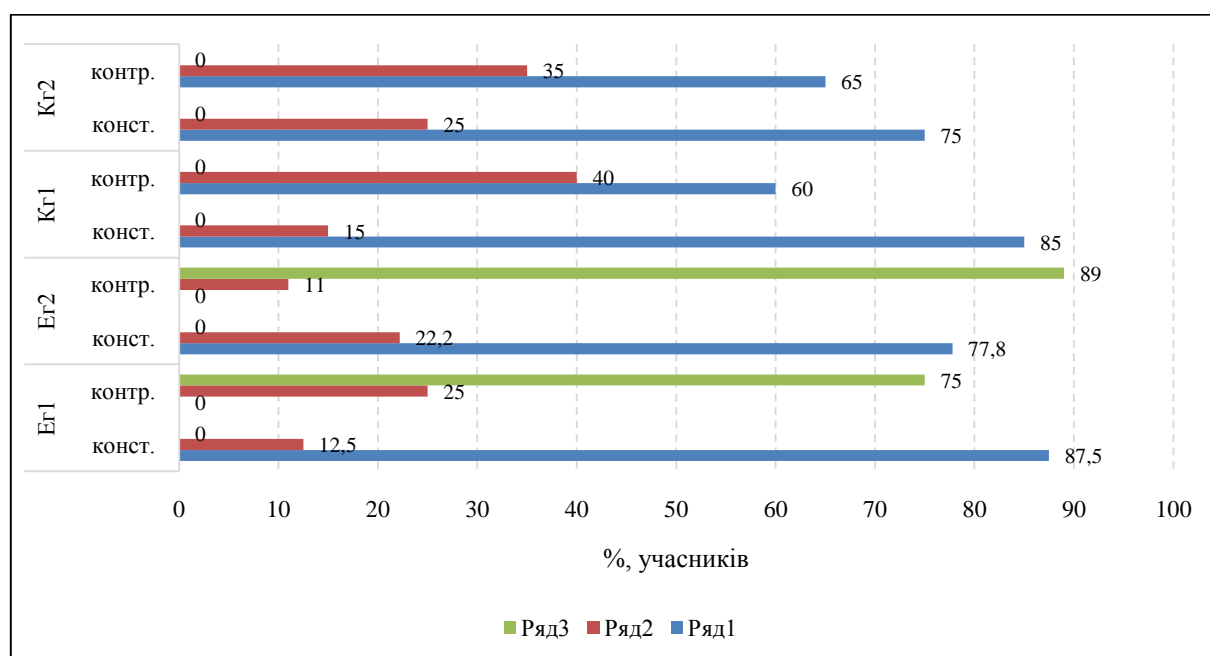


Рис. 3.2. Динаміка рівнів сформованості когнітивного компоненту професійної готовності майбутніх інженерів-програмістів до тестування ПЗ (ряд 1 – початковий рівень, 2 – достатній, 3 - високий)

Оскільки в структуру операційно-діяльнісного компонента готовності майбутніх інженерів-програмістів до тестування ПЗ входять як технічні, так і нетехнічні вміння, це вимагало здійснення окремих діагностичних процедур.

Так, виявлення рівня сформованості технічних умінь студентів експериментальних та контрольних груп відбувалось за результатами

експертної оцінки виконання ними завдання та тестування ПЗ. До складу команди експертів увійшли 3 досвідчені тестувальники проєктних команд. Оцінка виконаного завдання здійснювались експертами за 100-бальною шкалою за допомогою матриці експертної оцінки, показники якої відповідали показникам сформованості технічних умінь і складовим матриці професійних компетенцій ІТ-фахівців у сфері тестування ПЗ, що є внутрішнім документом ІТ-компанії. Початковий рівень визначався кількістю балів від 0 до 60; середній – від 61 до 89 балів та високий – від 90 балів і вище.

На констатувальному етапі педагогічного експерименту діагностування технічних умінь студентів (як Ег1, Ег2, так і Кг1, Кг2) з тестування ПЗ не здійснювалось, адже його проведення вбачалось недоцільним у зв'язку з відсутністю у них необхідних для цього професійних знань - орієнтаційної основи їх практичної діяльності.

За результатами контрольного етапу більшість студентів експериментальних груп виявили достатній рівень сформованості технічних умінь за результатами виконання завдання з тестування ПЗ (Ег1 – 62,5%, Ег2 – 72,3%); високий рівень технічних умінь був діагностований у 37,5% студентів Ег1 та 27,7% - Ег2 (табл. 3.4).

*Таблиця 3.4*

Динаміка показників сформованості технічних умінь з тестування ПЗ студентів експериментальних та контрольних груп до та після впровадження формувальних заходів, у %

Рівень	Ег1		Ег2		Кг1		Кг2	
	конст.	контр.	конст.	контр.	конст.	контр.	конст.	контр.
п	100	0	100	0	100	100	100	100
д	0	62,5	0	72,3	0	0	0	0
в	0	37,5	0	27,7	0	0	0	0

Що стосується студентів контрольних груп, то на контрольному етапі педагогічного експерименту вони не змогли впоратися із

практичним завданням – виявлений початковий рівень сформованості їх технічних умінь з тестування ПЗ (рис. 3.3).

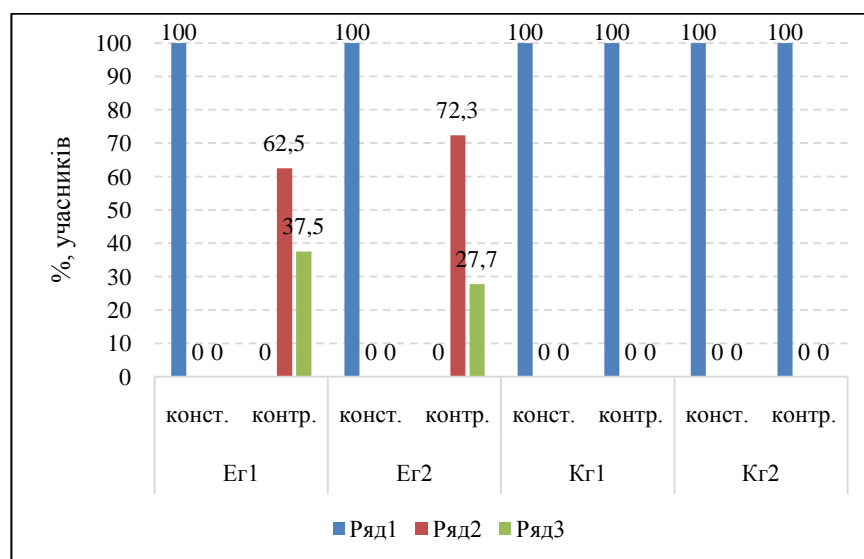


Рис. 3.3. Динаміка рівнів сформованості технічних умінь майбутніх інженерів-програмістів з тестування ПЗ (ряд 1 – початковий рівень, 2 – достатній, 3 – високий)

Вимірювання рівня розвиненості нетехнічних умінь майбутніх фахівців здійснювалось за двома методиками: *Методикою дослідження комунікативних та організаторських схильностей* (авт. Б. Федоришин), що надала змогу діагностувати комунікативні та організаторські вміння та *Шкалою самооцінки метакогнітивної поведінки ЛаКоста*, що забезпечила вимірювання метакогнітивних умінь студентів.

Для порівняння та узагальнення отриманих результатів за всіма методиками усі показники були приведені до вищезастосованої трирівневої системи оцінки – початкового, достатнього та високого рівнів. При чому, автором Методики дослідження комунікативних та організаторських схильностей Б. Федоришином зазначено про рівнозначність та взаємопоєднаність коефіцієнтів комунікативних та організаційних умінь при загальній оцінці ознаки. Тобто, для визначення загального коефіцієнта сформованості нетехнічних умінь майбутніх тестувальників ПЗ доцільно використовувати середньостатистичні значення.

Узагальнені результати діагностування нетехнічних умінь студентів експериментальних та контрольних груп подані у табл. 3.5.

Таблиця 3.5

Динаміка показників сформованості нетехнічних умінь студентів експериментальних та контрольних груп до та після впровадження формувальних заходів, у %

Рівень	Ег1		Ег2		Кг1		Кг2	
	конст.	контр.	конст.	контр.	конст.	контр.	конст.	контр.
комунікативні вміння								
п	18,7	6,25	27,8	5,6	20	15	25	30
д	50	37,5	38,9	61,2	60	60	45	50
в	31,25	56,25	22,2	33,3	20	25	30	20
організаційні вміння								
п	43,75	12,5	27,8	5,6	30	30	35	30
д	37,5	56,25	50	55,6	50	45	65	60
в	18,75	31,25	22,2	38,9	20	25	5	10
метакогнітивні вміння								
п	50	0	38,9	0	40	30	40	35
д	50	56,25	55,6	55,6	60	70	55	60
в	0	43,75	5,6	44,4	0	0	5	5

Як бачимо, за результатами первинного зрізу співвідношення рівнів сформованості зазначених ознак в усіх групах виявлялося за кривою нормального розподілу. Результати контрольного діагностування висвітлюють позитивну динаміку показників сформованості усіх груп нетехнічних умінь студентів експериментальних груп: значне зниження початкового рівня: 1) комунікативні вміння: в Ег1 з 18,5% до 6,25%; в Ег2 з 27,8% до 5,6% (рис. 3.4); 2) організаційні вміння: в Ег1 з 43,75% до 12,5%, в Ег2 з 27,8% до 5,6% (рис. 3.5); 3) метакогнітивні вміння: в Ег1 з 50% до 0%, в Ег2 з 38,9% до 0% (рис. 3.6).

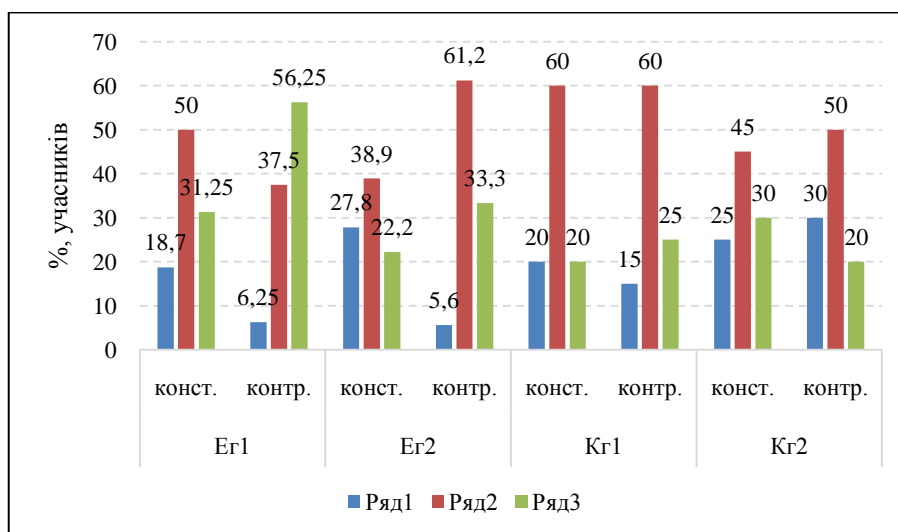


Рис. 3.4. Динаміка рівнів сформованості комунікативних умінь майбутніх інженерів-програмістів (ряд 1 – початковий рівень, 2 – достатній, 3 - високий)

Крім того, зафіксоване зростання показників високого рівня сформованості ознак: 1) комунікативні вміння: в Eг1 з 31,25% до 56,25%; в Eг2 з 22,3% до 33,3%; 2) організаційні вміння: в Eг1 з 18,75% до 31,25%, в Eг2 з 22,2% до 38,9%; 3) метакогнітивні вміння: в Eг1 з 0% до 25%, в Eг2 з 5,6% до 33,3%.

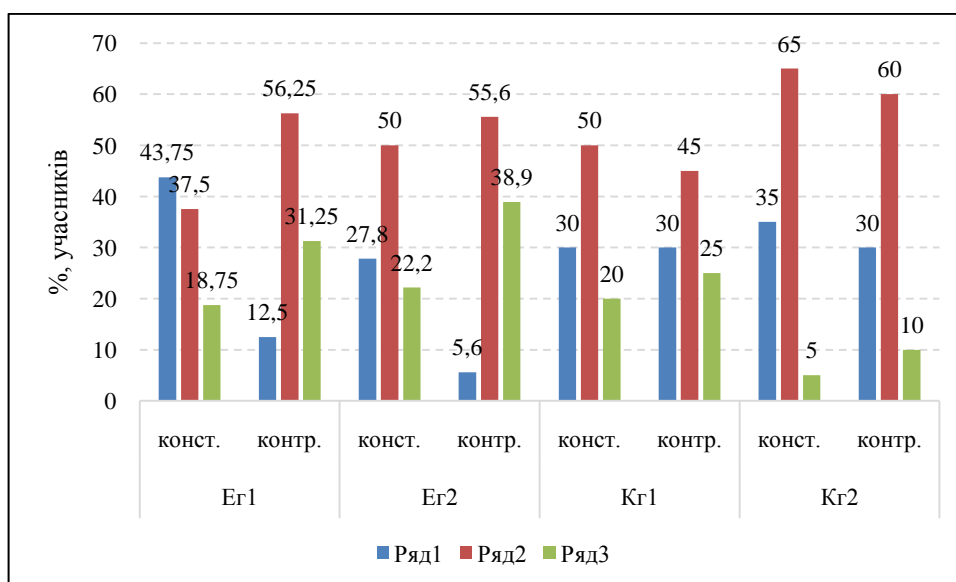


Рис. 3.5. Динаміка рівнів сформованості організаційних умінь майбутніх інженерів-програмістів (ряд 1 – початковий рівень, 2 – достатній, 3 - високий)

У студентів контрольної груп такої значної позитивної динаміки сформованості комунікативних, організаційних та метакогнітивних умінь

на контрольному етапі не виявлено. Хоча у цих групах також спостерігалось незначне зменшення початкового рівня та збільшення високого рівня сформованості нетехнічних вмінь, що пояснюється розвивальних характером діагностичних методик, а також наявним досвідом проходження опитувальників.

Необхідність отримання показників загальної динаміки рівнів професійної готовності майбутніх інженерів-програмістів до тестування ПЗ зумовлює потребу у підрахуванні загального коефіцієнта, що підраховується як арифметична середня коефіцієнтів кожного з компонентів готовності: ціннісно-мотиваційного, когнітивного та операційно-діяльнісного.

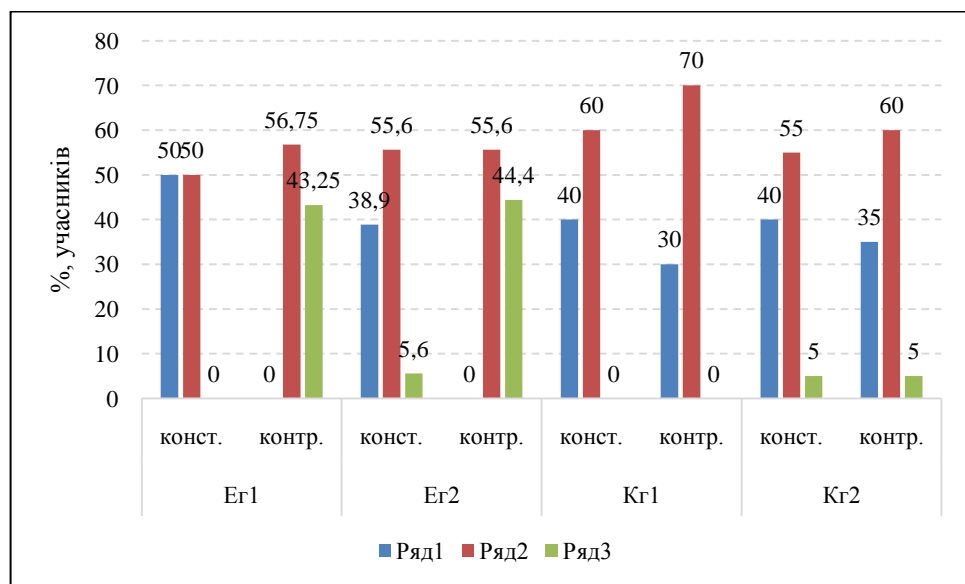


Рис. 3.6. Динаміка рівнів сформованості метакогнітивних умінь майбутніх інженерів-програмістів (ряд 1 – початковий рівень, 2 – достатній, 3 - високий)

Отже, загальний коефіцієнт професійної готовності майбутніх інженерів-програмістів до тестування ПЗ ( $K_{заг}$ ) дорівнює арифметичному середньому коефіцієнта ціннісно-мотиваційного компонента ( $K_{цм}$ ), когнітивного ( $K_{ког}$ ) та операційно-діяльнісного ( $K_{од}$ ) компонентів та підраховується за формулою:

$$K_{заг} = (K_{цм} + K_{ког} + K_{од}) / 3,$$

де:

$K_{zag}$  - загальний коефіцієнт професійної готовності майбутніх інженерів-програмістів до тестування ПЗ;

$K_{цм}$  – коефіцієнт ціннісно-мотиваційного компонента;

$K_{ког}$  – коефіцієнт когнітивного компонента;

$K_{од}$  - коефіцієнт операційно-діяльнісного компонента.

У свою чергу,  $K_{од}$  є комплексним коефіцієнтом-складовою, визначається за результатами опрацювання результатів різних методик діагностування. Отже, він визначається за формулою:

$$K_{од} = (K_{од1} + K_{од2}) / 2,$$

де:

$K_{од}$  - коефіцієнт операційно-діяльнісного компонента;

$K_{од1}$  – коефіцієнт сформованості технічних умінь;

$код2$  – коефіцієнт сформованості не технічних умінь. Причому він також є комплексним, визначається відповідно до рівнів сформованості комунікативних, організаційних та метакогнітивних умінь за формулою:

$$K_{од2} = (K_{ком} + K_{орг} + K_{мет}) / 3,$$

де:

$K_{од2}$  – коефіцієнт сформованості нетехнічних умінь;

$K_{ком}$  – коефіцієнт сформованості комунікативних умінь,

$K_{орг}$  - коефіцієнт сформованості організаційних

$K_{мет}$  - коефіцієнт сформованості метакогнітивних умінь.

Таким чином, загальний коефіцієнт загальний коефіцієнт професійної готовності майбутніх інженерів-програмістів до тестування ПЗ дорівнює:

$$K_{zag} = (K_{цм} + K_{ког} + (K_{тех} + (K_{ком} + K_{орг} + K_{мет}) / 3) / 2) / 3.$$

Кожний із вищезазначених коефіцієнтів підраховуємо за формулою:

$$K = (m * x_g + p * x_o + n * x_n) / Q$$

де

де:

$Q$  – загальний показник обсягу вибірки, який дорівнює сумі  $m$ ,  $n$ ,  $p$ ;  
 $m$  – кількість показників із високим рівнем сформованості ознаки;  
 $x_v$  – вага ознаки;  
 $p$  – кількість показників із достатнім рівнем сформованості ознаки;  
 $x_d$  – вага ознаки;  
 $n$  – кількість показників з початковим рівнем сформованості ознаки;  
 $x_n$  – вага ознаки.

Якщо вага показників із високим рівнем сформованості дорівнює 1, із достатнім – 1/2, з початковим – 0, то одержуємо формулу:

$$K = (m + 1/2 * p) / Q.$$

Здійснене підрахування коефіцієнтів професійної готовності майбутніх інженерів-програмістів до тестування ПЗ за ціннісно-мотиваційним, когнітивним та операційно-діяльним компонентами дає змогу визначити загальний коефіцієнт сформованості за результатами констатувального та контрольного вимірювання (табл. 3.6).

Таблиця 3.6

Динаміка коефіцієнтів професійної готовності майбутніх інженерів-програмістів до тестування ПЗ за результатами педагогічного експерименту

Компонент професійної готовності (коефіцієнт)	Ег1		Ег2		Кг1		Кг2	
	Конст.	Контр.	Конст.	Контр.	Конст.	Контр.	Конст.	Контр.
Ціннісно-мотиваційний (Кцм)	0,43	0,75	0,41	0,83	0,45	0,42	0,4	0,5
Когнітивний (Кког)	0,125	0,875	0,11	0,94	0,075	0,2	0,125	0,175
Операційно-діяльним (Код)	0,173	0,695	0,2	0,655	0,208	0,23	0,2	0,2
Загальний (Кзаг)	0,24	0,77	0,24	0,80	0,24	0,28	0,24	0,29

Виходячи з того, що: якщо коефіцієнт знаходиться в межах від 0,00 до 0,32, то він визначає загальний початковий рівень сформованості ознаки, якщо між 0,33 до 0,66 – достатній, між 0,67 до 1,00 – високий рівень. Отже, можна стверджувати, що за результатами педагогічного експерименту спостерігається значна позитивна динаміка коефіцієнтів

професійної готовності майбутніх інженерів-програмістів до тестування ПЗ експериментальних груп з початкового до високого рівнів (Ег1: з 0,24 до 0,77; Ег2: з 0,24 до 0,80) (рис. 3.7).

Для оцінки достовірності різниці коефіцієнтів професійної готовності майбутніх інженерів-програмістів до тестування ПЗ на констатувальному та контрольному етапах педагогічного дослідження експериментальних груп був застосований кореляційний аналіз за допомогою IBM SPSS Statistics, версія 23.

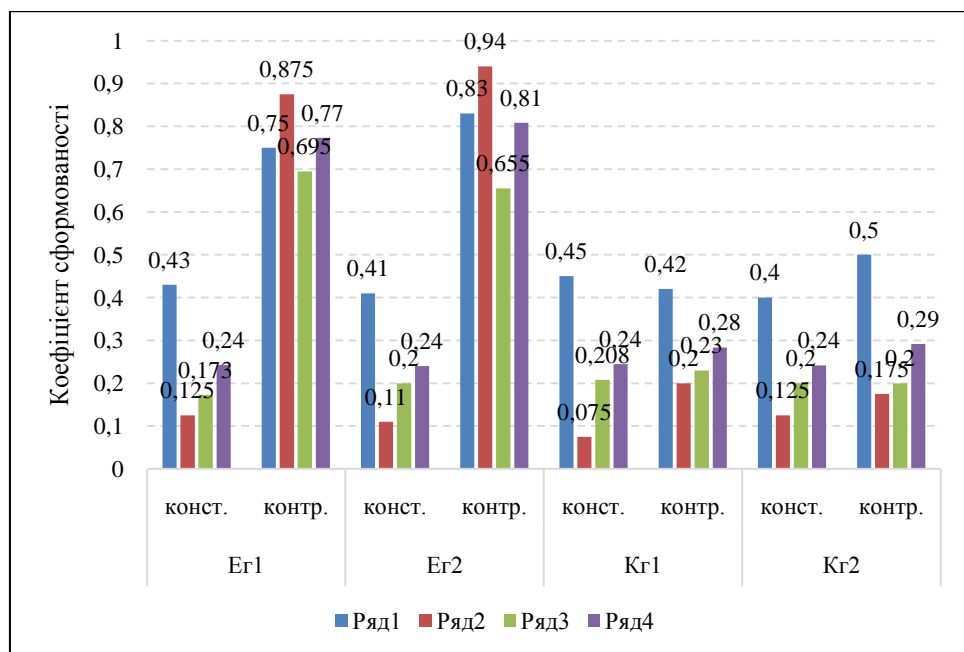


Рис. 3.7. Динаміка коефіцієнтів сформованості готовності майбутніх інженерів-програмістів до тестування ПЗ (ряд 1 – коефіцієнт ціннісно-мотиваційного компоненту, 2 – когнітивного, 3 – операційно-діяльнісного, 4 – загальний коефіцієнт)

Перевірка достовірності одержаних результатів здійснена з використанням t-критерія Стьюдента. Оскільки значення є парними та з рівною кількістю вимірювань, то використовується формула визначення достовірності різниці середніх у випадку залежних вибірок:

$$Sd = \sqrt{\frac{\sum d_i^2 - \frac{(\sum d_i)^2}{n}}{n \cdot (n - 1)}}$$

де,

$d$  – різниця між результатами в кожній порівнюваній парі коефіцієнтів професійної готовності до та після проведення формувальних заходів;

$\sum d$  – різниця між результатами в кожній парі коефіцієнтів,

$\sum d^2$  – сума квадратів часткових різниць,

$n$  – число коефіцієнтів.

Розрахунок  $t_{\text{емп}}$  здійснюємо за формулою:

$$t_{\text{емп}} = \frac{\sum d}{Sd}$$

Прикінцеві розрахунки подані у табл. 3.7.

Отримане емпіричне значення  $t$ -критерія Стьюдента ( $t_{\text{емп}}$ ) дорівнює 8,415, що є значно вищим за табличне значення ( $t_{\text{табл}} = 2,571$ ), при кількості ступенів вільності  $k = (n-1) = 5$  та рівнем значущості 0,05.

Таблиця 3.7

Розрахункова матриця (експериментальні групи)

	x	y	d	d <sup>2</sup>
	0,43	0,75	-0,32	0,102 4
	0,125	0,875	-0,75	0,562 5
	0,173	0,695	-0,522	0,272 484
	0,41	0,83	-0,42	0,176 4
	0,11	0,94	-0,83	0,688 9
	0,2	0,655	-0,455	0,207 025
<b><math>\Sigma</math></b>			<b>-3,297</b>	<b>2,009 709</b>
<b>d</b>	-0,5495			
<b>Sd</b>	0,06532336			

Відтак, висунута гіпотеза ( $H_1$ ) про те, що зростання коефіцієнтів компонентів професійної готовності майбутніх інженерів-програмістів до тестування ПЗ – учасників експериментальних груп спричинено впливом експериментальних заходів, приймається; різниця між коефіцієнтами є достовірною.

Порівняння коефіцієнтів професійної готовності майбутніх інженерів-програмістів до тестування ПЗ – учасників контрольних груп засвідчило про відсутність статистично значущої їх різниці: значення  $t$ -критерія Стьюдента ( $t_{\text{емп2}}$ ) дорівнює 0,923, що є значно нижчим за табличне значення ( $t_{\text{табл}} = 2,571$ ), при кількості ступенів вільності  $k=(n-1)=5$  та рівнем значущості 0,05.

Отже, отримана статистично значуща різниця коефіцієнтів професійної готовності майбутніх інженерів-програмістів до тестування ПЗ експериментальних груп до та після впровадження педагогічних заходів, засвідчує про ефективність визначених та запроваджених науково-методичних засад їх підготовки в умовах неформальної освіти.

### **Висновки до третього розділу**

Узагальнення результатів здійсненого експериментального дослідження із запровадження визначених науково-методичних засад професійної підготовки майбутніх інженерів-програмістів до тестування ПЗ в умовах неформальної освіти дає змогу сформулювати такі висновки за розділом.

1. Педагогічний експеримент, відповідно до існуючих вимог, було реалізовано за лонгітудною формою, в три етапи: 1) констатувальний – спрямований на встановлення первинного рівня професійної готовності майбутніх фахівців; 2) формувальний, який передбачав організацію підготовки згідно розробленої структурно-функціональної моделі і реалізацію всіх педагогічних заходів; 3) контрольний, метою якого було проведення вторинного діагностування рівня сформованості професійної готовності майбутніх інженерів-програмістів до тестування ПЗ після реалізованих формувальних заходів.

2. У зв'язку з відсутністю вже створеного та апробованого комплексу діагностичних методик було розроблено власний комплекс діагностичних

засобів, що включає: 1) для діагностування ціннісно-мотиваційного компоненту професійної готовності майбутніх інженерів-програмістів до тестування ПЗ - Методику «Мій шлях широкий» (визначення психологічної готовності до трудової діяльності), авт. Л. Матвеева, І. Вибойщик, Д. Мякушкін, та Методику вивчення кар'єрних орієнтацій «Якоря кар'єри», авт. Е. Шейн (адаптація В. Чикер, В. Вінокурової); 2) для визначення рівня сформованості когнітивного компоненту - авторський Тест оцінки знань з тестування ПЗ; 3) для вимірювання рівня сформованості операційно-діяльнісного компоненту - експертну оцінку сформованості технічних умінь, Методику дослідження комунікативних та організаторських схильностей (КОС-2), авт. Б. Федоришин, та Шкалу самооцінки метакогнітивної поведінки ЛаКоста.

3. Узагальнення усіх отриманих даних, наявність статистично значущої різниці коефіцієнтів професійної готовності майбутніх інженерів-програмістів до тестування ПЗ експериментальних груп до та після впровадження педагогічних заходів, засвідчує про ефективність визначених та запроваджених науково-методичних засад їх підготовки в умовах неформальної освіти.

Основні положення розділу висвітлені у наукових працях [31; 35; 193].

## ВИСНОВКИ

У дисертації теоретично та експериментально досліджено науково-методичні засади, що забезпечують успішність професійної підготовки майбутніх інженерів-програмістів до тестування ПЗ в умовах неформальної освіти. Одержані результати підтвердили правомірність гіпотези, покладеної в його основу, а реалізовані мета й завдання надають змогу зробити загальні висновки.

1. Здійснений аналіз сучасних вимог до професійної діяльності та професійного розвитку інженерів-програмістів, їх кваліфікаційних категорій та функціональних обов'язків, змісту професійних завдань за різними методологіями і типами тестування на різних його етапах, а також результати здійсненого емпіричного дослідження серед фахівців з тестування ПЗ різних ІТ-компаній, дали підстави для виокремлення таких специфічних особливостей професійної діяльності інженера-програміста з тестування ПЗ як: 1) первинна відповідальність за якість продукту на всіх етапах його розробки, що у свою чергу вимагає здійснення аналізу та проведення максимальної кількості перевірок ПЗ; 2) метасистемний характер – вимагає від тестувальника професійних компетенцій як програміста (мати досвід програмування, знати програмний код), так і бізнес-аналітика, менеджера проекту (знати його організаційну структуру, мати організаційні та комунікативні вміння); 3) необхідність здійснення аналізу продукту з декількох позицій: з точки зору замовника програмного продукту, розробника ПЗ, а також, - користувача продуктом; 4) творчий характер - вимагає постійного пошуку шляхів перевірки, оптимізації тестового покриття, прийняття рішень в умовах невизначеності.

*Професійна готовність майбутніх інженерів-програмістів з тестування ПЗ* визначена системним утворенням, інтегративним особистісно-професійним явищем, що є результатом їх цілеспрямованої, спеціально організованої професійної підготовки в умовах неперервної

освіти, є передумовою професійної компетентності та визначає успішність їх подальшої професійної діяльності і професійного розвитку, та складається з таких компонентів як: 1) *ціннісно-мотиваційний* (визначається мотивацією професійної діяльності та подальшого їх професійного розвитку і навчання, ціннісним ставленням до виконання професійних обов'язків – тестування ПЗ, спрямованістю на досягнення командного результату); 2) *когнітивний* (віддзеркалює систему загальних та спеціальних професійних знань); 3) *операційно-діяльнісний* (складається з технічних - відображають завдання/етапи професійної діяльності тестувальника ПЗ та *нетехнічних вмінь* - відображають вимоги до їх мисленнєвої, комунікативної, регулятивної та метапізнавальної діяльності.

2. Здійснення аналізу зарубіжного та вітчизняного досвіду професійної підготовки (включаючи її нормативно-правове забезпечення, освітні програми та плани) майбутніх-інженерів програмістів до професійної діяльності в цілому та до тестування ПЗ зокрема дає підстави стверджувати про таке:

1) її характерними ознаками в зарубіжних закладах вищої освіти є: забезпечення фундаментальної та технічної підготовки у гармонійній єдності з практичним навчанням – отримання студентами ґрунтовного практичного досвіду в спеціалізованих сферах ІТ, у тому числі - у сфері тестування ПЗ; метапредметне спрямування – розширення змісту завдяки суміжним галузям та сферам діяльності (менеджмент, безпека, торгівля, медіа тощо); зорієнтованість навчальних форм і методів на виробничий процес розробки ПЗ, що вимагає проведення тематичних досліджень, локальних та загальних квазіпрофесійних проєктних заходів та ін.; забезпечення цілеспрямованого розвитку нетехнічних професійних знань та умінь студентів у процесі групових, проєктних форм, командної роботи, навчальних та виробничих практик тощо; ефективне використання ресурсу неформальної освіти: обов'язковості проходження тривалої виробничої практики на ІТ-підприємствах;

2) у вітчизняних закладах вищої освіти спостерігається: домінування теоретичної, фундаментальної підготовки над практичною, що зумовлена у тому числі застарілістю матеріально-технічного та програмного забезпечення освітнього процесу; ригідність оновлення освітньо-професійних програм, що загострює протиріччя між потребою ІТ-індустрії у кваліфікованих фахівцях та неготовністю випускників закладів вищої освіти до роботи в умовах сучасних ІТ-компаній; недостатня зорієнтованість змісту навчання на іншомовну (передусім, англомовну) підготовку студентів; недостатній рівень співпраці закладів вищої освіти та роботодавців, що у поєднанні з проблемами матеріально-технічного та програмного забезпечення освітнього процесу, ускладнюють створення необхідних умов для практичної підготовки студентів; недостатня спрямованість на підготовку майбутніх фахівців до діяльності у спеціалізованих ІТ-сферах, у тому числі – тестування ПЗ.

3. Розроблено й апробовано науково-методичні засади, що забезпечують ефективність формування готовності майбутніх інженерів-програмістів до тестування ПЗ в умовах неформальної освіти та поєднують:

1) структурно-функціональну модель освітнього процесу, яка, ґрунтуючись на положеннях: діалектичного матеріалізму, прагматизму та конструктивізму; системного, синергетичного, інформаційного (цифрового) підходів; діяльнісного, контекстного, техніко-технологічного, компетентнісного, особистісного та метакогнітивного підходів, забезпечує інтеграцію та синхронізацію неформальної і формальної підготовки на рівні цільового, змістовно-технологічного і результативного компонентів;

2) авторський курс «Основи тестування програмного забезпечення», спрямований на формування орієнтовної основи для подальшої професійної діяльності з тестування ПЗ - актуалізацію загальних професійних знань майбутніх фахівців, оволодіння спеціальними

професійними знаннями з тестування ПЗ, а також базовими технічними вміннями у процесі квазіпрофесійної діяльності;

3) зміст та технологію організації практичної підготовки на базі ІТ-компанії на засадах індивідуального та групового менторства, що передбачає вирішення майбутніми фахівцями навчально-професійних завдань, побудованих на внутрішній проблематиці ІТ-компанії на навчально-адаптаційній фазі; та їх роботу в реальних умовах професійної діяльності - проєктної ІТ-команди - на виробничій фазі;

4) розвиток нетехнічних умінь завдяки впровадженню тренінгів розвитку мисленнєвих та метакогнітивних умінь, комунікативних та організаційних здібностей, а також мітапів, які забезпечують актуалізацію мотиваційних ресурсів майбутніх фахівців щодо їх професійного саморозвитку в ІТ-сфері.

4. У ході експериментальної перевірки отримано статистично значущу різницю коефіцієнтів професійної готовності майбутніх інженерів-програмістів до тестування ПЗ експериментальних груп до та після впровадження педагогічних заходів ( $t_{\text{емп}}=8,415$  при  $t_{\text{табл}}= 2,571$ ), що засвідчує про ефективність визначених та запроваджених науково-методичних засад їх підготовки в умовах неформальної освіти. Теоретично обґрунтовані й експериментально перевірені науково-методичні засади підготовки майбутніх інженерів-програмістів до тестування ПЗ в умовах неформальної освіти упроваджено в освітню практику на основі:

- 1) авторського курсу «Основи тестування програмного забезпечення»;
- 2) практичної підготовки на засадах індивідуального та групового менторства за навчально-адаптаційною і виробничою фазами;
- 3) тренінгової програми розвитку нетехнічних умінь («Тренінг-дизайн мислення», «Тренінг професійної комунікації») та мітапів, що забезпечують актуалізацію мотиваційних ресурсів майбутніх фахівців щодо їх професійного саморозвитку в сфері ІТ.

Таким чином, розв'язання завдань дослідження зумовило досягнення його мети – наукового обґрунтування науково-методичних засад, що забезпечують успішність підготовки майбутніх інженерів-програмістів до тестування ПЗ в умовах неформальної освіти.

Проведене дослідження не вичерпує всіх аспектів проблеми й засвідчує необхідність її подальшої розробки за такими найбільш перспективними напрямками, як: професійна підготовка майбутніх інженерів-програмістів в умовах дуальної освіти; професійна підготовка майбутніх інженерів-програмістів у закладах вищої освіти до проєктної діяльності; розвиток нетехнічних умінь фахівців ІТ-сфери в умовах післядипломної освіти.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Авраменко А.С., Авраменко В.С., Косенюк Г.В. Тестування програмного забезпечення. Навчальний посібник. Черкаси: ЧНУ імені Богдана Хмельницького, 2017. 284 с.
2. Академічний тлумачний словник української мови. URL: [http://sum.in.ua/s/ghotovnistj\\_s](http://sum.in.ua/s/ghotovnistj_s) (дата звернення: 15.01.2021).
3. Ахновська І.О. Відкрита та неформальна освіта: компаративний аналіз. *Економіка і організація управління*. 2019. № 4 (36). С. 66-76.
4. Бабанін С. Статистика розвитку ІТ-ринку в США, Україні й світі. *Щоквартальний науково-інформаційний журнал «Статистика України»*. 2013. №1(60). С.22-28.
5. Бабій Г.В. Аналіз вимог до особистісних та професійних якостей інженерів із програмного забезпечення в контексті формування готовності до професійного спілкування. *Science and Education a New Dimension: Pedagogy and Psychology*. 2013. № 7. С. 17-21.
6. Бабкин О.В., Варламов А.А., Горшунов Р.А., Дос Е.В., Кропачев А.В., Зуев Д.О. Зарубежный опыт профессиональной подготовки программистов. URL: <https://cyberleninka.ru/article/n/zarubezhnyy-opyt-professionalnoy-podgotovki-programmistov> (дата звернення 01.12.20).
7. Бабушко С.Р. Європейський досвід підтвердження результатів неформальної освіти. URL: [http://reposit.unisport.edu.ua/bitstream/handle/787878787/1623/%D0%91%D0%B0%D0%B1%D1%83%D1%88%D0%BA%D0%BE\\_%D1%80%D0%B5%D0%B7%D1%83%D0%BB%D1%8C%D1%82%D0%B0%D1%82%D0%B8\\_%D0%9D%D0%A4%D0%9E.pdf?sequence=1&isAllowed=y](http://reposit.unisport.edu.ua/bitstream/handle/787878787/1623/%D0%91%D0%B0%D0%B1%D1%83%D1%88%D0%BA%D0%BE_%D1%80%D0%B5%D0%B7%D1%83%D0%BB%D1%8C%D1%82%D0%B0%D1%82%D0%B8_%D0%9D%D0%A4%D0%9E.pdf?sequence=1&isAllowed=y) (дата звернення 03.12.20).

8. Бейзер Б. Тестирование чёрного ящика. Технологии функционального тестирования программного обеспечения и систем. СПб.: Питер, 2004. 320 с.
9. Бейч Э. Консалтинговый бизнес: основы профессионализма. СПб.: Питер, 2006. 272 с.
10. Беларуский технический университет. Кафедра «Программное обеспечение информационных систем и технологий». URL: <http://www.bntu.by/fitr-poisit.html>
11. Беларуский технический университет. Факультет информационных технологий и робототехники. URL: <http://priem.bntu.by/ru/pk/specialty/fitr/#400101>
12. Бельмаз Я.М. Роль інституту наставництва в професійному розвитку викладачів вищої школи в США та Великій Британії. *Педагогіка формування творчої особистості у вищій і загальноосвітній школах*. 2015. Вип. 41 (94). С. 484-492.
13. Бех І.Д. Особистісно зорієнтоване виховання: Науково-метод. посібник, К.: ІЗМН, 1998. 204 с.
14. Биков В.Ю., Спирін О.М., Пінчук О.П. Проблеми та завдання сучасного етапу інформатизації освіти. *Наукове забезпечення розвитку освіти в Україні: актуальні проблеми теорії і практики (до 25-річчя НАПН України)*. 2017. С. 191-198.
15. Бодров В. А. Психология профессиональной пригодности. М.: ПЕР СЭ, 2006. 511 с.
16. Божко Ю.П. Професійна успішність програмістів. *Актуальні проблеми соціології, психології, педагогіки*. 2013. №4 (21). URL: <https://core.ac.uk/download/pdf/229856215.pdf> (дата звернення 03.12.20).
17. Бойченко М.А. Менторство як форма підтримки обдарованих і талановитих у країнах ЄС та США. *Педагогічні науки: теорія, історія, інноваційні технології*. 2014. № 7 (41). С. 3-9.

18. Бородіна О.О., Альошин С.П., Гафіяк А.М. Тестування програмного забезпечення та його складові. URL: <https://sworld.education/simpoz11/17.pdf> (дата звернення 03.12.20).
19. Боярська-Хоменко А.В. Визнання результатів неформального навчання як основа розвитку безперервної освіти в Чеській республіці. *Modern education - accessibility, quality, recognition*. 2018 С. 37-40.
20. Брусенко Н.В. Розробка програми менторингу на підприємстві: основні складові. *Економіка та управління підприємством*. 2014. Вип. I (53). С. 195-222.
21. Бузмаков В.І. ИТ-консалтинг как составляющая менеджмент-консалтинга. URL: <http://www.management.com.ua/ims/ims034.html> (дата звернення 03.12.20).
22. Войти в IT: какие бесплатные курсы предлагают украинцам ведущие IT-компании. URL: <https://tech.liga.net/technology/article/voyti-v-it-kakie-besplatnye-kursy-predlagayut-ukraintsam-veduschie-it-kompanii/section4#page> (дата звернення 03.12.20).
23. Глазунова О.Г., Гуржій А.М., Волошина Т.В., Корольчук В.І., Пархоменко О.В. Неформальна освіта майбутніх фахівців з інформаційних технологій: організація, контент, інструменти. *Фізико-математична освіта*. 2020. Вип.1(23). С. 29-35.
24. Гончаренко С.У. Педагогічні дослідження: методологічні поради молодим науковцям. К.-Вінниця: ДОВ Вінниця, 2008. 278 с.
25. Гончаров В.В. В поисках совершенства управления: руководство для высшего управленческого персонала. М.: МНИИПУ, 2016. 539 с.
26. Горленко В.М. Неформальна освіта як чинник розвитку інформаційно-комунікаційної компетентності вихователів дошкільного навчального закладу. *Науковий вісник Ужгородського університету. серія: «Педагогіка. соціальна робота»*. 2017. Вип. 1 (40). С. 67-73.

27. Григорьев Д. Бизнес-тренинг: как это делается М., Манн, Иванов и Фербер. 2014. 288 с.
28. Гура О.І. Психолого-педагогічна компетентність викладача вищого навчального закладу: теоретико-методологічний аспект. Запоріжжя: ГУ ЗІДМУ, 2006. 322 с.
29. Гура О. Аналіз зарубіжного досвіду впровадження системи дуальної підготовки майбутніх фахівців. *European Humanities Studies: State & Society*. 2017. Issue 4. С. 31-43.
30. Гура О. Актуальні проблеми професійної підготовки фахівців ІТ-сфери. *Електронний збірник наукових праць ЗОІППО*. 2016. № 3(25). URL: <https://drive.google.com/file/d/0B6QknaQCh-IHTFpkeVpSeHhBdGM/view> (дата звернення 03.12.20).
31. Гура О. Матриця компетенцій (skill matrix) як орієнтир підготовки майбутніх ІТ-фахівців. *European Humanities Studies: State & Society*. 2018. Issue 6. С. 181-191.
32. Гура О. Основні сфери реалізації фахівців ІТ на сучасному ринку праці: аналіз професійних вимог. *Науковий вісник Миколаївського національного університету імені В. О. Сухомлинського. Педагогічні науки*. 2017. Вип. 2 (57). С. 166-170.
33. Гура О. Особливості впровадження системи дуальної підготовки майбутніх фахівців: порівняльний аналіз. *III Всеукраїнська науково-практична конференція з міжнародною участю “Педагогіка вищої школи: досвід і тенденції розвитку”*, Запорізький Національний Університет, м. Запоріжжя, Україна, 11 травня 2018 р. С. 41-42.
34. Гура О. Особливості основних сфер реалізації ІТ фахівця на сучасному ринку праці. *Електронний збірник наукових праць ЗОІППО*. 2017. Вип. 1 (27). URL: <https://drive.google.com/file/d/0B6QknaQCh-IHWGtSWEhiYXJGTIU/view> (дата звернення 03.12.20).
35. Гура О. Особливості розвитку м'яких навичок студентів ІТ-

- спеціальностей засобами навчальних SCRUM проектів. *ScienceRise: Pedagogical Education*. 2019. №4(31). С. 8-15.
36. Гура О. Особливості співбесіди як форми контролю якості освіти студентів ІТ спеціальностей. *Педагогічні науки та освіта*. 2016. Вип. 18-19. С. 29-34.
  37. Гура О. Підготовка інженерів-програмістів до тестування програмного забезпечення в умовах неформальної освіти: проблеми та шляхи реалізації. *Педагогічні науки: теорія, історія, інноваційні технології : наук. журнал / голов. ред. А.А. Сбруюва*. Суми: Вид-во СумДПУ імені А. С. Макаренка. 2020. № 7 (101). С. 55-62.
  38. Гура О. Професійна підготовка фахівців ІТ сфери на засадах матриці компетенцій (skills matrix.) *Україна в гуманітарних та соціально-економічних вимірах. Матеріали III Всеукраїнської наукової конференції (30-31 березня 2018р. м. Дніпро) ч. II*. Наук.ред. О.Ю. Висоцький. Дніпро: «Охотник», 2018. 268 с. С. 103-105.
  39. Деркач А. А. Акмеологические основы развития профессионала. М., Воронеж: НПО МОДЭК, 2004. 752 с.
  40. Длугунович Н.А. Soft skills як необхідна складова підготовки ІТ-фахівців. *Вісник Хмельницького національного університету*. 2014. №6. С. 239-242.
  41. Дрейфусова модель набуття навичок. URL: <https://web.archive.org/web/20120315053756/http://www.sld.demon.co.uk/dreyfus.pdf/> (дата звернення 03.12.20).
  42. Енциклопедія освіти / Акад.пед.наук України; головний редактор В.Г. Кремень. К.: Юрінком Інтер, 2008. 1040 с.
  43. Ершов А.П. О человеческом и эстетическом факторах программирования. *Кибернетика*. 1972. №5. С. 95-99. URL: <http://ershov.iis.nsk.su/ru/node/791295> (дата звернення 03.12.20).
  44. Терьохіна Н.О. Організаційно-педагогічні засади розвитку неформальної освіти дорослих у США (остання чверть ХХ – початок

- XXI століття): автореф. дис. к.п.н. (13.00.01 – загальна педагогіка та історія педагогіки). Суми. 2015. 21 с.
45. Єлагін В.П. Система оцінки якості неформальної освіти в Україні. URL: <http://apdu.journal.kharkiv.ua/index.php/apdu/article/view/153/154>. (дата звернення 03.12.20).
46. Єлагін В.П., Благовестов М.О. Особливості та тенденції розвитку системи неформальної освіти в Україні. *Державне будівництво*. 2019. № 1. С. 1-11.
47. Єлагін В.П., Благовестов М.О. Система оцінки якості неформальної освіти в Україні. *Актуальні проблеми державного управління*. 2020. №1(57). С. 261-270.
48. Журавльов О. В., Сімачов О. А. Статистичне дослідження ринку ІТ-послуг в Україні. *Статистика України*. 2018. №4. С. 25-33.
49. Загородній А.Г. Аутсорсинг та його вплив на витрати підприємства . *Фінанси України*. 2009. №9. С. 87-97.
50. Зарплати українських розробників - июнь-июль 2018. URL: <https://dou.ua/lenta/articles/salary-report-june-july-2018/> (дата звернення 03.12.20).
51. Зінченко С. Психологічні особливості неформальної освіти дорослих. *Освіта дорослих: теорія, досвід, перспективи*. 2009. Вип. 1. С. 154-160.
52. Іванова Л.В., Скорнякова О.В. «Soft skills» як важлива складова конкурентоспроможності фахівця з інформаційних технологій. *Молодий вчений*. 2018. № 12 (64). URL: <http://molodyvcheny.in.ua/files/journal/2018/12/22.pdf> (дата звернення 03.12.20).
53. ІТ в Україні: куди ми рухаємося. URL: <https://dou.ua/lenta/columns/future-of-it-ukraine/> (дата звернення 03.12.20).

54. IT в Україні: якими є головні виклики для галузі. URL: <https://mind.ua/openmind/20181689-it-v-ukrayini-yakimi-e-golovni-vikliki-dlya-galuzi> (дата звернення 03.12.20).
55. Калбертсон Р., Браун К., Кобб Г. Быстрое тестирование. М.: «Вильямс», 2002. 374 с.
56. Канер К., Фолк Дж., Нгуен Енг Кек. Тестирование программного обеспечения. Фундаментальные концепции менеджмента бизнесприложений. Киев: ДиаСофт, 2001. 544 с.
57. Кар'єра в IT. URL: <https://dou.ua/lenta/tags/%D0%9A%D0%B0%D1%80%D1%8C%D0%B5%D1%80%D0%B0%20%D0%B2%20IT/> (дата звернення 03.12.20).
58. Каракозов С.Д., Петров Д.А., Худжина М.В. Формирование основной образовательной программы в условиях приведения в соответствие требований ФГОС высшего образования профессиональным стандартам (на примере направления подготовки бакалавров «Информатика и вычислительная техника» и профессионального стандарта «Программист»). Преподаватель XXI века. №1(4). С. 22-34.
59. Карпов А.В. Психология рефлексивных механизмов деятельности. М.: Изд-во «Институт психологии РАН», 2004. 424 .
60. Карпов А.В., Карпов А.А. Системная методология как основа разработки проблемы метакогнитивных способностей личности. *Системная психология и социология*. 2014. № 3(11). С. 11-19.
61. Карпов А.В., Карпов А.А., Карабущенко Н.Б., Иващенко А .В. Динамика метакогнитивных детерминант управленческой деятельности в процессе профессионализации. *Экспериментальная психология*. 2018. Т. 11. №1. С. 49-60.
62. Карпов А.В., Скитяева И.М. Психология метакогнитивных процессов личности. М.: Изд-во «Институт психологии РАН», 2005. 352 с.

63. Касич А. О. Стартапи як форма підприємницької діяльності: поняття, значення, зарубіжний досвід . URL: [http://www.market-infr.od.ua/journals/2020/39\\_2020\\_ukr/18.pdf](http://www.market-infr.od.ua/journals/2020/39_2020_ukr/18.pdf) (дата звернення 03.12.20).
64. Кирій В.В. Дослідження ІТ-галузі як важеля економічного розвитку країни і Харківського регіону. *Економіка і суспільство*. 2018. №19. С. 763-769.
65. Клищ Н.Н., Январев В.А. Наставничество на государственной службе - новая технология профессионального развития государственных служащих(зарубежный и российский опыт наставничества на государственной службе). М.: Изд. дом Высшей школыэкономики, 2014. 64 с.
66. Кобилянський А. Войти в ІТ: какие бесплатные курсы предлагают украинцам ведущие ІТ-компании. URL: <https://tech.liga.net/technology/article/voyti-v-it-kakie-besplatnye-kursy-predlagayut-ukrainsam-veduschie-it-kompanii> (дата звернення 03.12.20).
67. Коломієць А. Фундаменталізація вищої технічної освіти за кордоном: проблеми та перспективи. *Педагогіка безпеки*. 2018. № 1. С. 69-74.
68. Колычев Н.М. Лекция о лекции: учебное пособие. Москва; Берлин: Директ-Медиа, 2019. 126 с.
69. Компютерна академія ШАГ. URL: <https://od.itstep.org> (дата звернення 03.12.20).
70. Конюхов С.Л. Аналіз закордонних досліджень із проблем навчання майбутніх інженерів-програмістів об'єктно-орієнтованому програмуванню. *Збірник наукових праць Мелітопольського державного педагогічного університету імені Богдана Хмельницького. Педагогічні науки*. 2018. Вип. LXXXII. Том.2, С. 143-147.

71. Кравченко Г.Ю. Технологія організації проектної діяльності студентів вищих навчальних закладів. *Безпека людини в сучасних умовах : матер. доп. 7-ї Міжнар. наук.-метод. конф. та 105-ї Міжнар. конф. Європейської асоціації безпеки (EAS)*. Харків : НТУ «ХП», 2015. С. 138-146.
72. Краевский В. В. *Методология педагогики : новый этап : [учеб. пособие для студ. высш. учеб. заведений]* / В. В. Краевский, Е. В. Бережнова. М. : академия, 2008. 400 с.
73. Крашеніннік І.В., Осадчий В.В. Дослідження мотиваційного компоненту фахових компетентностей майбутніх інженерів-програмістів. URL: <http://eprints.mdupu.org.ua/id/eprint/10250/> (дата звернення 03.12.20).
74. Крашеніннік І.В., Осадчий В.В. Короткострокові програми навчання інженерів-програмістів у закладах формальної і неформальної освіти України. *Проблеми інженерно-педагогічної освіти*. 2017. № 54-55. С. 72-82.
75. Криспин Л., Джанет Грегори Дж. Гибкое тестирование: практическое руководство для тестировщиков ПО и гибких команд. М.: «Вильямс», 2010. 464 с.
76. Круглик В., Осадчий В. Структура професійної компетентності майбутнього інженера-програміста. *Педагогічний дискурс*. 2016. Вип. 21. С. 69-74.
77. Круглик В.С. Аналіз програм підготовки інженерів-програмістів у вищих навчальних закладах США. *Науково-педагогічний журнал «Молодь і ринок»*. 2016. №9 (140). Дрогобич: ДДПУ ім.І.Франка. С. 43-49.
78. Круглик В.С. Система підготовки майбутніх інженерів-програмістів до професійної діяльності у вищих навчальних закладах. Монографія. Мелітополь: МДПУ ім. Б. Хмельницького, 2017. 384 с.

79. Круглик В.С. Аналіз змісту та організації підготовки фахівців з програмної інженерії в університетах США. *Педагогічний дискурс: зб. наук. праць*. За ред. І.М. Шоробура. Хмельницький: ХГПА, 2016. Вип. 20. С. 107-115.
80. Круглик В.С., Осадчий В.В. Аналіз змісту професійної підготовки інженерів-програмістів. *Проблеми інженерно-педагогічної освіти*. 2016. № 5. С. 2-53.
81. Куликов С. Тестирование программного обеспечения. Базовый курс. URL: [https://svyatoslav.biz/software\\_testing\\_book/](https://svyatoslav.biz/software_testing_book/)
82. Курсы программирования CYBERBIONIC SYSTEMATICS. URL: <https://edu.cbsystematics.com/ru> (дата звернення 03.12.20).
83. Кучерук О.Я. Компетентнісний підхід у підготовці майбутніх інженерів-програмістів. URL: <https://naukajournal.org/index.php/naukajournal/article/view/170> (дата звернення 03.12.20).
84. Лебедев Сергей Алексійович. Біографія. URL: [https://uk.wikipedia.org/wiki/Лебедев\\_Сергий\\_Олексійович](https://uk.wikipedia.org/wiki/Лебедев_Сергий_Олексійович) (дата звернення 03.12.20).
85. Лебедь Г.М. Історико-педагогічний вимір змісту фахової підготовки майбутніх програмістів у другій половині ХХ - початку ХХІ століття. *Педагогічний дискурс: Збірник наукових праць/ редкол. І.М. Шоробура*. Хмельницький: ХГПА, 2017. Вип. 22. С. 85-92.
86. Левитов Н. Д. О психических состояниях человека. М.: Просвещение, 1967. 343 с.
87. Леонтьев Д.А. Психология смысла. Психология смысла. Природа, строение и динамика смысловой реальности. М.: Смысл, 2003. 488 с.
88. Лернер И.Я. Дидактические основы методов обучения. М.: Педагогика, 1991. 186 с.
89. Липаев В.В. Тестирование программ. М.: Радио и связь, 1986. 296 с.

90. Литвинова С.Г., Проскура С.Л. Підготовка фахівців з інформаційних технологій у закладах вищої освіти: стан, проблеми і перспективи. *Інформаційні технології в освіті*. 2018. №2(35). С.72-88.
91. Литвинова С.Г., Проскура С.Л. Формування професійної компетентності майбутніх бакалаврів комп'ютерних наук. *Фізико-математична освіта: науковий журнал*. Суми: СумДПУ ім. А.С.Макаренка, 2019. Вип. 2(20) С. 137-146.
92. Майерс Г., Баджетт Т., Сандлер К. Искусство тестирования программ, 3-е издание. М.: «Диалектика», 2012. 272 с.
93. Макгрегор Дж., Сайкс Д. Тестирование объектно-ориентированного программного обеспечения К: Диасофт, 2002. 432с.
94. Мачинська Н.І., Стельмах С.С. Сучасні форми організації навчального процесу у вищій школі: навчально-методичний. Львів: Львівський державний університет внутрішніх справ, 2012. 180 с.
95. МГТУ им. Баумана. Учебные планы ВО. URL: <https://bmstu.ru/abitur/studies-life/edu-programs/> (дата звернення 03.12.20).
96. МГТУ им. Баумана. Факультеты и направления. URL: <https://bmstu.ru/abitur/general/presentations/> (дата звернення 03.12.20).
97. Медзевровский И. Проблемы подготовки специалистов для ИТ-индустрии. URL: <http://hrliga.com/index.php?module=profession&op=view&id=250> (дата звернення 03.12.20).
98. Морозова М. Е. Наставничество як процес формування особистості молодого спеціаліста. *Економіка та управління: історія, реалії та перспективи*. 2016. № 1. С. 1-12.
99. Мухина Т .Г. Активные и интерактивные образовательные технологии (формы проведения занятий) в высшей школе: учебное пособие. Н. Новгород: ННГАСУ, 2013. 97 с.

100. Нагорний В.В. Інтерактивна лекція як сучасна форма викладання дисципліни у вищій школі. *Актуальні питання підготовки майбутніх фармацевтичних та медичних фахівців в умовах сучасної освіти*. Житомир: КВНЗ «Житомирський базовий фармацевтичний коледж ім. Г.С. Протасевича», 2016. С. 187-189.
101. Нагорний В. В., Нагорна Н. О., Сінченко Д. М. Інтерактивна лекція як сучасна форма викладання дисципліни у вищій школі. URL: [http://dspace.zsmu.edu.ua/bitstream/123456789/4077/1/%D0%9D%D0%B0%D0%B3%D0%BE%D1%80%D0%BD%D0%B8%D0%B9%20%D0%92%D0%92\\_2016\\_187-189.pdf](http://dspace.zsmu.edu.ua/bitstream/123456789/4077/1/%D0%9D%D0%B0%D0%B3%D0%BE%D1%80%D0%BD%D0%B8%D0%B9%20%D0%92%D0%92_2016_187-189.pdf) (дата звернення 03.12.20).
102. Наумчук О. Визначення професійних якостей майбутніх інженерів програмістів у галузі комп'ютерних мереж. *Науковий вісник мелітопольського педагогічного університету. Серія: Педагогіка*. 2015. №1(14). С. 353-358.
103. Національний класифікатор України. Класифікатор професій ДК003-2010. URL: <https://zakon.rada.gov.ua/rada/show/va327609-10#Text> (дата звернення 03.12.20).
104. Неформальна освіта: кращі практики і проекти : журнал / [гол. ред. Н.П.Павлик, відп. ред. К.А. Марчук]. Житомир: Вид-во Житомирського держ. унту імені І. Франка, 2019. Вип. 2. 130 с.
105. Образовательный стандарт высшего образования. Высшее образование Первая ступень. Специальность 1-400501 информационные системы и технологии (по направлениям). Квалификация зависит от направления специальности (в ред. постановлений Минобразования от 04.02.2015 №2, от 07.04.2016 №21) Министерство образования Республики Беларусь. Минск. URL: <https://elib.bsu.by/bitstream/123456789/165127/1/%D0%9E%D0%A1%D0%92%D0%9E%201-40%2005%2001-2013.pdf> (дата звернення 17.11.20)

106. Огієнко О.І., Терьохіна Н.О. Неформальна освіта дорослих у Сполучених Штатах Америки: історія і сучасність: монографія. Суми, 2019. 195 с.
107. Огреніч М.А. Інтерактивна лекція як одна з новітніх фор навчання. URL: <http://dspace.pdpu.edu.ua/bitstream/123456789/2945/1/Ogrenich.pdf> (дата звернення 03.12.20).
108. Ожегов С.И. Словарь русского языка. М.: Оникс-ЛИТ, Мир и образование, 1972. 1376 с.
109. Осадчий В.В., Крашеніннік І.В. Формування змісту освітніх програм підготовки майбутніх інженерів-програмістів за скороченим терміном навчання на основі аналізу ринку праці. *Інформаційні технології і засоби навчання*. 2017. Том 58, №2. С. 11-25.
110. Освітньо-професійна програма «Інженерія програмного забезпечення». Перший (бакалаврський) рівень за спеціальністю 121 Інженерія програмного забезпечення. НТУ «ХПІ». Харків, 2019. URL: <https://web.kpi.kharkov.ua/asu/121-inzheneriya-programnogo-zabezpechennya/> (дата звернення 03.12.20).
111. Основы современного тестирования программного обеспечения, разработанного на С#. Учебное пособие / В.П.Котляров, Т.В.Коликова; под ред.В.П.Котлярова. СПб, 2004. 170 с.
112. Особистісний і професійний розвиток дорослих: теорія і практика: [монографія] / авт. кол.: Аніщенко О.В., Баніт О.В., Василенко О.В., Волярська О.С., Дорошенко Н.І., Зінченко С.В., Сігаєва Л. Є.; за ред. Аніщенко О.В. К.: ІООД НАПН України, 2016. 354 с.
113. Павленко П.М. Проблемні питання підготовки ІТ-фахівців для промислових підприємств України. URL: [http://avia.nau.edu.ua/doc/2011/3/avia2011\\_3\\_1.pdf](http://avia.nau.edu.ua/doc/2011/3/avia2011_3_1.pdf) (дата звернення 03.12.20).

114. Павлик Н.П. Неформальна освіта у системі освіти України. *Освітологічний дискурс*. 2016. № 2 (14). С. 27-37.
115. Павлик Н.П. Теорія і практика організації неформальної освіти молоді: [Навчальний посібник]. Житомир: Вид-во ЖДУ ім. І.Франка, 2017. 162 с.
116. Пашко О. Навчання дорослих: виклики, специфіка, інтерактивні методи. Український досвід в регіональному економічному розвитку. Львів, 2014. 108 с.
117. Пермінова С . Проблеми та перспективи розвитку ІТ -індустрії в Україні. URL: <http://www.spilnota.net.ua/ru/article/id-1880/> (дата звернення 03.12.20).
118. Поліщук І.В. Менторство як ефективний механізм розвитку професійного потенціалу державних службовців в Україні. *Держава та регіони*. Серія: Державне управління. 2017. №1 (57). С. 16-20.
119. Попершняк С.В. Проблеми підготовки ІТ-спеціалістів. *Системи обробки інформації*. 2010. Вип. 7(88). С. 127-130.
120. Приказ Минобрнауки России от 12.03.2015 №229 «Об утверждении федерального государственного образовательного стандарта высшего образования по направлению подготовки 09.03.04 Программная инженерия (уровень бакалавриата)». URL: <http://fgosvo.ru/uploadfiles/fgosvob/090304.pdf> (дата звернення 03.12.20).
121. Програми тренінгів з розвитку комунікативності: посібник / Н.В. Лук'янчук. К., 2012. 130 с.
122. Професійний стандарт «Фахівець з розробки програмного забезпечення». URL: <https://mon.gov.ua/storage/app/media/vyshcha/IT-prof-standarty/6-ps-rozrobnik-pz-13.12.2014.pdf> (дата звернення 03.12.20).
123. Профорієнтаційна робота психолога / Упорядник Т. Гончаренко, 2007. 128 с.

124. Гришко Л.В. Про формування професійних якостей майбутнього програміста в процесі навчання основ програмування. *Вісник черкаського університету*, 2011. Вип. 194. С. 129-136.
125. Психология и педагогика / Под. ред. А.А. Бодалева, В.И. Жукова, Л.Г. Лаптева, В.А. Слостенина. М.: Изд-во Института Психотерапии, 2002. 585 с.
126. Радченко В. В. Менторство в системі післядипломної освіти лікаря. *Медична освіта*. 2019. № 1 С. 116-120.
127. Развитие субъекта образования: проблемы, подходы, методы исследования / Под ред. Е.Д. Божович. М.: ПЕРСЭ, 2005. 400 с.
128. Рантюк І.І., Вакалюк Т.А. Світовий і вітчизняний досвід використання інформаційно-комунікаційних технологій у неформальній освіті з управління проектами співробітників ІТ-компаній. *Інформаційно-комунікаційні технології в освіті*. 2019. Вип.16. Т. 2. С. 172-177.
129. Рейкін В .С. Консалтинг в Україні: оцінка стану та тенденції розвитку. URL: [http://market-infr.od.ua/journals/2020/39\\_2020\\_ukr/18.pdf](http://market-infr.od.ua/journals/2020/39_2020_ukr/18.pdf) (дата звернення 03.12.20).
130. Рейтинг университетов мира по подготовке программистов. URL: <https://www.topuniversities.com/university-rankings/university-subject-rankings/2019/computer-science-information-systems> (дата звернення 03.12.20).
131. Рекомендации по преподаванию программной инженерии и информатики в университетах. Software Engineering 2004: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering Computing Curricula 2004: Computer Science: пер. с англ. М.:ИНТУИТ.РУ «Интернет-Университет Информационных Технологий», 2007. 462 с.
132. Рекомендации по преподаванию программной инженерии и информатики в университетах. Software Engineering 2014: Curriculum

- Guidelines for Undergraduate Degree Programs in Software Engineering Computing Curricula 2014. URL: <https://www.acm.org/binaries/content/assets/education/se2014.pdf> (дата звернення 03.12.20).
133. Рекомендації ради Європи щодо ключових компетентностей для освіти впродовж життя від 17.01.2018р. URL: <https://ec.europa.eu/education/sites/education/files/recommendation-key-competences-lifelong-learning.pdf> (дата звернення: 15.01.2021).
134. Рекомендації ради про визнання неформального й інформального навчання. URL: [http://ipq.org.ua/upload/files/files/06\\_Biblioteka/01\\_Normativna\\_baza/01\\_Viznznya\\_neformalnogo\\_navchannya/01\\_Mignarodni\\_dokumenti/council\\_recommendations\\_20\\_dec\\_2012\\_ukr.pdf](http://ipq.org.ua/upload/files/files/06_Biblioteka/01_Normativna_baza/01_Viznznya_neformalnogo_navchannya/01_Mignarodni_dokumenti/council_recommendations_20_dec_2012_ukr.pdf) (дата звернення 03.12.20).
135. Рівний рівному. Навчальний посібник. К.: Інша освіта, 2015. 57 с.
136. Розвиток Української ІТ-індустрії. Аналітичний звіт. URL: [https://ko.com.ua/files/u125/Ukrainian\\_IT\\_Industry\\_Report\\_UKR.pdf](https://ko.com.ua/files/u125/Ukrainian_IT_Industry_Report_UKR.pdf) (дата звернення 03.12.20).
137. Рынок IT-образования: курсы и учебные центры подготовили свыше 30 тыс. студентов в 2015 году. URL: <https://dou.ua/lenta/articles/it-schools-rankings-2015/#shag> (дата звернення 03.12.20).
138. Рынок IT-образования: курсы и учебные центры подготовили свыше 35 тыс. студентов в 2016 году. URL: <https://dou.ua/lenta/articles/it-schools-rankings-2016/#shag> (дата звернення 03.12.20).
139. Савош В.О. Засобово-орієнтовані складники системи неперервної освіти: різноаспектний аналіз формальної, інформальної та неформальної освіти. *Педагогічний альманах*. 2020. Вип. 45. С. 35-41.
140. Сандецька О.О. Проблеми оцінювання освітніх досягнень дорослих у Польщі. *Педагогічний альманах*. 2011. Вип. 12. Ч. 3. С. 175-178.
141. Седов В.Є. Обґрунтування критеріїв та показників сформованості фахової компетентності майбутніх інженерів-програмістів у системі

- магістратури. *Міжнар. наук. журнал «ScienceRice» Pedagogical Education*. 2016. № 8. С. 38-43.
142. Сергієнко В.В. Філософські проблеми наукового пізнання: [навчальний посібник]. Кременчук: Кременчуцький національний університет імені Михайла Остроградського, 2011. 103 с.
143. Сікора Я.Б., Якимчук Б.Л. Модель підготовки майбутнього вчителя інформатики на основі принципів дуальної освіти. *Наукові записки. Серія: Педагогічні науки*. Вип. 177, Ч. II. С. 88-91.
144. Сеницын С.В., Налютин Н.Ю. Верификация программного обеспечения. М.: БИНОМ, 2008. 368 с.
145. Ситжанова А.М., Лабужская Т.И. «Менторство» как одна из форм индивидуальной траектории при обучении кадров. Методические проблемы развития практико-ориентированного обучения на управленческих и экономических направлениях подготовки. URL: [https://conference.osu.ru/assets/files/conf\\_info/conf14/s4.pdf](https://conference.osu.ru/assets/files/conf_info/conf14/s4.pdf) (дата звернення 03.12.20).
146. Сидоренко Е.В. Тренинг коммуникативной компетентности в деловом взаимодействии. Спб., 2004. 208 с.
147. Сидоренко Е.В. Технологии тренерской работы. М.: Университет «Синергия». 2018. 320 с.
148. Слободянюк І.А. Тренінг партнерського спілкування. К., 2010. 48 с.
149. Словник-довідник з професійної педагогіки / За ред. А.В. Семенової. Одеса: Пальміра, 2006. 221 с.
150. Советский энциклопедический словарь / Гл.ред. А.М. Прохоров; рекол.: А.А.Гусев и др. Изд. 4-е. М.: Сов. Энциклопедия, 1987. 1600 с.
151. Спиридонов В.Ф., Кукушкина Ю.А. Критическое мышление как фактор профессиональной компетентности программистов, *Психология. Журнал высшей школы экономики*. 2008. Т.5. №1. С. 165-174.

152. Стандарт вищої освіти України. Рівень вищої освіти: перший (бакалаврський), галузь знань 12 – Інформаційні технології, спеціальність – 121-Інженерія програмного забезпечення. Наказ МОН №1166 від 29.10.2018. URL: <https://mon.gov.ua/storage/app/media/vishcha-osvita/zatverdzeni%20standarty/12/21/121-inzheneriya-programnogo-zabezpechennya-bakalavr.pdf> (дата звернення 03.12.20).
153. Стандарт вищої освіти України. Рівень вищої освіти: другий (магістерський), галузь знань 12 – Інформаційні технології, спеціальність – 121-Інженерія програмного забезпечення. Наказ МОН №1424 від 17.11.2020. URL: <https://mon.gov.ua/ua/osvita/visha-osvita/naukovo-metodichna-rada-ministerstva-osviti-i-nauki-ukrayini/zatverdzeni-standarti-vishoyi-osviti> (дата звернення 03.12.20).
154. Сухоняк С. Сутність та особливості аутсорсингу. *Економічний часопис Східноєвропейського національного університету імені Лесі Українки*. 2016. №2. С. 73-79. URL: [http://nbuv.gov.ua/UJRN/echscenu\\_2016\\_2\\_15](http://nbuv.gov.ua/UJRN/echscenu_2016_2_15).
155. Тейзе Є. Українські ІТ-стартапи: п'ять історій успіху. URL: <https://www.dw.com/uk/українські-іт-стартапи-пять-історій-успіху/a-50847383> (дата звернення 03.12.20).
156. Теренко О. Північноамериканський досвід визнання результатів неформальної та інформальної освіти дорослих. *Педагогічні науки: теорія, історія, інноваційні технології*. 2018. №9 (83). С. 48-58.
157. Терьохіна Н.О. Неформальна освіта як важлива складова системи освіти дорослих. *Порівняльно-педагогічні студії*. 2014. № 2-3 (20-21). С. 109-114.
158. Терьохіна Н.О. Особливості розвитку неформальної освіти дорослих в Україні. *Педагогічні науки: теорія, історія, інноваційні технології*. 2013. № 3(29).

159. Терьохіна Н.О. Педагогічне осмислення неформальної освіти як складової системи освіти дорослих. URL: [https://repository.sspu.edu.ua/bitstream/123456789/9004/1/terokhina\\_Pedahohichne.pdf](https://repository.sspu.edu.ua/bitstream/123456789/9004/1/terokhina_Pedahohichne.pdf) (дата звернення 03.12.20).
160. Терьохіна Н.О. Розвиток неформальної освіти дорослих в Україні <http://repo.snau.edu.ua/bitstream/123456789/4994/1/%D0%A2%D0%B5%D1%80%D1%8C%D0%BE%D1%85%D1%96%D0%BD%D0%B0%20%D0%9D.%20%D0%9E.%20%D0%A0%D0%BE%D0%B7%D0%B2%D0%B8%D1%82%D0%BE%D0%BA%20%D0%BD%D0%B5%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D1%97%20%D0%BE%D1%81%D0%B2%D1%96%D1%82%D0%B8.pdf>. режим (дата звернення 03.12.20).
161. Терюханова І.М., Дрозач М.І., Стульпінас Н.К. Проблеми та перспективи розвитку наставництва в Україні. *SOCIOПРОСТІР: міждисциплінарний електронний збірник наукових праць з соціології та соціальної роботи*. 2017. №6. С. 31-38.
162. Технології роботи організаційних психологів: Навч.посіб. для студентів вищ.навч.закл. та слухачів ін.післядипл.освіти / За наук. ред., Л.М. Карамушки. К: ІНКОС, 2005. 366 с.
163. Тонгуш В.В., Роцектаева У.Ю. Наставничество как один из методов обучения и развития персонала. *Международный научный журнал «Инновационная наука»*. 2017. №04-2. С. 190-192
164. Управление персоналом организации. Практикум: Учебное пособие /Под ред.А.Я. Кибанова. М.: ИНФРА-М, 2012.
165. Холодная М.А. Психология интеллекта. Парадоксы исследования. СПб: Питер, 2002. 272 с.
166. Цибизова Т.Ю., Августан О.М., Сергеев Д.А., Марданов С.А. «Менторство» как элемент методики работы преподавателя в системе высшего профессионального образования. Современные

- проблемы науки и образования. 2017. № 4. URL: <http://www.science-education.ru/ru/article/view?id=26646> (дата обращения: 30.12.2020).
167. Чутчева Е.Б. Собеседование: подготовка и проведение. URL: <http://www.psychology.ru/library/649> (дата звернення 03.12.20).
168. Шапочкіна О.В. Сучасні тенденції розвитку неформальної освіти майбутніх учителів у Німеччині. Автореф. дис. ... канд. пед. наук : спец. 13.00.04 - теорія і методика професійної освіти. Київський університет імені Бориса Грінченка, 2012. 20 с.
169. Шульдова С.Г. Международные стандарты, проблемы и перспективы подготовки ИТ-специалистов. *Инновационные образовательные технологии*. 2016. №4(48). С. 31-37.
170. Шустак Ю.І. Неформальна освіта дорослих у нормативно-правовому полі України. *Наукові записки Бердянського державного педагогічного університету Серія: Педагогічні науки*. 2016. Вип. 2. 270 с. С. 233-239.
171. Щедролосьєв Д.Є. Компетентнісний підхід до підготовки інженерів-програмістів. *Інформаційні технології і засоби навчання*. 2011. №4. (24). URL: [http://www.nbuv.gov.ua/old\\_jrn/e-journals/ITZN/2011\\_4/11schdey.pdf](http://www.nbuv.gov.ua/old_jrn/e-journals/ITZN/2011_4/11schdey.pdf) (дата звернення 03.12.20).
172. Щедролосьєв Д.Є. Особливості підготовки ІТ фахівців в українських вищих навчальних закладах. *Комп'ютер у школі та сім'ї*. 2010. №8. С. 12-15.
173. Щербатенко О. ІТ-галузь в Україні: path of dependence. URL: <https://blog.liga.net/user/asherbatenko/article/29045> (дата звернення 03.12.20).
174. An improved Assessment of Personality Traits in Software Engineering/ A. S. Sodiya, H. O. Longe, L. O. Omatosho. 2007.
175. Beck K., Beedle M., Bennekum A., Cockburn A., Cunningham W., Fowler M., Grenning J., Highsmith J., Hunt A., Jeffries R., Kern J., Marick B., Martin R., Mellor S., Schwaber K., Sutherland J, Thomas D.

- (2001). Manifesto for Agile Software Development. URL: <http://www.agilemanifesto.org/> (дата звернення: 30.06.2019).
176. Burnstein I. Practical Software Testing. A process-oriented approach. Springer-Verlag, New York, 2003. 732 p
177. Cal Poly. College of Engineering. Overview. URL: <http://catalog.calpoly.edu/collegesandprograms/collegeofengineering> (дата звернення 03.12.20).
178. Cal Poly. Computer Science and Software Engineering. URL: <http://catalog.calpoly.edu/collegesandprograms/collegeofengineering/computersciencesoftwareengineering/#courseinventory> (дата звернення 03.12.20).
179. Chao Georgia T. Formal Mentoring: Lessons Learned From Past Practice. *Professional Psychology: Research and Practice*. 2009, Vol. 40, No. 3, 314-320.
180. Cinque M. Lost in translation. Soft skills development in European countries. *Tuning Journal*. 2016. 3 (2). 389-427.
181. Crispin L., Gregory J.: Agile testing. Addison-Wesley, Upper Saddle River. 2014.
182. Cusumano, M., Selby, R.: Microsoft secrets. Touchstone Book, New York 1998.
183. Deakin University. Bachelor of Software Engineering (Honours): program description. URL: [https://www.deakin.edu.au/course/bachelor-software-engineering-honours#tab\\_1--2](https://www.deakin.edu.au/course/bachelor-software-engineering-honours#tab_1--2) (дата звернення 03.12.20).
184. Deakin University. SIT314 - Software Architecture and Scalability for Internet-Of-Things. Unit details. URL: <https://www.deakin.edu.au/courses-search/unit.php?unit=SIT314> (дата звернення 03.12.20).
185. Deakin University. SIT707 - Software Quality and Testing. Unit details. URL: <https://www.deakin.edu.au/courses-search/unit.php?unit=SIT707> (дата звернення 03.12.20).

186. Dr. Karla Satchwell Mentoring Literature Review 55 с. August 2006  
URL: <http://www.humanservices.alberta.ca/documents/mentoring-literature-review.pdf> (дата звернення 03.12.20).
187. E-Mentoring in Global Software Development Teams: Success Factors to Develop a Common Culture. URL: [https://www.researchgate.net/publication/259148685\\_E-Mentoring\\_in\\_Global\\_Software\\_Development\\_Teams\\_Success\\_Factors\\_to\\_Develop\\_a\\_Common\\_Culture#fullTextFileContent](https://www.researchgate.net/publication/259148685_E-Mentoring_in_Global_Software_Development_Teams_Success_Factors_to_Develop_a_Common_Culture#fullTextFileContent) (дата звернення 03.12.20).
188. Estonian University of applied science. Software development and entrepreneurship. URL: [https://www.euas.eu/study%20programs/program/software\\_development\\_and\\_entrepreneurship](https://www.euas.eu/study%20programs/program/software_development_and_entrepreneurship) (дата звернення 03.12.20).
189. Estonian University of applied science. Software testing (SE-018). URL: [https://www.euas.eu/study%20programs/program/software\\_development\\_and\\_entrepreneurship?design=aineprogramm&file=ainekaarteng.php&id=SE-018](https://www.euas.eu/study%20programs/program/software_development_and_entrepreneurship?design=aineprogramm&file=ainekaarteng.php&id=SE-018) (дата звернення 03.12.20).
190. Faheem Ahmed, Luiz Fernando Capretz, Salah Bouktif, Piers Campbell. Soft Skills and Software Development: A Reflection from the Software Industry. *International Journal of Information Processing and Management*. 2015. 4(3):171-191.
191. Freeman J. Mentoring gifted pupils: an international view. *Educating Able Children*. 2001. №5. P. 6-12. URL: [http://www.joanfreeman.com/pdf/Mentoring\\_Educ\\_Able\\_01.pdf](http://www.joanfreeman.com/pdf/Mentoring_Educ_Able_01.pdf). (дата звернення 03.12.20).
192. Glenford J. Myers. The Art of Software Testing, Second Edition. Business Data Processing: A Wiley Series (Book 28), Hoboken, New Jersey. 2012.
193. Gura Jr. O., Gura O., Gura T., Chernikova L. Research on metacognitive skills of software testers: a problem statement. Proceedings of the 16th International Conference on ICT in Education, Research and Industrial

- Applications. Integration, Harmonization and Knowledge Transfer. Volume II: Workshops. Part IV: V International Workshop on Professional Retraining and Life-Long Learning using ICT: Person-oriented Approach (3L-Person). Kharkiv, Ukraine, October 06-10, 2020, 607-618.
194. Haselberger D., Oberhuemer P., Pérez E., Cinque M., Capasso F.D. (2016). Mediating Soft Skills at Higher Education Institutions (ModEs project: Lifelong Learning Programme. 2012), 67, accessed March, 31.
  195. How to Reduce the Cost of Software Testing / ed. : M. Heusser, G. Kulkarni. Boca Raton; London; New York : Taylor & Francis Group, 2012. 340 p.
  196. Jia S. and Yang C. Teaching software testing based on CDIO. World Trans. on Engng. and Technol. Educ, 2013. 1, 4, 476-479.
  197. Kenton W. Research and Development (R&D). URL: <https://www.investopedia.com/terms/r/randd.asp> (дата звернення 03.12.20).
  198. Kirsten M Poulsen Mentoring programmes: Learning opportunities for mentees, for mentors, for organisations and for society. URL: [https://www.researchgate.net/publication/263601999\\_Mentoring\\_programmes\\_Learning\\_opportunities\\_for\\_mentees\\_for\\_mentors\\_for\\_organisations\\_and\\_for\\_society](https://www.researchgate.net/publication/263601999_Mentoring_programmes_Learning_opportunities_for_mentees_for_mentors_for_organisations_and_for_society) (дата звернення 03.12.20).
  199. KTH Royal Institute of Technology. Bachelor's Programme in Information and Communication. URL: <https://www.kth.se/student/kurser/program/TCOMK-20202.pdf?l=en> (дата звернення 03.12.20).
  200. KTH Royal Institute of Technology. Courses Part of Programme. URL: <https://www.kth.se/student/kurser/kurser-inom-program?l=en> (дата звернення 03.12.20).
  201. Lytvynova S. Organization of Independent Studying of Future Bachelors in Computer Science within Higher Education Institutions of Ukraine.

- Proceedings of the 14th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer. 2018. Vol-2104. P. 348-358. URL: [http://ceur-ws.org/Vol-2104/paper\\_160.pdf](http://ceur-ws.org/Vol-2104/paper_160.pdf) (дата звернення 03.12.20).
202. Lewis W., Dobbs D., Veerapillai G. Software testing and continuous quality improvement. 2008.
203. Main Academy. URL: <https://mainacademy.ua/> (дата звернення 03.12.20).
204. Michigan Tech. Departmental Courses. URL: [https://www.banweb.mtu.edu/pls/owa/studev.stu\\_ctg\\_utils.p\\_display\\_classes\\_facbio?ps\\_department=CS&PS\\_STYLE\\_DEPT=computing&ps\\_level=UG&ps\\_faculty=all](https://www.banweb.mtu.edu/pls/owa/studev.stu_ctg_utils.p_display_classes_facbio?ps_department=CS&PS_STYLE_DEPT=computing&ps_level=UG&ps_faculty=all) (дата звернення 03.12.20).
205. Michigan Tech. The Enterprise program. BS Computer Science-Computer Science SCS2. URL: <https://www.mtu.edu/computing/undergraduate/computer-science/computer-science-flowchart.pdf> (дата звернення 03.12.20).
206. Michigan Tech. The Enterprise program. URL: <https://www.mtu.edu/enterprise/involved/students/> (дата звернення 03.12.20).
207. Nanyang Technological University. 4-year BEng (CS) Programme with Professional Internship (PI). URL: <http://scse.ntu.edu.sg/Programmes/CurrentStudents/Undergraduate/Documents/2020/CS/AY2021%20SCSE%20CS.pdf> (дата звернення 03.12.20).
208. Nanyang Technological University. Double Degree in Business. URL: <http://scse.ntu.edu.sg/Programmes/CurrentStudents/Undergraduate/Pages/nbsdd.aspx> (дата звернення 03.12.20).
209. Postlethwaite B.E., Schaffer R.H. Creating a Mentoring Program That Works. *Human Resources*. 2019. VOLUME 22. ISSUE 3. URL:

- <https://gbr.pepperdine.edu/2019/12/creating-a-mentoring-program-that-works/> (дата звернення 03.12.20).
210. Raitskaya L.K., Tikhonova. E.V. (2018). Perceptions of Soft Skills by Russia's University Lecturers and Students in the Context of the World Experience. *RUDN Journal of Psychology and Pedagogics*, 15 (3), 350-363.
  211. Robinson J. Mentoring program. Guidance and Program Plan. U.S. Department of Energy Office of Learning and Workforce Development. URL:  
[https://www.energy.gov/sites/prod/files/2014/11/f19/DOE\\_Mentoring%20Guidance%20%20Pgrm%20Plan2\\_0.pdf](https://www.energy.gov/sites/prod/files/2014/11/f19/DOE_Mentoring%20Guidance%20%20Pgrm%20Plan2_0.pdf) (дата звернення 03.12.20).
  212. Schraw G., Dennison R.S. Assessing metacognitive awareness. *Contemporary Educational Psychology*, 1994. 19 (4), 460-475.
  213. Schwaber K., Sutherland J. The Scrum Guide. URL:  
<https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf>. (дата звернення 03.12.20).
  214. Tech ecosystem guide for Ukraine 2019. URL: [https://data.unit.city/tech-guide/Tech\\_Ecosystem\\_Guide\\_To\\_Ukraine\\_En-1.1.pdf](https://data.unit.city/tech-guide/Tech_Ecosystem_Guide_To_Ukraine_En-1.1.pdf) (дата звернення 03.12.20).
  215. The interconnection of learning ability and the organization of metacognitive processes and traits of personality/ Karpov Alexander A., Karpov Anatoliy V., Karabushchenko Natalia B., Ivashchenko Aleksandr V. *Psychology in Russia: State of the Art*. Vol. 10, Issue 1, 2017.
  216. Tinoco-Giraldo Harold, Eva María Torrecilla Sánchez, Francisco José García-Peñalvo. E-Mentoring in Higher Education: A Structured Literature Review and Implications for Future Research. *Sustainability*. 2020, 12.
  217. Trainer E.H., Kalyanasundaram A., Herbsleb J.D. E-Mentoring for Software Engineering: A Socio-technical Perspective. *Engineering, Computer Science 2017, IEEE/ACM 39th International Conference on*

- Software Engineering: Software Engineering Education and Training Track (ICSE-SEET). URL: <https://www.semanticscholar.org/paper/E-Mentoring-for-Software-Engineering%3A-A-Perspective-Trainer-Kalyanasundaram/6752449bc351b883c0240f0b6236870b44c42819> (дата звернення 03.12.20).
218. Wallack M. 5 Companies Showcasing Successful Mentorship Programs. URL: <https://www.gloat.com/blog/5-companies-showcasing-successful-mentorship-programs/> (дата звернення 03.12.20).
219. Weinberg G. M. Perfect Software And Other Illusions about Testing / G. M. Weinberg [S. 1.] : Dorset House, 2011. 200 p.
220. Xiaofeng Wang., Juan Garbajosa., Ademar Aguiar.: Agile Processes in Software Engineering and Extreme Programming: 19th International Conference, XP 2018, Porto, Portugal, May 21-25, 2018, Proceedings. Springer International Publishing. 2018.
221. Yarkova T. A., Cherkasova I. I. 2016. Forming Soft Skills of Students in the Implementation of Professional Standard of a Teacher. *Tyumen State University Herald. Humanities Research. Humanitates*. Vol. 2, no 4, pp. 222-234.

# ДОДАТКИ

## Додаток А

**ЗАТВЕРДЖЕНО**

Наказ Міністерства  
освіти і науки України

29.10.2018 № 1166

### СТАНДАРТ ВИЩОЇ ОСВІТИ УКРАЇНИ

**РІВЕНЬ ВИЩОЇ ОСВІТИ** Перший (бакалаврський) рівень  
( назва рівня вищої освіти )

**СТУПІНЬ ВИЩОЇ ОСВІТИ** Бакалавр  
( назва ступеня вищої освіти )

**ГАЛУЗЬ ЗНАНЬ** 12 – Інформаційні технології  
( шифр та назва галузі знань )

**СПЕЦІАЛЬНІСТЬ** 121 – Інженерія програмного забезпечення  
( код та найменування спеціальності )

*Видання офіційне*

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**Київ  
2018**

## I Прембула

Стандарт вищої освіти України: перший (бакалаврський) рівень, галузь знань 12 – Інформаційні технології, спеціальність 121 – Інженерія програмного забезпечення.

Затверджено і введено в дію наказом Міністерства освіти і науки України від 29.10.2018 № 1166

Стандарт розроблено членами підкомісії зі спеціальності 121 «Інженерія програмного забезпечення» Науково-методичною комісією 8 з інформаційних технологій, автоматизації та телекомунікацій сектору вищої освіти Науково-методичної ради Міністерства освіти і науки України:

Нікітченко Микола Степанович, <i>голова підкомісії</i> Дудар Зоя Володимирівна, <i>заступник голови підкомісії</i>	доктор фізико-математичних наук, завідувач кафедри теорії та технології програмування Київського національного університету імені Тараса Шевченка; кандидат технічних наук, завідувач кафедри програмної інженерії Харківського національного університету радіоелектроніки;
Глибовець Андрій Миколайович, <i>секретар підкомісії</i> Байбуз Олег Григорович	кандидат фізико-математичних наук, доцент кафедри мережних технологій Національного університету «Києво-Могилянська академія»; доктор технічних наук, завідувач кафедри математичного забезпечення ЕОМ Дніпропетровського національного університету імені Олеся Гончара;
Ворочек Ольга Григорівна	кандидат технічних наук, доцент кафедри програмної інженерії Харківського національного університету радіоелектроніки;
Дивак Микола Петрович	доктор технічних наук, декан факультету комп'ютерних технологій Тернопільського національного економічного університету
Дичка Іван Андрійович	доктор технічних наук, декан факультету прикладної математики Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського»;
Омельчук Людмила Леонідівна	кандидат фізико-математичних наук, доцент кафедри теорії та технології програмування Київського національного університету імені Тараса Шевченка;
Петрик Михайло Романович	доктор фізико-математичних наук, завідувач кафедри програмної інженерії Тернопільського національного технічного університету імені Івана Пулюя;
Сидоров Микола Олександрович	доктор технічних наук, завідувач кафедри інженерії програмного забезпечення Національного авіаційного університету;

Ткачук Микола Вячеславович	доктор технічних наук, професор кафедри програмної інженерії та інформаційних технологій управління Національного технічного університету «Харківський політехнічний інститут»;
Туркін Ігор Борисович	доктор технічних наук, завідувач кафедри інженерії програмного забезпечення Національного аерокосмічного університету імені М. Є. Жуковського «Харківський авіаційний інститут»;
Шинкаренко Віктор Іванович	доктор технічних наук, завідувач кафедри комп'ютерних інформаційних технологій Дніпропетровського національного університету залізничного транспорту імені академіка В. Лазаряна.

Стандарт розглянуто та схвалено на засіданні підкомісії зі спеціальності 121 «Інженерія програмного забезпечення» Науково-методичної комісії № 8 з інформаційних технологій, автоматизації та телекомунікацій Науково-методичної ради Міністерства освіти і науки України (протокол № 3 від 06.12.2016 р.).

Стандарт розглянуто на засіданні сектору вищої освіти Науково-методичної ради Міністерства освіти і науки України (протокол від 22.11.2016 р. № 9).

Фахову експертизу проводили :

Кобозєва Алла Анатоліївна	доктор технічних наук, професор, завідувач кафедри інформатики та управління захистом інформаційних систем Одеського національного політехнічного університету;
Яковина Віталій Степанович	доктор технічних наук, професор, завідувач кафедри програмного забезпечення Національного університету «Львівська політехніка».
Єршов Сергій Володимирович	Учений секретар Інституту кібернетики імені В. М. Глушкова НАН України, доктор-фізико-математичних наук, старший науковий співробітник.

Методичну експертизу проводили:

Захарченко Вадим Миколайович	доктор технічних наук, професор, проректор з науково-педагогічної роботи Національного університету «Одеська морська академія»;
Золотарьова Ірина Олександрівна	кандидат економічних наук, доцент, професор кафедри інформаційних систем Харківського національного економічного університету імені Семена Кузнеця;
Калашнікова Світлана Андріївна	доктор педагогічних наук, професор, директор Інституту вищої освіти НАПН України;
Таланова Жаннета Василівна	доктор педагогічних наук, доцент, старший науковий співробітник, менеджер з аналітичної роботи

Національного Еразмус+ офісу в Україні.

Стандарт розглянуто Державною службою спеціального зв'язку та захисту інформації України та Федерацією роботодавців України.

Стандарт розглянуто після надходження всіх зауважень та пропозицій та схвалено на засіданні підкомісії зі спеціальності 121 «Інженерія програмного забезпечення» Науково-методичної комісії № 8 з інформаційних технологій, автоматизації та телекомунікацій Науково-методичної ради Міністерства освіти і науки України (протокол № 5 від 14.05.2018 р.).

## II Загальна характеристика

Рівень вищої освіти	Перший (бакалаврський) рівень
Ступінь вищої освіти	Бакалавр
Галузь знань	12 – Інформаційні технології
Спеціальність	121 – Інженерія програмного забезпечення
Обмеження щодо форм навчання	Обмеження відсутні
Освітня кваліфікація	Бакалавр з інженерії програмного забезпечення за спеціалізацією (зазначити назву спеціалізації за наявності)
Кваліфікація в дипломі	Ступінь вищої освіти – Бакалавр Спеціальність – 121 Інженерія програмного забезпечення Спеціалізація – (зазначити назву спеціалізації за наявності) Освітня програма – (зазначити назву)
Опис предметної області	<i>Об'єкт:</i> програмне забезпечення, процеси, інструментальні засоби та ресурси розробки, супроводження та забезпечення якості програмного забезпечення. <i>Ціль навчання:</i> підготовка фахівців, здатних ставити і розв'язувати завдання, що пов'язані з розробкою, супроводженням та забезпеченням якості програмного забезпечення. <i>Теоретичний зміст предметної області:</i> базові математичні, інформаційні, фізичні, економічні положення щодо створення і супроводження програмного забезпечення; основи доменного аналізу, моделювання, проектування, конструювання, супроводження програмного забезпечення. <i>Методи, методика та технології:</i> методи та технології розробки програмного забезпечення; збирання, обробки та інтерпретації результатів досліджень з інженерії програмного забезпечення. <i>Інструменти та обладнання:</i> програмно-апаратні та інструментальні засоби розробки, супроводження та експлуатації програмного забезпечення.

Академічні права випускників	Можливість навчання за програмою другого (магістерського) рівня вищої освіти. Набуття додаткових кваліфікацій в системі післядипломної освіти.
------------------------------	--

### **III Обсяг кредитів ЄКТС, необхідний для здобуття першого (бакалаврського) ступеня вищої освіти:**

- на базі повної загальної середньої освіти становить 240 кредитів ЄКТС;
- на базі ступеня «молодший бакалавр» (освітньо-кваліфікаційного рівня «молодший спеціаліст») заклад вищої освіти має право визнати та перезарахувати не більше ніж 60 кредитів ЄКТС, отриманих в межах попередньої освітньої програми підготовки молодшого бакалавра (молодшого спеціаліста) за спеціальністю 121 «Інженерія програмного забезпечення» та не більше ніж 30 кредитів ЄКТС, отриманих в межах попередньої освітньої програми підготовки молодшого бакалавра (молодшого спеціаліста) за іншими спеціальностями.

Мінімум 50 % обсягу освітньої програми має бути спрямовано на забезпечення загальних та спеціальних (фахових) компетентностей за спеціальністю, визначених цим Стандартом вищої освіти.

### **IV Перелік компетентностей випускника**

Інтегральна компетентність	Здатність розв'язувати складні спеціалізовані завдання або практичні проблеми інженерії програмного забезпечення, що характеризуються комплексністю та невизначеністю умов, із застосуванням теорій та методів інформаційних технологій.
Загальні компетентності	<p>K01. Здатність до абстрактного мислення, аналізу та синтезу.</p> <p>K02. Здатність застосовувати знання у практичних ситуаціях.</p> <p>K03. Здатність спілкуватися державною мовою як усно, так і письмово.</p> <p>K04. Здатність спілкуватися іноземною мовою як усно, так і письмово.</p> <p>K05. Здатність вчитися і оволодівати сучасними знаннями.</p> <p>K06. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.</p> <p>K07. Здатність працювати в команді.</p> <p>K08. Здатність діяти на основі етичних міркувань.</p> <p>K09. Прагнення до збереження навколишнього середовища.</p> <p>K10. Здатність діяти соціально відповідально та свідомо.</p> <p>K11. Здатність реалізувати свої права і обов'язки як члена суспільства, усвідомлювати цінності громадянського (вільного демократичного) суспільства та необхідність його сталого розвитку, верховенства права, прав і свобод людини і громадянина в Україні.</p> <p>K12. Здатність зберігати та примножувати моральні, культурні, наукові цінності і досягнення суспільства на</p>

	<p>основі розуміння історії та закономірностей розвитку предметної області, її місця у загальній системі знань про природу і суспільство та у розвитку суспільства, техніки і технологій, використовувати різні види та форми рухової активності для активного відпочинку та ведення здорового способу життя.</p>
<p>Спеціальні (фахові, предметні) компетентності</p>	<p>K13. Здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення.</p> <p>K14. Здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування.</p> <p>K15. Здатність розробляти архітектури, модулі та компоненти програмних систем.</p> <p>K16. Здатність формулювати та забезпечувати вимоги щодо якості програмного забезпечення у відповідності з вимогами замовника, технічним завданням та стандартами.</p> <p>K17. Здатність дотримуватися специфікацій, стандартів, правил і рекомендацій в професійній галузі при реалізації процесів життєвого циклу.</p> <p>K18. Здатність аналізувати, вибирати і застосовувати методи і засоби для забезпечення інформаційної безпеки (в тому числі кібербезпеки).</p> <p>K19. Володіння знаннями про інформаційні моделі даних, здатність створювати програмне забезпечення для зберігання, видобування та опрацювання даних.</p> <p>K20. Здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення.</p> <p>K21. Здатність оцінювати і враховувати економічні, соціальні, технологічні та екологічні чинники, що впливають на сферу професійної діяльності.</p> <p>K22. Здатність накопичувати, обробляти та систематизувати професійні знання щодо створення і супроводження програмного забезпечення та визнання важливості навчання протягом всього життя.</p> <p>K23. Здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення.</p> <p>K24. Здатність здійснювати процес інтеграції системи, застосовувати стандарти і процедури управління змінами для підтримки цілісності, загальної функціональності і надійності програмного забезпечення.</p> <p>K25. Здатність обґрунтовано обирати та освоювати</p>

	інструментарій з розробки та супроводження програмного забезпечення. К26. Здатність до алгоритмічного та логічного мислення.
--	---

**V Нормативний зміст підготовки здобувачів вищої освіти, сформульований у термінах результатів навчання**

ПР01.	Аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки.
ПР02.	Знати кодекс професійної етики, розуміти соціальну значимість та культурні аспекти інженерії програмного забезпечення і дотримуватись їх в професійній діяльності.
ПР03.	Знати основні процеси, фази та ітерації життєвого циклу програмного забезпечення.
ПР04.	Знати і застосовувати професійні стандарти і інші нормативно-правові документи в галузі інженерії програмного забезпечення.
ПР05.	Знати і застосовувати відповідні математичні поняття, методи доменного, системного і об'єктно-орієнтованого аналізу та математичного моделювання для розробки програмного забезпечення.
ПР06.	Уміння вибирати та використовувати відповідну задачі методологію створення програмного забезпечення.
ПР07.	Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення.
ПР08.	Вміти розробляти людино-машинний інтерфейс.
ПР09.	Знати та вміти використовувати методи та засоби збору, формулювання та аналізу вимог до програмного забезпечення.
ПР10.	Проводити передпроектне обстеження предметної області, системний аналіз об'єкта проектування.
ПР11.	Вибирати вихідні дані для проектування, керуючись формальними методами опису вимог та моделювання.
ПР12.	Застосовувати на практиці ефективні підходи щодо проектування програмного забезпечення.
ПР13.	Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань.
ПР14.	Застосовувати на практиці інструментальні програмні засоби доменного аналізу, проектування, тестування, візуалізації, вимірювань та документування програмного забезпечення.
ПР15.	Мотивовано обирати мови програмування та технології розробки для розв'язання завдань створення і супроводження програмного забезпечення.

ПР16.	Мати навички командної розробки, погодження, оформлення і випуску всіх видів програмної документації.
ПР17.	Вміти застосовувати методи компонентної розробки програмного забезпечення.
ПР18.	Знати та вміти застосовувати інформаційні технології обробки, зберігання та передачі даних.
ПР19.	Знати та вміти застосовувати методи верифікації та валідації програмного забезпечення.
ПР20.	Знати підходи щодо оцінки та забезпечення якості програмного забезпечення.
ПР21.	Знати, аналізувати, вибирати, кваліфіковано застосовувати засоби забезпечення інформаційної безпеки (в тому числі кібербезпеки) і цілісності даних відповідно до розв'язуваних прикладних завдань та створюваних програмних систем.
ПР22.	Знати та вміти застосовувати методи та засоби управління проектами.
ПР23.	Вміти документувати та презентувати результати розробки програмного забезпечення.
ПР24.	Вміти проводити розрахунок економічної ефективності програмних систем.

## VI Форми атестації здобувачів вищої освіти

Форми атестації здобувачів вищої освіти	Атестація здійснюється у формі публічного захисту кваліфікаційної роботи.
Вимоги до кваліфікаційної роботи	<p>Кваліфікаційна робота передбачає розв'язання спеціалізованого завдання або практичної задачі інженерії програмного забезпечення, що характеризуються комплексністю та невизначеністю умов, із застосуванням теорій та методів інформаційних технологій.</p> <p>У кваліфікаційній роботі не може бути академічного плагіату, фальсифікації та списування.</p> <p>Кваліфікаційна робота має бути оприлюднена на офіційному сайті закладу вищої освіти або його підрозділу, або у репозитарії закладу вищої освіти.</p> <p>Оприлюднення кваліфікаційних робіт, що містять інформацію з обмеженим доступом, здійснювати у відповідності до вимог чинного законодавства.</p>

## VII Вимоги до наявності системи внутрішнього забезпечення якості вищої освіти

У закладі вищої освіти повинна функціонувати система забезпечення закладом вищої освіти якості освітньої діяльності та якості вищої освіти

(система внутрішнього забезпечення якості), яка передбачає здійснення таких процедур і заходів:

- 1) визначення принципів та процедур забезпечення якості вищої освіти;
- 2) здійснення моніторингу та періодичного перегляду освітніх програм;
- 3) щорічне оцінювання здобувачів вищої освіти, науково-педагогічних і педагогічних працівників закладу вищої освіти та регулярне оприлюднення результатів таких оцінювань на офіційному веб-сайті закладу вищої освіти, на інформаційних стендах та в будь-який інший спосіб;
- 4) забезпечення підвищення кваліфікації педагогічних, наукових і науково-педагогічних працівників;
- 5) забезпечення наявності необхідних ресурсів для організації освітнього процесу, у тому числі самостійної роботи студентів, за кожною освітньою програмою;
- 6) забезпечення наявності інформаційних систем для ефективного управління освітнім процесом;
- 7) забезпечення публічності інформації про освітні програми, ступені вищої освіти та кваліфікації;
- 8) забезпечення дотримання академічної доброчесності працівниками закладів вищої освіти та здобувачами вищої освіти, у тому числі створення і забезпечення функціонування ефективного системи запобігання та виявлення академічного плагіату;
- 9) інших процедур і заходів.

Система забезпечення закладом вищої освіти якості освітньої діяльності та якості вищої освіти (система внутрішнього забезпечення якості) за поданням закладу вищої освіти оцінюється Національним агентством із забезпечення якості вищої освіти або акредитованими ним незалежними установами оцінювання та забезпечення якості вищої освіти на предмет її відповідності вимогам до системи забезпечення якості вищої освіти, що затверджуються Національним агентством.

## **VIII Перелік нормативних документів, на яких базується Стандарт вищої освіти**

- Закон України від 01.07.2014 р. № 1556-VII «Про вищу освіту» [Режим доступу: <http://zakon5.rada.gov.ua/laws/show/2145-19>];
- Закон України від 05.09.2017 р. «Про освіту» – [Режим доступу: <http://zakon5.rada.gov.ua/laws/show/2145-19>];
- Постанова Кабінету Міністрів України від 29.04.2015 р. № 266 «Про затвердження переліку галузей знань і спеціальностей, за якими здійснюється підготовка здобувачів вищої освіти» [Режим доступу: <http://zakon4.rada.gov.ua/laws/show/266-2015-п>];
- Постанова Кабінету Міністрів України від 30.12.2015 р. № 1187 «Про затвердження Ліцензійних умов провадження освітньої діяльності» [Режим доступу: <http://zakon4.rada.gov.ua/laws/show/1187-2015-п/page>];
- Постанова Кабінету Міністрів України від 23.11.2011 р. № 1341 «Про

затвердження Національної рамки кваліфікацій» [Режим доступу: <http://zakon4.rada.gov.ua/laws/show/1341-2011-п>];

- Національний класифікатор України: «Класифікація видів економічної діяльності» ДК 009: 2010 [Режим доступу: <http://www.ukrstat.gov.ua/>];

- Національний класифікатор України: «Класифікатор професій» ДК 003: 2010 ДК 003:2010 [Режим доступу: <http://www.dk003.com/>];

-

### **Інші рекомендовані джерела**

- Стандарти і рекомендації щодо забезпечення якості в Європейському просторі вищої освіти (ESG) [Режим доступу: [http://ihed.org.ua/images/doc/04\\_2016\\_ESG\\_2015.pdf](http://ihed.org.ua/images/doc/04_2016_ESG_2015.pdf)];

- International Standard Classification of Education (ISCED 2011): UNESCO Institute for Statistics [Режим доступу: <http://www.uis.unesco.org/education/documents/isced-2011-en.pdf>];

- ISCED Fields of Education and Training 2013 (ISCED-F 2013): UNESCO Institute for Statistics [Режим доступу: <http://www.uis.unesco.org/Education/Documents/isced-fields-of-education-training-2013.pdf>].

- Методичні рекомендації щодо розроблення стандартів вищої освіти, затверджені наказом Міністерства освіти і науки України від 01.06.2017 р. № 600 (у редакції наказу Міністерства освіти і науки України від 21.12.2017 р. № 1648), схвалені сектором вищої освіти Науково-методичної Ради Міністерства освіти і науки України (протокол від 29.03.2016 № 3);

- Розроблення освітніх програм. Методичні рекомендації [Режим доступу: [http://ihed.org.ua/images/doc/04\\_2016\\_rozroblennya\\_osv\\_program\\_2014\\_tempus-office.pdf](http://ihed.org.ua/images/doc/04_2016_rozroblennya_osv_program_2014_tempus-office.pdf)];

- Національний освітній глосарій: вища освіта [Режим доступу: [http://ihed.org.ua/images/doc/04\\_2016\\_glossariy\\_Visha\\_osvita\\_2014\\_tempus-office.pdf](http://ihed.org.ua/images/doc/04_2016_glossariy_Visha_osvita_2014_tempus-office.pdf)];

- Розвиток системи забезпечення якості вищої освіти в Україні: інформаційно-аналітичний огляд [Режим доступу: [http://ihed.org.ua/images/doc/04\\_2016\\_Rozvitok\\_sisitemi\\_zabesp\\_yakosti\\_VO\\_UA\\_2015.pdf](http://ihed.org.ua/images/doc/04_2016_Rozvitok_sisitemi_zabesp_yakosti_VO_UA_2015.pdf)];

- Європейська кредитна трансферна накопичувальна система: Довідник користувача [Режим доступу: [http://ihed.org.ua/images/doc/04\\_2016\\_ECTS\\_Users\\_Guide-2015\\_Ukrainian.pdf](http://ihed.org.ua/images/doc/04_2016_ECTS_Users_Guide-2015_Ukrainian.pdf)].

- EQF-LLL – European Qualifications Framework for Lifelong Learning [Режим доступу: [https://ec.europa.eu/ploteus/sites/eac-eqf/files/brochexp\\_en.pdf](https://ec.europa.eu/ploteus/sites/eac-eqf/files/brochexp_en.pdf)];

- QF-EHEA – Qualification Framework of the European Higher Education Area [Режим доступу: <http://www.ehea.info/article-details.aspx?ArticleId=67>];

- TUNING (для ознайомлення зі спеціальними (фаховими) компетентностями та прикладами стандартів [Режим доступу: <http://www.unideusto.org/tuningeu/>].

## **Пояснювальна записка**

Стандарт вищої освіти містить компетентності, що визначають специфіку підготовки бакалаврів зі спеціальності 121 – Інженерія програмного забезпечення та результати навчання, які виражають що саме студент повинен знати, розуміти та бути здатним виконувати після успішного завершення освітньої програми. Вони узгоджені між собою та відповідають дескрипторам Національної рамки кваліфікацій. Таблиця 1 показує відповідність визначених Стандартом компетентностей та дескрипторів НРК. В таблиці 2 показана відповідність програмних результатів навчання та компетентностей.

Заклад вищої освіти самостійно визначає перелік дисциплін, практик та інших видів освітньої діяльності, необхідний для набуття означених Стандартом компетентностей.

Наведений в Стандарті перелік компетентностей і результатів навчання не є вичерпним. Заклади вищої освіти при формуванні профілю освітніх програм можуть вказувати додаткові компетентності і програмні результати навчання.

Заклад вищої освіти має право вводити додаткові форми атестації здобувачів вищої освіти.

Таблиця 1.

## Матриця відповідності визначених Стандартом компетентностей дескрипторам НРК

<b>Класифікація компетентностей за НРК</b>	<b>Знання</b> <b>Зн1</b> Концептуальні знання, набуті у процесі навчання та професійної діяльності, включаючи певні знання сучасних досягнень <b>Зн2</b> Критичне осмислення основних теорій, принципів, методів і понять у навчанні та професійній діяльності	<b>Уміння</b> <b>Ум1</b> Розв'язання складних непередбачуваних задач і проблем у спеціалізованих сферах професійної діяльності та/або навчання, що передбачає збирання та інтерпретацію інформації (даних), вибір методів та інструментальних засобів, застосування інноваційних підходів	<b>Комунікація</b> <b>К1</b> Донесення до фахівців і нефахівців інформації, ідей, проблем, рішень та власного досвіду в галузі професійної діяльності <b>К2</b> Здатність ефективно формувати комунікаційну стратегію	<b>Автономія та відповідальність</b> <b>АВ1</b> Управління комплексними діями або проектами, відповідальність за прийняття рішень у непередбачуваних умовах <b>АВ2</b> Відповідальність за професійний розвиток окремих осіб та/або груп осіб <b>АВ3</b> Здатність до подальшого навчання з високим рівнем автономності
<b>Загальні компетентності</b>				
К01. Здатність до абстрактного мислення, аналізу та синтезу.		<b>Ум1</b>		
К02. Здатність застосовувати знання у практичних ситуаціях.	<b>Зн1</b>	<b>Ум1</b>		
К03. Здатність спілкуватися державною мовою як усно, так і письмово.		<b>Ум1</b>	<b>К2</b>	
К04. Здатність спілкуватися іноземною мовою мовою як усно, так і письмово.		<b>Ум1</b>	<b>К2</b>	
К05. Здатність вчитися і оволодівати сучасними знаннями.		<b>Ум1</b>		<b>АВ3</b>

К06. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.		<b>Ум1</b>		
К07. Здатність працювати в команді.			<b>К1</b>	<b>АВ1</b>
К08. Здатність діяти на основі етичних міркувань.			<b>К2</b>	<b>АВ2</b>
К09. Прагнення до збереження навколишнього середовища.			<b>К1</b>	<b>АВ2</b>
К10. Здатність діяти соціально відповідально та свідомо.			<b>К1</b>	<b>АВ2</b>
К11. Здатність реалізувати свої права і обов'язки як члена суспільства, усвідомлювати цінності громадянського (вільного демократичного) суспільства та необхідність його сталого розвитку, верховенства права, прав і свобод людини і громадянина в Україні.			<b>К1</b>	
К12. Здатність зберігати та примножувати моральні, культурні, наукові цінності і досягнення суспільства на основі розуміння історії та закономірностей розвитку предметної області, її місця у загальній системі знань про природу і суспільство та у розвитку суспільства, техніки і технологій, використовувати			<b>К2</b>	<b>АВ2</b>

різні види та форми рухової активності для активного відпочинку та ведення здорового способу життя.				
<b>Спеціальні (фахові) компетентності</b>				
К13. Здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення.		<b>Ум1</b>		<b>АВ1</b>
К14. Здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування.		<b>Ум1</b>		
К15. Здатність розробляти архітектуру, модулі та компоненти програмних систем.		<b>Ум1</b>		<b>АВ1</b>
К16. Здатність формулювати та забезпечувати вимоги щодо якості програмного забезпечення у відповідності з вимогами замовника, технічним завданням та стандартами.		<b>Ум1</b>	<b>К1</b>	
К17. Здатність дотримуватися специфікацій, стандартів, правил і рекомендацій в професійній галузі при реалізації процесів життєвого циклу.	<b>Зн1</b>			<b>АВ1</b>

K18. Здатність аналізувати, вибирати і застосовувати методи і засоби для забезпечення інформаційної безпеки (в тому числі кібербезпеки).	<b>Зн2</b>	<b>Ум1</b>		
K19. Володіння знаннями про інформаційні моделі даних та системи, здатність створювати програмне забезпечення для зберігання, видобування та опрацювання даних.	<b>Зн1</b>	<b>Ум1</b>		
K20. Здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення.	<b>Зн2</b>	<b>Ум1</b>		
K21. Здатність оцінювати і враховувати економічні, соціальні, технологічні та екологічні чинники, що впливають на сферу професійної діяльності.		<b>Ум1</b>		<b>АВ1</b>
K22. Здатність накопичувати, обробляти та систематизувати професійні знання щодо створення і супроводження програмного забезпечення та визнання важливості навчання протягом всього життя.	<b>Зн1</b>			
K23. Здатність реалізовувати фази та ітерації життєвого		<b>Ум1</b>		<b>АВ1</b>

циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення.				
K24. Здатність здійснювати процес інтеграції системи, застосовувати стандарти і процедури управління змінами для підтримки цілісності, загальної функціональності і надійності програмного забезпечення.		<b>Ум1</b>		
K25. Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення.		<b>Ум1</b>		<b>АВ1</b>
K26. Здатність до алгоритмічного та логічного мислення	<b>Зн1</b>	<b>Ум1</b>		

Таблиця 2.

Матриця відповідності визначених Стандартом результатів навчання та компетентностей

Програмні результати навчання		Інтегральна компетентність	Компетентності																								
			Загальні компетентності										Спеціальні (фахові) компетентності														
			K01	K02	K03	K04	K05	K06	K07	K08	K09	K10	K11	K12	K13	K14	K15	K16	K17	K18	K19	K20	K21	K22	K23	K24	K25
<b>ПРО 1</b>	Аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки.	Здатність розв'язувати складні спеціалізовані завдання або практичні проблеми інженерії програмного забезпечення, що характеризуються комплексністю та невизначеністю умов, із			+	+	+	+													+		+				
<b>ПРО 2</b>	Знати кодекс професійної етики, розуміти соціальну значимість та культурні аспекти інженерії програмного забезпечення і дотримуватись їх в професійній			+						+	+	+	+	+				+					+				











<b>2</b>	застосовувати методи та засоби управління проектами.																		
<b>ПР2 3</b>	Вміти документувати та презентувати результати розробки програмного забезпечення.			+	+														
<b>ПР2 4</b>	Вміти проводити розрахунок економічної ефективності програмних систем.		+								+								

## Додаток Б

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ  
ІМЕНІ СЕМЕНА КУЗНЕЦЯ

УХВАЛЕНО  
Рішення вченої ради  
ХНЕУ ім. С. Кузнеця  
Протокол № 6 від 31.01.2020 р.



ВВЕДЕНО В ДІЮ

Наказ ректора ХНЕУ ім. С. Кузнеця  
№ 31 від 31.01.2020 р.

В. С. Пономаренко

ПОЛОЖЕННЯ ПРО ПОРЯДОК ВИЗНАННЯ РЕЗУЛЬТАТІВ  
НЕФОРМАЛЬНОЇ ТА ІНФОРМАЛЬНОЇ ОСВІТИ У  
ХАРКІВСЬКОМУ НАЦІОНАЛЬНОМУ ЕКОНОМІЧНОМУ  
УНІВЕРСИТЕТІ ІМЕНІ СЕМЕНА КУЗНЕЦЯ

Харків

2020

## 1. Загальні положення

1.1. Положення про порядок визнання результатів навчання, отриманих у неформальній та інформальній освіті в Харківському національному економічному університеті ім. С. Кузнеця розроблено відповідно до вимог Законів України «Про вищу освіту», «Про освіту».

1.2. Результати навчання та компетентності, необхідні для присудження освітніх та/або присвоєння професійних кваліфікацій, можуть досягатися та здобуватися у системі формальної, неформальної чи інформальної освіти.

1.3. Формальна освіта - це освіта, яка здобувається за освітніми програмами відповідно до визначених законодавством рівнів освіти, галузей знань, спеціальностей (професій) і передбачає досягнення здобувачами освіти визначених стандартами освіти результатів навчання відповідного рівня освіти та здобуття кваліфікацій, що визнаються державою.

1.4. Неформальна освіта - це освіта, яка здобувається, як правило, за освітніми програмами та не передбачає присудження визнаних державою освітніх кваліфікацій за рівнями освіти, але може завершуватися присвоєнням професійних та/або присудженням часткових освітніх кваліфікацій.

1.5. Інформальна освіта (самоосвіта) - це освіта, яка передбачає самоорганізоване здобуття особою певних компетентностей, зокрема під час повсякденної діяльності, пов'язаної з професійною, громадською або іншою діяльністю, родиною чи дозвіллям.

1.6. Відповідно до ст. 8 Закону України «Про освіту» результати навчання, здобуті шляхом неформальної освіти та / або інформальної, визнаються в системі формальної освіти в порядку, визначеному законодавством.

1.7. До формування національним агентством кваліфікацій вимог до процедур визнання результатів неформального (інформального) навчання Університет має право самостійно запроваджувати відповідні процедури у межах своєї автономії згідно з частиною третьою статті 2 Закону України «Про освіту».

1.8. Право на визнання результатів навчання у неформальній або інформальній освіті поширюється на здобувачів усіх рівнів вищої освіти.

## **2. Порядок визнання результатів навчання здобутих у неформальній та інформальній освіті**

2.1. Визнання результатів навчання, які здобуті у неформальній або інформальній освіті дозволяється для дисциплін, які починають викладатися з другого семестру першого року підготовки в Університеті за певним рівнем освіти. При цьому визнання результатів проводиться у семестрі, який передусє семестру, у якому згідно з навчальним планом конкретної освітньої програми передбачено вивчення дисципліни, що перезараховується. Обмеження зроблено з врахуванням ймовірності здобувача вищої освіти не підтвердити свої результати навчання у неформальній та інформальній освіті.

2.2. Визнання результатів навчання здобутих у неформальній та інформальній освіті поширюється на базові (обов'язкові) та вибіркові дисципліни навчального плану за освітньою програмою.

2.3. Університет може визнати результати навчання здобуті у неформальній чи інформальній освіті в обсязі, що, як правило, не перевищує 10% від загального обсягу кредитів передбачених освітньою програмою.

2.4. Дане Положення поширюється на випадки, коли отриманні у неформальній або інформальній освіті підтверджуючі документи не визначають обсяг відпрацьованих кредитів (годин) та/або результати оцінювання рівня здобутих компетентностей.

2.5. Визнання Університетом результатів навчання, які здобуті у неформальній чи інформальній освіті передбачає:

2.5.1. Подання заяви.

2.5.1.1. Здобувач вищої освіти звертається із заявою до ректора Університету з проханням про визнання результатів навчання, здобутих у неформальній або інформальній освіті.

2.5.1.2. У заяві зазначається: яких саме результатів навчання набув здобувач вищої освіти у інформальній або неформальній освіті, в який саме спосіб він це зробив (хто надав йому освітні, професійні послуги), яке навчальне навантаження (види та форми активності, кількість годин тощо) при цьому він виконав, на визнання результатів навчання з якої саме дисципліни навчального плану за освітньою програмою Університету здобувач вищої освіти претендує.

2.5.1.3. До заяви можуть додаватися будь-які документи (сертифікати, свідоцтва, освітні програми тощо), які підтверджують ті результати навчання, які здобувач вищої освіти отримав.

2.5.2. Формування предметної комісії.

2.5.2.1. За поданням заступника керівника (проректора з науково-педагогічної роботи) згідно з розподілом функціональних обов'язків

видається наказ ректора, яким визначається:

склад предметної комісії, що визначає можливість визнання результатів навчання, здобутих у неформальній чи інформальній освіті;

форми та строки проведення атестації здобувача вищої освіти для визнання результатів навчання набутих у неформальній або інформальній освіті.

2.5.2.2. До предметної комісії мають входити: заступник керівника (проректор з науково-педагогічної роботи) згідно з розподілом функціональних обов'язків; декан факультету, за яким закріплений здобувач вищої освіти; завідувач кафедри, за якою закріплений здобувач вищої освіти; гарант освітньої програми, на якій навчається здобувач вищої освіти; науково-педагогічні працівники, які викладають дисципліни, що пропонуються до перезарахування на основі визнання результатів навчання у неформальній або інформальній освіті; представники наукового товариства студентів, аспірантів, докторантів і молодих вчених Університету (тільки для аспірантів), представники молодіжної організації (для студентів).

### 2.5.3. Атестація.

2.5.3.1. Предметна комісія визначає метод оцінювання результатів навчання відповідно до навчального плану. Здобувача вищої освіти ознайомлюють з програмою навчальної дисципліни та переліком питань, які виносяться на підсумкове оцінювання. Якщо навчальним планом передбачено виконання письмової роботи по даній дисципліні (курсової роботи, есе, реферату, тощо), то здобувача вищої освіти ознайомлюють з переліком тем письмової роботи. Також здобувача вищої освіти ознайомлюють з критеріями оцінювання та правилами оскарження результатів.

2.5.3.2. Предметна комісія дає 10 робочих днів для підготовки здобувача вищої освіти до підсумкового контролю (з кожної дисципліни) та 20 робочих днів для написання письмової роботи (за наявності).

2.5.3.3. Підсумковий контроль проходить у вигляді екзамену у відповідності до «Положення про проведення письмових екзаменів у Харківському національному економічному університеті імені Семена Кузнеця». Екзаменаційний білет та критерії оцінювання формуються викладачами кафедри, які викладають зазначену дисципліну та затверджуються протоколом засідання кафедри, де працюють викладачі. Як правило, використовуються білети, що розроблені для здобувачів вищої освіти, що вивчають дану дисципліну(и) в рамках навчального плану. Критерії оцінювання формуються виходячи зі 100 балів. Екзаменаційний білет та критерії оцінки, підписані викладачами, що викладають дисципліну та завідувачем кафедри, де працюють викладачі, погоджують з заступником

керівника (проректором з науково-педагогічної роботи).

2.5.3.4. Комісія виставляє підсумкову оцінку за шкалою ЄКТС. Якщо здобувач вищої освіти отримав менше 60 балів, то йому не зараховуються результати навчання у неформальній або інформальній освіті.

2.5.4. Прийняття рішення.

2.5.4.1. За підсумками роботи предметна комісія формує протокол про зарахування чи не зарахування відповідної дисципліни (із зазначенням назви дисципліни, загальної кількості годин/кредитів, оцінки та підстави щодо перезарахування).

2.5.4.2. Рішення предметної комісії доводиться до відома ректора, після чого видається відповідний наказ, на підставі якого здобувачу вищої освіти перезараховується вказана дисципліна(и), вносяться данні до індивідуального навчального плану, відомостей успішності, академічної довідки.

2.5.4.3. Після визнання набутих здобувачем вищої освіти результатів навчання у неформальній або інформальній освіті, він звільняється від вивчення перезарахованої дисципліни у наступному семестрі підготовки в Університеті.

2.5.5. Оскарження результатів.

2.5.5.1. У разі негативного висновку предметної комісії щодо визнання результатів навчання здобувач вищої освіти має право звернутися з заявою про апеляцію до ректора Університету. Ректор створює наказом апеляційну комісію у складі заступника керівника (проректора з науково-педагогічної діяльності) та науково-педагогічних працівників, які не входили до предметної комісії.

2.5.5.2. Апеляційна комісія за результатами розгляду скарги приймає обґрунтоване рішення про повне або часткове задоволення скарги чи про залишення поданої скарги без задоволення.

### **3. Прикінцеві положення**

3.1. Університет інформує Міністерство освіти і науки України про всі випадки трансферу кредитів у випадку визнання результатів у неформальній або інформальній освіті в обсязі понад 30 кредитів.

3.2. Положення вводиться в дію наказом ректора Університету.

3.3. Усі зміни та доповнення до цього Положення розглядаються та ухвалюються на засіданні вченої ради Університету та вводяться в дію наказом ректора.

## Додаток В

### Програма авторського курсу «Основи тестування програмного забезпечення»

#### 1. Мета і завдання спецкурсу

Метою викладення курсу є формування у студентів системи спеціалізованих теоретичних, організаційних і технічних знань в сфері розробки та тестування програмного забезпечення.

Основними завданнями вивчення курсу є:

- розкриття сутності тестування програмного забезпечення, а також технічних і організаційних аспектів його впровадження в процес розробки;
- сприяння професійному самовизначенню студентів-програмістів за рахунок визначення організаційної та процесуальної ролі тестувальника програмного забезпечення в команді розробки;
- формування стійких знань щодо теорії тестування програмного забезпечення, його структурних складових, виробничих практик, звітної документації.

В результаті вивчення даного курсу студент повинен знати:

- зміст базових понять дисципліни: *тестування, контроль якості, вимоги, методологія розробки, планування, естімація, тест-план, тест-стратегія, тест-кейс, чек-лист, дефект, звіт, автоматизація, фреймворк;*
- теоретичні основи тестування програмного забезпечення, класифікацію його форм і видів;
- поняття методології розробки програмного забезпечення, їх класифікація та найбільш популярні представники;
- сутність тест-дизайну, техніки написання тестових сценаріїв;
- сутність, структуру, правила оформлення і контекст використання основних документів тестування;
- організаційні аспекти планування процесу тестування і естімації робочих завдань;
- поняття автоматизації, технічні та структурні особливості її впровадження в процес тестування.

Вміти:

- розкривати сутність основних понять і концептуальних положень дисципліни;
- аналізувати і тестувати вимоги до програмного забезпечення;
- застосовувати і аргументувати вибір технік тест-дизайну;
- створювати, оформлювати та реалізовувати тестові сценарії;
- застосовувати техніки естімації і оцінювати трудовитрати завдань з тестування;

- створювати, оформлювати та аналізувати звіти про дефекти і тестуванні;

- аргументувати вибір інструментів для автоматизації тестування.

Міждисциплінарні зв'язки з програмою підготовки студентів у закладах вищої освіти: основи алгоритмів, математика, об'єктно-орієнтоване програмування, вступ до тестування, основи управління проєктами, архітектура і проєктування.

## 2. Структура навчальної дисципліни

Назви тематичних розділів і тем	Кількість годин			
	усього	у тому числі		
		л	пр.	сам. р.
Модуль 1. Вступ до тестування програмного забезпечення	6	2	-	4
Модуль 2. Тестування в контексті сучасних методологій розробки	16	4	4	8
Модуль 3. Тест-дизайн	16	4	4	8
Модуль 4. Планування і естимація	16	4	4	8
Модуль 5. Дефекти і звітність	8	2	2	4
Модуль 6. Автоматизація тестування	16	4	4	8
Фінальний проєкт	12	-	-	12
<b>Усього годин</b>	<b>90</b>	<b>20</b>	<b>18</b>	<b>52</b>

## 3. Програма навчальної дисципліни

### Модуль 1. Вступ до тестування програмного забезпечення

Визначення понять якості, контролю якості, тестування. Історія появи галузі. Основні передумови виділення тестування програмного забезпечення в окремий процес. Цілі і завдання тестування.

### Модуль 2. Тестування в контексті сучасних методологій розробки

Визначення методології розробки програмного забезпечення. Передумови виникнення підходів до організації процесу розробки. Класифікація сучасних методологій розробки програмного забезпечення. Роль і особливості етапів тестування в основних представників.

### Модуль 3. Тест-дизайн

Поняття тест-дизайну. Еволюція формалізації процесу тестування. Види і типи тестування, функціональне / нефункціональне тестування, види нефункціонального тестування. Поняття «чек лист». Техніки складання чек-листів. Поняття «тест кейс». Особливості написання тест-кейсів. Техніки тест-дизайну.

### Модуль 4. Планування і естимація

Визначення понять планування і естимації в контексті сучасних методологій розробки програмного забезпечення. Техніки та підходи до планування і естимації. Формалізація планування процесу тестування. Тест-план і тест-стратегія.

### Модуль 5. Дефекти і звітність

Поняття дефекту. Види дефектів. Життєвий цикл дефекту в процесі розробки. Звіт про дефект: оформлення, поля, особливості в контексті різних проектів.

#### Модуль 6. Автоматизація тестування

Поняття автоматизації тестування. Місце автоматизації в процесі розробки. Види і підходи автоматизації тестування. Огляд інструментів автоматизації тестування. Поняття «фреймворк». Підходи до написання фреймворків для автоматизації тестування. Особливості створення інфраструктури для автоматизації.

#### **4. Темі лекційних занять**

Модуль	Назва лекції	Обсяг
Вступ до тестування програмного забезпечення	1. Вступ до тестування програмного забезпечення	2
Тестування в контексті сучасних методологій розробки	2. Тестування в контексті сучасних методологій розробки	2
	3. Робота з вимогами	2
Тест-дизайн	4. Види і типи тестування	2
	5. Техніки тест дизайну	2
Планування та естимація	6. Планування. Тест план і тест стратегія	2
	7. Техніки естимації	2
Дефекти та звітність	8. Дефекти та звітність	2
Автоматизація тестування	9. Введення в автоматизацію тестування. Сторонні інструменти для автоматизації	2
	10. Архітектура фреймворка автоматизації тестування	2
Разом		20

#### **5. Темі практичних занять**

Модуль	Назва теми практичного заняття	Обсяг
Тестування в контексті сучасних методологій розробки	1. Візуалізація сучасних методологій розробки програмного забезпечення і місця тестування в ньому	2
	2. Робота з вимогами	2
Тест-дизайн	3. Види і типи тестування	2
	4. Техніки тест дизайну	2
Планування та естимація	5. Планування. Тест план і тест стратегія	2
	6. Техніки естимації	2
Дефекти та звітність	7. Дефекти та звітність	2
Автоматизація тестування	8. Введення в автоматизацію тестування. Сторонні інструменти для автоматизації	2
	9. Архітектура фреймворка автоматизації тестування	2
Разом		18

#### **6. Індивідуальні завдання для самостійної роботи**

Модуль	Назва теми та завдання	Обсяг
Вступ до тестування програмного забезпечення	<i>Введення в тестування програмного забезпечення</i>	4

	<i>Завдання:</i> Ознайомлення з рекомендованою профільною літературою [1-4]	
Тестування в контексті сучасних методологій розробки	<i>Тестування в контексті сучасних методологій розробки</i> <i>Завдання:</i> Запропонувати не пов'язану з ІТ сферу виробництва або послуг, процес якої може бути представлений в форматі методології Scrum, визначити та опрацювати проектні ролі, описати глобальні процеси, підготувати схему ітерацій роботи команди (відповідно до вимог)	4
	<i>Робота з вимогами</i> <i>Завдання:</i> Надати позитивні та негативні приклади до кожної характеристики вимог (відповідно до вимог)	4
Тест-дизайн	<i>Види і типи тестування</i> <i>Завдання:</i> Створити та оформити приклади сценаріїв тестування програмного продукту згідно з кожним з розглянутих видів тестування. (відповідно до вимог)	4
	<i>Техніки тест дизайну</i> <i>Завдання:</i> підготувати набір тест-кейсів, що описують сценарії тестування функціональності комерційного додатку з умовою використання оптимальних технік тест-дизайну. (відповідно до вимог)	4
Планування та естімація	<i>Планування. Тест план і тест стратегія</i> <i>Завдання:</i> розробити тест-план і тест-стратегію для обраного веб-додатку (відповідно до вимог)	4
	<i>Техніки естімації</i> <i>Завдання:</i> провести командну декомпозицію завдань до самостійної роботи спецкурсу. Фасилітувати процес естімації та планування завдань. (відповідно до вимог)	4
Дефекти та звітність	<i>Дефекти та звітність</i> <i>Завдання:</i> перенести створений в рамках практичного заняття «Тест дизайн» набір тест кейсів в систему тест-менеджменту, виконати тест кейси, сформувані звіт та створити звіти дефектів. (відповідно до вимог)	4
Автоматизація тестування	<i>Введення в автоматизацію тестування.</i> <i>Сторонні інструменти для автоматизації</i>	4

	<i>Завдання:</i> створити набір тестових сценаріїв для веб-сервісу за власним вибором, автоматизувати її за рахунок використання інструменту Postman. (відповідно до вимог)	
	<i>Архітектура фреймворка автоматизації тестування</i> <i>Завдання:</i> Розробити фреймворк з автоматизації сценаріїв для веб-сервісу за власним вибором. (відповідно до вимог)	4
Разом		40

## 7. Вимоги до виконання індивідуальних завдань самостійної роботи

### 1. Тестування в контексті сучасних методологій розробки

#### *Завдання:*

Запропонувати не пов'язану з ІТ сферу виробництва або послуг, процес якої може бути представлений в форматі методології Scrum, визначити та опрацювати проектні ролі, описати глобальні процеси, підготувати схему ітерацій роботи команди.

#### Вимоги:

- вибір не пов'язаного безпосередньо з ІТ бізнес-процесу чи сфери;
- підготування таблиці відповідності командної ролі у методології Scrum і спеціалізації фахівця;
- підготування таблиці відповідності командного процесу у методології Scrum і командній взаємодії/процесу у обраній області, з описом обов'язків кожної ролі;
- підготувати схему ітерацій роботи;

Наприклад:

Предметна галузь: приватна кондитерська

Scrum роль	Роль у процесу
Scrum-майстер	Шеф-кондитер
Замовник	Клієнт, що заздалегідь замовив торт на весілля
...	...

Scrum процес	Виробничий процес	Опис процесу
Sprint planning	Планування закупівлі інгредієнтів, заготовки напівфабрикатів та виконання замовлень	Шеф-кухар презентує перелік замовлень на наступний тиждень та план необхідних обсягів підготування коржів, пастили та крему для тортів. Спеціаліст з закупівлі презентує список інгредієнтів, що попадають під ризик бути відсутніми у магазині протягом

		наступного тижня ...
...	...	...

## **2. Робота з вимогами**

*Завдання:*

Надати позитивні та негативні приклади до кожної характеристики якості вимог до програмного забезпечення.

Вимоги:

- обрати функціонал популярного додатку у відкритому доступі;
- мінімум три приклади до кожної характеристики якості вимоги у форматі таблиці. Наприклад:

Функціональність: коментарі до новини на порталі ukr.net

<b>Характеристика</b>	<b>Негативний приклад</b>	<b>Позитивний приклад</b>
Атомарність	Зареєстрований користувач може лишити коментар до новини та поставити йому «лайк»	1. Зареєстрований користувач може лишити коментар до новини 2. Зареєстрований користувач може поставити «лайк» коментарю до новини
...	...	...

## **3. Види і типи тестування**

*Завдання:*

Створити та оформити приклади сценаріїв тестування програмного продукту згідно з кожним з розглянутих видів тестування.

Вимоги:

- вибір популярного програмного продукту, що знаходиться у відкритому доступі;
- оформлення таблиці, що містить типи тестування та короткий чек-лист відповідно до типу;
- кожен тип тестування повинен включати чек-ліст мінімум з 5 позитивних та 2 негативних (де це можливо) перевірок;

Наприклад:

Продукт: додаток Google Maps для платформи iOS

<b>Тип тестування</b>	<b>Перевірки</b>
Тестування локалізації	1. Після зміни мови на Українську пункти головного меню відображаються українською мовою 2. Після зміни мови на Українську назва збережених маршрутів залишається незмінною ...
...	...

## **4. Техніки тест дизайну**

*Завдання:*

підготувати набір тест-кейсів, що описують сценарії тестування функціональності комерційного додатку з умовою використання оптимальних технік тест-дизайну.

Вимоги:

- цільова функціональність для тестування – форма реєстрації нового користувача інтернет-магазину Rozetka;
- оформлення тест-кейсів у таблиці Excel відповідно до формату, представленого у лекції;
- групування тестів за використаними техніками тест-дизайну з короткою письмовою аргументацією.

### **5. Планування. Тест план і тест стратегія**

*Завдання:*

розробити тест-план для обраного веб-додатку

Вимоги:

- вибір популярного програмного продукту, що знаходиться у відкритому доступі;
- письмовий опис вхідних даних (на розсуд студента): склад команди, доступні технічні ресурси, терміни тестування;
- використання шаблону IEEE 829.

### **6. Техніки естімації**

*Завдання:*

Провести командну декомпозицію завдань до самостійної роботи спецкурсу. Фасилітувати процес естімації та планування завдань, надати командне планування завдань на один тиждень.

Вимоги:

- розбитися на команди від двох до чотирьох студентів;
- рівномірно розподілити завдання зі списку самостійної роботи спецкурсу між членами команди;
- кожний з членів команди декомпозує свої завдання на список атомарних, детально їх описує та готує до процесу естімації;
- на практичному занятті студенти презентують свої завдання, естімують загальний список завдань з використанням техніки Planning poker, та максимально ефективно розподіляють їх на один робочий тиждень при умовах 8-годинного робочого дня та поточного складу команди;
- результати планування презентуються аудиторії та викладачу;

### **7. Дефекти та звітність**

*Завдання:*

перенести створений в рамках практичного заняття «Тест дизайн» набір тест кейсів в систему тест-менеджменту, виконати тест кейси, сформувати звіт та створити звіти дефектів.

Вимоги:

- зареєструвати власний проект у системі qatouch.com, створити інфраструктуру;
- перенести тест-кейси, створенні в завданні 4, включаючи всі поля та ієрархічну структуру;
- додати до списку тест-кейсів один задалегідь невірний тест, що симулюватиме дефект;
- створити Test Run, додати у нього тест кейси для проходження;
- виконати тести, сформувати звіт дефекту та звіт про проходження тестового набору;

### ***8. Вступ до автоматизації тестування. Сторонні інструменти для автоматизації.***

*Завдання:*

створити набір тестових сценаріїв для веб-сервісу за власним вибором, автоматизувати його за рахунок використання інструменту Postman.

Вимоги:

- обраний сервіс повинен бути побудованим за підходом REST або SOAP, знаходитися у відкритому доступі та мати документацію. (Наприклад, <https://www.weatherbit.io>);
- наявність підготовленого **письмово** (у електронному форматі) чек-листа, що покриває **мінімум 5 API запитів**, серед яких є **мінімум 1 параметризований**;
- кожен API запит повинен бути протестованим відповідно до чек-листу та включати перевірки статусу відповіді сервісу та відповідності переданих параметрів отриманим результатам.;
- хост сервісу повинен бути сконфігурований та збережений як Postman Environment.;
- усі запити повинні бути збережені у Postman Collection;
- загальна кількість заавтоматизованих тестів повинна дорівнювати **мінімум 20-ти**;
- до презентації та захисту роботи на практичному занятті повинні бути представлені: чек-лист, postman collection, postman environment с демонстрацією проходження тестів безпосередньо під час презентації роботи.

### ***9. Архітектура фреймворка автоматизації тестування***

*Завдання:*

Розробити фреймворк з автоматизації сценаріїв для веб-сервісу за власним вибором.

Вимоги:

- використання обраного сервісу та створеного чек-листа у завданні 8;
- відповідність API запитів перевіркам з завдання 8;
- вибір будь-якої об'єктно-орієнтованої мови за бажанням студента, за умови наявності в ній інструмента для створення unit-тестів (наприклад Java, Javascript, C#) та http-клієнту;
- наявність окремих функціонального, проміжного та тестового шарів;
- забороняється використовувати готові фреймворки для тестування;
- менеджмент сторонніх бібліотек за рахунок інструменту для побудови проекту відповідно до обраної мови програмування (наприклад, Apache Maven для мови Java);

## **8. Вимоги до виконання фінального проекту**

*Завдання:*

розробити тест-план для обраного веб-додатку, згідно з яким підготувати набір тест-кейсів, що описують сценарії тестування функціональності додатку з умовою використання оптимальних технік тест-дизайну, автоматизувати набір приймальних тестів (смоук-тестів) за рахунок використання інструменту Postman або власноруч розробленого фреймворку.

Вимоги:

- обраний сервіс повинен бути побудованим за підходом REST або SOAP, знаходитися у відкритому доступі та мати документацію. (Наприклад, <https://www.weatherbit.io>);
- обраний сервіс повинен відрізнятися від використаного в завданнях 8 та 9;
- стратегія тестування, використані техніки тест-дизайну, типи тестування та покриття як мануальними, так і автоматизованими тестами повинні відповідати написаному тест-плану;
- набір тест-кейсів повинен бути ієрархічно структурованим та збереженим у системі тест-менеджменту за вибором (наприклад, qatouch.com);
- вибір підходу та інструментів автоматизації повинен бути описаним і аргументованим в тест-плані.

## **9. Перелік питань для фінального інтерв'ю**

1. Що таке тестування?
2. Що таке контроль якості? Відмінність від поняття «тестування».
3. В чому полягає сутність поняття «методологія розробки»?
4. У чому відмінність монолітних від гнучких методологій розробки?
5. На підставі яких ознак можна зробити висновок про те, що методологія Scrum - гнучка?
6. Назвіть командні ролі в методології Scrum.

7. Назвіть основні процесні наради методології Scrum. Опишіть їх призначення і підсумкові артефакти.
8. Назвіть характеристики якості вимоги до програмного забезпечення. Наведіть приклади коректних і некоректних вимог.
9. У чому відмінність функціонального і нефункціонального тестування?
10. Наведіть класифікацію видів нефункціонального тестування.
11. Наведіть класифікацію рівнів тестування.
12. Опишіть основні складові частини тест-кейса.
13. Опишіть особливості техніки тест-дизайну на ваш вибір. Наведіть приклад функціональності, в якій вона буде максимально ефективною.
14. У чому різниця понять «планування» і «естимация»?
15. Назвіть основні техніки командної естимачії завдань.
16. Перелічіть складові частини тест плану. Дайте коротку характеристику кожної частини.
17. Опишіть основні складові звіту про дефект.
18. У яких випадках впровадження автоматизації принесе вигоду?
19. Які перевірки потрібно автоматизувати в першу чергу? Які тести автоматизувати не вигідно?
20. Назвіть переваги і недоліки автоматизації тестування за рахунок самостійно розробленого фреймворку.

### 10. Види контролю і система накопичення балів

Контроль навчальної діяльності здійснюється за допомогою системи оцінювання за 100-бальною шкалою, за результатами поточного і підсумкового контролю.

Співвідношення між поточним і підсумковим контролем у загальній оцінці навчальної діяльності студента з дисципліни становить 60/40:

- максимальна кількість балів за результатами поточного контролю – 60;
- максимальна кількість балів за підсумковий контроль (захист фінального проекту та інтерв'ю) – 40.

№	Тема практичного заняття	Кількість балів
1	Тестування в контексті сучасних методологій розробки	5
2	Робота з вимогами	5
3	Види і типи тестування	5
4	Техніки тест дизайну	5
5	Планування. Тест план і тест стратегія	8
6	Техніки естимачії	8
7	Дефекти та звітність	8
8	Введення в автоматизацію тестування. Сторонні інструменти для автоматизації	8
9	Архітектура фреймворка автоматизації тестування	8
<b>Усього</b>		<b>60</b>

№	Вид роботи	Кількість балів
1	Захист фінального проекту	20
2	Технічне інтерв'ю	20
<b>Усього</b>		<b>40</b>

## 11. Рекомендована література

1. Glenford J. Myers. The Art of Software Testing, Second Edition. John Wiley & Sons, Inc., Hoboken, New Jersey. 2004
2. Куликов, С. С. Тестирование программного обеспечения. Базовый курс / С. С. Куликов. Минск: Четыре четверти, 2020. 312 с.
3. Савин Р. Тестирование Dot Com, или Пособие по жестокому обращению с багами в интернет-стартапах. М.: Дело, 2007. 312 с.
4. Lisa Crispin, Janet Gregory. Agile Testing: A Practical Guide for Testers and Agile Team. -Addison-Wesley, 2009. 533с.
5. Agile-манифест разработки программного обеспечения. URL: <https://agilemanifesto.org/iso/ru/manifesto.html>
6. Essential Scrum: A Practical Guide to the Most Popular Agile Process by Kenneth S. Rubin Addison-Wesley Professional, 2012 452
7. Test Plan Tutorial: A Guide To Write A Software Test Plan Document From Scratch. URL: <https://www.softwaretestinghelp.com/how-to-write-test-plan-document-software-testing-training-day3/>
8. Test plan template (IEE 829). URL: <https://jmpovedar.files.wordpress.com/2014/03/ieee-829.pdf>
9. Postman learning center, Documentation. URL: <https://learning.postman.com/docs/getting-started/introduction/>

**РЕЦЕНЗІЯ**  
**на авторську програму спецкурсу**  
**«Вступ до тестування програмного забезпечення»**  
**для студентів 3-4 курсу спеціальності 121 Програмна інженерія**

Інформаційні технології та розробка програмного забезпечення на сьогоднішній день є, мабуть, однією з найбільш динамічних і креативних сфер сучасного бізнесу, а підготовка висококваліфікованих фахівців індустрії становиться об'єктивною потребою розвитку світової економіки. Однак, зі стрімким розвитком сучасних технологій, задача підготовки молодих інженерів-програмістів стає пропорційно складнішою. Сучасна ІТ-індустрія має властивість стрімко і непередбачувано змінюватися: технології втрачають свою актуальність, виробники програмного забезпечення оновлюють свої продукти, а методології та процеси розробки програмного забезпечення адаптуються під умови світової економіки. Система освіти не встигає швидко адаптуватися до змістовних та організаційних змін індустрії, а пошук альтернативних шляхів підготовки інженерів-програмістів стає актуальною проблемою, що і зумовлює підвищений інтерес до програми, представленої до рецензії.

Однією з головних завдань підготовки студентів до роботи у ІТ індустрії є формування системи спеціалізованих теоретичних, організаційних і технічних знань в сфері розробки та тестування програмного забезпечення, на що й націлена програма теоретичного спецкурсу «Вступ до тестування програмного забезпечення». В рамках викладення матеріалу розглядаються найбільш важливі теоретичні аспекти сфери тестування програмного забезпечення, такі як зміст основних понять дисципліни, теоретичні основи тестування програмного забезпечення, його класифікація та види, організаційні аспекти впровадження процесу тестування, правила і контекст роботи з основними документами, поняття автоматизації, технічні та структурні особливості її впровадження в процес тестування.

Аспект практичного опробування та закріплення матеріалу викладений у програмі виробничої практики, що спрямована на поглиблення теоретичних знань, отриманих під час проходження спецкурсу, а також отримання навичок та вмінь з тестування програмного забезпечення у реальному проектному середовищі.

Актуальність теоретичних матеріалів, оптимальність структури викладення лекцій, доцільність практичних завдань, їх відповідність реальним індустріальним задачам та іноваційність форм контролю і зворотного зв'язку підтверджують високий рівень організації освітнього процесу в умовах програми «Вступ до тестування програмного забезпечення», що дозволяє рекомендувати її до впровадження в систему вищої освіти.

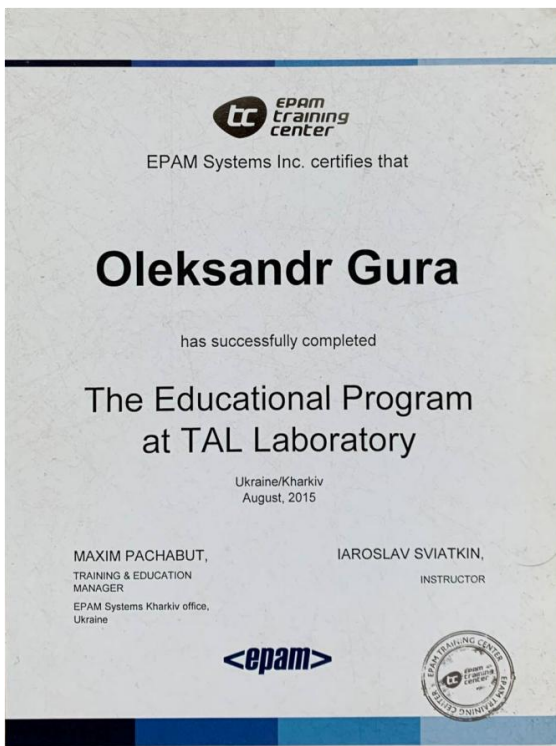
**Генеральний директор**  
**ТОВ «ЕПАМ СИСТЕМЗ»**



С.В. Рожок



Рис. В.1. Учасники авторського курсу «Основи тестування ПЗ»





RD Unit, EPAM Systems Inc. Kharkiv office  
thanks:

# Oleksandr Gura

for his active participation in developing  
young EPAM specialists

**QA Lab**  
As Mentor and Lecturer

Ukraine/Kharkiv  
July, 2017

MAXIM PACHABUT,  
Senior Resource  
Development Manager  
EPAM Systems Kharkiv office,  
Ukraine



RD Unit, EPAM Systems Inc. Kharkiv office  
thanks:

# Oleksandr Gura

for active participation in developing  
young EPAM specialists

**QA Lab**  
As Lecturer

Ukraine/Kharkiv  
September, 2018

MAXIM PACHABUT,  
Resource Development  
Manager  
EPAM Systems Kharkiv office,  
Ukraine



## Додаток Г

### Програма практичної підготовки майбутніх інженерів-програмістів до тестування ПЗ на засадах менторства

#### 1. Мета і завдання практики

**Метою** виробничої педагогічної практики є закріплення та поглиблення теоретичних знань, отриманих під час проходження курсу «Основи тестування програмного забезпечення», а також отримання навичок та вмінь з тестування ПЗ у реальному проєктному середовищі.

Основними **завданнями** виробничої педагогічної практики є:

- поглиблення та закріплення теоретичних знань, отриманих студентами протягом проходження курсу;
- ознайомлення з досвідом використання різноманітних прийомів та методів тестування професіоналами та набуття власного досвіду;
- формування розуміння структури, методології та особливостей проєктного процесу;
- розвиток нетехнічних умінь (навичок командної роботи, професійного спілкування, метакогнітивних та регулятивних умінь).

#### 2. Структура практики

№	Етап	Кількість тижнів	Семестр	Курс
1.	Навчально – адаптаційний етап	8	6 або 8	3-4
2.	Виробничий етап	8	6 або 8	3-4

#### 3. Зміст практики

Виробнича практика на засадах менторства проводиться після проходження курсу «Основи тестування програмного забезпечення» для студентів 3(4) курсу спеціальності 121 «Програмна інженерія» та складається з двох етапів:

- Навчально – адаптаційний етап;
- Виробничий етап.

*Перший етап*, навчально-адаптаційний, передбачає практичну діяльність студентів протягом 8 тижнів, змістом якої є вирішення професійних завдань, побудованих на внутрішній проблематиці ІТ-організації, які більшою мірою мають навчальний характер.

Основними завданнями, які виконують студенти протягом виробничого етапу практики є:

1. Завдання 1 «Адміністрування і робота з системами управління проектами». Створити повну інфраструктуру управління проектом «Менторство» в системі Atlassian JIRA, що включає в себе безпосередній проект, команду розробки (студента і його ментора), документацію проекту, список завдань (відповідно до програми менторства), планування ітерацій (відповідно до розкладу практичних занять) і дошки відстеження активних завдань.

2. Завдання 2 «Тестування сервісного шару веб-додатків». Покрити публічний веб-сервіс прогнозу погоди Weatherbit.io тестовими сценаріями, з подальшим їх проходженням засобами веб-клієнта Postman і формуванням звіту.

3. Завдання 3 «Тестування користувацького інтерфейсу веб-додатків». Створити чек-лист і оформити повний набір тестових сценаріїв для функціональності «кошик» інтернет-магазину rozetka.ua з подальшим їх проходженням і формуванням звіту.

4. Завдання 4 «Тестування мобільних додатків». Підготувати три набори тестових сценаріїв для функціональності «кошик» мобільного інтернет-магазину Rozetka з урахуванням особливостей кожної з трьох платформ: мобільний додаток Rozetka для платформи Android, аналогічний додаток для платформи iOS і мобільну версію веб-сайту, доступну для браузера цільового пристрою. Після складання описаних вище наборів тестових сценаріїв, потрібно пройти кожен з наборів за рахунок емуляції або використання відповідних пристроїв з подальшим формуванням звітів.

5. Завдання 5 «Автоматизація тестування сервісного шару веб-додатків». Створити інфраструктуру для автоматизації веб-сервісу прогнозу погоди Weatherbit.io за рахунок використання об'єктно-орієнтованої мови Java.

6. Завдання 6 «Автоматизація тестування інтерфейсу веб-додатків». Створити інфраструктуру для автоматизації призначеного для користувача інтерфейсу веб-додатків за рахунок використання самостійно розробленого фреймворка з використанням об'єктно-орієнтованого програмування.

7. Завдання 7 «Тестування продуктивності веб-додатків». Створити і розгорнути на локальній машині примітивний веб-додаток засобами об'єктно-орієнтованої мови програмування Java та інструменту SpringBoot, яке складається з одного REST сервісу, а також розробити сценарії і плани навантаження, на підставі чого написати алгоритми для інструменту

тестування продуктивності Apache JMeter з подальшим їх запуском і формуванням звіту.

8. Завдання 8 «Адміністрування і робота з системами безперервної інтеграції і безперервного розгортання». Розгорнути систему Jenkins на локальному комп'ютері, провести її попередню настройку і оптимізацію. Для тестового додатка на технології SpringBoot, розробленого в рамках завдання номер 7, пропонується написати один автоматизований тест засобами інструменту JUnit, після чого створити інфраструктуру його розгортки засобами інструменту Jenkins, з попередньою установкою і налаштуванням останнього.

*Другий етап* практичної підготовки студентів на засадах менторства – виробничий. Він реалізується в реальних умовах професійної діяльності: студент є членом проєктної команди, яка забезпечує реалізацію актуального виробничого проєкту.

Основними завданнями, які виконують студенти протягом виробничого етапу практики є:

- Дослідження форми та методології організації процесу розробки у команді;
- Участь у формальних та неформальних командних заходах, таких як щоденний скрам, планування, ретроспектива та ін.;
- Робота з проєктною та тестовою документацією;
- Виконання задач з тестування;
- Виконання задач з розробки автоматизованих тестів (якщо передбачено стратегією тестування на проєкті).

#### **4. Методичні рекомендації**

##### **4.1 Методичні рекомендації до навчально-адаптаційного етапу**

Перший етап побудований на засадах менторства, що має як індивідуальний, так і груповий характер.

Індивідуальне менторство реалізується системно, як в формальних, так і неформальних умовах за невизначеним («плаваючим») графіком. Частота спілкування та взаємодії ментора і підопічного, більшою мірою залежить від ініціативи останнього, і відбувається за принципом постановки певної проблеми, що виникла в процесі професійної діяльності в організації, яка є актуальною і потребує нагального вирішення.

Групове менторство забезпечують куратори груп, які є визнаними фахівцями з проблематики тієї чи іншої професійної сфери, за якою сформовано перелік завдань, що виконують майбутні фахівці з тестування

на першому етапі навчально-практичної діяльності. Лекція-консультація передбачає наступний сценарій:

- Огляд матеріалу з проблематики, за якою працюють студенти, у тому числі акцент робиться на сформованих завданнях і на тих результатах, які мають бути отримані; аналізуються труднощі, які можуть виникати в процесі роботи;

- Розгляд професійних ускладнень, які ментор (ментори) фіксує на рівні індивідуальної роботи, і висвітлення можливих шляхів їх вирішення;

- Відповіді на питання слухачів. Завершується заняття загальним обговоренням за схемою рефлексії.

#### **4.1.1 Вимоги до виконання завдань навчально-адаптаційного етапу**

1. «Адміністрування і робота з системами управління проектами».

*Завдання:*

Створити повну інфраструктуру управління проектом «Менторство» в системі Atlassian JIRA, що включає в себе безпосередній проект, команду розробки (студента і його ментора), документацію проекту, список завдань (відповідно до програми менторства), планування ітерацій (відповідно до розкладу практичних занять) і дошки відстеження активних завдань.

*Вимоги:*

- Підняття локального серверу JIRA;
- Створення дошки для слідкування за поточними завданнями;
- Занесення всіх завдань даної програми у якості беклогу;
- Заповнення ітерацій завданнями згідно з програмою;
- Створення окремих користувачів для студента та ментора.

2. «Тестування сервісного шару веб-додатків».

*Завдання:*

Покрити публічний веб-сервіс прогнозу погоди Weatherbit.io тестовими сценаріями, з подальшим їх проходженням засобами веб-клієнта Postman і формуванням звіту.

*Вимоги:*

- Наявність чек-листа та узгодження його з ментором до процесу написання тестових сценаріїв;
- Формат та місце збереження тест-кейсів визначаються ментором;
- Postman колекції та конфігурація повинні бути збережені в системі контролю версій.

### 3. «Тестування користувацького інтерфейсу веб-додатків».

#### *Завдання:*

Створити чек-лист і оформити повний набір тестових сценаріїв для функціональності «кошик» інтернет-магазину rozetka.ua з подальшим їх проходженням і формуванням звіту.

#### *Вимоги:*

- Наявність чек-листа та узгодження його з ментором до процесу написання тестових сценаріїв;
- Формат та місце збереження тест-кейсів визначаються ментором.

### 4. «Тестування мобільних додатків».

#### *Завдання:*

Підготувати три набори тестових сценаріїв для функціональності «кошик» мобільного інтернет-магазину Rozetka з урахуванням особливостей кожної з трьох платформ: мобільний додаток Rozetka для платформи Android, аналогічний додаток для платформи iOS і мобільну версію веб-сайту, доступну для браузера цільового пристрою. Після складання описаних вище наборів тестових сценаріїв, потрібно пройти кожен з наборів за рахунок емуляції або використання відповідних пристроїв з подальшим формуванням звітів.

#### *Вимоги:*

- Наявність чек-листа та узгодження його з ментором до процесу написання тестових сценаріїв;
- Формат та місце збереження тест-кейсів визначаються ментором;
- Тестування кожної з платформ повинне бути оформлене як окреме завдання та відображено на дошці.

### 5. «Автоматизація тестування сервісного шару веб-додатків».

#### *Завдання:*

Створити інфраструктуру для автоматизації веб-сервісу прогнозу погоди Weatherbit.io за рахунок використання об'єктно-орієнтованої мови Java.

#### *Вимоги:*

- Створення власного фреймворку для автоматизації тестів;
- Відповідність заавтоматизованих тестів результатам завдання 2;
- Генерація HTML звіту як результат проходження тестів;
- Результати роботи повинні бути збережені в системі контролю версій.

## 6. «Автоматизація тестування інтерфейсу веб-додатків».

### *Завдання:*

Створити інфраструктуру для автоматизації призначеного для користувача інтерфейсу веб-додатків за рахунок використання самостійно розробленого фреймворка з використанням об'єктно-орієнтованого програмування.

### *Вимоги:*

- Створення власного фреймворка для автоматизації тестів;
- Відповідність заавтоматизованих тестів результатам завдання 3;
- Генерація HTML звіту як результат проходження тестів;
- Результати роботи повинні бути збережені в системі контролю версій.

## 7. «Тестування продуктивності веб-додатків».

### *Завдання:*

Створити і розгорнути на локальній машині примітивний веб-додаток засобами об'єктно-орієнтованої мови програмування Java та інструменту SpringBoot, яке складається з одного REST сервісу, а також розробити сценарії і плани навантаження, на підставі чого написати алгоритми для інструменту тестування продуктивності Apache JMeter з подальшим їх запуском і формуванням звіту.

### *Вимоги:*

- Власноручне написання тестового REST-сервісу без використання готових інфраструктур та додатків;
- Тестові сценарії та конфігураційні файли повинні бути збережені в системі контролю версій.

## 8. «Адміністрування і робота з системами безперервної інтеграції і безперервного розгортання».

### *Завдання:*

Розгорнути систему Jenkins на локальному комп'ютері, провести її попередню настройку і оптимізацію. Для тестового додатка на технології SpringBoot, розробленого в рамках завдання номер 7, пропонується написати один автоматизований тест засобами інструменту JUnit, після чого створити інфраструктуру його розгортки засобами інструменту Jenkins, з попередньою установкою і налаштуванням останнього.

### *Вимоги:*

- Створення сценаріїв розгортання за рахунок написання pipeline-скриптів;

- Розміщення pipeline-скриптів у системі контролю версій;
- Отримання актуального коду сервісу з системи контролю версій перед розгортанням
- Ініціація оновлення та розгортання сервісу за розкладом.

#### 4.1.2 Критерії оцінювання завдань навчально-адаптаційного етапу

Контроль навчальної діяльності здійснюється ментором за допомогою системи оцінювання за 100-бальною шкалою, за результатами поточного контролю.

Кожне завдання може бути оцінено ментором максимум у 12,5 балів, що в сумі дозволяє студенту отримати максимум 100 балів наприкінці етапу.

Оцінка включає у себе наступні складові:

- Якість виконання завдання
- Вкладання в терміни виконання завдання
- Якість аргументації використання тих чи інших інструментів та підходів до виконання завдання

	Тема /Завдання	Кількість балів
	Адміністрування і робота з системами управління проектами	12.5
	Тестування сервісного шару веб-додатків	12.5
	Тестування користувацького інтерфейсу веб-додатків	12.5
	Тестування мобільних додатків	12.5
	Автоматизація тестування сервісного шару веб-додатків	12.5
	Автоматизація тестування інтерфейсу веб-додатків	12.5
	Тестування продуктивності веб-додатків	12.5
	Адміністрування і робота з системами безперервної інтеграції і безперервного розгортання	12.5
	<b>Усього</b>	<b>100</b>

#### 4.2 Методичні рекомендації до виробничого етапу

**Другий етап** практичної підготовки студентів на засадах менторства – виробничий. Він реалізується згідно розробленої програми, в реальних умовах професійної діяльності: студент є членом проектної команди, яка забезпечує реалізацію актуального виробничого проекту.

На цьому етапі роботи передбачено менторство в індивідуальній формі, але в двох варіаціях: продовження закріпленням за студентом Ментором індивідуальних консультацій у сталий спосіб та проектно-

спрямоване професійне консультування (консультації за проблематикою проекту), що здійснює лідер команди.

Основними аспектами взаємодії студента з лідером команди є:

- організація процесу впровадження в проектну команду;
- надання структурованої інформації щодо технічних та організаційних аспектів роботи команди;
- постановка проектних завдань згідно з методологією роботи та налагодженими процесами в команді;
- перевірка та аналіз результатів роботи студента;
- консультування та підтримка в процесі виконання проектних завдань та участі у формальних командних заходах;

Метою індивідуальних консультацій, що проводяться з ментором, закріпленим на попередньому етапі практики, є:

- Консультація та підтримка у контексті загальних або технічних аспектів роботи;
- Консультація щодо нетехнічних аспектів роботи, таких як взаємодія з членами проектною командою, звітність.

#### **4.2.1 Критерії оцінювання результатів виробничого етапу**

Оцінювання результатів виробничого етапу відбувається за 100-бальною шкалою на підставі відгука лідера проектною командою за наступними критеріями:

- 90-100 балів («відмінно») виставляється у разі своєчасного виконання студентом повного обсягу проектних завдань з максимальною якістю та відсутністю помилок, активній та конструктивній участі у всіх формальних командних заходах, безперешкодній комунікації з членами команди та висунуті ідей щодо покращення процесу тестування;

- 75-90 балів («добре») передбачає, що всі проектні завдання повністю виконані без суттєвих помилок та можуть бути прийняті без необхідності значних доробок з боку інших членів команди. Студент приймає активну участь у формальних командних заходах та вільно комунікує з членами команди;

- 60-75 балів («задовільно») передбачає, що суттєва частина проектних завдань виконана з помилками та потребує значних виправлень чи консультацій з боку членів команди. Студент приймає пасивну позицію у формальних командних заходах та не завжди здатен налагодити комунікацію з колегами;

- 0-60 балів («незадовільно») ставиться, коли студент не виконав значної частини проєктних завдань, допускає суттєві помилки та не в змозі приймати участь у формальних командних заходах.

### **5. Підведення підсумків практики**

Підсумки практики підводяться згідно з результатами проходження двох етапів за 100-бальною шкалою, де оцінка першого етапу з коефіцієнтом 0,4 додається до оцінки другого етапу з коефіцієнтом 0,6, результуючі у максимум 100 балів.

№	Етап практики	Кількість балів
1	Навчально-адаптаційний	40 (оцінка * 0.4)
2	Виробничий етап	60 (оцінка * 0.6)
<b>Усього</b>		<b>100</b>

**Додаток Е**  
**Програма Тренінгів розвитку нетехнічних умінь майбутніх інженерів-програмістів**

**Додаток Е1**  
**Програма Тренінгу-дизайн мислення**

Тренінг-дизайн мислення (далі - тренінг) є складовою тренінгової програми, спрямованої на розвиток нетехнічних умінь майбутніх тестувальників ПЗ.

Метою тренінга є розвиток у майбутніх інженерів-програмістів таких груп нетехнічних умінь як мисленнєві та метакогнітивні, а саме:

- мисленнєві вміння (аналітичність, оперативність та гнучкість мислення, нестандартно мислити, самостійність мислення);
- метакогнітивні вміння (вміння проблематизації, проєктувальні вміння, планування, прийняття рішення в умовах невизначеності), включаючи рефлексивні вміння (аналізувати, оцінювати власну діяльність, знаходити помилки у неї та коригувати власну мисленнєву діяльність).

Тренінг розрахований на 16 годин, за своєю організаційною структурою складається з трьох основних частин: 1) вступної, 2) основної та 3) завершальної.

**Програма тренінгових занять**

№	Назва та зміст активності	Орієнтовний час
I	Вступна частина	
1	Ситуативне привітання. Розподіл учасників по командам	10 хв.
2	Мета та завдання тренінгу	5 хв.
3	Правила тренінгової роботи: створюємо самі. Презентація правил. Створюємо паркування запитань та ідей.	20 хв.
4	Креативний бейджик: створення бейджика з предметів навколо	15 хв.
5	Я-предмет: презентація себе завдяки таємному предмету. Я – майбутній інженер-програміст	20 хв.
6	Очікування учасників: від індивідуальних до командних очікувань	15 хв.
7	Дорожня карта тренінгу: групова робота	30 хв.
8	Презентація команд учасників: створюємо свій власний логотип. Презентація логотипу команди	20 хв.
9	Що я знаю про своє мислення? Самодіагностика	15 хв.
10	Інформаційний блок: мисленнєва діяльність	10 хв.
11	Завдання для самостійної роботи	5 хв.
II	Основна частина	
1	Ситуативне привітання. Розподіл учасників по командам	10 хв.
2	Рефлексивний аналіз завдання для самостійної роботи	15 хв.
3	Занурення у проблему. Інформаційне повідомлення: рівні	30 хв.

	мисленнєвої діяльності: мисленнєві та метакогнітивні уміння	
4	Командна робота: SMART-технологія в дії	20 хв.
5	K-W-L charts (відповіді на запитання: Що я знаю? Що я хочу узнати? Як я дізнався про це? Що я маю вивчити? Як я маю знайти це? Що я хочу знати ще?)	30 хв.
6	Вправи на зняття напруги, розвиток саморегуляції (австралійський дощ, оркестр та ін.)	20 хв.
7	Розподіл учасників по новим командам	10 хв.
8	Командна робота: аналізуємо кейс «Перший робочий день тестувальника ПЗ»	30 хв.
9	Самодіагностика індивідуальних метакогнітивних стратегій	20 хв.
10	Створення індивідуальної та командної ментальної карти з Лего «Пошук ефективних метакогнітивних стратегій»	40 хв.
11	Вправи на зняття напруги, розвиток саморегуляції (австралійський дощ, оркестр та ін.)	20 хв.
12	Інформаційний блок: Механізми метакогнітивної діяльності тестувальника ПЗ	20 хв.
13	Завдання для самостійної роботи	5 хв.
III	Завершальна частина	
1	Ситуативне привітання. Розподіл учасників по командам	10 хв.
2	Рефлексивний аналіз завдання для самостійної роботи	15 хв.
3	Командна робота: створюємо проєкт мисленнєвої діяльності інженера-програміста з тестування ПЗ. Презентація проєктів	40 хв.
4	Що я не знаю про своє мислення?	15 хв.
5	Рефлексивний Я	20 хв.
6	Аналіз дорожньої карти тренінгу. Загальна рефлексія	30 хв.
8	Валіза: скетчбук	10 хв.
9	Завершення тренінгової роботи	10 хв.

### РЕКОМЕНДОВАНА ЛІТЕРАТУРА

1. Гелб М.Д. Как мыслить подобно Леонардо да Винчи: Упражнения, рабочая тетрадь. Мн.: ООО «Попурри», 2000. 145 с.
2. Громько Ю.В. Метапредмет «Знак» / Ю.В. Громько. М.: Пушкинский дом, АО «Московский учебник», 2001. 159 с.
3. Інтелектуальний розвиток дорослих у віртуальному освітньому просторі: монографія / М.Л. Смульсон, Ю.М. Лотоцька, М.М. Назар, П.П. Дітюк, І.Г. Коваленко-Кобилянська [та ін.]; за ред. М.Л. Смульсон. К.: Педагогічна думка, 2015. 221 с.
4. Карпов А.В., Скитяева И.М. Психология метакогнитивных процессов личности. М.: Изд-во «Институт психологии РАН», 2005. 352 с.
5. Карпов А.В. Рефлексия в структуре метакогнитивной организации субъекта. Рефлексивные процессы и управление. 2004. №1. Т.4. С. 99-109.
6. Кудрявцев В.Т. Методы интуитивного поиска технических решений (методы анализа проблем и поиска решений в технике). М.: Речной транспорт, 1991. 110 с.
7. Нельке М. Техники креативности. М.: Омега-Л, 2007. 144 с.
8. Тюков А.А. Схемы и схематизация. URL: <http://www.fondgp.ru/lib/conferences/2007/notes/5>
9. Швалб Ю.М. Целеполагающее сознание (психологические модели и исследования). К.: Миллениум, 2003. 152 с.
10. Щедровицкий Г.П. Мышление – Понимание – Рефлексия. М.: Наследие ММК, 2005. 800 с.

## Додаток Е2

### Програма Тренінгу професійної комунікації

Тренінг професійної комунікації (далі - тренінг) є складовою тренінгової програми, спрямованої на розвиток нетехнічних умінь майбутніх тестувальників ПЗ.

Метою тренінга є розвиток у майбутніх інженерів-програмістів таких груп нетехнічних умінь як комунікативні та регулятивні, а саме: 1) вміння працювати у команді, підтримувати позитивну атмосферу, домовлятися, комунікувати з членами команди та з замовником, обґрунтовувати і відстоювати свою позицію, вміння індивідуального підходу, неконфліктність; 2) уміння самоконтролю власної діяльності, самоконтролю власних емоцій, активізації працездатності; вміння керувати часом, бути зосередженим та уважним.

Тренінг розрахований на 16 годин, за своєю організаційною структурою складається з трьох основних частин: 1) вступної, 2) основної та 3) завершальної.

#### Програма тренінгових занять

№	Назва та зміст активності	Орієнтовний час
I	Вступна частина	
1	Ситуативне привітання. Розподіл учасників по командам	10 хв.
2	Мета та завдання тренінгу	5 хв.
3	Правила тренінгової роботи: створюємо самі. Презентація правил. Створюємо паркування запитань та ідей.	20 хв.
4	Професійний бейджик: створення бейджика з предметів навколо	15 хв.
5	Презентація себе: Я – невербальними засобами спілкування	20 хв.
6	Очікування учасників: від індивідуальних до командних очікувань	15 хв.
7	Дорожня карта тренінгу: групова робота	30 хв.
8	Презентація команд учасників: створюємо свій власний логотип. Презентація логотипу команди	20 хв.
9	Що я знаю про свою поведінку та стиль спілкування? Самодіагностика	15 хв.
10	Інформаційний блок: професійна комунікація: вербальне та невербальне спілкування та техніки його активізації	10 хв.
11	Завдання для самостійної роботи	5 хв.
II	Основна частина	
1	Ситуативне привітання. Розподіл учасників по командам	10 хв.
2	Рефлексивний аналіз завдання для самостійної роботи	15 хв.
3	Занурення у проблему. Інформаційне повідомлення: стилі спілкування, особливості спілкування в проєктній команді	30 хв.
4	Командна робота: створюємо модель успішного комунікатора	40 хв.

5	Презентація моделей, створення спільної моделі успішного комунікатора	20 хв.
6	Вправи на зняття напруги, розвиток саморегуляції (австралійський дощ, оркестр та ін.)	20 хв.
7	Розподіл учасників по новим командам	10 хв.
8	Командна робота: аналізуємо кейс «Перший робочий день тестувальника ПЗ: комунікативний та організаційний аспект»	30 хв.
9	Самодіагностика комунікативних та регулятивних умінь	20 хв.
10	Створення індивідуальної та командної ментальної карти з Лего «Проектне завдання з різних рольових позицій»	40 хв.
11	Вправи на зняття напруги, розвиток саморегуляції (австралійський дощ, оркестр та ін.)	20 хв.
12	Інформаційний блок: Механізми керування часом. Блокатори та винищувачі часу та їх запобіжники	20 хв.
13	Командна робота: Створюємо ефективний робочий простір. Презентація та обговорення	20 хв.
14	Завдання для самостійної роботи	5 хв.
III	Завершальна частина	
1	Ситуативне привітання. Розподіл учасників по командам	10 хв.
2	Рефлексивний аналіз завдання для самостійної роботи	15 хв.
3	Командна робота: створюємо проєкт комунікативної та саморегулятивної діяльності інженера-програміста з тестування ПЗ. Презентація проєктів	40 хв.
4	Що я не знаю про свої особливості комунікації в команді?	15 хв.
5	Рефлексивний Я: щоденні цілі та етапи їх досягнення	20 хв.
6	Аналіз дорожньої карти тренінгу. Загальна рефлексія	30 хв.
8	Подарунки: що беремо із собою	10 хв.
9	Завершення тренінгової роботи	10 хв.

#### РЕКОМЕНДОВАНА ЛІТЕРАТУРА

222. Авидон І., Гончукова О. Тренінги формування команди. Матеріали для підготовки і проведення. СПб.: Речь. 2008. 280 с.
223. Ігри та вправи для тренінгів (зі скарбнички неформальної освіти). URL: <https://www.gurt.org.ua/articles/43009/>
224. Кокерелл Ли Магія тайм-менеджмента. Как все успевать и жить в свое удовольствие. 2015. 192 с.
225. Сидоренко Е.В. Тренінг комунікативної компетентності в деловом взаємодії. Спб., 2004. 208 с.
226. Сидоренко Е.В. Технологии тренерской работы. М.: Университет «Синергия». 2018. 320 с.
227. Технологія тренінгу / Упоряд.: О. Главник, Г. Бевз/ За заг. ред. С. Максименко. К.: Главник, 2005. 112 с.
228. Посібник для тренера. Разом до майбутнього. К., 2010. 91 с.

## Додаток Ж

### Авторський опитувальник (пілотне дослідження)

#### Шановний колего!

Просимо Вас взяти участь у якості експерта у дослідженні, метою якого є визначення вимог до професійної підготовки майбутніх фахівців з тестування програмного забезпечення у закладах вищої освіти.

#### *I. Будь ласка, зазначте Ваш:*

Вік \_\_\_\_ Стать \_\_\_\_

Спеціальність за дипломом

Стаж роботи у сфері ІТ \_\_\_\_\_ Професійний рівень (title)

Кількість ІТ компаній, в яких Ви працювали

#### *II. На Ваш погляд:*

- 1) Чим відрізняється діяльність фахівця з тестування програмного забезпечення в ІТ-компанії від інших ІТ спеціалізацій
- 2) Які нетехнічні професійні якості (soft skills) у першу чергу сприяють успішній професійній кар'єрі фахівця з тестування програмного забезпечення?
- 3) Чи вважаєте Ви достатнім рівень професійної підготовки у закладі вищої освіти для подальшої роботи фахівцем з тестування програмного забезпечення? (зазначте чому саме)
- 4) Як покращити професійну підготовку майбутніх фахівців з тестування програмного забезпечення у закладах вищої освіти?
- 5) Чи є ефективною дуальна форма (проходження стажування в ІТ компанії паралельно із навчанням у ЗВО) підготовки майбутніх фахівців ІТ сфери? (зазначте чому саме)

#### *III.*

Вам пропонуються наступні твердження, що стосуються особливостей Вашого мислення та способів вирішення проблем. Оцініть, будь-ласка ці твердження за шкалою (обведіть кружечком потрібну цифру)

		<i>ні</i>	<i>скоріше ні</i>	<i>не знаю</i>	<i>скоріше так</i>	<i>так</i>
1	Періодично я запитую себе, чи досягаю я своїх життєвих цілей	1	2	3	4	5
2	Я розглядаю декілька альтернатив вирішення проблеми перед тим, як обрати остаточний варіант	1	2	3	4	5
3	Я намагаюсь використовувати способи та методи вирішення проблем, які спрацьовували раніше	1	2	3	4	5
4	Я обираю такий темп вирішення завдання, щоб мати достатньо часу	1	2	3	4	5

5	Я усвідомлюю свої інтелектуальні переваги та обмеження	1	2	3	4	5
6	Я думаю про те, яка інформація мені згодиться перед тим, як почати виконання завдання	1	2	3	4	5
7	Я можу оцінити наскільки добре виконав роботу у той момент, коли вона закінчена	1	2	3	4	5
8	Перед тим, як виконувати роботу, я чітко визначаю її мету	1	2	3	4	5
9	Я уповільнюю тем роботи, коли зіштовхуюсь з важливою для себе інформацією	1	2	3	4	5
10	Я знаю, яка саме інформація особлива важлива у моїй роботі	1	2	3	4	5
11	Працюючи над проблемою я час від часу запитую себе, чи розглянув я всі альтернативи її вирішення	1	2	3	4	5
12	Я здатний добре структурувати інформацію	1	2	3	4	5
13	Я свідомо концентруюсь на важливій для мене інформації	1	2	3	4	5
14	Я точно знаю, з якою метою використовую різні стратегії вирішення проблем	1	2	3	4	5
15	Я краще засвоюю інформацію, коли знаю щось відносно самої теми	1	2	3	4	5
16	Я знаю, чого очікує від мене викладач	1	2	3	4	5
17	Я добре запам'ятовую нову інформацію	1	2	3	4	5
18	Я використовую різні стратегії залежно від ситуації	1	2	3	4	5
19	Я запитую себе, чи існував легший шлях зробити завдання після того, як воно було виконане	1	2	3	4	5
20	Я здатний контролювати якість рішень, які я прийняв	1	2	3	4	5
21	Час від часу я «обертаюсь назад», що допомагає мені краще зрозуміти значущі для мене відносини	1	2	3	4	5
22	Я ставлю себе запитання, наскільки гарне прийняте рішення перед тим, як почати його здійснення	1	2	3	4	5
23	Я обмірковую декілька способів вирішення проблеми та обираю оптимальніший	1	2	3	4	5
24	Закінчивши роботу (виконавши завдання) я підводжу підсумки тому, що я зробив	1	2	3	4	5
25	Коли я в чомусь не можу розібратися, я звертаюсь за допомогою до інших	1	2	3	4	5
26	Я можу замотивувати себе навчатися, коли мені це необхідно	1	2	3	4	5
27	Я усвідомлюю, які стратегії використовую, коли приймаю рішення	1	2	3	4	5
28	Приймаючи важливе рішення, я аналізую ефективність використаних мною стратегій	1	2	3	4	5
29	Я використовую свої інтелектуальні переваги для компенсації своїх слабкостей	1	2	3	4	5
30	Я концентрую увагу на значенні та практичній цінності нової інформації	1	2	3	4	5
31	Я створюю власні приклади, щоб краще осмислити інформацію	1	2	3	4	5
32	Я можу точно цінити ступінь своєї компетентності у той чи іншій сфері	1	2	3	4	5

33	Я автоматично застосовую ефективні стратегії вирішення завдання	1	2	3	4	5
34	Вивчаючи щось нове, я час від часу роблю паузу та запитую себе, наскільки добре я розумію матеріал	1	2	3	4	5
35	Я знаю, в якому випадку кожна з використаних мною стратегій буде найбільш ефективною	1	2	3	4	5
37	Коли вирішення завдання закінчене, я запитую себе, чи досягнуті поставлені цілі	1	2	3	4	5
38	Я роблю малюнки та діаграми, які допомагають краще зрозуміти проблему, над якою працюю	1	2	3	4	5
39	Після того, як завдання зроблено, я запитую себе, чи врахував я інші можливі варіанти його вирішення	1	2	3	4	5
40	Я намагаюсь перевести нову інформацію в доступну для мене форму	1	2	3	4	5
41	Коли я не можу щось зрозуміти, я змінюю спосіб роботи з інформацією	1	2	3	4	5
42	Я уважно читаю інструкцію перед тим, як розпочати виконувати завдання	1	2	3	4	5
43	Коли я читаю про щось нове, я співвідношу це з тим, що мені відомо у цій сфері	1	2	3	4	5
44	Я переглядаю свої пропозиції, коли затрудняюсь в рішенні проблеми	1	2	3	4	5
45	Я організую свій час так, щоб досягнути своїх цілей найкращим способом	1	2	3	4	5
46	Я краще навчаюсь, коли тема мені цікава	1	2	3	4	5
47	Я намагаюсь розбити роботу на певну кількість окремих завдань	1	2	3	4	5
48	Я концентруюсь на загальному смислі роботи у більшій мірі, ніж на її деталях	1	2	3	4	5
49	Я схильний запитувати себе, наскільки успішно я просуваюсь, коли вивчаю щось нове	1	2	3	4	5
50	Я запитую себе чи достатньо я вивчив ніж потрібно перед тим як завершити завдання	1	2	3	4	5
51	Якщо нова інформація недостатньо зрозуміла для мене, я повертаюсь до неї для того, щоб ще раз переосмислити	1	2	3	4	5
52	Коли я читаю текст, я декілька разів перечитую складні для мого розуміння абзаци	1	2	3	4	5

Будь ласка, оцініть мисленнєві стратегії, які Ви використовуєте у своїй професійній діяльності, за наступною шкалою (обведіть кружечком потрібну цифру)

		<i>дуже часто</i>	<i>часто</i>	<i>іноді</i>	<i>рідко</i>	<i>дуже рідко</i>
1	Стратегічне планування (цілеспрямоване планування, моніторинг та оцінка діяльності)	5	4	3	2	1
2	Формулювання запитань (свідоме формулювання запитань, що звернені до пробілів у тій чи іншій	5	4	3	2	1

	сфері знання)					
3	Свідоме прийняття рішення (прогнозування ефекту та наслідків кожного варіанта вибору)	5	4	3	2	1
4	Диференційна оцінка (рефлексивна оцінка власних дій за різними критеріями)	5	4	3	2	1
5	Осмилення досягнень (співвіднесення суб'єктивних досягнень з об'єктивним зворотним зв'язком)	5	4	3	2	1
6	Подолання суб'єктивних обмежень (усвідомлення можливостей вирішення складних завдань та наполегливий свідомий пошук рішення)	5	4	3	2	1
7	Перефразування та резюмування отриманої інформації (переосмилення ідей, що приходять)	5	4	3	2	1
8	Визначення когнітивної поведінки (визначення використаних когнітивних стратегій та їх значущості для вирішення завдання)	5	4	3	2	1
9	Визначення термінології (формулювання точних визначень первинно розмитих, багатозначних або погано зрозумілих термінів)	5	4	3	2	1
10	Рольові ігри (прогривання позиції партнера по спілкуванню/замовника, мисленнєвий діалог з ним)	5	4	3	2	1
11	Ведення щоденника (письмова фіксація власних думок)	5	4	3	2	1
12	Моделювання (побудова мисленнєвих репрезентацій досвіду) та схематизація (побудова схем явищ та процесів)	5	4	3	2	1

*Дякуємо за співробітництво!*

## Додаток 3

### Додаток 31

#### Методики діагностики ціннісно-мотиваційного компоненту

Опитувальник «Мій шлях широкий»

(визначення психологічної готовності до трудової діяльності)

#### Інструкція

Шановний колего! Уважно прочитайте кожне судження. Якщо ви вважаєте, що воно стосовно вас є правильним (так) або наближене до нього (швидше так), поставьте у відповідній графі бланку (+). Відповідно, якщо судження неправильне (ні) або наближене до нього (швидше ні), також у відповідній графі бланку поставьте (+). Відповідайте швидко і щиро. Немає поганих або хороших відповідей. Всі запропоновані судження спрямовані допомогти зрозуміти, чи готові ви розпочати трудову діяльність за обраною спеціальністю. Успіху!

#### Бланк відповідей

№ судження	Варіант відповіді			
	Так	Швидше так	Швидше ні	Ні
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				

1. Я думаю, що люди, з якими я спілкуюсь, вважають мене впевненою в собі людиною.
2. Розпочинаючи роботу, я звик(ла) аналізувати умови, в яких мені необхідно буде працювати.
3. Після виконання будь-якої роботи я звик(ла) оцінювати не тільки кінцевий результат, але і ті, що отримую в процесі роботи.

4. Я схильний(а) відмовитись від задуманого, якщо іншим здається, що я розпочав(ла) не так.
5. Навіть при виконанні відповідальної роботи мені не потрібен чийсь контроль.
6. Я однаково старанно виконую як цікаву роботу, так і ту, яка мене не зацікавила.
7. Для успішного виконання відповідальної роботи я потребую контролю збоку інших.
8. Як правило мій робочий день проходить безсистемно (немає розподілу роботи, часу, плану, чіткого розпорядку).
9. За наявності вибору я віддаю перевагу роботі менш відповідальній, але й менш цікавій.
10. По завершенні будь-якої роботи я звик(ла) обов'язково перевіряти, чи правильно я її зробив(ла).
11. Я обов'язково повертаюсь до розпочатої справи, навіть тоді, коли мене ніхто не контролює.
12. Сумніви в успіхах нерідко змушують мене відмовитись від справи.
13. Мені часто не вистачає наполегливості для досягнення мети.
14. Мої плани ніколи не розходяться з моїми можливостями.
15. Як правило, будь-які рішення я приймаю, тільки порадившись з кимось.
16. Часто мені буває важко змусити себе зосередитись на якомусь завданні чи роботі.
17. Коли я захоплений(на) якоюсь справою, мені буває важко переключитись на виконання іншої.
18. Я схильний(на) відмовитись від роботи, яка мені не вдається.

Ключ

Так, швидше так: 1,5,11,14.

Ні, швидше ні: 2, 3, 4, 6, 7, 8, 9, 10 12 13 15,16,17,18.

### **Інтерпретація результатів**

*Від 0 до 3 балів.* Ви і зараз сумніваєтесь, чи правильно обрали саме цю професію. Вам важко буває самому об'єктивно оцінювати результати своєї діяльності (знаходити помилки, шукати шляхи їх виправлення).

*Від 4 до 6 балів.* Ви не сумніваєтесь у виборі саме цієї професії. Успішно справляєтесь із добре знайомою роботою, намагаєтесь виконувати її якомога краще. Впевненість у свої сили допоможе вам так само добре виконувати роботу із складними і відповідальними завданнями. Наполегливість і критичність, які успішно у вас розвиваються, допоможуть глибоко і правильно оцінювати свої можливості. І ви обов'язково станете фахівцем своєї справи.

*Від 7 і більше балів.* Ви впевнені, що вибрали саме цю професію. У вас добре сформовані такі риси характеру, як цілеспрямованість, зібраність, самостійність, здатність критично оцінювати свою діяльність. Ви вмієте знаходити кращі заходи для вирішення складних завдань. Впевненість у своїх силах, відповідальність за виконувану роботу, намагання зробити її дуже добре розвинули у вас звичку аналізувати причини невдач та успіхів у роботі, вміння приймати правильне рішення. Саме ці риси і формують з вас кваліфікованого фахівця, цілком готового до трудової діяльності.

**Методика вивчення кар'єрних орієнтацій «Якоря кар'єри»,**  
*авт. Е. Шейн (адаптація В. Чикер, В. Вінокурової)*

«Якорі кар'єри» - це ціннісні орієнтації, соціальні установки, інтереси, соціально зумовлені спонукання до діяльності, характерні для певної людини. Тест дозволяє виявити наступні кар'єрні орієнтації: професійна компетентність, менеджмент, автономія, стабільність, служіння, виклик, інтеграція стилів життя, підприємництво.

**Інструкція.** Будь ласка, дайте відповідь на питання тесту.

Питання 1-21: Наскільки важливим для Вас є кожне з наступних тверджень? Варіанти відповідей: 1 - абсолютно не важливо, чи 2, 3, 4, 5, 6, 7, 8, 9, 10 - винятково важливо.

Питання 22-40: Наскільки Ви згодні з кожним з наступних тверджень? Варіанти відповідей: 1 - абсолютно не згоден, 2,3,4,5,6,7,8,9,10 - повністю згоден.

**Опитувальний бланк**

1		11		21		31	
2		12		22		32	
3		13		23		33	
4		14		24		34	
5		15		25		35	
6		16		26		36	
7		17		27		37	
8		18		28		38	
9		19		29		39	
10		20		30		40	
						41	

По кожній кар'єрній орієнтації підраховується кількість балів. Для цього необхідно, користуючись ключем, підсумовувати бали по кожній орієнтації та отриману суму розділити на кількість питань. Таким чином визначається провідна кар'єрна орієнтація.

**Ключ до тесту**

- Професійна компетентність: 1, 9, 17, 25, 33.
- Менеджмент: 2, 10, 18, 26, 34.
- Автономія (незалежність): 3, 11, 19, 27, 35.
- Стабільність роботи: 4, 12, 36.

- Стабільність місця проживання: 20, 28, 41.
- Служіння: 5, 13, 21, 29, 37.
- Виклик: 6, 14, 22, 30, 38.
- Інтеграція стилів життя: 7, 15, 23, 31, 39.
- Підприємництво: 8, 16, 24, 32, 40.

№ п/п	Кар'єрні орієнтації	Сумарний бал	Середнє значення
	Професійна компетентність		
	Менеджмент		
	Автономія		
	Стабільність роботи		
	Стабільність місця проживання		
	Служіння		
	Виклик		
	Інтеграція стилів життя		
	Підприємництво		

**Професійна компетентність.** *Бути професіоналом, майстром у своїй справі.*

Ця орієнтація пов'язана з наявністю здібностей і талантів у певній галузі. Люди з такою орієнтацією бувають особливо щасливі, коли досягають успіху в професійній сфері, але швидко втрачають інтерес до роботи, яка не дозволяє розвивати їхні здібності. Навряд чи їх зацікавить навіть значно більш висока посада, якщо вона не пов'язана з їх професійними компетенціями. Вони шукають визнання своїх талантів, що має виражатися в статусі, відповідному їх майстерності. Багато хто з цієї категорії відкидають роботу керівника, управління розглядають як необхідну умову для просування у своїй професійній сфері.

**Менеджмент.** *Управляти - людьми, проектами, бізнес-процесами і т.п.*

Орієнтація особистості на інтеграцію зусиль інших людей, повнота відповідальності за кінцевий результат і з'єднання різних функцій організації. Можливості для лідерства, високого доходу, підвищених рівнів відповідальності і внесок в успіх своєї організації є ключовими цінностями і мотивами. Найголовніше для них - управління: людьми, проектами, якими бізнес-процесами - це в цілому не має принципового значення. Центральне поняття їх професійного розвитку - влада, усвідомлення того, що від них залежить прийняття ключових рішень. Причому для них не є

принциповим управління власним проектом або цілим бізнесом, скоріше навпаки, вони більшою мірою орієнтовані на побудову кар'єри в найманій менеджменті, але за умови, що їм будуть делеговані значні повноваження. Людина з такою орієнтацією буде вважати, що не досяг мети своєї кар'єри, поки не займе посаду, на якій буде управляти різними сторонами діяльності підприємства.

**Автономія** (незалежність) *Головне в роботі - це свобода і незалежність.*

Первинна турбота особистості з такою орієнтацією - звільнення від організаційних правил, приписів і обмежень. Вони відчують труднощі, пов'язані з встановленими правилами, процедурами, робочим днем, дисципліною, формою одягу і т.д. Вони люблять виконувати роботу своїм способом, темпом і за власними стандартами. Вони не люблять, коли робота втручається в їхнє приватне життя, тому воліють робити незалежну кар'єру власним шляхом. Вони скоріше виберуть низькосортну роботу, ніж відмовляться від автономії та незалежності. Для них першочергове завдання розвитку кар'єри - отримати можливість працювати самостійно, самому вирішувати, як, коли і що робити для досягнення тих чи інших цілей. Кар'єра для них - це, насамперед, спосіб реалізації їх свободи, тому будь-які рамки і суворе підпорядкування відштовхнуть їх навіть від зовні привабливою вакансії. Така людина може працювати в організації, яка забезпечує достатній ступінь свободи.

**Стабільність роботи.** *Стабільна, надійна робота на тривалій час.*

Ці люди відчують потребу в безпеці, захисті і можливості прогнозування і будуть шукати постійну роботу з мінімальною ймовірністю звільнення. Ці люди ототожнюють свою роботу зі своєю кар'єрою. Їх потреба в безпеці і стабільності обмежує вибір варіантів кар'єри. Авантюрні або короткострокові проекти і тільки що стають на ноги компанії їх, швидше за все, не приваблюють. Вони дуже цінують соціальні гарантії, які може запропонувати роботодавець, і, як правило, їх вибір місця роботи пов'язаний саме з тривалим контрактом і стабільним становищем компанії на ринку. Такі люди відповідальність за управління своєю кар'єрою перекладають на наймача. Часто дана ціннісна орієнтація поєднується з невисоким рівнем домагань.

**Стабільність місця проживання.** *Головне - жити у своєму місті* (мінімум переїздів, відряджень). Важливіше залишитися на одному місці проживання, ніж отримати підвищення або нову роботу на новій місцевості. Переїзд для таких людей неприйнятний, і навіть часті відрядження є для них негативним чинником при розгляді пропозиції про роботу.

**Служіння.** *Втілювати в роботі свої ідеали і цінності.* Дана ціннісна орієнтація характерна для людей, що займаються справою через бажання реалізувати у своїй роботі головні цінності. Вони часто орієнтовані більше

на цінності, ніж на вимагаються в даному виді роботи здібності. Вони прагнуть приносити користь людям, суспільству, для них дуже важливо бачити конкретні плоди своєї роботи, навіть якщо вони і не виражені в матеріальному еквіваленті. Основна теза побудови їх кар'єри - отримати можливість максимально ефективно використовувати їх таланти і досвід для реалізації суспільно важливої мети. Люди, орієнтовані на служіння, товариські і часто консервативні. Людина з такою орієнтацією не працюватиме в організації, яка ворожа його цілям і цінностям.

**Виклик.** *Зробити неможливе - можливим, вирішувати унікальні завдання.* Ці люди вважають успіхом подолання нездоланих перешкод, рішення нерозв'язних проблем або просто виграш. Вони орієнтовані на те, щоб "кидати виклик". Для одних людей виклик представляє більш важка робота, для інших це - конкуренція і міжособистісні відносини. Вони орієнтовані на рішення свідомо складних завдань, подолання перешкод заради перемоги в конкурентній боротьбі. Вони відчувають себе процвітаючими тільки тоді, коли постійно залучені до вирішення важких проблем чи в ситуацію змагання. Кар'єра для них - це постійний виклик їх професіоналізму, і вони завжди готові його прийняти. Соціальна ситуація найчастіше розглядається з позиції "виграшу - програшу". Процес боротьби і перемога важливіша для них, ніж конкретна область діяльності або кваліфікація. Новизна, різноманітність і виклик мають для них дуже велику цінність, і, якщо все йде дуже просто, їм ставати нудно.

**Інтеграція стилів життя.** *Збереження гармонії між особистим життям і кар'єрою.* Для людей цієї категорії кар'єра має асоціюватися із загальним стилем життя, врівноважуючи потреби людини, сім'ї та кар'єри. Вони хочуть, щоб організаційні відносини відбивали б повагу до їх особистих і сімейних проблем. Вибирати і підтримувати певний спосіб життя для них важливіше, ніж домагатися успіху в кар'єрі. Розвиток кар'єри їх приваблює тільки в тому випадку, якщо вона не порушує звичний їм стиль життя і оточення. Для них важливо, щоб все було урівноважене - кар'єра, сім'я, особисті інтереси і т.п. Жертвувати чимось одним заради іншого їм явно не властиво. Такі люди зазвичай у своїй поведінці проявляють конформність (тенденція змінювати свою поведінку залежно від впливу інших людей, з тим, щоб воно відповідало думку оточуючих)

**Підприємництво.** *Створювати нові організації, товари, послуги.* Цим людям подобається створювати нові організації, товари або послуги, які можуть бути ототожені з їх зусиллями. Працювати на інших - це не їх, вони - підприємці за духом, і мета їх кар'єри - створити щось нове, організувати свою справу, втілити в життя ідею, цілком належить тільки їм. Вершина кар'єри в їхньому розумінні - власний бізнес.



## Додаток 32

### Авторський Тест оцінки знань з тестування ПЗ

Тест спрямований на визначення рівня сформованості когнітивного компоненту професійної готовності майбутніх тестувальників ПЗ – їх загально професійних та спеціально професійних знань. Тест складається з 40. У тесті за своєю формою наявні як закриті запитання (15 з 20 з кожної частини), так і відкриті запитання (5 з 20). Кожна правильна відповідь на закриті запитання оцінюється в 1 бал, за відкриті запитання – 2 бали.

#### Блок 1. Загальні теоретичні питання

1. В чому полягає сутність поняття «життєвий цикл ПЗ»?  
*Життєвий цикл ПЗ - це послідовність стадій, які проходить програмний продукт від появи ідеї до її реалізації в кодї, імплементації в бізнес-середовищі і подальшої підтримки.*
2. Підхід до розробки програмного забезпечення, орієнтований на використання ітеративної розробки, динамічне формування вимог і забезпечення їх реалізації в результаті постійної взаємодії всередині самоорганізованих робочих груп, називається:  
А. Гнучким  
Б. Монолітним  
В. Водоспадним  
Г. Модульним
3. До командних ролей методології Scrum не відноситься:  
А. Майстер  
Б. Секретар  
В. Команда розробки  
Г. Власник продукту
4. Сутність наради Backlog Grooming полягає у:  
А. Обміні інформацією про прогрес виконання завдань  
Б. Обговоренню минулої ітерації розробки  
В. Естимації завдань до виконання у наступній ітерації  
Г. Приоритизації і перевірки актуальності ще невиконаних завдань
5. Story points це:  
А. Кількість ще не запланованих завдань.  
Б. Короткі історії про процес розробки.  
В. Одиниці виміру складності і трудовитрат.  
Г. Узагальнені вимоги до функціональності програмного продукту.
6. У чому різниця понять «планування» і «естімація»?  
*Естімація - процес оцінки трудовитрат для виконання завдання, а планування - визначення часу і контексту виконання завдання на підставі його пріоритету та результатів естімації.*
7. До командних технік естімації відноситься:  
А. Planning poker

*Б. Estimation roulette*

*В. S.T.A.R.*

*Г. Equivalent division*

8. Делегування це:

*А. Процес відправлення завдання на доопрацювання через незадовільні результати*

*Б. Процес передачі повноважень або завдання іншому члену команди*

*В. Процес спільного виконання завдання*

*Г. Процес презентації результатів завдання замовнику*

9. «Спагетті-кодом» називають:

*А. Заплутану і неінтуїтивну архітектуру програмного забезпечення*

*Б. Інструменти програмування, розроблені в італійських ІТ-компаніях*

*В. Свіжі і популярні практики побудови архітектури програмного забезпечення*

*Г. Основний базис коду програмного продукту, на якому будується високорівнева логіка*

10. До кваліфікаційних рівнів працівника в сфері ІТ не відноситься:

*А. Junior*

*Б. Chief*

*В. Elder*

*Г. Middle*

11. Для чого потрібні системи контролю версій?

*Для контролю та управління змінами в вихідному коді програмного забезпечення з плином часу*

12. До популярних систем контролю версій не відноситься:

*А. Git*

*Б. Mercurial*

*В. Subversion*

*Г. Gradle*

13. До централізованих систем контролю версій відноситься:

*А. Git*

*Б. Mercurial*

*В. Subversion*

*Г. Gradle*

14. Команда git pull:

*А. Зберігає локальні зміни*

*Б. Проводить злиття локальної гілки з віддаленою*

*В. Завантажує вміст віддаленого репозиторія та поновлює їм локальний*

*Г. Оновлює список створених гілок у віддаленому репозиторії*

15. Створення гілок у системі контролю версій дозволяє:

*А. Одночасно керувати декількома версіями програмного продукту і синхронізувати їх зміни між собою*

*Б. Підвищувати якість програмного продукту за рахунок залучення нових користувачів в процес тестують*

*В. Анулювати свіжі зміни до програмного продукту в разі виявлення критичного дефекту*

*Г. Підвищити швидкість розробки за рахунок паралельної роботи з віддаленим репозиторієм*

16. В чому полягає відмінність між Continuous Integration та Continuous Delivery?

*Continuous Integration має на увазі практику моментального додавання новоствореного коду в головний репозиторій в міру його готовності.*

*Continuous Delivery є наступним етапом процесу, який передбачає автоматичне оновлення функціоналу додатка в міру потрапляння нового коду в головний репозиторій.*

17. СУБД потрібна для:

*А. Усунення блокуючих дефектів*

*Б. Управління базою даних*

*В. Управління бізнес-додатками*

*Г. Укріплення бізнес динаміки*

18. До типів операції JOIN не відноситься:

*А. LEFT*

*Б. BOTTOM*

*В. RIGHT*

*Г. INNER*

19. До принципів ООП не відноситься:

*А. Інкапсуляція*

*Б. Сегрегація*

*В. Наслідування*

*Г. Поліморфізм*

20. Чим відрізняються протоколи HTTP та HTTPS?

*У HTTPS використовується шифрування за допомогою криптографічних протоколів - SSL і TLS. Фактично це той же протокол передачі даних, що і HTTP, але з додаванням надбудови шифрування.*

## **Блок 2. Тестування ПЗ**

1. В чому відмінність понять «тестування» та «контроль якості»?

*Тестування - процес безпосередньо виявлення дефектів, в той час як контроль якості - комплекс заходів, спрямований на всі етапи розробки програмного забезпечення, який фокусується більше на превентивних засобах, що запобігають появі дефектів.*

2. До типів тестування за доступом до архітектури продукту не відноситься:

*А. Black box*

*Б. Grey box*

*В. White box*

Г. Green box

3. Перевірка зовнішнього вигляду веб-сторінки відноситься до:

*А. Unit testing*

*Б. Performance testing*

*В. GUI testing*

*Г. Usability testing*

4. Тестові сценарії, що призводять до одної поведінки системи, об'єднують в :

*А. Тест-план*

*Б. Клас еквівалентності*

*В. Функціональний набір*

*Г. Не об'єднують*

5. Тестування без планування і документації називають:

*А. Black box testing*

*Б. Ad-hoc testing*

*В. Functional testing*

*Г. Volume testing*

6. Який вид тестування потребує повторного проходження вже існуючих сценаріїв для тестування функціоналу на предмет змін?

*А. Модульне*

*Б. Регресійне*

*В. Ієрархічне*

*Г. Повторне*

7. На якому етапі життєвого циклу не можуть з'явитися дефекти?

*А. Написання програмного коду*

*Б. Побудування архітектури*

*В. Створення документації*

*Г. Можуть з'явитися у всіх випадках вище*

8. Чи входить написання Unit тестів в обов'язки відділу тестування?

*А. Зазвичай, так*

*Б. Зазвичай, ні*

9. У чому відмінність функціонального і нефункціонального тестування?

*Функціональне тестування направлено на перевірку того, які функції ПЗ реалізовані, і того, наскільки вірно вони реалізовані.*

*Нефункціональне тестування – перевірка того, як функції роботають у контексті системи і взаємодії з користувачем.*

10. До обов'язкових елементів тест-кейсу не відноситься:

*А. Очікувана поведінка*

*Б. Пріоритет*

*В. Ідентифікатор*

*Г. Актуальна поведінка*

11. Техніка тест-дизайну, що спрямована на формування таких наборів даних, в яких кожне тестоване значення кожного з перевірених параметрів

хоча б один раз поєднується з кожним тестованим значенням всіх інших параметрів, називається:

*A. Equivalence Partitioning*

*B. Boundary Value Analysis*

*V. Descision Table*

*G. Pairwise*

12. Останнім етапом життєвого циклу дефекту є:

*A. Перевірений*

*B. Закритий*

*V. Виправлений*

*G. Розроблений*

13. Які перевірки потрібно автоматизувати в першу чергу?

*V першу чергу автоматизуються найбільш повторювані та важливі перевірки, такі як Smoke-набір, а також перевірки низькорівневого функціоналу чи нефункціональних вимог, що потребують спеціальних технічних інструментів, наприклад, тестування продуктивності.*

14. Selenium WebDriver – це інструмент для взаємодії з:

*A. HTML-сторінками*

*B. REST-запитами*

*V. Базами даних*

*G. Мобільними пристроями*

15. Для чого призначений Page object pattern?

*Для структурованого опису веб-сторінки чи її частини у кодї фреймворку для автоматизації.*

16. Apache JMeter – інструмент для автоматизації:

*A. Тестування продуктивності*

*B. Тестування коду на платформі Java*

*V. Інсталяційного тестування*

*G. Тестування інтерфейсу*

18. Інструмент Postman дозволяє:

*A. Відправляти веб-запити та аналізувати результати*

*B. Збирати статистику щодо активних користувачів мережі*

*V. Налаштовувати розсилку електронних листів*

*G. Обмінюватися захищеними повідомленнями з замовником*

19. Який вид тестування неможливо автоматизувати?

*A. Тестування продуктивності*

*B. Бета-тестування*

*V. Тестування білого ящика*

*G. Інсталяційне тестування*

20. Чи входить стратегія автоматизації у тест-план?

*A. Зазвичай, входить*

*B. Зазвичай, не входить*



## Методика дослідження комунікативних та організаторських схильностей

авт. Б. Федоришин

Дослідження комунікативних та організаторських схильностей (КОС) за допомогою тест-опитувальника КОС можна проводити і з одним досліджуваним, і з групою. Досліджуваним роздають тексти опитувальника, бланки для відповідей, зачитується інструкція.

Інструкція: «Запропонований вам тест містить 40 питань. Прочитайте їх та дайте відповіді на всі питання за допомогою бланка. На бланку надруковані номери питань. Якщо ваша відповідь на питання позитивна, тобто ви погоджуєтесь з тим, про що запитують у питанні, то на бланку відповідний номер обведіть кружечком. Якщо ваша відповідь негативна, тобто ви не погоджуєтесь, то відповідний номер закресліть. Слідкуйте, щоб номер питання і номер у бланку для відповідей співпадали. Майте на увазі, що питання носять загальний характер і не можуть містити всіх необхідних подробиць. Тому уявіть собі типові ситуації і не задумуйтесь над деталями. Не слід витратити багато часу на роздуми, відповідайте швидко. Можливо, на деякі питання вам буде важко відповісти. Тоді намагайтесь дати ту відповідь, якій ви надаєте перевагу. При відповіді на будь-яке з цих питань звертайте увагу на його перші слова та узгоджуйте свої відповіді з ними. Відповідаючи на питання, не прагніть створити заздалегідь приємне враження. Важлива ваша відвертість при відповіді».

Текст опитувальника

1. Чи багато у вас друзів, з якими ви постійно спілкуєтесь?
2. Чи часто вам вдається схилити більшість своїх знайомих до прийняття ними вашої думки?
3. Чи довго вас турбує почуття образи, яку завдали вам хтось із приятелів?
4. Чи завжди вам важко орієнтуватися у критичній ситуації, що склалась?
5. Чи є у вас прагнення до встановлення нових знайомств з різними людьми?
6. Чи подобається вам займатись громадською роботою?
7. Чи вірно, що вам приємніше та простіше проводити час з книжками або з якими-небудь іншими справами, ніж з людьми?
8. Якщо виникли які-небудь завади у здійсненні ваших намірів, то чи легко відступаєте від них?
9. Чи легко ви встановлюєте контакти з людьми, які значно старші за віком?
10. Чи любите ви придумувати та організовувати зі своїми приятелями різні ігри та розваги?
11. Чи важко вам включатись у нову для вас компанію?

12. Чи часто ви відкладаєте на інші дні ті справи, які потрібно було б виконати сьогодні?
13. Чи легко вам вдається встановлювати контакти з незнайомими людьми?
14. Чи прагнете ви добитися, щоб ваші приятелі, колеги діяли у відповідності з вашою думкою?
15. Чи важко ви адаптуєтесь у новому колективі?
16. Чи вірно, що у вас не буває конфліктів з колегами з причини невиконання ними своїх обов'язків, зобов'язань?
17. Чи прагнете ви при нагоді познайомитись та поговорити з новою людиною?
18. Чи часто у вирішенні важливих справ ви приймаєте ініціативу на себе?
19. Чи дратують вас оточуючі люди і чи хочеться вам побути одному?
20. Чи правда, що ви зазвичай погано орієнтуєтесь при незнайомих умовах?
21. Чи подобається вам постійно знаходитись серед людей?
22. Чи виникає у вас роздратування, якщо вам не вдається закінчити розпочату справу?
23. Чи зазнаєте ви почуття утруднення, незручності, якщо прийдеться проявити ініціативу, щоб познайомитись з новою людиною?
24. Чи правда, що ви стомлюєтесь від частого спілкування з приятелями?
25. Чи любите ви брати участь у колективних іграх?
26. Чи часто ви проявляєте ініціативу при вирішенні питань, що стосуються інтересів ваших друзів, знайомих?
27. Чи правда, що ви відчуваєте себе невпевнено серед малознайомих людей?
28. Чи вірно, що ви рідко прагнете довести свою правоту?
29. Чи вважаєте ви, що вам не дуже важко внести пожвавлення в малознайому компанію?
30. Чи брали ви участь у громадській роботі в школі?
31. Чи прагнете ви обмежити коло своїх знайомих невеликою кількістю людей?
32. Чи вірно, що ви не прагнете відстоювати свою думку або рішення, якщо воно не було відразу прийняте вашими товаришами?
33. Чи відчуваєте себе невимушено, попавши у незнайому вам компанію?
34. Чи охоче ви приступаєте до організації різних заходів для своїх товаришів?
35. Чи правда, що ви не відчуваєте себе достатньо впевненим та спокійним, коли потрібно говорити що-небудь великій групі людей?
36. Чи часто ви спізнюєтесь на ділові зустрічі, побачення?
37. Чи вірно, що у вас багато друзів?
38. Чи часто ви потрапляєте у центр уваги своїх товаришів?

39. Чи часто ви соромитесь, почуваетесь незручно при спілкуванні з малознайомими людьми?

40. Чи правда, що ви не дуже впевнено почуваете себе в оточенні великої групи своїх приятелів?

Бланк для відповідей являє собою чотири колонки з цифрами, що означають номери питань та пронумеровані від 1 до 40 у певному порядку, точно такому, як на зразку.

Обробка результатів. Мета обробки результатів - отримання індексів комунікативних та організаторських здібностей. Для цього відповіді досліджуваного зіставляють із дешифратором і підраховують кількість збігів ок-ремо за комунікативними та організаторськими схильностями. У дешифраторі враховується за рядками розташування номерів питань у бланку для відповідей.

5	9	13	17	21	25	29	33	37	
2	6	10	14	18	22	26	30	34	38
3	7	11	15	19	23	27	31	35	39
4	8	12	16	20	24	28	32	36	40

#### Дешифратор

Схильності	Відповіді	
	позитивні	негативні
Комунікативні	номера питань 1-го рядка	номера питань 2-го рядка
Організаторські	номера питань 3-го рядка	номера питань 4-го рядка

$$K_k = \frac{K_x}{20}; \quad K_o = \frac{O_x}{20}, \text{ де}$$

$K_k$  — коефіцієнт комунікативних схильностей;

$K_o$  — коефіцієнт організаційних схильностей;

Щоб визначити рівень комунікативних та рівень організаторських здібностей, потрібно порахувати їх коефіцієнти. Коефіцієнти являють собою відношення кількості збігів відповідей тієї чи іншої схильності до максимально можливої кількості збігів, у даному випадку - до 20. Формули для підрахунку такі:

$K_x$  та  $O_x$  - кількість співпадаючих з дешифратором відповідей, відповідно до комунікативних та організаційних схильностей.

Аналіз результатів

У ході аналізу результатів спочатку дають оцінку рівня комунікативних та організаторських схильностей досліджуваного. Для цього користуються шкалою оцінок.

#### Шкала оцінок комунікативних та організаторських схильностей

$K_o$	Шкала оцінки	
0,10–0,45	0,2–0,55	1
0,46–0,55	0,56–0,65	2

0,56–0,65	0,66–0,70	3
0,66–0,75	0,71–0,80	4
0,75–1,00	0,81–1,00	5

Рівень розвитку комунікативних та організаторських схильностей характеризується за допомогою оцінок за шкалою наступним чином. Досліджувані, що отримали оцінку 1, - це люди з низьким рівнем прояву комунікативних та організаторських схильностей.

Досліджувані, що отримали оцінку 2, мають комунікативні та організаційні схильності нижче середнього рівня. Вони не прагнуть до спілкування, почувають себе скуто у новій компанії, колективі, надають перевагу проводити час наодинці, обмежують свої знайомства, відчують труднощі у встановленні контактів з людьми і у виступах перед аудиторією, погано орієнтуються у незнайомій ситуації, не відстоюють свою думку, важко переживають образи. У багатьох справах вони намагаються уникати прояву самостійних рішень та ініціативи.

Для досліджуваних з оцінкою 3 характерний середній рівень прояву комунікативних та організаторських схильностей. Вони прагнуть до контактів з людьми, не обмежують коло своїх знайомств, відстоюють свою думку, планують свою роботу, однак потенціал їх схильностей не відрізняється високою стійкістю. Ця група досліджуваних потребує подальшої серйозної та планомірної виховної роботи по формуванню та розвитку комунікативних та організаторських схильностей.

Досліджувані, що отримали оцінку 4, відносяться до групи з високим рівнем прояву комунікативних та організаторських схильностей. Вони не губляться у новій обстановці, швидко знаходять друзів, постійно прагнуть розширити коло знайомих, займаються громадською діяльністю, допомагають близьким, друзям, проявляють ініціативу у спілкуванні, з задоволенням беруть участь у організації громадських заходів, здатні приймати самостійні рішення у важкій ситуації. Все це вони роблять не за примусом, а згідно внутрішнім прагненням.

Досліджуваним, що отримали вищу оцінку 5, притаманний дуже високий рівень прояву комунікативних та організаторських схильностей. Вони відчують потребу у комунікативній та організаторській діяльності і активно прагнуть до неї, швидко орієнтуються у складних ситуаціях, невимушено поведуть себе у новому колективі, це ініціативні люди, які віддають перевагу у важливих справах або у складній ситуації приймати самостійні рішення, відстоюють свою думку та досягають, щоб її сприймали інші. Вони можуть внести позитивний внесок у незнайому компанію, люблять організовувати різні ігри, заходи, настійливі у діяльності, яка їх приваблює, і самі шукають такі справи, які б задовольняли їх потребу у комунікації та організаційній діяльності.

Бланк відповідей

1		11		21		31	
2		12		22		32	
3		13		23		33	
4		14		24		34	
5		15		25		35	
6		16		26		36	
7		17		27		37	
8		18		28		38	
9		19		29		39	
10		20		30		40	

Ключі. Графи, виділені жирною лінією вирізаються, при накладенні трафарету-дешифратора на «Лист питань» у них видно відповіді випробуваного, які порівнюються з дешифраційними.

#### Комунікативні схильності

1	+		11	-		21	+		31	-	
2			12			22			32		
3	-		13	+		23	-		33	+	
4			14			24			34		
5	+		15	-		25	+		35	-	
6			16			26			36		
7	-		17	+		27	-		37	+	
8			18			28			38		
9	+		19	-		29	+		39	-	
10			20			30			40		

#### Організаторські схильності

1	-		11	+		21	-		31	+	
2	+		12	-		22	+		32	-	
3	+		13	-		23	+		33	-	
4	-		14	+		24	-		34	+	
5	-		15	+		25	-		35	+	
6	+		16	-		26	+		36	-	
7	+		17	-		27	+		37	-	
8	-		18	+		28	-		38	+	
9	-		19	+		29	-		39	+	

## Шкала самооцінки метакогнітивної поведінки ЛаКоста (адаптація А. Карпова)

Методика рекомендована для визначення рівня сформованості метакогнітивних стратегій у ситуації цілеспрямованого навчання та для діагностики профілю метакогнітивних стратегій у фахівців, які використовують їх у професійній діяльності. Шкала складається з 12 тверджень, які визначають ступінь генералізації метакогнітивних умінь, оскільки піддослідний обирає частоту їх використання (5 - дуже часто; 4 – часто; 3 – іноді; 2 – рідко; 1 – дуже рідко) у різних ситуаціях. Результат підраховується як проста сума балів з максимальною до 60 (до 21 балів – низький рівень, 22-43 бали – середній рівень, від 44 і вище – високий рівень).

Інструкція: Будь ласка, оцініть мисленнєві стратегії, які Ви використовуєте, за наступною шкалою (обведіть кружечком потрібну цифру): використовую: 5 - дуже часто; 4 – часто; 3 – іноді; 2 – рідко; 1 – дуже рідко.

		<i>дуже часто</i>	<i>часто</i>	<i>іноді</i>	<i>рідко</i>	<i>дуже рідко</i>
1	Стратегічне планування (цілеспрямоване планування, моніторинг та оцінка діяльності)	5	4	3	2	1
2	Формулювання запитань (свідоме формулювання запитань, що звернені до пробілів у тій чи іншій сфері знання)	5	4	3	2	1
3	Свідоме прийняття рішення (прогнозування ефекту та наслідків кожного варіанта вибору)	5	4	3	2	1
4	Диференційна оцінка (рефлексивна оцінка власних дій за різними критеріями)	5	4	3	2	1
5	Осмислення досягнень (співвіднесення суб'єктивних досягнень з об'єктивним зворотним зв'язком)	5	4	3	2	1
6	Подолання суб'єктивних обмежень (усвідомлення можливостей вирішення складних завдань та наполегливий свідомий пошук рішення)	5	4	3	2	1
7	Перефразування та резюмування отриманої інформації (переосмислення ідей, що приходять)	5	4	3	2	1
8	Визначення когнітивної поведінки (визначення використаних когнітивних стратегій та їх значущості для вирішення завдання)	5	4	3	2	1
9	Визначення термінології (формулювання точних визначень первинно розмитих, багатозначних або погано зрозумілих термінів)	5	4	3	2	1
10	Рольові ігри (прогривання позиції партнера по спілкуванню/замовника, мисленнєвий діалог з ним)	5	4	3	2	1

11	Ведення щоденника (письмова фіксація власних думок)	5	4	3	2	1
12	Моделювання (побудова мисленневих репрезентацій досвіду) та схематизація (побудова схем явищ та процесів)	5	4	3	2	1

Результатом за даною методикою є проста сума балів, що визначає рівень сформованості метакогнітивних та когнітивних умінь в ситуації цілеспрямованого навчання або для діагностування профілю метакогнітивних стратегій у фахівців, які використовують метакогнітивні стратегії у своїй професійній діяльності.

**Додаток К**  
**СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА**  
**Наукові праці, в яких опубліковані основні наукові результати**  
**дисертації**

***Статті в наукових виданнях, включених до міжнародних  
наукометричних баз Scopus та Web of Science***

1. Gura Jr. O., Gura O., Gura T., Chernikova L. Research on metacognitive skills of software testers: a problem statement. *Proceedings of the 16th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer. Volume II: Workshops. Part IV: V International Workshop on Professional Retraining and Life-Long Learning using ICT: Person-oriented Approach (3L-Person).* Kharkiv, Ukraine, October 06-10, 2020, 607-618. (*Scopus*)

***Статті в періодичних наукових виданнях інших держав***

2. Гура О. Аналіз зарубіжного досвіду впровадження системи дуальної підготовки майбутніх фахівців. *European Humanities Studies: State & Society.* 2017. Issue 4. С. 31-43. (*Index Copernicus*)

3. Гура О. Матриця компетенцій (skill matrix) як орієнтир підготовки майбутніх ІТ-фахівців. *European Humanities Studies: State & Society.* 2018. Issue 6. С. 181-191. (*Index Copernicus*)

***Статті в наукових фахових виданнях України***

4. Гура О. Основні сфери реалізації фахівців ІТ на сучасному ринку праці: аналіз професійних вимог. *Науковий вісник Миколаївського національного університету імені В. О. Сухомлинського. Педагогічні науки.* 2017. Вип. 2 (57). С. 166-170. (*Index Copernicus*)

5. Гура О. Особливості розвитку м'яких навичок студентів ІТ-спеціальностей засобами навчальних SCRUM проектів. *ScienceRise: Pedagogical Education.* 2019. №4(31). С. 8-15. (*Index Copernicus*)

6. Гура О. Підготовка інженерів-програмістів до тестування програмного забезпечення в умовах неформальної освіти: проблеми та шляхи реалізації. *Педагогічні науки: теорія, історія, інноваційні технології: наук. журнал* / голов. ред. А.А. Сбруюва. Суми: Вид-во СумДПУ імені А.С.Макаренка. 2020. № 7 (101). С. 55-62. (*Index Copernicus*)

#### ***Статті в інших наукових виданнях***

7. Гура О. Особливості співбесіди як форми контролю якості освіти студентів ІТ спеціальностей. *Педагогічні науки та освіта*. 2016. Вип. 18-19. С. 29-34.

#### ***Матеріали конференцій***

8. Гура О. Актуальні проблеми професійної підготовки фахівців ІТ-сфери. *Електронний збірник наукових праць ЗОІППО*. 2016. № 3(25). URL: <https://drive.google.com/file/d/0B6QknaQCh-IHTFpkeVpSeHhBdGM/view>

9. Гура О. Особливості основних сфер реалізації ІТ фахівця на сучасному ринку праці. *Електронний збірник наукових праць ЗОІППО*. 2017. Вип. 1(27). URL: <https://drive.google.com/file/d/0B6QknaQCh-IHWGtSWEhiYXJGTIU/view>

10. Гура О. Особливості впровадження системи дуальної підготовки майбутніх фахівців: порівняльний аналіз. *Педагогіка вищої школи: досвід і тенденції розвитку*. Матеріали III Всеукраїнської науково-практичної конференції з міжнародною участю. Запоріжжя: Запорізький Національний Університет, 2018. С. 41-42.

11. Гура О. Професійна підготовка фахівців ІТ сфери на засадах матриці компетенцій (skills matrix). *Україна в гуманітарних та соціально-економічних вимірах*. Матеріали III Всеукраїнської наукової конференції (30-31 березня 2018р. м. Дніпро). ч. II. Наук.ред. О.Ю. Висоцький. Дніпро: «Охотник», 2018. 268 с. С. 103-105.

## ВІДОМОСТІ ПРО АПРОБАЦІЮ РЕЗУЛЬТАТІВ ДИСЕРТАЦІЇ

Основні положення та результати дисертації були оприлюднені на науково-практичних конференціях, зокрема:

- *всукраїнського рівня*: «Науково-методичні засади професійного розвитку фахівця у системі неперервної освіти» (Запоріжжя, 2016); «Неперервна освіта нового сторіччя: досягнення та перспективи» (Запоріжжя, 2017); «Педагогіка вищої школи: досвід і тенденції розвитку» (Запоріжжя, 2018); «Україна в гуманітарних та соціально-економічних вимірах» (Дніпро, 2018);

- *міжнародного рівня*: «Вища освіта України у контексті інтеграції до Європейського освітнього простору» (Київ, 2017, 2018); International Workshop on Professional Retraining and Life-Long Learning using ICT: Person-oriented Approach (3L-Person) (Kharkiv, 2020).

Результати дослідження *впроваджено* в освітній процес Запорізького національного університету, Національного технічного університету «Харківський політехнічний інститут», Харківського національного економічного університету імені Семена Кузнеця, Хмельницької гуманітарної педагогічної академії, а також у кадрову та рекрутингову діяльність Товариства з обмеженою відповідальністю «ЕПАМ СИСТЕМЗ».



УКРАЇНА

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

вул. Жуковського, 66, м. Запоріжжя, МСП-41, 69600, Україна  
тел.: (061) 764-45-46, факс: (061) 228-75-08, e-mail: [znu@znu.edu.ua](mailto:znu@znu.edu.ua), Код ЄДРПОУ 02125243

ЗС. 10. 2020 № 01.01-13/131

На № \_\_\_\_\_

від \_\_\_\_\_

### ДОВІДКА

про впровадження результатів дисертаційного дослідження аспіранта кафедри загальної та прикладної фізики Запорізького національного університету Гури Олександра Олександровича за темою «Підготовка майбутніх інженерів-програмістів до тестування програмного забезпечення в умовах неформальної освіти» (спеціальність 015 Професійна освіта (за спеціалізаціями).

Впродовж 2016 – 2020 років на базі Запорізького національного університету Гурою Олександром Олександровичем в межах підготовки фахівців спеціальності 121 Інженерія програмного забезпечення (освітні програми: «Програмна інженерія», «Програмне забезпечення систем») здійснювалась дослідно-експериментальна робота, спрямована на вдосконалення системи підготовки майбутніх інженерів-програмістів до тестування програмного забезпечення в умовах неформальної освіти (практичної діяльності).

Результати роботи знайшли своє відображення в напрямках:

1) проведення емпіричних досліджень:

- пілотних досліджень, спрямованих на визначення актуальності та ресурсного забезпечення освітнього процесу;
- анкетування суб'єктів освітнього процесу;
- діагностика рівня готовності майбутніх фахівців до тестування програмного забезпечення;

2) контент-аналізу нормативно-методичної документації, що регламентує зміст підготовки (навчальні плани, логіко-структурні схеми, освітньо-професійні програми, програми практичної підготовки);

3) апробації авторських формувальних заходів (окремі теми спецкурсу з основ тестування програмного забезпечення; програми тренінгів з розвитку м'яких навичок та ін.).

Науково-теоретичні та емпіричні здобутки Гури О.О. неодноразово висвітлювались у фахових і зарубіжних наукових виданнях, доповідались на науково-практичних конференціях, засіданнях кафедри й отримали позитивну оцінку.

Проректор з науково-педагогічної роботи

Ю.О. Каганов





МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ХМЕЛЬНИЦЬКА ОБЛАСНА РАДА  
ХМЕЛЬНИЦЬКА ГУМАНІТАРНО-ПЕДАГОГІЧНА АКАДЕМІЯ

вул. Проскурівського підпілля, 139, м. Хмельницький, 29013,  
тел./факс: (0382) 72-09-23, 65-65-52, тел.: 79-53-55, 79-59-45  
E-mail: kgpa@ukr.net Код ЄДРПОУ 02138872

Від 21.12.20 № 634  
на \_\_\_\_\_ від \_\_\_\_\_

**ДОВІДКА**

про впровадження результатів дисертаційного дослідження аспіранта кафедри загальної та прикладної фізики Запорізького національного університету Гури Олександра Олександровича за темою «Підготовка майбутніх інженерів-програмістів до тестування програмного забезпечення в умовах неформальної освіти»

Протягом 2018-2020 років на базі Хмельницької гуманітарно-педагогічної академії в процесі професійної підготовки майбутніх фахівців галузі знань 01 Освіта/Педагогіка спеціальності 011 Освітні, педагогічні науки були реалізовані компоненти програми науково-дослідної роботи О.О. Гури, спрямованої на визначення науково-методичних засад підготовки майбутніх інженерів-програмістів до тестування програмного забезпечення в умовах неформальної освіти.

Зокрема, дисертантом здійснено контент-аналіз нормативно-методичної документації, яка регламентує зміст підготовки майбутніх фахівців ІТ сфери (навчальні плани, логіко-структурні схеми, освітньо-професійні програми), проведено опитування здобувачів освіти.

Практична апробація авторських принципів інтеграції формальної та неформальної освіти (врахування можливостей позааудиторної підготовки, поєднання різноманітних методів і форм роботи), програми тьюторства, діагностичного інструментарію засвідчили про доцільність їх впровадження в процес професійної підготовки майбутніх фахівців.

Результати науково-дослідної роботи О.О. Гури обговорювались на засіданнях кафедри менеджменту освіти та педагогіки вищої школи й отримали позитивну оцінку (протокол № 3 від 05 жовтня 2020 року).

Проректор з наукової роботи,  
доктор педагогічних наук,  
професор



Олександр ГАЛУС



УКРАЇНА

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ**  
**ІМЕНІ СЕМЕНА КУЗНЕЦЯ**

61166, м. Харків, пр. Науки, 9-А, тел. (057) 702-03-04, факс: (057) 702-07-17  
E-mail: post@hneu.edu.ua, http://www.hneu.edu.ua

№ 10/86-15-186 від 14.12.2020

На № \_\_\_\_\_ від \_\_\_\_\_

**ДОВІДКА**

про впровадження результатів дисертаційного дослідження аспіранта кафедри загальної та прикладної фізики Запорізького національного університету  
Гури Олександра Олександровича  
на тему: «Підготовка майбутніх інженерів-програмістів до тестування програмного забезпечення в умовах неформальної освіти»  
(спеціальність 015 Професійна освіта (за спеціалізаціями).

Апробація науково-методичних розробок О.О. Гури здійснювалась в освітньому процесі підготовки фахівців спеціальності 121 «Інженерія програмного забезпечення», 122 «Комп'ютерні науки», 126 «Інформаційні системи та технології» ступеня бакалавр кафедрою інформаційних систем Харківського національного економічного університету ім. С. Кузнеця протягом 2017-2020 років у формах:

1) аналіз контенту нормативно-методичної документації (навчальних планів, освітньо-професійних програм, методичних рекомендацій до практичної підготовки та ін.);

2) пілотного дослідження стану підготовки майбутніх ІТ фахівців в сфері тестування програмного забезпечення; апробації методів діагностики рівня їх копетентностей; апробації науково-методичних розробок в умовах неформальної освіти, зокрема:

- методичних розробок для дисциплін «Якість програмного забезпечення та тестування», «Технології розробки та тестування програмного забезпечення»;
- тренінгу розвитку метакогнітивних вмінь ("soft skills");
- програми менторства;
- технології співбесіди, як форми контролю якості освіти та ін.

Результати впровадження науково-методичних розробок О.О. Гури в освітній процес бакалаврів Харківського національного економічного університету ім. С. Кузнеця дають підстави стверджувати, що вони є значущими для підготовки ІТ фахівців в сфері тестування програмного забезпечення.

Зав. кафедри ІС

Ірина УШАКОВА

В.о. ректора

Володимир ПОНОМАРЕНКО





Проректор з науково-педагогічної  
роботи Національного технічного  
університету «ХПІ»

Геннадій ХРИПУНОВ

« 15 » 12 2020 р.  
№ 01/37-425

### ДОВІДКА

про впровадження результатів дисертаційного дослідження аспіранта кафедри загальної та прикладної фізики Запорізького національного університету Гури Олександра Олександровича за темою «Підготовка майбутніх інженерів-програмістів до тестування програмного забезпечення в умовах неформальної освіти» (спеціальність 015 Професійна освіта (за спеціалізаціями).

Результати дисертаційної роботи О.О. Гури впроваджувалися впродовж 2017-20 рр. у навчальний процес підготовки фахівців спеціальностей 123 Комп'ютерна інженерія, 125 Кібербезпека Національного технічного університету «Харківський політехнічний інститут» при викладанні навчальних дисциплін «Захист інформації» загальним обсягом 194 академічних години та «Комп'ютерні системи та їх тестування» загальним обсягом 164 академічних години

Зокрема пройшли апробацію:

1) принципи інтеграції формальної та неформальної освіти на рівнях предметному, який передбачає поєднання змісту авторського спецкурсу «Основи тестування програмного забезпечення та контролю якості», що реалізовувався на засадах неформальної підготовки зі змістом профільних професійних дисциплін; інтеграції методів, яка зумовлює поєднання різноманітних методів навчання заради досягнення максимального дидактичного ефекту; інтеграція форм, яка базується на компіляції різних форм освітнього процесу (формальних/неформальних) як відповідно до мети і змісту навчальних дисциплін, так і вимог сучасної ІТ сфери;

2) елементи розробленої авторської моделі формування готовності майбутніх фахівців до тестування програмного забезпечення в процесі неформального навчання: комплекс навчально-методичних матеріалів (мультимедійні презентації та методичні розробки занять); організаційно-методичні засади проектної практики на принципах дуальної освіти; тренінги та сценарії ділових ігор.

Результати впровадження науково-методичних розробок Гури Олександра Олександровича отримали позитивну оцінку викладацького складу кафедри обчислювальної техніки та програмування Національного технічного університету «Харківський політехнічний інститут».

Завідувач кафедри «ОТП»,  
д.т.н., професор

Сергій СЕМЕНОВ

Декан факультету «КІТ»,  
к.е.н., доцент

Максим ГЛАВЧЕВ

Проф. каф. «ОТП» НТУ «ХПІ»,  
д.т.н., професор

Георгій КУЧУК



Товариство з обмеженою відповідальністю «ЕПАМ СИСТЕМЗ»

вул. Фізкультури 28, Київ, 03150, Україна | Т: +38 044 390 54 57 Ф: +38 044 390 08 61  
e-mail: [info@epam.com](mailto:info@epam.com) | web: [www.epam.com](http://www.epam.com)

ЄДРПОУ 33880213, п/р UA583808050000000026000383792 в ПАТ «Райффайзен Банк Аваль»

Вих. № 126

від 15 лютого 2021 року

### ДОВІДКА

про впровадження результатів дисертаційного дослідження аспіранта кафедри педагогіки та психології освітньої діяльності Запорізького національного університету Гури Олександра Олександровича за темою «Підготовка майбутніх інженерів-програмістів до тестування програмного забезпечення в умовах неформальної освіти» (спеціальність 015 Професійна освіта (за спеціалізаціями).

Впродовж 2016 – 2020 років на базі EPAM Systems аспірантом кафедри педагогіки та психології освітньої діяльності Запорізького національного університету Гурюю Олександром Олександровичем в межах розробки та реалізації кадрової та рекрутингової політики здійснювалась дослідно-експериментальна робота, спрямована на вдосконалення системи підготовки майбутніх інженерів-програмістів до тестування програмного забезпечення в умовах неформальної освіти (практичної діяльності).

Результати роботи знайшли своє відображення в напрямках:

- 1) проведення емпіричних досліджень, спрямованих на визначення рівня готовності майбутніх фахівців до роботи в реальних умовах;
- 2) розробці й впровадженню авторської моделі підготовки, що передбачає інтеграцію контенту професійної діяльності тестувальника в зміст формальної професійної освіти; використання дидактичних можливостей формувальних заходів ІТ організацій інтегрованих в систему професійного навчання на робочому місці (тренінги; співбесіди; аналіз професійних ситуацій тощо);
- 3) реалізації системи менторства;
- 4) організації та управління проектною діяльністю.

Застосування розроблених діагностичних і навчально-методичних матеріалів сприяло підвищенню рівня готовності майбутніх фахівців ІТ сфери до тестування програмного забезпечення, й, в цілому, рівня їх професійної підготовки.

Генеральний директор ТОВ «ЕПАМ СИСТЕМЗ»



С.В. Рожок