

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ**

ПАРХОМЕНКО ОЛЕКСАНДРА В'ЯЧЕСЛАВІВНА



УДК 004.413:378.4:372.8

**ВИКОРИСТАННЯ ГНУЧКИХ МЕТОДОЛОГІЙ РОЗРОБКИ
ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ У ПІДГОТОВЦІ МАЙБУТНІХ
ПРОГРАМІСТІВ**

13.00.04 «Теорія та методика професійної освіти»

Автореферат дисертації на здобуття наукового ступеня
кандидата педагогічних наук

Київ – 2021

Дисертацією є кваліфікаційна наукова праця на правах рукопису

Роботу виконано у Національному університеті біоресурсів і природокористування України Міністерства освіти і науки України

Науковий керівник доктор педагогічних наук, професор
Глазунова Олена Григорівна,
Національний університет біоресурсів
і природокористування України,
декан факультету інформаційних технологій

Офіційні опоненти: доктор педагогічних наук, професор
Осадчий В'ячеслав В'ячеславович,
Мелітопольський державний педагогічний
університет імені Богдана Хмельницького,
завідувач кафедри комп'ютерних наук

кандидат педагогічних наук, доцент
Антонюк Дмитро Сергійович,
Державний університет
«Житомирська політехніка»,
доцент кафедри інженерії програмного
забезпечення

Захист відбудеться «13» травня 2021 року о 10⁰⁰ годині на засіданні спеціалізованої вченої ради Д 26.004.18 у Національному університеті біоресурсів і природокористування України за адресою: 03041, м. Київ, вул. Героїв Оборони, 15, навчальний корпус № 3, кімната 213

З дисертацією можна ознайомитися у науковій бібліотеці Національного університету біоресурсів і природокористування України за адресою: 03041, м. Київ, вул. Героїв Оборони, 13, навчальний корпус № 4, кімната 41а

Автореферат розіслано «09» квітня 2021 року

Вчений секретар
спеціалізованої вченої ради



І. В. Сопівник

ЗАГАЛЬНА ХАРАКТЕРИСТИКА РОБОТИ

Актуальність теми. Сучасний етап розвитку світового суспільства тісно пов'язаний зі стрімким розвитком інформаційних технологій. Складність інформаційних систем невідомо зростає, відповідно набувають актуальності питання щодо ефективного управління процесами створення, тестування та впровадження таких систем. Змінюються сфери застосування, вимоги до програмного забезпечення, підходи до розробки та інструменти. Інформаційні технології покривають майже всі сфери людської діяльності. Все більше з'являється розробок у галузях BigData, інтернету речей, штучного інтелекту, віртуальної реальності, хмарних технологій, підвищується попит на фахівців з програмування, програмної інженерії, проектування та розробки інформаційних систем.

В таких умовах швидкого розвитку суспільства виникає потреба у підготовці фахівців з інформаційних технологій, які володіють сучасними, затребуваними на ринку праці технологіями. Однак, традиційна система освіти має певні проблеми з швидким впровадженням нових технологій у навчальний процес, що пов'язані, насамперед, з необхідністю оновлення змістової складової та технологічної бази. Дослідження статей, форумів, професійних спільнот і власний досвід роботи з колективом ІТ-фахівців показує, що для підтримки професіоналізму програміст повинен бути обізнаним з безліччю нових технологій, знати кращі рішення тих або інших завдань (best practice), використовувати допоміжні інструменти, що пришвидшують процес розробки, вміти працювати в команді, та всіляко поповнювати свої знання і вміння. Актуальною стає проблема підготовки фахівців у закладі вищої освіти відповідно до вимог ринку праці, формування в них фахових компетентностей. Отже, завдання сучасної освіти – відповідати запитам ринку інформаційних технологій та здійснювати підготовку фахівців, які володітимуть актуальними знаннями і компетентностями, що закладені у освітні стандарти підготовки.

Важливим є й той факт, що сучасне програмування є колективним, і корисність окремого програміста тісно пов'язана з його корисністю для всієї команди, а тому вимагає від особи навичок роботи у команді, лідерських якостей, певних знань з галузі психології й управління. Колективність процесу розробки та складність розроблюваних систем спричиняють появу і поширення специфічних технологій, методів, підходів до управління проектами з розробки програмного забезпечення. Керування розробки і підтримкою складних інформаційних систем і програмних продуктів в умовах невизначеності спричиняє застосування нових підходів і інструментів. Крім цього, особливістю успішного програміста є не деякий фіксований набір знань, умінь і навичок у певній галузі, а сформованість спектру компетентностей. Побудова навчального процесу саме на засадах компетентнісного підходу є найефективнішою для забезпечення підготовки фахівця з інженерії програмного забезпечення відповідно до сучасних вимог вищої освіти та суспільства.

У рамках компетентнісного підходу крім формування певного набору професійних знань, умінь і навичок у галузі інженерії програмного забезпечення увага приділяється також формуванню таких якостей, як робота в команді, лідерські якості, відповідальність, здатність до рефлексії, здатність до самостійного навчання й освоєння нових технологій протягом життя, самоосвіта, планування діяльності, логічне й алгоритмічне мислення, цілеспрямованість, наполегливість, уміння самостійно ухвалювати рішення, швидко адаптуватися до нових умов і постійно навчатися. Крім того, виникає затребуваність у специфічних знаннях психології й менеджменту, зокрема, управління проектами в рамках гнучких методологій розробки програмного забезпечення.

Одним із шляхів вдосконалення процесу підготовки майбутніх програмістів є використання гнучких методологій у навчанні. Загальною перевагою використання гнучких методологій розробки програмних продуктів є те, що студенти не лише отримують можливість познайомитися з сучасними методами розробки, а й використати їх при виконанні проєктів з розробки програмного забезпечення. При цьому створюються умови, наближені до реальних, для здобуття фахових компетентностей з розробки програмного забезпечення безпосередньо під час навчання. Проєктне навчання майбутніх програмістів необхідно реалізовувати з використанням сучасних методологій, та на реальних практичних кейсах.

Аналізом специфіки завдань у галузі програмування і питанням професійних якостей програмістів у різний час займалися науковці В. Концедайло, В. Сєдов, А. Стрюк, Т. Гончаренко, О. Наумук, В. Осадчий, К. Осадча, І. Сердюк, D. Saito, A. Takebaya-shi, Т. Волошина, Ts. Yamaura, С. Конюхов, D. Coppit, P. Kamthan, О. Баранюк, О. Спірін, В. Єремєєв, В. Круглик та ін. Дослідженнями стану розвитку та ролі інформаційних технологій в Україні і світі займалися В. Геєць, А. Мазаракі, А. Маслов, В. Міщенко, В. Плескач, Т. Тадаскіна, О. Чубукова, С. Андрєєв, Д. Белл, П. Друкер, Я. Жаліло, М. Кастельс, Е. Тоф-Флер, Ю. Христюк, Л. Довгань, А. Козинець, І. Кондіус, Р. Винничук, Т. Склярук, Т. Taivalsaari, A. Mikkonen, A. Begel, J. Bosch, M. A. Storey та ін. Використання гнучких методологій розробки програмного забезпечення в навчанні висвітлюють у своїх працях педагоги G. Lang, J. Sharp, Т. Яковишина, V. Razmov, R. Anderson, A. H. W. Chun, J. Wengroff, D'Souza M., P. Rodrigues, Н. Гульчевская, A. Hulshult, T. Krehbiel та ін.

На основі здійсненого аналізу виявлено низку суперечностей, що супроводжують сучасний стан підготовки майбутніх інженерів-програмістів: між станом розвитку інформаційно-комунікаційних технологій у світі та Україні, що відображається у вимогах роботодавців та недостатнім рівнем необхідних компетентностей, що формуються у студентів під час навчання у закладах вищої освіти; різноманітністю інструментів і методів гнучкої розробки програмного забезпечення та недостатнім навчально-методичним забезпеченням науково-педагогічних працівників з рекомендаціями щодо їх добору і використання у навчальному процесі; можливостями використання

інструментів і методів гнучкої розробки програмного забезпечення майбутніми програмістами та недостатнім рівнем розробленості відповідних методик використання інструментів і методів гнучкої розробки програмного забезпечення в навчальному процесі. Одна із непростих проблем полягає у застосуванні гнучких підходів до інженерії програмного забезпечення під час навчального процесу, оскільки в межах семестрової навчальної дисципліни неможливо якісно виконати розробки програмного продукту відповідно до принципів гнучких методологій. Зміст освіти в сфері програмування також вимагає постійного вдосконалення, тож, постає питання щодо мов програмування, підходів і технологій, що використовуються наразі в сфері інформаційних технологій, які варто включати в освітні програми, та як зробити процес навчання більш ефективним.

Актуальність окресленої проблеми і необхідність вирішення зазначених суперечностей зумовили вибір теми дисертації.

Зв'язок роботи з науковими програмами, планами, темами.

Дисертацію виконано відповідно до наукового напрямку, що входить до плану науково-дослідної роботи кафедр комп'ютерних наук, інформаційних систем і технологій факультету інформаційних технологій Національного університету біоресурсів і природокористування України. Дисертаційне дослідження проведено в рамках виконання науково-дослідної теми «Створення моделі гібридного веб-орієнтованого середовища доставки навчального контенту в умовах відкритої університетської освіти» (номер державної реєстрації 0120U102156).

Мета та завдання дослідження. Мета роботи – теоретично обґрунтувати, розробити й експериментально перевірити ефективність методики використання гнучких методологій розробки програмного забезпечення у підготовці майбутніх програмістів.

Для досягнення поставленої мети необхідно було вирішити такі завдання:

– проаналізувати вітчизняні та зарубіжні дослідження щодо підготовки майбутніх програмістів та вимоги ринку праці;

– проаналізувати гнучкі методології розробки програмного забезпечення, інструменти гнучкої розробки та визначити можливості їх використання у професійній підготовці програмістів, обґрунтувати критерії добору методів та інструментів гнучкої розробки програмного забезпечення для використання у професійній підготовці майбутніх програмістів;

– уточнити поняття «agile-навчання», обґрунтувати зміст поняття та умови застосування у підготовці майбутніх програмістів;

– розробити модель методики використання гнучких методологій розробки програмного забезпечення у підготовці майбутніх програмістів;

– експериментально перевірити методику використання гнучких методологій розробки програмного забезпечення у підготовці майбутніх програмістів та розробити методичні рекомендації.

Об'єкт дослідження – професійна підготовка майбутніх програмістів.

Предмет дослідження – методика використання гнучких методологій розробки програмного забезпечення у підготовці майбутніх програмістів.

Методи дослідження. Для досягнення поставленої мети й вирішення завдань використано такі методи дослідження. Теоретичні методи: аналіз та узагальнення наукових та науково-методичних джерел з проблеми дослідження, законодавчої та нормативної документації з питань вищої освіти, вітчизняного та закордонного досвіду навчання гнучким методологіям розробки програмного забезпечення, та використання гнучких підходів для організації навчального процесу, власного педагогічного досвіду. Емпіричні методи: анкетування, бесіди з викладачами, студентами, педагогічне спостереження за процесом застосування гнучких методів розробки програмних продуктів; моделювання навчального процесу і педагогічного експерименту в умовах застосування гнучких методологій розробки та гнучких методів навчання; методи математичної статистики для опрацювання результатів педагогічного експерименту (критерії Пірсона та Стюдента для доведення гіпотези про значимість різниці отриманих результатів в контрольній і експериментальній групах, дисперсійний аналіз для перевірки груп на однорідність).

Наукова новизна одержаних результатів дослідження полягає в тому, що:

вперше теоретично обґрунтовано та розроблено модель методики використання гнучких методологій розробки програмного забезпечення у підготовці майбутніх програмістів, яка складається з шести блоків: цільовий, концептуальний, методико-технологічний, змістовно-процесуальний, діагностичний і результативний; і має на меті підвищити рівень сформованості компетентностей з розробки програмного забезпечення з використанням гнучких методологій;

уточнено поняття «agile-навчання» – як процес навчання, в якому використовуються принципи, методи і інструменти гнучких методологій розробки програмного забезпечення в навчальних цілях та побудований на основі коротких проектних циклів, які називаються «спринтами», в яких повністю планується, проектується, будується, випробовується, переглядається та запускається повна версія навчального проекту;

удосконалено:

– критерії добору методів гнучкої розробки програмного забезпечення для використання у професійній підготовці майбутніх програмістів, зокрема, запропоновано показники щодо функціональності інструментів гнучкої розробки для структурно-функціонального та навчально-комунікаційного критеріїв;

– рівні, критерії та показники сформованості професійних компетентностей з розробки програмного забезпечення у майбутніх фахівців з програмування щодо використання методологій та інструментів гнучкої розробки програмного забезпечення;

– зміст підготовки майбутніх програмістів з розробки програмного забезпечення в частині доповнення змісту навчальних дисциплін та навчальних практик.

Практичне значення одержаних результатів дослідження полягає в тому, що: розроблено та перевірено ефективність методики використання

гнучких методологій розробки програмного забезпечення у підготовці майбутніх програмістів, яка базується на трьох етапах: підготовчому – вивчення гнучких методологій розробки; практичному – застосування гнучких методів навчання та розробки програмного забезпечення під час проектно-технологічної практики; кваліфікаційному – виконання кваліфікаційної роботи під керівництвом викладача з використанням методів гнучкої розробки; розроблено та впроваджено у навчальний процес закладів вищої освіти: методичні рекомендації щодо використання гнучких методологій розробки програмного забезпечення у підготовці майбутніх програмістів; розроблено та впроваджено у навчальний процес електронний навчальний курс з проектно-технологічної практики та модуля електронного навчального курсу з дисципліни «Конструювання програмного забезпечення».

Результати дослідження впроваджено в навчальний процес Національного університету біоресурсів і природокористування України, Рівненського коледжу Національного університету біоресурсів і природокористування України, Національного педагогічного університету імені М. Драгоманова, Сумського державного університету, Київського університету імені Бориса Грінченка.

Особистий внесок автора. У статтях, опублікованих у співавторстві, автору належать: аналіз поняття гнучких методологій розробки, аналіз домінуючого навчального стилю майбутніх програмістів, добір хмарних сервісів і ресурсів для використання на різних етапах організації перевернутого навчання майбутніх ІТ-фахівців, обґрунтування та розробка моделі процесу використання сервісів для проектної командної роботи майбутніх ІТ-фахівців, аналіз наявних платформ та системи автоматизованої перевірки завдань із програмування та їх застосування для формування професійних компетентностей майбутніх програмістів, аналіз можливостей хмарних ресурсів та сервісів Microsoft для неформальної освіти студентів ІТ-фаху, зокрема ресурси щодо інженерії програмного забезпечення, обґрунтування переваг сервісу GitHub для організації проектного навчання майбутніх програмістів та результати експериментального дослідження, обґрунтування та опис методики використання сервісу GitHub в підготовці майбутніх програмістів, аналіз і опис застосування гнучких методологій розробки програмного забезпечення в навчанні майбутніх програмістів.

Апробація результатів дисертації. Основні положення і результати дослідження доповідались та обговорювались на: II Міжнародній науково-практичній конференції «Глобальні та регіональні проблеми інформатизації в суспільстві і природокористуванні '2014» (м. Київ, 2014 р.); IX Міжнародній науково-практичній конференції молодих вчених «Інформаційні технології: економіка, техніка, освіта» (м. Київ, 2018 р.); VII Міжнародній науково-практичній конференції «Глобальні та регіональні проблеми інформатизації в суспільстві і природокористуванні 2019» (м. Київ, 2019 р.); XII International Conference on Mathematics, Science and Technology Education (ICon-MaSTEd 2020) (м. Кривий Ріг, 2020 р.); The International Conference on Sustainable Futures: Environmental, Technological, Social and Economic Matters 2020 (ICSF2020) (м. Кривий Ріг, 2020 р.); 16th International Conference on ICT in Research,

Education and Industrial Applications (ICTERI 2020) (м. Харків, 2020 р.); VIII Міжнародній науково-практичній конференції «Глобальні та регіональні проблеми інформатизації в суспільстві і природокористуванні 2020» (м. Київ, 2020 р.).

Публікації. За матеріалами дисертаційного дослідження опубліковано 10 наукових праць, з яких 2 статті у наукових фахових виданнях України, у тому числі включених до міжнародних наукометричних баз даних, 4 статті у наукових виданнях, включених до міжнародних наукометричних баз даних Scopus/Web of Science, 4 тези наукових доповідей.

Структура та обсяг дисертації. Дисертація складається з анотацій, переліку умовних позначень, скорочень і термінів, вступу, трьох розділів, висновків, списку використаних джерел і додатків. Загальний обсяг дисертації становить 272 сторінки. Робота містить 17 таблиць та 41 рисунок. Список використаних джерел налічує 208 найменувань, з яких 65 англійською мовою.

ОСНОВНИЙ ЗМІСТ ДИСЕРТАЦІЇ

У **вступі** обґрунтовано вибір теми, визначено об'єкт, предмет, мету та завдання дослідження, розкрито її наукову новизну, теоретичне та практичне значення роботи.

У першому розділі «**Теоретичні основи підготовки майбутніх програмістів**» проаналізовано поняття та складові підготовки фахівців з інженерії програмного забезпечення, здійснено аналіз організації підготовки майбутніх програмістів у вітчизняних та закордонних університетах, аналіз гнучких методів та інструментів розробки програмного забезпечення, визначено критерії добору для їх використання у навчальному процесі.

Проаналізовано стандарти підготовки фахівців з програмування – міжнародні (eCF) та державні (Державний стандарт підготовки фахівців за спеціальністю 121 «Інженерія програмного забезпечення»). В результаті виокремлено дев'ять фахових компетентностей з розробки програмного забезпечення (визначення вимог, моделювання і проєктування, розробка архітектури, забезпечення якості, життєвий цикл та підходи до розробки програмного забезпечення, інформаційні моделі даних, накопичення знань).

З метою добору методологій та інструментів розробки програмного забезпечення було проаналізовано сучасні методології розробки програмного забезпечення, інструменти гнучкої розробки, визначено базові поняття, критерії та показники добору інструментів для розробки програмного забезпечення. Аналізуючи наукові праці щодо підходів до розробки програмного забезпечення, під методологією розробки програмного забезпечення будемо розуміти сукупність методів, застосовуваних на різних стадіях життєвого циклу розробки програмного забезпечення, що мають спільний філософський підхід та, відповідно до цього підходу, дозволяють забезпечити найкращу ефективність процесів розробки. Поряд з прогнозованими та адаптивними методологіями значне місце на ринку розробки програмних продуктів займають гнучкі методології розробки програмного забезпечення (Agile software

development). В результаті аналізу джерел та на основі власного досвіду під гнучкою методологією розробки програмних продуктів будемо розуміти комплекс підходів до розробки програмного забезпечення, що орієнтований на використання ітеративних розробок, динамічне формування вимог до проєкту, забезпечення реалізації в результаті безперервного взаємодії всередині робочих самоорганізованих груп, що складаються з фахівців різного профілю.

Використання гнучких методологій та підходів до розробки програмного забезпечення змінює процес розробки, та впливає відповідно на всі її етапи. Це зумовлює використання спеціальних програм і сервісів, крім власне середовищ розробки – де програмісти пишуть код програм. Процес розробки ділиться на ітерації, спринти, завдання і виникає необхідність слідкувати за виконанням вимог. В результаті дослідження встановлено, що серед респондентів (якими були представники компаній і підприємств з різних країн, що використовують agile-методології для розробки програмного забезпечення та керування роботою команд різного профілю) найбільш популярна agile-методологія, до якої відносять такі різновиди: Скрам (58 %), Скрам доповнений Канбаном – Скрамбан (10 %), Скрам доповнений Екстремальним програмуванням (8 %).

Для завдань з розробки програмного забезпечення використовують програмні менеджери проєктів – сервіси, застосунки. В гнучких методологіях – це канбан дошки, які називають «канбан-дошка», «дошка проєктів», «agile-дошка» в залежності від інтерфейсу продукту. Головним призначенням таких сервісів – є візуалізація задач, поділ їх на етапи, що мають вигляд колонок на дошці, пріоритизація задач, призначення учасників проєкту виконання певних задач, обмеження задач у роботі. Крім цього ітеративна розробка і постійні зміни спричиняють використання систем контролю версій для всіх проєктів з розробки програмного забезпечення. Системи контролю версій використовуються для: збереження і відслідковування змін в програмному коді, можливості додавати нові частини програмного забезпечення, не зашкоджуючи роботі вже існуючих та можливість повертатися до минулих версій коду. У результаті експертної оцінки, до якої було залучено 21 експерт з ІТ-компаній, було обрано чотири інструменти гнучкої розробки програмного продукту, а саме Git (140 балів), GitHub (135 балів), Gogs (119 балів) та GitLab (96 балів). На другому етапі опитування експертів було визначено показники добору інструментів гнучкої розробки програмного продукту, за попередньо визначеними критеріями: за структурно-функціональним критерієм: наявність репозитарію; можливість керування версіями; наявність редактора коду; наявність візуалізації виконаної роботи; наявність дошки задач; можливість інтеграції; підтримка платформи; за навчально-комунікаційним критерієм: можливість створення проєкту з іншими учасниками; можливість призначення задач користувачам; можливість обговорення коду; спільнота користувачів для обміну досвідом; безкоштовна версія. Обґрунтування критеріїв добору засобів гнучкої розробки на різних етапах формування професійних

компетентностей з розробки програмного забезпечення забезпечує ефективне застосування гнучких методологій у процесі підготовки майбутніх фахівців з інженерії програмного забезпечення.

У другому розділі **«Методика використання гнучких методологій розробки програмного забезпечення у підготовці майбутніх програмістів»** розглянуто суть гнучкого agile-навчання та наукові підходи до його застосування при підготовці майбутніх інженерів-програмістів, модель методики використання гнучких методологій для формування професійних компетентностей з розробки програмного забезпечення у майбутніх фахівців з програмування, обґрунтовано етапи методики використання гнучких методологій розробки програмного забезпечення у підготовці майбутніх програмістів.

Обґрунтована та розроблена модель використання гнучких методологій розробки програмних продуктів у підготовці майбутніх програмістів складається з шести блоків: цільовий, концептуальний, методико-технологічний, змістовно-процесуальний, діагностичний і результативний (рис. 1). Розроблена модель має на меті описати методику використання гнучких методологій програмного забезпечення у підготовці майбутніх програмістів. Крім того, для досягнення навчальних цілей використовуються гнучкі підходи і принципи в організації навчальної діяльності, задля забезпечення кращих навчальних результатів в умовах сучасної вищої освіти.

До цільового блоку відносяться мета та завдання, яких хочемо досягти. Навчання в закладах вищої освіти України ґрунтується на компетентнісному підході, тому за мету було обрано підвищення рівня сформованості компетентностей з розробки програмного забезпечення з використанням гнучких методологій. Для майбутніх інженерів програмістів фаховими компетентностями є саме компетентності з розробки програмного забезпечення, як провідної діяльності. Для досягнення цієї мети було поставлено завдання: організувати навчальний процес з використанням принципів гнучких методологій та використанням гнучких методологій розробки програмного забезпечення на різних етапах підготовки майбутніх програмістів.

В маніфесті гнучкої розробки, на який спираємось як на концептуальну основу гнучких методологій, основою є ідеї і принципи, а в розробленій моделі концептуальний блок формують стандарти, рамки кваліфікацій і рекомендації стейкхолдерів, компетентності з розробки програмного забезпечення, принципи і теорії навчання та гнучкі методології. Це ті ідеї, засади, концепції і документи, якими будемо керуватися для досягнення поставленої мети. До стандартів віднесено затверджений Міністерством освіти і науки України стандарт вищої освіти за спеціальністю 121 «Інженерія програмного забезпечення» першого (бакалаврського) рівня вищої освіти та відповідну освітню програму, а також навчальний план підготовки майбутніх фахівців з інженерії програмного забезпечення за вищезазначеною спеціальністю, розроблені і затверджені Національним університетом біоресурсів і природокористування України. З цього стандарту вищої освіти виокремлено компетентності, що відносяться до розробки програмного забезпечення.

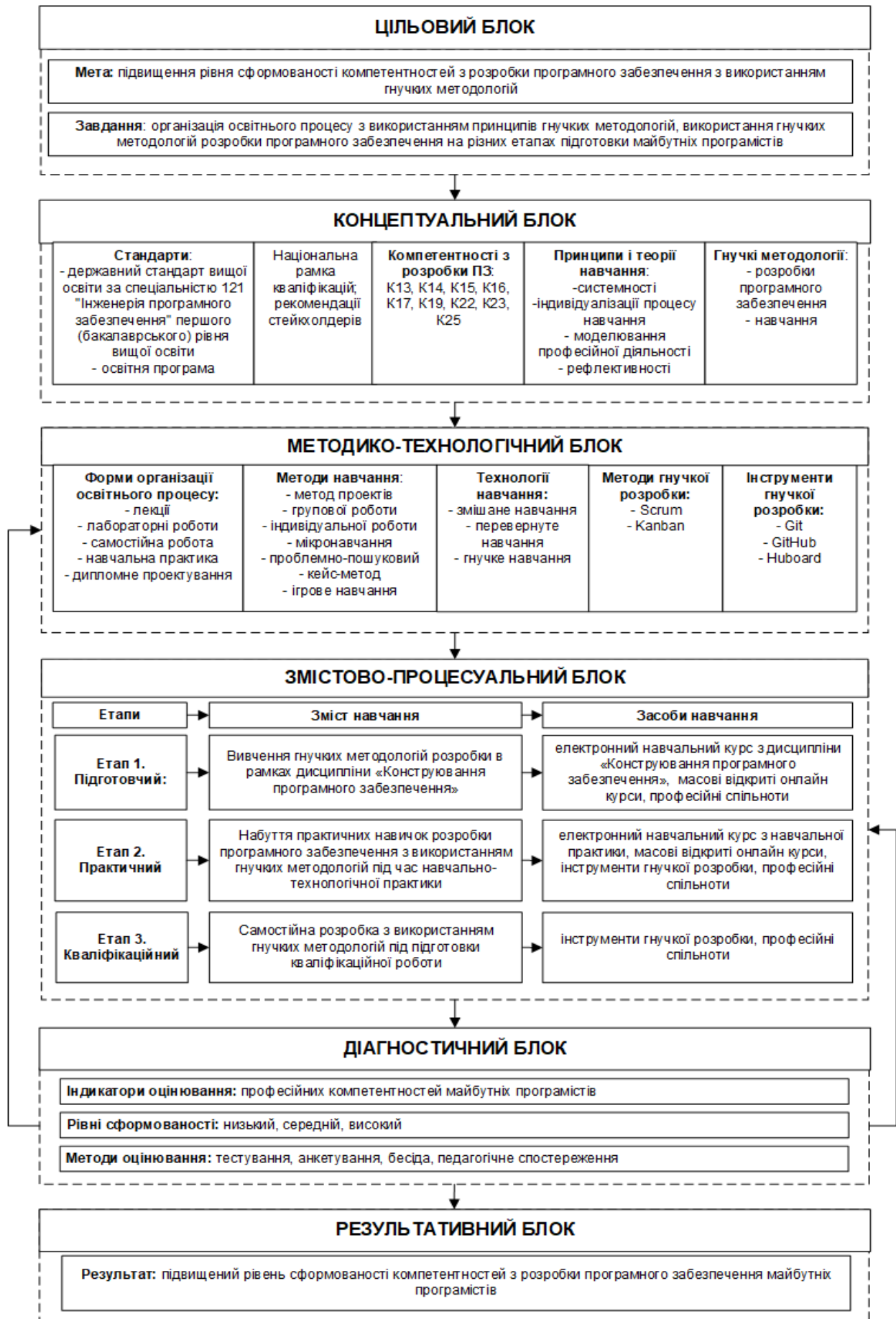


Рис. 1. Модель методики використання гнучких методологій розробки програмного забезпечення у підготовці майбутніх програмістів

Обґрунтування методики використання гнучких методологій розробки програмних продуктів у підготовці майбутніх програмістів здійснювалося на основі принципів і теорій навчання, зокрема: принцип системності, індивідуалізації процесу навчання, моделювання професійної діяльності та рефлексивності. Крім того, для обґрунтування методики було використано методології розробки програмного забезпечення, що відносяться до гнучких. Гнучкі методології розробки ґрунтуються на agile-маніфесті, що складається з основних ідей і принципів, основою яких є готовність до змін і постійна взаємодія учасників процесу. Ці принципи та ідеї були також адаптовані різними науковцями та освітянами під освітній процес і навчальні цілі. Гнучкі технології навчання передбачають динамічність, гнучкість, режим найбільшого сприяння для реалізації індивідуальних інтересів, можливостей і здібностей студентів. У розробленій методиці передбачено керування цими принципами та ідеями як для організації навчального процесу, так і для розробки програмних продуктів.

До методико-технологічного блоку віднесено: форми організації освітнього процесу, методи навчання, технології навчання, методи гнучкої розробки та інструменти гнучкої розробки. До форм організації освітнього процесу належать: аудиторні заняття – лекції, лабораторні роботи, навчальна практика, позааудиторні – самостійна робота та дипломне проектування.

Серед методів навчання у розробленій моделі запропоновано використати: метод проєктів, групової роботи, індивідуальної роботи, мікронавчання, проблемно-пошуковий, кейс-метод та ігрове навчання. Метод проєктів передбачає вирішення певної проблеми шляхом планування і виконання практичних завдань – проєктів з метою здобуття практичних навичок і поглиблення знань. Метод проєктів гармонічно поєднується з методом групової роботи. Під час групової роботи студенти об'єднуються задля досягнення визначених навчальних цілей, на противагу індивідуальній роботі, де студент самостійно вирішує поставлені завдання. Мікронавчання (Microlearning) має за свою суть розбиття інформації на маленькі шматки, легкі для усвідомлення, що можна опанувати за час до 15 хвилин. Мікронавчання робить навчання більш доступним в будь-якому місці в будь-який час, що знаходить своє місце у змішаному і перевернутому навчанні. Проблемно-пошуковий метод передбачає здійснення аналізу різних ситуацій і проблем, спочатку формулюючи проблему, здійснюючи аналіз проблеми, та пошук можливих рішень. Формулювання проблеми і пошук можливих рішень відповідають етапам збору і формулювання вимог та моделювання в життєвому циклі програмного забезпечення. Проблемно-пошуковий метод часто зустрічається в літературі як проблемне навчання. Для реалізації розробленої методики застосовано інший вид проблемного навчання – рольові ігри, коли крім визначення проблеми студентам дається ще роль, з позиції якої вони шукають рішення. Рольова гра відіграє важливу роль в активізації навчання. Рольова гра є одним з методів навчання, що допомагає студентам справлятися з невизначеністю. Рольова гра ставить їх у ситуацію, що включає обмеження, які існують у реальному світі. В Скрамі та інших гнучких методологіях, всі учасники поділені на ролі в середині команд, що робить рольову гру

важливим інструментом для розвитку компетентностей майбутніх програмістів. Всі методи навчання спрямовані на вирішення дидактичних завдань щодо розробки програмного забезпечення.

Технології навчання, які використані у розробленій моделі, відрізняються від традиційного навчання своєю гнучкістю і адаптивністю до потреб студента та навчальних цілей: змішане навчання, перевернуте навчання та гнучке навчання (agile-навчання). Застосування методів гнучкої розробки програмного забезпечення при підготовці майбутніх програмістів тісно пов'язане з методами навчання, які при цьому застосовуються. Важливим чинником при цьому є підбір таких методів навчання, які будуть максимально інтегруватися з гнучкими методами розробки. В цьому контексті проаналізовано та порівняно підходи Agile (гнучке навчання що застосовує agile-цінності та принципи гнучких методологій розробки програмного забезпечення в навчальних цілях для кращої співпраці та гнучкості) та ADDIE (класичний підхід в навчальному дизайні що розшифровується як: аналіз, проектування, розробка, впровадження та оцінка) у навчальному проектуванні. При цьому визначено, що використовується два терміни для гнучкого навчання: «agile learning» і «flexible learning». Термін «flexible learning» – є більш широким у значенні навчання, що є гнучким в усіх формах, де учень самостійно вирішує що, де, коли і як буде вивчати, тож надалі під «гнучким навчанням» будемо розуміти саме це. Термін «agile learning» відноситься до гнучких методологій розробки програмного забезпечення, а саме адаптації принципів, та ідей Agile-маніфесту, а також Agile-методологій та їх окремих інструментів до навчального процесу, тому для його визначення будемо використовувати термін «agile-навчання». Оскільки гнучке навчання застосовує процеси та принципи гнучкої розробки програмного забезпечення до контексту навчання, будемо визначати «agile-навчання» як: процес навчання, в якому використовуються принципи, методи і інструменти гнучких методологій розробки програмного забезпечення в навчальних цілях та побудований на основі коротких проектних циклів, які називаються «спринтами», в яких повністю планується, проектується, будується, випробовується, переглядається та запускається повна версія навчального проекту. Протягом декількох спринтів студенти ітеративно розширюють та вдосконалюють розроблений навчальний проект. Гнучке «agile-навчання» відрізняється від традиційного проектного навчання, яке характерне лінійним процесом виконання розробки.

Навчання з використанням гнучкості багато в чому нагадує змішане навчання (blended learning), але воно перекладає більше контролю та відповідальності на студентів. Це дає їм свободу досліджувати різні навчальні середовища та з'ясувати, що для них найкраще підходить. Також це допомагає студентам розвивати свої здібності до самооцінки та відстеження свого прогресу.

Для розробки програмного забезпечення обрано методи гнучкої розробки Скрам (Scrum) та Канбан (Kanban), які відносяться до методологій гнучкої розробки програмного забезпечення (Agile). Скрам є найбільш популярним фреймворком серед ІТ-компаній та набуває популярності в керуванні проектами

в інших сферах. Канбан являє собою більш конкретні практики, якими, як показали дослідження, ефективно можна доповнити методологію Скрам. Agile, Скрам і Канбан віднесли до методів, а не до методологій, оскільки будемо брати основні методи з цих методологій, які є об'ємними і їх впровадження та масштабування їх під навчальні проєкти повністю є окремою об'ємною задачею.

Серед інструментів гнучкої розробки на основі експертної оцінки було обрано Git, GitHub та Huboard. Git – єдиною системою контролю версій на основі зліпків системи, що зробило її всесвітнім лідером серед систем контролю версій. Постійне оновлення програмного коду, призводить до необхідності надійного збереження написаного програмного коду, та можливості в будь-який момент повернутися до іншої версії. Крім того, Git має систему галуження для копіювання часини коду у власний репозиторій і відправки у основний, після того як новий фрагмент буде дописано і протестовано, при цьому злиття гілок контролюється і захищене від конфліктів версій. Проте Git не має графічного інтерфейсу, тому було обрано GitHub для зручності і наочності роботи з репозиторіями. Крім цього, GitHub є соціальною мережею серед розробників, що дозволяє ділитися своїми розробками, дивитися, що розробляють інші учасники, приймати участь у відкритих проєктах з розробки програмного забезпечення з відкритим кодом (open source) і створювати колективні проєкти. GitHub має редактор коду та можливість інтегрувати інші сервіси, прив'язавши їх до свого репозиторію і навіть до конкретних задач. Huboard представляє собою інструмент для керування, менеджменту проєктів у вигляді канбан-дошки. Сервіс легко інтегрується з GitHub і є безкоштовним, що стало ключовим для його вибору.

До змістовно-процесуального блоку відноситься зміст, етапи та засоби навчання. Для реалізації методики визначено три етапи, для кожного з яких обґрунтовується відповідний зміст навчання з використанням гнучких методологій розробки програмного забезпечення та засоби, які будуть використані в процесі навчання:

Етап 1 – Підготовчий: включає в себе ознайомлення з гнучкими методологіями в рамках дисципліни «Конструювання програмного забезпечення» під час лекційних і лабораторних робіт та самостійної роботи студентів. Теоретичний матеріал для студентів розміщується в електронному навчальному курсі дисципліни, що доступний студентам онлайн в будь-який час з будь-якого місця. За принципами перевернутого навчання перед аудиторним заняттям студентам пропонується ознайомитися з матеріалом, а під час заняття його обговорювати з викладачем, який охоплює весь передбачений матеріал користуючись методами проблемної лекції, кейс методом з елементами рольової гри. Для вирішення завдань студент може користуватися матеріалами з електронного навчального курсу, професійних спільнот, де розробники діляться цінним передовим досвідом, що не може забезпечити жоден підручник, та масовими онлайн курсами, запропонованими викладачем, або обраними самостійно.

Етап 2 – Практичний: відбувається під час проєктно-технологічної практики, де студенти виконують груповий проєкт за методологією Скрам, використовуючи також інструменти і підходи Канбан, обрані вище інструменти гнучкої розробки та керуються принципами та ідеями маніфесту гнучкої розробки. Перед виконанням проєкту для студентів проводяться практичні заняття та дві рольові гри за темою розробки програмних продуктів з використанням гнучких методологій. Темі для навчальних проєктів і задачі визначають команди самостійно під час мозкового штурму. Під час проєкту студенти працюють в командах за визначеними ролями, які обирають самостійно, здійснюючи розробку ітеративно. Студенти самі визначають цілі на кожну ітерацію і ставлять собі задачі щодо розробки програмного продукту, знаходять потрібний матеріал на масових відкритих онлайн курсах, професійних спільнотах та в електронному навчальному курсі з навчально-технологічної практики. Після завершення і презентації проєкту проводиться ретроспектива як форма рефлексії – оцінюється успішність виконання проєкту, що можна було покращити, що було добре, чому навчилися і що зрозуміли студенти під час практики.

Етап 3 – Кваліфікаційний. Студенти працюють самостійно над виконанням кваліфікаційної роботи під керівництвом викладача. Роботи організовується за принципами гнучкої розробки та гнучкого навчання. Робота планується ітераціями і розбивається на задачі, які пріоритизуються. Оскільки найважливіші задачі виконуються першими, а процес роботи прозорий, то підвищується вірогідність якісного виконання студентських кваліфікаційних робіт готовими у відведений строк. Студент з викладачем зустрічаються в аудиторії або онлайн для представлення результатів роботи і обговорення подальших задач. Отримання зворотного зв'язку частіше і гнучке планування задач мотивує до постійного темпу розробки.

Отже, на першому етапі студенти знайомляться з гнучкими методологіями розробки; на другому етапі знайомляться з ними глибше на практиці, навчаються за методологіями розробки програмних продуктів Скрам, Канбан та використовують на практиці інструменти гнучкої розробки, виконуючи групові проєкти; на третьому етапі студенти самостійно використовують всі набуті знання, виконуючи кваліфікаційну роботу. Компетентнісний підхід передбачає орієнтацію всіх компонентів навчального процесу на здобуття майбутніми фахівцями компетентностей, необхідних для професійної діяльності. Основна ідея полягає в тому, що головний результат освіти – це не окремі знання, вміння та навички, а комплексне засвоєння знань і способів практичної діяльності, що забезпечать людині можливість успішно реалізувати себе в різних галузях своєї життєдіяльності. Тому увага сконцентрована на практичній підготовці, як найбільш ефективному чиннику формування професійних компетентностей майбутніх програмістів.

До діагностичного блоку відносяться розроблені індикатори оцінювання професійних компетентностей з програмування майбутніх інженерів програмістів, рівні сформованості компетентностей: низький, середній, високий, та методи оцінювання: тестування, анкетування, бесіда, педагогічне

спостереження. Основними методами оцінювання обрано: анкетування, тестування, аналіз результатів успішності та педагогічне спостереження. Знання та уміння вимірюватимуться за допомогою анкетування, тестування та аналізу результатів успішності. Комунікація, автономія та відповідальність, які є серед дескрипторів не відносяться до обраних компетентностей. Відповідно до стандарту вищої освіти за спеціальністю 121 «Інженерія програмного забезпечення» першого (бакалаврського) рівня вищої освіти кожній компетентності відповідає один або декілька дескрипторів: знань, умінь, комунікації, автономії відповідальності, що можуть мати різні рівні. Отже, для оцінки рівнів сформованості компетентностей студентів будемо оцінювати їх знання та уміння. Для цього було розроблено індикатори до кожної обраної компетентності з розробки програмного забезпечення відповідно.

У результативному блоці відображено очікуваний результат розробленої методики: підвищення рівня сформованості компетентностей з розробки програмного забезпечення майбутніх програмістів.

У третьому розділі **«Експериментальна перевірка методики використання гнучких методологій розробки програмного забезпечення у підготовці майбутніх програмістів»** було розкрито організацію, зміст дослідно-експериментальної роботи, проаналізовано результати констатувального, формувального етапів.

На першому, констатувальному, етапі (2018–2019 рр.) проаналізовано наукові дослідження з підготовки майбутніх програмістів, вимоги ринку праці, використання гнучких методологій розробки програмного забезпечення, обґрунтовано модель методики використання гнучких методологій розробки програмного забезпечення у підготовці майбутніх програмістів; визначено рівні сформованості компетентностей з програмування в процесі навчання майбутніх програмістів.

На другому, формувальному, етапі (2019–2020 рр.) було обґрунтовано та апробовано методику використання гнучких методологій розробки програмного забезпечення у підготовці майбутніх програмістів, а саме: коригування робочих навчальних програм з дисциплін «Конструювання програмного забезпечення», проєктно-технологічної практики; розробка електронних навчальних курсів. Формувальний етап експерименту здійснювався відповідно до 3-етапної методики. На підготовчому етапі здійснювалося ознайомлення з гнучкими методологіями всіх студентів в рамках дисципліни «Конструювання програмного забезпечення» під час лекційних і лабораторних робіт та самостійної роботи студентів. В результаті першого етапу було сформовано експериментальну та контрольну групу. Для перевірки груп на однорідність було застосовано дисперсійний аналіз. Під час практичного етапу студенти виконували груповий проєкт зі створення програмного продукту в ході проєктно-технологічної практики. При цьому, в експериментальній групі було використано гнучкі методології розробки та відповідні інструменти, а в контрольній групі для розробки програмного продукту студенти використовували каскадну модель (лінійний проєкт). На кваліфікаційному етапі студенти експериментальних груп працювали

самостійно над виконанням кваліфікаційної роботи під керівництвом викладача за принципами гнучкої розробки та гнучкого навчання, а також з використання інструментів гнучкої розробки програмних продуктів.

Дослідження динаміки змін рівня сформованості компетентностей з програмування майбутніх програмістів проводилося в умовах проходження проектно-технологічної практики та розробки кваліфікаційної роботи.

Для оцінки рівня сформованості фахових компетентностей студентів було розроблено технічне тестування, що складається з 90 питань різних видів: одновибіркові і багатовибіркові питання, відкриті питання, співставлення, результати якого подано у табл. 1. З метою доведення статистичної гіпотези про значимість різниці отриманих результатів в контрольній і експериментальній групах було використано таблиці спряженості, критерії Пірсона та Стюдента. Сформовано нульову гіпотезу: про відсутність відмінностей між рівнями сформованості компетентностей з розробки програмного забезпечення студентами контрольних і експериментальних груп та альтернативну гіпотезу про наявність значимих відмінностей між рівнями сформованості компетентностей з програмування студентами контрольних і експериментальних груп. Розраховане значення критерія Пірсона за результатами констатувального зрізу (для рівня значимості 0,05 та 2 ступенів свободи) менше від критичного: $0,741 < 5,991$, отже, нульову гіпотезу про відсутність відмінностей між рівнями сформованості компетентностей з розробки програмного забезпечення у студентів контрольних і експериментальних груп на початку експерименту приймаємо.

Таблиця 1

**Розподіл студентів у контрольних та експериментальних групах
за рівнями компетентностей з програмування
за результатами констатувального та кінцевого зрізів**

Рівень	Констатувальний зріз (кількість студентів)		Кінцевий зріз (кількість студентів)	
	контрольна група	експериментальна група	контрольна група	експериментальна група
Низький	24	32	18	14
Середній	86	88	80	59
Високий	72	77	84	124
Всього	182	197	182	197

Якщо на початку експерименту за результатами констатувального зрізу характеристики (рівні сформованості компетентностей з розробки програмного забезпечення у студентів) контрольних та експериментальних груп збігаються і підтверджується нульова гіпотеза, то вже наприкінці експерименту за результатами кінцевого зрізу доведена достовірність значущих відмінностей характеристик контрольних та експериментальних груп, що підтверджує альтернативну гіпотезу (розраховане значення критерію Пірсона більше від критичного та становить 10,783).

Порівняльний розподіл студентів у контрольних та експериментальних групах за рівнями сформованості компетентностей з розробки програмного

забезпечення подано на рис. 2. Порівнюючи рівні сформованості компетентностей з розробки програмного забезпечення майбутніх програмістів у експериментальних групах на початку та наприкінці педагогічного експерименту, спостерігаємо збільшення частки студентів, які мають високий та середній рівні сформованості зазначеної компетентності.

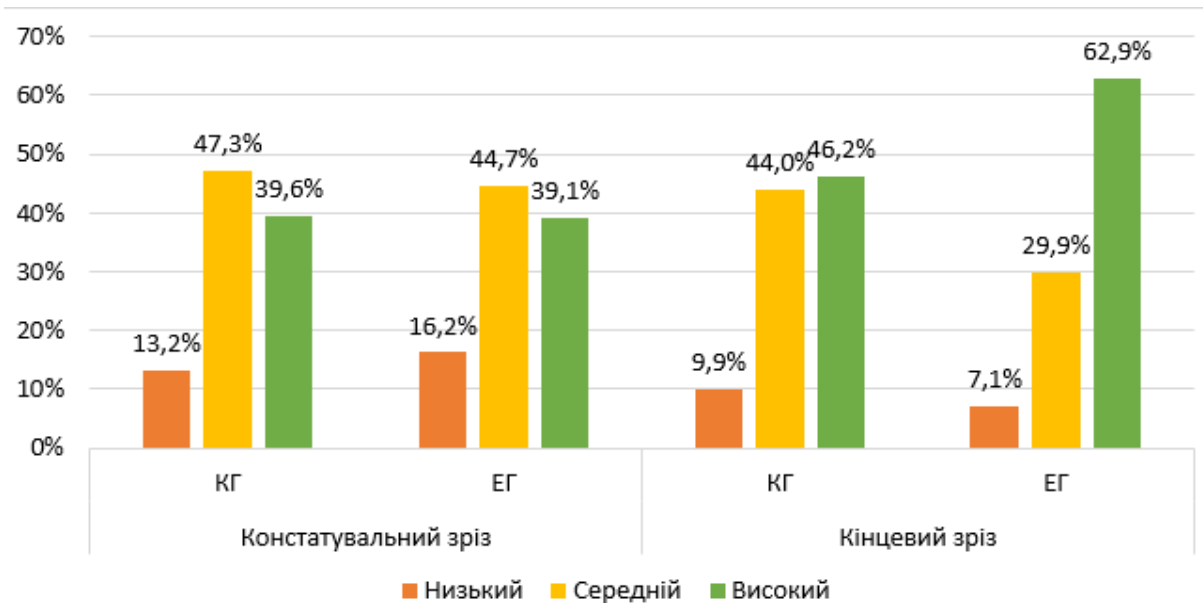


Рис. 2. Діаграма порівняльного розподілу студентів у контрольних та експериментальних групах за рівнями сформованості компетентностей з програмування

Крім цього, було проведено опитування студентів після самостійного проходження третього етапу – написання кваліфікаційної роботи, де студентам пропонувалося працювати над проектом, використовуючи гнучкі методології, що підтвердило вищий рівень сформованості компетентностей з розробки програмного забезпечення у студентів експериментальних груп. Особливої уваги заслуговують закономірності, що випливають з результатів другого та третього етапів в розрізі конкретних компетентностей. Експериментальна перевірка методики свідчить про підвищення рівня сформованості професійних компетентностей з розробки програмного забезпечення в експериментальних групах, особливо компетентностей, що стосуються визначення вимог до програмного забезпечення, забезпечення якості програмного забезпечення та реалізації етапів життєвого циклу.

Для підтвердження отриманих результатів після третього етапу було проведено оцінювання рівня сформованості компетентностей з розробки програмного забезпечення та висунуто гіпотезу, що середнє значення рівня сформованості даних компетентностей у контрольній та експериментальній групах не відрізняється. Перевірка даної гіпотези за критерієм Стюдента показала, що фактичне значення критерію $t_{\phi}=3,5$ перевищує критичний $t_{кр}=2,12$ для заданого рівня значимості (0,05), який необхідний для відхилення нульової гіпотези про рівність двох середніх. Отже, можна зробити висновок, що різниця середніх оцінок сформованості компетентностей з розробки програмного

забезпечення між контрольною та експериментальною групами статистично значима і становить 14,34 %.

Таким чином проведений педагогічний експеримент підтвердив гіпотезу дослідження, а аналіз його результатів підтвердив ефективність розробленої методики.

ВИСНОВКИ

Відповідно до мети та завдань дисертаційного дослідження отримано наступні основні результати: з'ясовано стан розробленості проблеми застосування гнучких методологій навчання розробки програмного забезпечення у підготовці майбутніх програмістів, уточнено понятійний апарат дослідження; проаналізовано наявні гнучкі методи розробки програмних продуктів та гнучкі методи навчання; визначено критерії та показники добору інструментів гнучкої розробки для використання у процесі підготовки майбутніх програмістів, рівні сформованості фахових компетентностей з розробки програмного забезпечення майбутніх програмістів; розроблено модель методики використання гнучких методологій розробки програмного забезпечення; експериментально перевірено педагогічну доцільність методики застосування гнучких методологій розробки програмного забезпечення у підготовці майбутніх програмістів.

Одержані результати дослідження дають підстави зробити наступні висновки.

1. Гнучкі методології розробки програмного забезпечення є невід'ємною складовою підготовки майбутніх програмістів. Під гнучкою методологією розробки програмного забезпечення варто розуміти комплекс підходів до розробки програмного забезпечення, що орієнтований на використання ітеративних розробок, динамічне формування вимог до проєкту, забезпечення реалізації в результаті безперервної взаємодії всередині робочих самоорганізованих груп, що складаються з фахівців різного профілю. Побудовано узагальнену систематизацію методів гнучкої розробки, у результаті чого було визначено методи Скрам та Канбан для використання у навчальному процесі майбутніх програмістів. В результаті аналізу вітчизняного та закордонного досвіду підготовки майбутніх програмістів у закладах вищої освіти встановлено, що важливою є розробка критеріїв добору засобів гнучкої розробки на різних етапах формування професійних компетентностей з розробки програмного забезпечення.

2. Добір інструментів гнучкої розробки програмного забезпечення доцільно здійснювати на основі таких критеріїв та показників: структурно-функціональний критерій (наявність репозиторію; можливість керування версіями; наявність редактора коду; наявність візуалізації виконаної роботи; наявність дошки задач; можливість інтеграції; підтримка платформи), навчально-комунікаційний критерій (можливість створення проєкту з іншими учасниками; можливість призначення задач користувачам; можливість обговорення коду; спільнота користувачів для обміну досвідом; безкоштовна

версія). Водночас, встановлено суттєву залежність результату оцінювання програмних засобів за пропонованими критеріями і показниками від ваги показників і, як наслідок, потребу добору інструментів гнучкої розробки на різних етапах підготовки майбутніх програмістів у кожному окремому випадку.

3. Виокремлено фахові компетентності майбутніх фахівців з програмування щодо розробки програмного забезпечення: здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення; здатність брати участь у проєктуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування; здатність розробляти архітектури, модулі та компоненти програмних систем; здатність формулювати та забезпечувати вимоги щодо якості програмного забезпечення у відповідності з вимогами замовника, технічним завданням та стандартами; здатність дотримуватися специфікацій, стандартів, правил і рекомендацій в професійній галузі при реалізації процесів життєвого циклу; володіння знаннями про інформаційні моделі даних, здатність створювати програмне забезпечення для зберігання, видобування та опрацювання даних; здатність накопичувати, обробляти та систематизувати професійні знання щодо створення і супроводження програмного забезпечення та визнання важливості навчання протягом всього життя; здатність реалізовувати фази та ітерації життєвого циклу програмного забезпечення на основі відповідних моделей і підходів розробки програмного забезпечення. Діагностування рівня сформованості фахових компетентностей з розробки програмного забезпечення доцільно здійснювати на основі розроблених індикаторів за трьома рівнями сформованості: низький, середній, високий.

4. Гнучкі методології розробки програмного забезпечення ґрунтуються на Agile-маніфесті, що складається з основних ідей і принципів, основою яких є готовність до змін і постійна взаємодія учасників процесу. Ці принципи і ідеї були також адаптовані під освітній процес і навчальні цілі в підготовці майбутніх програмістів для розробки програмних продуктів та використовувати методології, що відносяться до гнучких. Гнучке «agile-навчання» застосовує процеси та принципи гнучкої розробки програмного забезпечення до контексту навчання і визначено як процес навчання, що побудований на основі коротких проєктних циклів, які називаються «спринтами», в яких повністю планується, проєктується, будується, випробовується, переглядається та запускається повна версія розробленого навчального проєкту. Гнучкі технології навчання передбачають динамічність, гнучкість, режим найбільшого сприяння для реалізації індивідуальних інтересів, можливостей і здібностей студентів.

5. Модель методики використання гнучких методологій розробки програмних продуктів у підготовці майбутніх програмістів складається з шести блоків: цільовий, концептуальний, методико-технологічний, змістовно-процесуальний, діагностичний і результативний, і має на меті підвищити рівень сформованості компетентностей з розробки програмного забезпечення з використанням гнучких методологій за допомогою визначеного змісту, форм,

методів навчання та інструментів протягом трьох етапів: підготовчого, практичного та кваліфікаційного. Застосування гнучких методологій розробки програмних продуктів у підготовці майбутніх програмістів доцільно здійснювати за розробленою методикою, в основу якої покладено послідовну реалізацію трьох етапів. На першому етапі – підготовчому, – відбувається вивчення гнучких методологій в рамках дисципліни «Конструювання програмного забезпечення» під час лекційних і лабораторних робіт та самостійної роботи студентів. На другому етапі – практичному, – відбувається застосування гнучких методів розробки під час навчально-технологічної практики, де студенти виконують груповий проєкт за методологією Скрам, використовуючи також інструменти і підходи Канбан, інструменти гнучкої розробки та керуються принципами та ідеями маніфесту гнучкої розробки. На третьому – кваліфікаційному етапі, студенти працюють самостійно над виконанням кваліфікаційної роботи під керівництвом викладача. Роботи організовується за принципами гнучкої розробки та гнучкого навчання.

6. Експериментальна перевірка методики свідчить про підвищення рівня сформованості професійних компетентностей з розробки програмного забезпечення в експериментальних групах, особливо компетентностей, що стосуються визначення вимог до програмного забезпечення, забезпечення якості програмного забезпечення, та реалізації етапів життєвого циклу. Аналіз результатів після другого та третього етапів методики дає підстави стверджувати про зростання рівня сформованості компетентностей з розробки програмного забезпечення у експериментальних групах на 14,34 %, порівняно з контрольними групами.

Перспективи подальших досліджень у даному напрямку полягають, на наш погляд, у розв'язанні таких проблем: теоретичне обґрунтування та розробка методики гнучкого навчання майбутніх фахівців з інформаційних технологій за технологією «agile – навчання»; науково-методичне обґрунтування застосування гнучких методів навчання під час реалізації змішаного та дистанційного навчання.

СПИСОК ОПУБЛІКОВАНИХ ПРАЦЬ ЗА ТЕМОЮ ДИСЕРТАЦІЇ

Статті у наукових фахових виданнях України,

у тому числі включених до міжнародних наукометричних баз даних

1. 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. *(Здобувачем проаналізовано можливості хмарних ресурсів та сервісів Microsoft для неформальної освіти студентів ІТ-фаху, зокрема ресурси щодо інженерії програмного забезпечення).*

2. Волошина Т., Глазунова О., Гуржій А., **Пархоменко О.**, Корольчук В. Платформи та системи автоматизованої перевірки завдань з програмування: аналіз, критерії добору та приклад використання. Відкрите освітнє е-середовище сучасного університету. 2020. № 8. С. 154–164. *(Здобувачем проаналізовано наявні платформи та системи автоматизованої перевірки завдань*

із програмування та їх застосування для формування професійних компетентностей майбутніх програмістів).

**Статті у наукових виданнях,
включених до міжнародних наукометричних баз даних
Scopus/Web of Science**

3. Glazunova O., Voloshyna T., Korolchuk V., **Parhomenko O.** Cloud-oriented environment for flipped learning of the future IT specialists. E3SWC 2020. Vol. 166. № 10014. URL: https://www.e3s-conferences.org/articles/e3sconf/abs/2020/26/e3sconf_icsf2020_10014/e3sconf_icsf2020_10014.html *(Здобувачем досліджено хмарні сервіси і ресурси для використання на різних етапах організації перевернутого навчання майбутніх IT фахівців).*

4. Glazunova O., Voloshyna T., Gurzhii A., Korolchuk V., **Parhomenko O.**, Sayapina T., Semyhinivska T. Cloud resources and services for development of self-educational competence of future IT specialists: business process modelling and examples of using. CEUR Workshop Proceedings. 2020. № 2732. P. 591–606. URL: <http://ceur-ws.org/Vol-2732/20200591.pdf>. *(Здобувачем обґрунтовано та розроблено моделі процесу використання сервісів для проектної командної роботи майбутніх IT-фахівців).*

5. Glazunova O., Morze N., Golub B., Burov O., Voloshyna T., **Parhomenko O.** Learning Style Identification System: Design and Data Analysis. CEUR Workshop Proceedings. 2020. № 2732. P. 793–807. URL: <http://ceur-ws.org/Vol-2732/20200793.pdf>. *(Здобувачем проаналізовано домінуючий навчальний стиль майбутніх програмістів).*

6. Glazunova O., **Parhomenko O.**, Korolchuk V., Voloshyna T. The effectiveness of GitHub cloud services for implementing a programming training project: students' point of view. Journal of Physics: Conference Series. 2021. Vol. 1840. URL: <https://iopscience.iop.org/article/10.1088/1742-6596/1840/1/012030>. *(Здобувачем обґрунтовано та описано методики використання сервісу GitHub в підготовці майбутніх програмістів, проаналізовано і описано застосування гнучких методологій розробки програмного забезпечення в навчанні майбутніх програмістів, проаналізовано домінуючий навчальний стиль майбутніх програмістів).*

Тези наукових доповідей

7. **Пархоменко О. В.**, Кодинець М. А. Переваги і недоліки гнучкої розробки програмного забезпечення. Глобальні та регіональні проблеми інформатизації в суспільстві і природокористуванні '2014: II Міжнародна науково-практична конференція. Київ, 2014. С. 68–70. *(Здобувачем проаналізовано поняття гнучких методологій розробки).*

8. Пархоменко О. В. Сучасні підходи до розробки програмного забезпечення в циклі дисциплін з програмування. Інформаційні технології: економіка, техніка, освіта '2018: IX Міжнародна науково-практична конференція молодих вчених. Київ, 2018. С. 289–290.

9. Пархоменко О. В. Використання GitHub для формування професійних та особистісних компетентностей у майбутніх програмістів. Глобальні та регіональні проблеми інформатизації в суспільстві і природокористуванні '2019: II Міжнародна науково-практична конференція. Київ, 2019. С. 248–251.

10. Корольчук В., **Пархоменко О.** Хмарний сервіс github як ефективний інструмент для проектної роботи майбутніх програмістів. Глобальні та регіональні проблеми інформатизації в суспільстві і природокористуванні '2020: III Міжнародна науково-практична конференція. Кривий Ріг, 2020. С. 177–180. *(Здобувачем обґрунтовано переваги сервісу GitHub для організації проектного навчання майбутніх програмістів та результати експериментального дослідження).*

АНОТАЦІЯ

Пархоменко О. В. Використання гнучких методологій розробки програмного забезпечення у підготовці майбутніх програмістів. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня кандидата педагогічних наук за спеціальністю 13.00.04 «Теорія та методика професійної освіти». Національний університет біоресурсів і природокористування України. Київ, 2021.

У дисертації досліджується проблема використання гнучких методологій розробки програмного забезпечення у підготовці майбутніх програмістів. Уточнено сутність основних понять дослідження та поняття «agile-навчання». Теоретично обґрунтовано вибір інструментів і методів гнучкої розробки для впровадження в навчальний процес майбутніх програмістів. Висвітлено сучасні тенденції розробки програмного забезпечення, популярні гнучкі методології розробки програмного забезпечення, їх принципи, методи, інструменти та їх використання у навчанні. Розроблено критерії добору інструментів гнучкої розробки. Розроблено та перевірено ефективність методики використання гнучких методологій розробки програмного забезпечення у підготовці майбутніх програмістів, що має на меті підвищити рівень сформованості компетентностей з розробки програмного забезпечення з використанням гнучких методологій, та базується на трьох етапах: «Підготовчому» – вивчення гнучких методологій розробки; «Практичному» – застосування гнучких методів навчання та розробки під час навчально-технологічної практики; «Кваліфікаційному» – виконання кваліфікаційної роботи під керівництвом викладача з використання методів гнучкої розробки проекту.

Ключові слова: гнучкі методології розробки програмного забезпечення, agile-навчання, компетентності з програмування, інженерія програмного забезпечення.

АННОТАЦИЯ

Пархоменко О. В. Использование гибких методологий разработки программного обеспечения в подготовке будущих программистов. – Квалификационная научная работа на правах рукописи.

Диссертация на соискание ученой степени кандидата педагогических наук по специальности 13.00.04 «Теория и методика профессионального образования». Национальный университет биоресурсов и природопользования Украины. Киев, 2021.

В диссертации исследуется проблема использования гибких методологий разработки программного обеспечения в подготовке будущих программистов. Уточнена сущность основных понятий исследования и понятия «agile-обучения». Теоретически обоснован выбор инструментов и методов гибкой разработки для внедрения в учебный процесс будущих программистов. Освещены современные тенденции разработки программного обеспечения, популярные гибкие методологии разработки программного обеспечения, их принципы, методы, инструменты и их использование в обучении. Разработаны критерии отбора инструментов гибкой разработки. Разработана и проверена эффективность методики использования гибких методологий разработки программного обеспечения в подготовке будущих программистов, с целью повысить уровень сформированности компетенций по разработке программного обеспечения с использованием гибких методологий, что базируется на трех этапах: «Подготовительном» – изучение гибких методологий разработки; «Практическом» – применение гибких методов обучения и разработки во время учебно-технологической практики; «Квалификационном» – выполнение квалификационной работы под руководством преподавателя по использованию методов гибкой разработки проекта.

Ключевые слова: гибкие методологии разработки программного обеспечения, agile-обучение, компетентности по программированию, инженерия программного обеспечения.

ANNOTATION

Parkhomenko O. V. The Use of Agile Software Development Methodologies in the Training of Future Software Developers. – Qualification scientific work on the rights of manuscripts.

The senior thesis on a scientific degree of the candidate of pedagogical sciences on a speciality 13.00.04 «Theory and Methods of Vocational Education Training». National University of Life and Environmental Sciences of Ukraine. Kyiv, 2021.

The senior thesis research investigates the problem of using agile software development methodologies in the training of future software developers.

Based on the analysis, a number of contradictions were identified that accompany the current state of training of future software engineers: between the state of development of information technologies in the world and Ukraine, which is reflected in the requirements of stakeholders and insufficient level of necessary

competencies developed in higher educational institutions; variety of tools and methods of agile software development and insufficient educational and methodological recommendations for their selection and use in the educational process; opportunities for the use of tools and methods of agile software development by future programmers and the insufficient level of development of appropriate methods of using tools and methods of agile software development in the educational process.

The essence of the basic concepts of research and the concept of agile-learning are specified. The choice of tools and methods of agile software development for introduction into the educational process of future software developers is theoretically substantiated. The modern trends of software development, popular agile methodologies of software development, their principles, methods, tools and their use in teaching and learning are covered. An agile software development tools were analyzed, two main areas were identified: agile project management tools (kanban-dashboards, agile-boards, project management systems), and version control systems, that are important for use in iterative software development in teams that is considered in agile software development. Criteria for selection of agile development tools have been developed. Based on the research the concept of agile-learning was determined as: learning process that uses the principles, methods and tools of agile methodologies for software development for educational purposes and is based on short project cycles, called «sprints», in which fully planned, designed, built, tested, the full version of the training project is reviewed and launched.

Developed and tested the effectiveness of the methodology of using agile methodologies of software development in the training of future programmers, which aims to increase the level of competencies in software development using agile methodologies. From the state standard of higher education of Ukraine in the specialty 121 «Software Engineering» of the first (bachelor's) level of higher education were selected competencies related to software development: requirements definition, modeling and design, architecture development, quality assurance, life cycle and approaches to software development, information data models, knowledge accumulation.

Model of the methodology of using agile methodologies of software development in the training of future programmers consist of three stages: «Preparatory» – study of agile development methodologies while studying the discipline «Construction and Design» in lectures, laboratory classes and independent work; «Practical» – the use of agile methods of teaching and development during technological practice to develop a group project using agile methodologies such as Scrum and Kanban to organize an agile team work on project; «Qualification» – the implementation of qualification work under the guidance of a supervisor using the methods of agile software development to develop a qualification work iteratively to prevent overtime and reach better results. The model of using agile methodologies of software development in the training of future programmers consists of six blocks: scope, conceptual, methodological, technological, procedural, diagnostic and effective, and aims

to increase the level of competencies in software development using flexible methodologies for using certain content, forms, teaching methods and tools.

To evaluate significance of using agile methodologies in educational process of software developers 3-levels evaluation system were designed. Conducted study showed rise of «middle» and «high» levels of professional competencies in experimental group compare to control group were using of agile methodologies weren't applied for technological practice. The survey conducted at the end of the experiment, after the Qualification stage, showed a significant increase in competencies related to the definition of software requirements, software quality assurance, and implementation of software life cycle stages.

Key words: agile software development, agile learning, programming competencies, software engineering.

Підписано до друку 09.04.2021 року. Формат 60x84\16
Ум. друк. арк. 0,9 Обл.-вид.арк. 0,9
Наклад 100 прим. Зам. № 210268

Віддруковано у редакційно-видавничому відділі НУБіП України
вул. Героїв Оборони, 15, Київ, 03041, тел.: 527-81-55, e-mail: nubip_druk@ukr.net
Свідоцтво суб'єкта видавничої справи ДК № 4097 від 17.06.2011

