

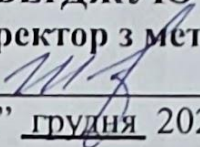
**ЗАКЛАД ВИЩОЇ ОСВІТИ
«УНІВЕРСИТЕТ КОРОЛЯ ДАНИЛА»**

Факультет суспільних і прикладних наук

Кафедра інформаційних технологій

ЗАТВЕРДЖУЮ

Проректор з методичної роботи

 **Ярослав ШТАНЬКО**
“ 19 ” грудня 2025 р.

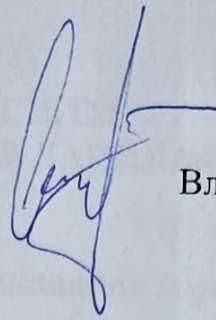
МЕТОДИ РОЗРОБКИ ПРОГРАМНИХ СИСТЕМ

СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Галузь знань:	12 Інформаційні технології
Спеціальність:	121 Інженерія програмного забезпечення
Освітньо-професійна (освітньо-наукова) програма:	Розробка та тестування програмного забезпечення
Освітній рівень:	(перший) бакалаврський
Статус дисципліни:	вибіркова
Мова викладання, навчання та оцінювання:	українська

РОЗРОБНИК:

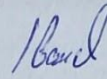
асистент кафедри ІТ



Владислав ГОЛУБЦОВ

ЗАТВЕРДЖЕНО:

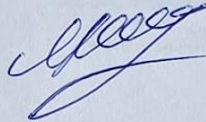
на засіданні кафедри ІТ
протокол № 5 від 18.12 2025 р.
PhD, заступник завідувача кафедри



Олександр ІВАНОВ

УЗГОДЖЕНО:

Гарант ОП
к.ф-м.н., доцент кафедри



Володимир МАКОВИШИН

СХВАЛЕНО:

на засіданні Науково-методичної ради, протокол № ____ від 19.12 2025 р.

e-mail	vladyslav.o.holubtsov@ukd.edu.ua
Номер аудиторії чи кафедри	206
Посилання на сайт	
Сторінка курсу в СДО	https://online.ukd.edu.ua/course/

ВСТУП

Анотація навчальної дисципліни

Навчальна дисципліна "Методи розробки програмних систем" є складовою освітньо-професійної програми підготовки фахівців за освітнім ступенем "бакалавр" галузі знань 12 "Інформаційні технології" спеціальності 121 "Інженерія програмного забезпечення", освітньої програми "Розробка та тестування програмного забезпечення".

Слухачі дисципліни повинні одержати теоретичні знання та практичні навички з використання методів та засобів конструювання програмного забезпечення в систематизованому вигляді для їх застосування на процесах розробки програмних систем на основі мови програмування Python, а також основам роботи в команді.

Мета навчальної дисципліни

Мета дисципліни – систематизація знань щодо особливостей застосування різних методологій та технологій розробки програм, формування у студента погляду аналітика, здатного зробити обґрунтований вибір методів та технологій для вирішення завдань різного типу, що вміє визначити критерії цього вибору.

В результаті вивчення дисципліни студент повинен **знати**:

- основні методології розробки програмного забезпечення та умови їх застосування;
- підходи до побудови технологій розробки на основі типових методологій;
- основні характеристики програмного продукту;
- методи організації та автоматизації процесу розробки;
- підходи до забезпечення якості програмного продукту та процесу розробки.

В результаті вивчення дисципліни студент повинен **вміти**:¹

- вибирати мови програмування для створення програмного забезпечення;
- виконувати розгортання та збірку програм;
- керувати процесом конструювання програм;
- обирати методику конструювання;
- застосовувати набуті знання на практиці.

¹ поняття вміти і знати повинні співвідноситися з програмними результатами навчання

Компетентності та результати навчання, яких набувають здобувачі освіти внаслідок вивчення навчальної дисципліни “Методи розробки програмних систем” (шифри та зміст компетентностей та програмних результатів навчання вказано відповідно до ОПП/ОНП «Розробка та тестування програмного забезпечення»).

Шифр та назва компетентності	Шифр та назва програмних результатів навчання
ЗК2. Здатність застосовувати знання у практичних ситуаціях	ПРН9. Знати та вміти використовувати методи збору та аналізу вимог
ЗК10. Здатність діяти соціально відповідально та свідомо	ПРН14. Застосовувати інструментальні засоби проектування, тестування та документування
ФК1. Здатність формулювати вимоги до ПЗ	ПРН20. Знати підходи до забезпечення якості ПЗ
ФК5. Здатність дотримуватися специфікацій та стандартів	ПРН23. Вміти документувати та презентувати результати розробки
ФК13. Здатність обґрунтовано обирати та освоювати інструментарій	

ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Курс	3		
Семестр	6		
Кількість кредитів ECTS	90		
Аудиторні навчальні заняття		денна форма	заочна форма
	лекції	20	4
	практичні	22	4
Самостійна робота (в годинах)		48	82
Форма підсумкового контролю	Залік		

Структурно-логічна схема вивчення навчальної дисципліни²:

Пререквізити	Постреквізити

² тільки для обов'язкових дисциплін

ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Перелік тем лекційного матеріалу

Змістовий модуль 1

Тема 1. Поняття програмної системи та програмного продукту (2 год.).

Вступ в предмет, мета, завдання. Опис робочого дня програміста. Поняття програмної системи та програмного продукту. Життєвий цикл програмного забезпечення: базові моделі.

Тема 2. Життєвий цикл програмного забезпечення: базові моделі (2 год.).

Поняття життєвого циклу ПЗ, їх призначення (Software Development Life Cycle, SDLC). Детальний опис та порівняння моделей: Каскадна (Waterfall), V, Ітеративна, Інкрементна, Спіральна. Практичні застосування. Переваги та недоліки.

Завдання для самостійної роботи. Побудувати схему життєвий цикл ПЗ для власного проєкту. Обрати модель ЖЦ та обґрунтувати вибір.

Тема 3. Підходи, фреймворки, методи управління: Agile, Scrum, Kanban, Lean (2 год.).

Знайомство з Agile та Agile Manifesto. Scrum (ролі, події, артефакти). Kanban та Lean-підходи. Extreme Programming (XP). Масштабування Agile (SAFe). Гібридні підходи до управління розробкою. Порівняльний аналіз Scrum та Kanban.

Завдання для самостійної роботи. Перенести схему життєвий цикл ПЗ під конкретний метод управління.

Тема 4. Організація та проєктування процесу розробки програмного забезпечення (2 год.).

Розглянемо підходи до організації роботи в програмному проєкті, управління вимогами до програмного забезпечення, методи проєктування та аналізу програмних систем. Вивчаються основні архітектурні рішення, а також інструментальні засоби, що використовуються на різних етапах розробки програмного забезпечення: Confluence, Swagger, Vision, JIRA.

Завдання для самостійної роботи. Сформулювати приклад вимог / user stories. Побудувати просту структуру документації проєкту (Vision, API, задачі).

Тема 5. Інструментальна підтримка, автоматизація та забезпечення якості розробки програмного забезпечення (2 год.).

Роль CASE-систем у підтримці етапів життєвого циклу ПЗ, класифікація: (Upper, Lower, Integrated). Інструменти автоматизації процесів розробки ПР, збірки, тестування та розгортання. Підходи до забезпечення якості програмного продукту та процесу розробки ПЗ. Використання стандартів і метрик якості.

Завдання для самостійної роботи. Визначити, які CASE-інструменти доречні для заданого проєкту. Підібрати інструменти автоматизації та якості ПЗ.

Тема 6. Аналіз проєктного досвіду та управління ризиками в розробці програмного забезпечення (2 год.).

Проаналізуємо приклади успішних проєктів та розробки ПЗ, розглянемо приклади аналіз невдалих проєктів та типових помилок. Проведемо класифікацію ризиків та причини їх виникнення у програмних проєктах. Закріпимо роль методології та процесів у мінімізації ризиків

Тема 7. Проєктування архітектури програмних систем (2 год.).

Розглянемо архітектурні стилі (моноліт, клієнт–сервер, мікросервіси) та принципи їх архітектурного проєктування. Порівняємо переваги та недоліки кожної архітектури. Поговоримо про компроміси архітектурних рішень.

Завдання для самостійної роботи. Обрати архітектурний стиль для проєкту. Аргументувати архітектурне рішення.

Тема 8. Оцінювання трудомісткості, строків і вартості розробки (2 год.).

Розглянемо роль оцінювання в управлінні програмними проєктами, поняття трудомісткості, строків і вартості розробки. Порівняльний аналіз методів оцінювання: експертного оцінювання, функціональних точок. Планування людських, часових та технічних ресурсів у програмному проєкті. Розподіл ролей і відповідальностей у команді.

Завдання для самостійної роботи. Виконати приблизну оцінку проєкту (експертно / функціональні точки). Спланувати ресурси команди.

Тема 9. Управління командою та комунікаціями в проєкті (2 год.).

Розглянемо роль команди в успішній реалізації програмного проєкту. Формування та розвиток проєктної команди. Відповідальність і розподіл ролей у проєкті. Конфлікти в команді та підходи до їх вирішення. Інструменти підтримки командної роботи.

Завдання для самостійної роботи. Змодельовати розподіл ролей у команді. Розглянути кейс конфлікту та запропонувати рішення.

Тема 10. Підтримка, супровід та розвиток програмних систем (2 год.).

Поговоримо про види супроводу програмних систем: коригувальний, адаптивний, удосконалювальний. Технічний борг та його вплив на розвиток програмного забезпечення. Забезпечення стабільності та надійності програмних систем під час супроводу. Роль документації та взаємодія команди підтримки з користувачами та замовниками.

Завдання для самостійної роботи. Класифікувати задачі супроводу ПЗ. Визначити підходи до зменшення технічного боргу.

Тема 1. Моделювання життєвого циклу ПЗ для заданого проєкту. Обґрунтування вибору моделі життєвого циклу. (2 год.).

Тема 2. Адаптація життєвого циклу ПЗ до Agile/Scrum/Kanban. Аналіз гібридних підходів. (4 год.).

Тема 3. Формування вимог і user stories. Побудова структури проєктної документації. Робота з інструментами: Confluence, Swagger, Vision, JIRA. (4 год.).

Тема 4. Аналіз CASE-систем та інструментів автоматизації. Вибір інструментів для проєкту. (2 год.).

Тема 5. Розгляд кейсів успішних і невдалих проєктів. Ідентифікація та класифікація ризиків. (2 год.).

Тема 6. Вибір архітектурного стилю. Обґрунтування архітектурного рішення. (2 год.).

Тема 7. Виконання оцінювання проєкту. Формування плану ресурсів команди. (2 год.).

Тема 8. Моделювання ролей у команді. Аналіз комунікаційних ситуацій і конфліктів. (2 год.).

Тема 9. Класифікація задач супроводу. Аналіз технічного боргу та шляхів його зменшення. (2 год.).

Зміст самостійної роботи здобувачів

Розподіл годин, виділених на вивчення дисципліни:

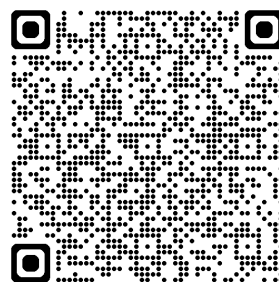
Найменування видів робіт	Розподіл годин за формами навчання	
	денна	заочна
Самостійна робота, год, у т.ч.:	48	82
Опрацювання матеріалу, викладеного на лекціях	8	13
Підготовка до практичних занять та контрольних заходів	12	21
Підготовка звітів з практичних робіт	8	13
Підготовка до поточного контролю	4	7
Опрацювання матеріалу, винесеного на самостійне вивчення	16	28

ПОЛІТИКА КУРСУ

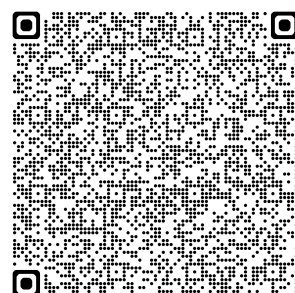
*Коротко, з покликанням на відповідну нормативну базу УКД, висвітлити питання:*³

1) щодо системи поточного і підсумкового контролю

Організація поточного та підсумкового семестрового контролю знань студентів, проведення практик та атестації, переведення показників академічної успішності за 100-бальною шкалою в систему оцінок за національною шкалою здійснюється згідно з “Положенням про систему поточного і підсумкового контролю, оцінювання знань та визначення рейтингу здобувачів освіти”. Ознайомитись з документом можна за [покликанням](#).



2) щодо оскарження результатів контрольних заходів

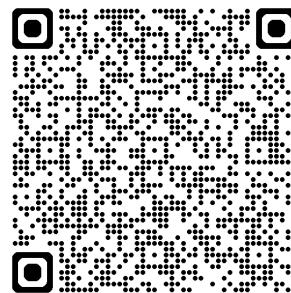


³ зміст пунктів може редагуватись з огляду на особливості курсу

Здобувачі вищої освіти мають право на оскарження оцінки з дисципліни отриманої під час контрольних заходів. Апеляція здійснюється відповідно до «Положення про політику та врегулювання конфліктних ситуацій». Ознайомитись з документом можна за [покликанням](#).

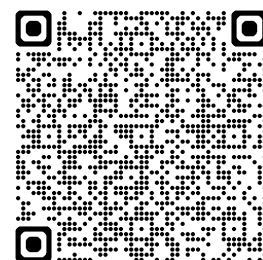
3) щодо відпрацювання пропущених занять

Згідно “Положення про організацію освітнього процесу” здобувач допускається до семестрового контролю з конкретної навчальної дисципліни (семестрового екзамену, диференційованого заліку), якщо він виконав усі види робіт, передбачені на семестр навчальним планом та силабусом/робочою програмою навчальної дисципліни, підтвердив опанування на мінімальному рівні результатів навчання (отримав ≥ 35 бали), відпрацював визначені індивідуальним навчальним планом всі лекційні, практичні, семінарські та лабораторні заняття, на яких він був відсутній. Ознайомитись з документом можна за [покликанням](#).



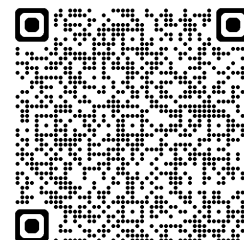
4) щодо дотримання академічної доброчесності

“Положення про академічну доброчесність” закріплює моральні принципи, норми та правила етичної поведінки, позитивного, сприятливого, доброчесного освітнього і наукового середовища, професійної діяльності та професійного спілкування спільноти Університету, викладання та провадження наукової (творчої) діяльності з метою забезпечення довіри до результатів навчання. Ознайомитись з документом можна за [покликанням](#).



5) щодо використання штучного інтелекту

“Положення про академічну доброчесність” визначає політику щодо використання технічних засобів на основі штучного інтелекту в освітньому процесі. Ознайомитись з документом можна за [покликанням](#).⁴ “Положення про систему запобігання та виявлення академічного плагіату, самоплагіату, фабрикації та фальсифікації академічних творів” містить рекомендації щодо використання в академічних текстах генераторів на основі штучного інтелекту. Ознайомитись з документом можна за [покликанням](#).



⁴ визначається політика використання ШІ в навчальній дисципліні - дозволене/заборонене, правила використання

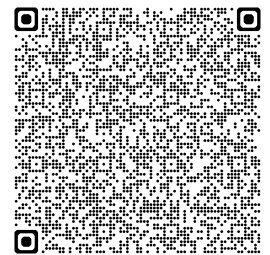
6) щодо використання технічних засобів в аудиторії та правила комунікації

Використання мобільних телефонів, планшетів та інших гаджетів під час лекційних та практичних занять дозволяється виключно у навчальних цілях (для уточнення певних даних, перевірки правопису, отримання довідкової інформації тощо). На гаджетах повинен бути активований режим «без звуку» до початку заняття. Під час занять заборонено надсилання текстових повідомлень, прослуховування музики, перевірка електронної пошти, соціальних мереж тощо. Під час виконання заходів контролю використання гаджетів заборонено (за винятком, коли це передбачено умовами його проведення). У разі порушення цієї заборони результат анулюється без права перескладання.

Комунікація відбувається через електронну пошту і сторінку дисципліни в Moodle.

7) щодо зарахування результатів навчання, здобутих шляхом формальної/інформальної освіти

Процедури визнання результатів навчання, здобутих шляхом формальної/інформальної освіти визначаються «Положенням про порядок визнання результатів навчання, здобутих шляхом неформальної та / або інформальної освіти». Ознайомитись з документом можна за [посиланням](#).⁵



МЕТОДИ НАВЧАННЯ

При вивченні дисципліни застосовується комплекс методів для організації навчання студентів з метою розвитку їх логічного та абстрактного мислення, творчих здібностей, підвищення мотивації до навчання та формування особистості майбутнього фахівця.

Програмний результат навчання⁶	<u>Метод навчання</u>	Метод оцінювання
ПРН9. Знати та вміти використовувати методи збору та аналізу вимог	розповідь-пояснення; практичні роботи; аналітичний; методи самостійної роботи вдома; дискусія, диспут	залік; поточний контроль; письмовий контроль

⁵ визначається перелік електронних та інших ресурсів та умови перезарахування

⁶ для вибіркових навчальних дисциплін вказується результат навчання

ПРН14. Застосовувати інструментальні засоби проектування, тестування та документування	комп'ютерні і мультимедійні методи; вправи; практичні роботи; методи самостійної роботи вдома; робота під керівництвом викладача	залік; поточний контроль; практичний контроль
ПРН20. Знати підходи до забезпечення якості ПЗ	розповідь-пояснення; вправи; практичні роботи; методи самостійної роботи вдома; кейс-метод	залік; поточний контроль; практичний контроль
ПРН23. Вміти документувати та презентувати результати розробки	бесіда; практичні роботи; методи самостійної роботи вдома; тренінгові заняття; бесіда-діалог	залік; поточний контроль; усний контроль

ОЦІНЮВАННЯ РЕЗУЛЬТАТІВ НАВЧАННЯ

Контрольні заходи⁷

(в разі потреби - розділити за семестрами)

<i>Вид</i>	<i>Зміст⁸</i>	<i>% від загальної оцінки</i>	<i>Бал</i>	
			<i>min</i>	<i>max</i>
Поточні контрольні заходи	всього	60	35	60
Підсумкові контрольні заходи	проект	40	25	40
Всього:		100	60	100

Процедура проведення контрольних заходів, а саме поточного контролю знань протягом семестру та підсумкового семестрового контролю, регулюється «Положенням про систему поточного та підсумкового контролю оцінювання знань та визначення рейтингу студентів».

Фіксація **поточного** контролю здійснюється в “Електронному журналі обліку успішності академічної групи” на підставі чотирибальної шкали - “2”; “3”; “4”; “5”. У разі відсутності студента на занятті виставляється “н”. За результатами поточного контролю у Журналі, автоматично визначається підсумкова оцінка, здійснюється підрахунок пропущених занять.

Усі пропущені заняття, а також негативні оцінки студенти зобов'язані відпрацювати впродовж трьох наступних тижнів. У випадку недотримання цієї норми, замість “н” в журналі буде виставлено “0” (нуль балів), без права перездачі. Відпрацьоване лекційне заняття в електронному журналі позначається літерою «в».⁹

Критерії оцінювання¹⁰ (за необхідності, поточного та/або підсумкового контролю)

До підсумкового контролю допускаються студенти які за результатами поточного контролю отримали не менше 35 балів. Усі студенти, що отримали 34 балів і менше, не допускаються до складання підсумкового контролю і на підставі укладання додаткового договору, здійснюють повторне вивчення

⁷ зміст редагується залежно від наповнення дисципліни

⁸ у випадку наявності видів роботи, які оцінюються окремо (проект, завдання тощо) прописується в окремому рядку; за відсутності - одним рядком визначається вся сукупність аудиторної роботи (опитування, поточні контрольні тощо) та визначається стандартне значення балів (35/60)

⁹ можна вказати теми чи завдання, які є обов'язковими до виконання, а також особисті підходи до оцінювання рівня знань здобувачів під час аудиторної роботи

¹⁰ критерії вказуються згідно з особливостями дисципліни.

дисципліни впродовж наступного навчального семестру. За результатами підсумкового контролю (диференційований залік/екзамен) студент може отримати 40 балів. Студенти, які під час підсумкового контролю отримали 24 бали і менше, вважаються такими, що не здали екзамен/диференційований залік і повинні йти на перездачу. Підсумковим контролем вивчення дисципліни є проект. Проект виконується згідно тематик та вимог.

Загальна семестрова оцінка з дисципліни, яка виставляється в екзаменаційних відомостях оцінюється в балах (згідно з **Шкалою оцінювання знань за ЄКТС**) і є сумою балів отриманих під час поточного та підсумкового контролю.

Підсумковий контроль з дисципліни Конструювання програмного забезпечення проводиться у вигляді екзамену.

Шкала оцінювання знань за ЄКТС:

Оцінка за національною шкалою	Рівень досягнень, %	Шкала ECTS
Національна диференційована шкала		
Відмінно	90 – 100	A
Добре	83 – 89	B
	75 – 82	C
Задовільно	67 – 74	D
	60 – 66	E
Незадовільно	35 – 59	FX
	0 – 34	F
Національна недиференційована шкала		
Зараховано	60 – 100	-
Не зараховано	0 – 59	-

Студенти, які не з'явилися на заліки/екзамени без поважних причин, вважаються такими, що одержали незадовільну оцінку.

РЕКОМЕНДОВАНІ ДЖЕРЕЛА ІНФОРМАЦІЇ¹¹

Основна література

¹¹ обов'язково: враховувати вимоги [ДСТУ 8302:2015](#) (відповідно до [Наказу № 65, від 4.03. 2016](#)), [рекомендації](#) Національного агентства з забезпечення якості вищої освіти, використовувати літературу за останні 5-7 років, наводити власні публікації за змістом навчальної дисципліни.

1. Прессман Р., Максим Б. Інженерія програмного забезпечення: практичний підхід, Р. Прессман, Б. Максим. - М.: McGraw-Hill, 2015. - 976 с.
2. Соммервілл І. Інженерія програмного забезпечення, І. Соммервілл. - К.: Pearson Education, 2016. - 816 с.
3. Боем Б. Економіка інженерії програмного забезпечення, Б. Боем. - М.: Prentice Hall, 1981. - 768 с.
4. Бек К. та ін. Маніфест гнучкої розробки програмного забезпечення, К. Бек та ін. - 2001. - 14 с.
5. Швабер К., Сазерленд Д. Посібник зі Scrum, К. Швабер, Д. Сазерленд. - Scrum.org, 2020. - 19 с.
6. Поппендік М., Поппендік Т. Lean-розробка програмного забезпечення: інструментарій Agile, М. Поппендік, Т. Поппендік. - М.: Addison-Wesley, 2003. - 240 с.
7. Ларман К. Agile та ітеративна розробка: посібник для менеджерів, К. Ларман. - М.: Addison-Wesley, 2004. - 368 с.
8. Басс Л., Клементс П., Казман Р. Практика архітектури програмного забезпечення, Л. Басс, П. Клементс, Р. Казман. - М.: Addison-Wesley, 2013. - 624 с.
9. IEEE Computer Society. SWEBOOK: Звід знань з інженерії програмного забезпечення. - IEEE, 2014. - 338 с.
10. ISO/IEC 12207. Процеси життєвого циклу програмного забезпечення. - ISO/IEC, 2017. - 142 с.

Електронні інформаційні ресурси

1. Прессман Р., Максим Б. Інженерія програмного забезпечення: практичний підхід, Р. Прессман, Б. Максим. - М.: McGraw-Hill, 2015. - 976 с. URL: <https://repository.unikom.ac.id/45085/1/Software%20Engineering%20-%20Roger%20S%20Pressman%20%5B5th%20edition%5D.pdf> (Дата звернення: 24.12.2025).
2. Соммервілл І. Інженерія програмного забезпечення, І. Соммервілл. - К.: Pearson Education, 2016. - 816 с. URL: <https://engineering.futureuniversity.com/BOOKS%20FOR%20IT/Software-Engineering-9th-Edition-by-Ian-Sommerville.pdf> (Дата звернення: 24.12.2025).
3. Соммервілл І. Software Engineering (8th Edition): навч. посіб. URL: <https://fita.vnua.edu.vn/wp-content/uploads/2013/06/Software-Engineering-By-Ian-Sommerville-8th-Edition.pdf> (Дата звернення: 24.12.2025).
4. Beck K., Beedle M., van Bennekum A. та ін. Маніфест гнучкої розробки програмного забезпечення. - 2001. URL: <https://agilemanifesto.org/iso/uk/principles.html> (Дата звернення: 25.12.2025).

5. Schwaber K., Sutherland J. The Scrum Guide: офіційний посібник зі Scrum. - Scrum.org, 2020. - 19 с. URL: <https://scrumguides.org/scrum-guide.html> (Дата звернення: 25.12.2025).
6. Poppendieck M., Poppendieck T. Lean Software Development: An Agile Toolkit: навч. посіб. - Addison-Wesley, 2003. - 240 с. URL: <https://www.lean.org/lexicon/lean-software-development> (Дата звернення: 26.12.2025).
7. IEEE Computer Society. Guide to the Software Engineering Body of Knowledge (SWEBOK). URL: <https://www.computer.org/education/bodies-of-knowledge/software-engineering> (Дата звернення: 27.12.2025).
8. ISO/IEC. ISO/IEC 12207: Процеси життєвого циклу програмного забезпечення. URL: <https://www.iso.org/standard/63712.html> (Дата звернення: 24.12.2025).
9. Програмна інженерія // Вікіпедія. URL: https://uk.wikipedia.org/wiki/Програмна_інженерія (Дата звернення: 27.12.2025).