

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА

ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра програмних систем і технологій

«ЗАТВЕРДЖУЮ»

Заступник декана

з навчально-виховної роботи

Наталія ТМЄНОВА

« 29 » серпня 2026 року

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
«ЯКІСТЬ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ТЕСТУВАННЯ»

для здобувачів вищої освіти

галузь знань	12 Інформаційні технології
спеціальність	121 Інженерія програмного забезпечення
освітній рівень	бакалавр
освітня програма	Інженерія програмного забезпечення
вид дисципліни	обов'язкова

Форма навчання	денна
Навчальний рік	2026/2027
Семестр	6
Кількість кредитів ECTS	6
Мова викладання, навчання та оцінювання	українська
Форма заключного контролю	іспит

Викладачі: **ВОВНА Олександр Володимирович**, доктор технічних наук, професор
МАРИНА Анна Сергіївна, кандидат економічних наук, доцент

Пролонговано: на 2027/2028 н.р. _____ (_____) «____» _____ 2027 р.
(підпис) (ПІБ) (дата)

на 2028/2029 н.р. _____ (_____) «____» _____ 2028 р.
(підпис) (ПІБ) (дата)

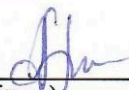
на 2029/2030 н.р. _____ (_____) «____» _____ 2029 р.
(підпис) (ПІБ) (дата)

Розробники:

ВОВНА Олександр Володимирович, д.т.н., професор, професор кафедри програмних систем і технологій

МАРИНА Анна Сергіївна, к.е.н., доцент, доцент кафедри програмних систем і технологій

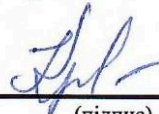
Завідувач кафедри
програмних систем і технологій

 проф. Олексій БИЧКОВ
(підпис)

Протокол № 18 від 17 червня 2026 року

Схвалено науково-методичною комісією факультету інформаційних технологій

Протокол № 11 від 26 червня 2026 року

Голова науково-методичної комісії  Ганна КРАСОВСЬКА
(підпис)

1. Мета дисципліни – формування у здобувачів освіти знань і практичних навичок щодо принципів, методів та засобів забезпечення якості програмних продуктів, включно з розробкою і застосуванням тестових стратегій, використанням інструментів автоматизованого тестування та оцінюванням відповідності програмного забезпечення сучасним стандартам і вимогам, сприяння засвоєнню методів та технологій організації процесів контролю якості та тестування програмного забезпечення на всіх етапах його життєвого циклу.

2. Попередні вимоги до опанування або вибору навчальної дисципліни:

1. Володіти теоретичними основами, які отримано під час вивчення нормативних дисциплін «Основи програмування», «Вступ до об'єктно-орієнтованого програмування», «Аналіз вимог до програмного забезпечення», «Архітектура та проектування програмного забезпечення».

2. Володіти навичками роботи в групах.

3. Анотація навчальної дисципліни / референс:

Дисципліна спрямована на формування у здобувачів системи теоретичних знань та практичних умінь, необхідних для забезпечення високої якості програмних продуктів і побудови ефективних процесів тестування на різних етапах життєвого циклу програмного забезпечення. У межах курсу розглядаються фундаментальні принципи та методологічні підходи до організації процесів тестування, включно з його фазами, технологіями та сучасними інструментами.

Особлива увага приділяється аналізу ключових проблем і завдань тестування, вивченню структурних, функціональних, стохастичних і мутаційних критеріїв вибору тестів, методам оцінювання покриття програмного коду та проектної документації. Розглядаються різні рівні тестування: модульне, інтеграційне, системне, регресійне, а також особливості їх застосування в умовах складних програмних систем.

Курс охоплює питання планування тестування, побудови стратегій і підходів до розробки тестів, розмежування ручної розробки та автоматизованої генерації тестових сценаріїв, а також організації автоматизованого тестового циклу. Вивчаються принципи документування результатів тестування, проведення оглядів, розробка та інтерпретація метрик якості. Особливо акцентується увага на аналізі методів відбору тестів та оцінюванні їх ефективності в контексті забезпечення надійності та стабільності програмних систем.

Елементи STEM-освіти реалізовано через інтеграцію науково обґрунтованих методів (Science), використання математичного апарату для формалізації критеріїв тестування та оцінки якості (Mathematics), застосування сучасних технологій автоматизації тестових процесів і інструментів контролю якості (Technology), а також розвиток інженерного мислення для побудови оптимальних стратегій тестування та управління життєвим циклом програмного забезпечення (Engineering). Такий підхід сприяє формуванню комплексних міждисциплінарних компетентностей, розвитку критичного мислення та здатності до інноваційного вирішення практичних завдань у сфері забезпечення якості програмних продуктів.

4. Завдання (навчальні цілі)

Інтегральна компетентність. Здатність розв'язувати складні спеціалізовані задачі та практичні проблеми у галузі інженерії програмного забезпечення, що характеризується комплексністю та неповною визначеністю умов із застосуванням теорій та методів інформаційних технологій.

Загальні компетентності:

ЗК01. Здатність до абстрактного мислення, аналізу та синтезу.

ЗК02. Здатність застосовувати знання у практичних ситуаціях.

ЗК04. Здатність спілкуватися іноземною мовою як усно, так і письмово.

ЗК05. Здатність вчитися та оволодівати сучасними знаннями та проводити самостійні дослідження на відповідному рівні.

ЗК06. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.

ЗК07. Здатність працювати у команді.

ЗК13. Здатність використовувати сучасні методології управління проектною розробкою програмного забезпечення.

Спеціальні (фахові, предметні) компетентності

ФК01. Здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення.

ФК02. Здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування.

ФК04. Здатність формувати та забезпечувати вимоги щодо якості програмного у відповідності з вимогами замовника, технічним завданням та стандартами.

ФК10. Здатність накопичувати, обробляти та систематизувати професійні знання щодо створення і супроводження програмного забезпечення та визнання важливості навчання протягом всього життя.

ФК11. Здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення.

5. Результати навчання за дисципліною:

Результат навчання (1. знати; 2. вміти; 3. комунікація; 4. автономність та відповідальність)		Методи викладання і навчання	Методи оцінювання	Відсоток у підсумковій оцінці з дисципліни
Код	Результат навчання			
1.1	Знати фундаментальні принципи забезпечення якості програмного забезпечення та методологію тестування.	Словесний метод (лекція, дискусія, співбесіда тощо); наочний метод (метод ілюстрацій і метод демонстрацій); практичний метод (лабораторні заняття, розрахункові роботи); відеометод у сполученні з інформаційним і технологіями та комп'ютерними засобами навчання (дистанційні, мультимедійні, веб-орієнтовані тощо); робота з навчально-методичною літературою та іншими джерелами інформації (конспектування, тезування, анотування, складання реферату, есе); самостійна робота (індивідуальні творчі завдання, розв'язання програмних завдань: домашні та лабораторні)	Стандартизовані тести; усний контроль (усне опитування); письмовий контроль (модульні контрольні роботи); захист лабораторних робіт, контроль рівня виконавських умінь для перевірки практичної підготовки, іспит	5%
1.2	Знати класифікацію видів, рівнів і технік тестування (модульне, інтеграційне, системне, регресійне, функціональне, навантажувальне тощо).			10%
1.3	Знати сучасні критерії відбору тестів, методи оцінювання покриття та показників ефективності тестування.			5%
1.4	Знати міжнародні стандарти та підходи до управління якістю програмних продуктів.			5%
1.5	Знати інструменти автоматизації тестування та принципи організації тестового циклу.			5%
2.1	Вміти планувати процес тестування з урахуванням специфіки проєкту.			5%
2.2	Вміти розробляти тестові стратегії, сценарії та дані для різних рівнів тестування.			10%
2.3	Вміти застосовувати інструменти автоматизації для підвищення ефективності та точності тестування.			5%
2.4	Вміти аналізувати результати тестів, виявляти дефекти та формувати рекомендації щодо їх усунення.			10%
2.5	Вміти здійснювати оцінювання ефективності методів і засобів тестування.			5%
3.1	Ефективно взаємодіяти у команді розробників, тестувальників та інших зацікавлених сторін			5%
3.2	Презентувати результати тестування та обґрунтовувати прийняті рішення			5%
3.3	Використовувати сучасні засоби документування й візуалізації даних для забезпечення прозорості процесу контролю якості			5%
4.1	Самостійно організовувати тестування у межах поставленого завдання			5%
4.2	Ухвалювати виважені рішення щодо вибору методів, інструментів і стратегій тестування			5%
4.3	Нести відповідальність за якість виконаних тестових процедур та достовірність отриманих результатів			5%
4.4	Дотримуватися професійної етики та стандартів у сфері забезпечення якості програмного забезпечення			5%

6. Співвідношення результатів навчання дисципліни із програмними результатами навчання

Результати навчання дисципліни	1	1	1	1	1	2	2	2	2	2	3	3	3	4	4	4	4
Програмні результати навчання	1	2	3	4	5	1	2	3	4	5	1	2	3	1	2	3	4
ПР-04. Знати і застосовувати професійні стандарти і інші нормативно-правові документи в галузі інженерії програмного забезпечення	+	+	+			+			+				+		+		+
ПР-14. Застосовувати на практиці інструментальні програмні засоби доменного аналізу, проектування, тестування, візуалізації, вимірювань та документування програмного забезпечення	+				+	+	+	+			+	+	+	+	+		
ПР-19. Знати та вміти застосовувати методи верифікації та валідації програмного забезпечення.	+		+		+				+	+				+	+		
ПР-20. Знати підходи щодо оцінки та забезпечення якості програмного забезпечення	+					+		+	+	+	+				+	+	+

7. Схема формування оцінки:

7.1. Форми оцінювання здобувачів вищої освіти:

Контроль знань здобувачів вищої освіти здійснюється за модульно-рейтинговою системою. Результати навчальної діяльності здобувачів вищої освіти оцінюються за 100-бальною шкалою. Робота в семестрі поділяється на два змістових модуля. Підсумковий контроль (іспит) – 40 балів.

У змістовий модуль 1 (ЗМ1) входять теми 1 – 8, а у змістовий модуль 2 (ЗМ2) – теми 9 – 17. Обов'язковим для допуску до іспиту є виконання та захист здобувачем вищої освіти усіх лабораторних робіт, що передбачено робочою програмою навчальної дисципліни. Під час цього сумарна мінімальна рекомендована кількість балів для допуску 36.

	ЗМ1/Частина 1	ЗМ2/Частина 2	іспит	Підсумкова оцінка
<i>Мінімум</i>	<i>18</i>	<i>18</i>	<i>24</i>	<i>60</i>
Максимум	30	30	40	100

До поточної підсумкової оцінки викладач може додати заохочувальні бали, які можна отримати активною роботою на лекціях, лабораторних заняттях, за виконання самостійних проєктів, участі в конференції з доповіддю за тематикою дисципліни. Заохочувальні бали можуть бути нараховані за проходження сертифікованих курсів з тестування / якості програмного забезпечення (до 5 балів). Максимальна кількість заохочувальних балів за семестр – 10 балів, які є складовою поточного оцінювання.

Оцінка за іспит не може бути меншою **24 балів** для отримання загальної позитивної оцінки за курс. Якщо здобувач на іспиті набрав менше 24 балів (60% від 40 балів, відведених на іспит), то вони не додаються до семестрової оцінки незалежно від кількості балів, отриманих під час семестру, а в екзаменаційній відомості у графі «результуюча оцінка» переноситься лише кількість балів, отриманих під час семестру.

7.2. Організація оцінювання:

Поточне семестрове оцінювання:

– оцінюванню підлягає модульна контрольна робота, звіти та захист лабораторних робіт, активна робота на лекціях і лабораторних заняттях;

– впродовж семестру за змістовними модулями 1 та 2 (теми 1-8 та 9-17) проводяться модульні контрольні роботи на лабораторних заняттях (здобувачі вищої освіти отримують завдання за варіантами та виконують їх письмово). Максимальна кількість балів за кожну з модульних контрольних робіт не перевищує 6, модульні контрольні роботи не перескладаються. Перевірка завдань, оприлюднення та аналіз результатів здійснюється не пізніше наступного лабораторного заняття;

– кожна лабораторна робота оцінюється в 6 бали, якщо всі завдання виконано, продемонстровано їх працездатність, звіт оформлено відповідно до вимог. Максимальна кількість балів за усі лабораторні роботи не перевищує 40. Кінцевим терміном здачі кожної лабораторної роботи є наступне лабораторне заняття після виконання зазначеної лабораторної роботи;

– у разі виконання проєктного STEM завдання, оцінювання проводиться відповідно до «Рекомендацій щодо оцінювання освітніх STEM-проєктів», затверджених на засіданні кафедри програмних систем і технологій (протокол № 16 від «01» травня 2026 р.): <https://pst.knu.ua/academics/bachelor/evaluation-stem-projects/>, кількість балів не перевищує 40;

– для допуску до іспиту обов'язковим є виконання та захист усіх запланованих лабораторних робіт, критично-розрахунковий мінімум для допуску до іспиту становить 36 балів.

Підсумкове оцінювання (у формі іспиту):

- екзаменаційний білет складається з тестової та практичної частин (виконуються письмово);
- максимальна кількість балів, які можуть бути отримані здобувачем вищої освіти на іспиті – 40 балів;
- здобувач не допускається до іспиту, якщо під час семестру набрав менше ніж 36 балів;
- для отримання загальної позитивної оцінки з дисципліни сума балів за іспит не може бути меншою 24.

7.3. Шкала відповідності:

Відмінно / Excellent	90 – 100
Добре / Good	75 – 89
Задовільно / Satisfactory	60 – 74
Незадовільно / Fail	0 – 59

8. Структура навчальної дисципліни

№ з/п	Назва теми	Кількість годин		
		лекції	лабора-торні роботи	С/Р
Змістовий модуль 1. Поняття якості програмного забезпечення та організація тестового процесу				
1	Тема 1. Історія предметної області, основні поняття і терміни.	2	-	7
2	Тема 2. Модель якості та надійності програмного забезпечення.	4	4	4
3	Тема 3. Метричні характеристики програм.	4	-	6
4	Тема 4. Якість у життєвому циклі розробки програмного забезпечення.	2	-	7
5	Тема 5. Тестування документації і вимог.	2	6	4
6	Тема 6. Вступ до тестування програмного забезпечення.	2	-	7
7	Тема 7. Процес тестування.	2	4	6
8	Тема 8. Дефекти. Причини, опис, відстеження.	2	-	7
9	Модульна контрольна робота 1	-	2	
Змістовий модуль 2. Методи, технології та інструменти тестування програмного забезпечення				
10	Тема 9. Модульне тестування.	2	4	6
11	Тема 10. Інтеграційне тестування.	2	-	7
12	Тема 11. Системне тестування.	2	4	6
13	Тема 12. Тестування об'єктно-орієнтованих програм.	2	-	7
14	Тема 13. Дослідницьке тестування.	2	4	6
15	Тема 14. Тестування інтерфейсу користувача.	2	-	6
16	Тема 15. Особливості тестування WEB-застосунків.	2	4	6
17	Тема 16. Тестування мобільних застосунків.	2	-	6
18	Тема 17. Автоматизація тестування.	2	4	6
	Модульна контрольна робота 2	-	2	
	Всього	38	38	104

Загальний обсяг 180 год., в тому числі:

- лекцій – 38 год.;
- лабораторні заняття – 38 год.;
- самостійна робота – 104 год.

Теми лабораторних занять

1. Розробка та тестування специфікації вимог.
2. Розробка тестового плану.
3. Написання тест-кейсів для тестування програмного забезпечення.
4. Тестування програмного забезпечення методом «чорної скриньки».
5. Пошук та документування дефектів. Написання баг-репортів.
6. Тестування WEB-застосунку.
7. Тестування Usability.
8. Оцінка якості розробленого програмного забезпечення.

Завдання для самостійної роботи

Тема 1. Етапи становлення та розвитку тестування програмного забезпечення. Основні поняття та визначення, що використовуються у сфері забезпечення якості ПЗ. Внесок IEEE, ISO та ISTQB у стандартизацію термінології з якості програмного забезпечення.

Тема 2. Сутність моделей якості ПЗ. Характеристики визначають надійність програмного забезпечення. Моделі якості, які застосовуються у процесі розробки та тестування.

Тема 3. Основні метрики, які використовуються для оцінки якості програмного забезпечення. Вимірювання та інтерпретація складності програм. Роль метрик у плануванні та проведенні тестування.

Тема 4. Інтеграція поняття якості у різних моделях життєвого циклу ПЗ. Відмінності між забезпеченням якості у каскадній та Agile-моделях. Фактори, які впливають на формування якості ПЗ на різних етапах розробки.

Тема 5. Основні методи перевірки коректності вимог. Рецензування проєктної документації. Виявлення суперечностей та неповноти у вимогах.

Тема 6. Цілі та завдання тестування ПЗ. Принципи тестування, що визначають ефективність цього процесу. Верифікація та валідація.

Тема 7. Основні етапи тестового процесу. Критерії початку і завершення тестування. Роль тест-плану у забезпеченні якості програмного продукту.

Тема 8. Типи дефектів у програмних системах. Опис дефектів у баг-трекінгових системах. Методи та інструменти для відстеження дефектів.

Тема 9. Сутність модульного тестування. Інструменти, які найчастіше застосовуються для модульного тестування.

Тема 10. Стратегії інтеграційного тестування. Взаємодія інтеграційного тестування з модульним і системним.

Тема 11. Об'єкт системного тестування. Різновиди системного тестування. Зв'язок системного тестування з вимогами замовника.

Тема 12. Особливості тестування ООП-систем. Використання UML-діаграм у процесі тестування. Тестування спадкування та поліморфізму.

Тема 13. Сутність дослідницького тестування. Техніки, які застосовуються для пошуку дефектів без формального сценарію. Ситуації, коли дослідницьке тестування є найбільш ефективним.

Тема 14. Критерії оцінки зручності та доступності інтерфейсу. Методи оцінки UI/UX. Забезпечення адаптивності інтерфейсу для різних пристроїв.

Тема 15. Фактори, що впливають на якість веб-застосунку. Методи перевірки безпеки веб-додатків. Тестування продуктивності веб-застосунків.

Тема 16. Відмінності мобільного тестування від веб- і десктопного. Особливості перевірки кросплатформених мобільних застосунків. Тестування push-сповіщень, геолокації та роботи в офлайн-режимі.

Тема 17. Найбільш поширені інструменти автоматизації тестування. Переваги та ризики впровадження автоматизації. Визначення сценаріїв для автоматизації та оцінка її ефективності.

РЕКОМЕНДОВАНА ЛІТЕРАТУРА:

Основна

1. ISO/IEC/IEEE 15288:2023 Systems and software engineering — System life cycle processes. <https://www.iso.org/standard/81702.html>
2. ISO/IEC 25010:2023 Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Product quality model. <https://www.iso.org/standard/78176.html>
3. ISO/IEC 25023:2016 Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Measurement of system and software product quality. <https://www.iso.org/ru/standard/35747.html>
4. Kulikov, S. Software Testing: Base Course. 3rd Edition. EPAM System. Book version 3.2.5 as of 2023-12-20. 280 p.
5. Добровольський Ю.Г. Інженерія надійності програмного забезпечення: навч. посібн. Чернівці, 2022. 126 с.
6. Цибульник С.О., Барандич К.С. Технології розроблення програмного забезпечення. Частина 1. Життєвий цикл програмного забезпечення: підручник. Київ, 2022. 270 с.
7. Крепич С.Я., Співак І.Я. Якість програмного забезпечення та тестування: базовий курс. навч. посіб. Тернопіль, 2020. 478с.
8. Авраменко А.С., Авраменко В.С., Косенюк Г.В. Тестування програмного забезпечення. навч. посіб. Черкаси, 2017. 284 с.

Додаткова

9. The Future of Software Quality Assurance. Springer International Publishing. 2019. – 257 p. https://www.google.com.ua/books/edition/The_Future_of_Software_Quality_Assurance/YRS_DwAAQBAJ?hl=en&gbpv=1&dq=Software+quality+and+testing&printsec=frontcover
10. Грицюк, Ю. І. (2022). Система комплексного оцінювання якості програмного забезпечення. Scientific Bulletin of UNFU, 32(2), 81-95. <https://doi.org/10.36930/40320213>
11. Васілевський О.М., Поджаренко В.О. Нормування показників надійності технічних засобів. https://web.posibnyky.vntu.edu.ua/firen/18vasilevsky,podzharenko_normuv_pokaz_nadijn_tech_zasobiv/z.htm
12. Яковина В., Мацелюх В., (2017). Огляд і аналіз моделей надійності програмного забезпечення. <https://science.lpnu.ua/sites/default/files/journal-paper/2018/jul/13773/17.pdf>
13. Rashid J., Mahmood T., Nisar M.W. A Study on Software Metrics and its Impact on Software Quality https://www.researchgate.net/publication/333292649_A_Study_on_Software_Metrics_and_its_Impact_on_Software_Quality
14. Модульне тестування (Unit Testing) в інженерії даних: Практичний посібник. <https://data-life-ua.com/ci-cd/unit-testing-v-data-engineer-praktychnyy-posibnyk/>
15. Бобков В. Б., Грудзинський Ю. Є., Крилов К. В. Програмування-2. Об'єктно-орієнтоване програмування. <https://ela.kpi.ua/server/api/core/bitstreams/b4b2edcc-c55a-4f74-bb81-9dc7d04a3a41/content>
16. A Guide to Mobile App Testing and Eliminating Bugs before the Product Release. <https://www.expertappdevs.com/blog/mobile-app-testing-guide>

Інформаційні ресурси

17. QATestLab <https://training.qatestlab.com/>
18. TestRail <https://www.testrail.com.ua/>