



МІЖНАРОДНИЙ ЄВРОПЕЙСЬКИЙ  
УНІВЕРСИТЕТ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ  
«ЄВРОПЕЙСЬКА ШКОЛА БІЗНЕСУ»

ЗАТВЕРДЖЕНО  
Директор ІНІ  
«Європейська школа бізнесу»

Юлія РЕМИГА

від «11» 09 2023 р.

М.П.

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ  
ЯКІСТЬ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ  
ТА ТЕСТУВАННЯ

Рівень вищої освіти

перший (бакалаврський)

(назва)

Спеціальність

121 Інженерія програмного забезпечення

(код і назва спеціальності)

Освітня програма

«Інженерія програмного забезпечення»

(назва освітньої програми)

Робоча програма навчальної дисципліни «**Якість програмного забезпечення та тестування**» складена на основі освітньо-професійної програми «Інженерія програмного забезпечення» для першого (бакалаврського рівня спеціальності 121 «Інженерія програмного забезпечення», затвердженої Вченою радою Університету «30» травня 2023 року, протокол № 4.

**Укладач програми:** Шевчук Борис Вікторович – кандидат педагогічних наук, доцент

**Рецензент:** к.ф.м. наук, доцент Шерман З.О.

**Гарант освітньої програми:**  Олександр НЕСТЕРЕНКО

Робочу програму навчальної дисципліни розглянуто та схвалено на засіданні кафедри інформаційних технологій, протокол №1 від «31» серпня 2023 р.

Розглянуто і схвалено Вченою радою Навчально-наукового інституту «Європейська школа бізнесу», протокол № 1 від «11» вересня 2023 р.

## ВСТУП

**Програма вивчення навчальної дисципліни «Якість програмного забезпечення та тестування»** складена відповідно до Стандарту вищої освіти України (далі – Стандарт) галузі знань 12 «Інформаційні технології» спеціальності 121 «Інженерія програмного забезпечення».

**Опис навчальної дисципліни (анотація).** Дана навчальна дисципліна є однією з дисциплін фахової підготовки майбутніх розробників програмного забезпечення.

Таблиця 1

Найменування показників	Галузь знань, напрям підготовки, освітній рівень	Характеристика навчальної дисципліни	
		денна форма навчання	заочна форма навчання
Кількість кредитів – 3	Галузь знань, 12 «ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ»	Нормативна	
Розділів – 1	Спеціальність: 121 «ІНЖЕНЕРІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ»	Рік підготовки	
Змістових розділів – 2		2023	
Індивідуальне науково-дослідне завдання:		Семестр	
		5	5
Загальна кількість годин – 90		Лекції	
		18	6
Тижневе навантаження: аудиторних – 2 самостійної роботи студента – 2	Освітній рівень: бакалавр	Практично-лабораторні	
		30	4
		Самостійна робота	
		72	110
		Вид контролю:	
		залік	

**Предметом вивчення навчальної дисципліни «Якість програмного забезпечення та тестування»** є вивчення сучасних парадигм та технологій забезпечення якості програмного забезпечення при його розробці. Вивчення стандартів тестування програмного забезпечення

**Міждисциплінарні зв'язки:** «Архітектура та проектування програмного забезпечення», «Основи програмування», «Об'єктно-орієнтоване програмування», «Моделювання та аналіз програмного забезпечення».

### 1. МЕТА ТА ЗАВДАННЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

1.1. **Метою** навчальної дисципліни «Якість програмного забезпечення та тестування» є формування сучасного рівня інформаційної та програмістської культури з основ теорії якості програмного забезпечення та тестування, оволодіння базовими методами побудови тестів та автоматизованими засобами тестування, процесами управління якістю

програмного забезпечення, термінологією та основами верифікації, набуття практичних навичок самостійного тестування за стандартами якості програмного забезпечення і створення якісного програмного забезпечення для розв'язування різноманітних задач у професійній діяльності.

1.2. **Основними завданнями** вивчення навчальної дисципліни «Якість програмного забезпечення та тестування» є:

- оволодіння методами тестування, верифікації і валідації;
- вивчення підходів до створення звітності по проблемах при розробці програмного забезпечення;
- ознайомлення з сучасними статистичними методами та інструментальними засобами контролю якості.

1.3. **Компетентності та результати навчання**, формуванню яких сприяє дисципліна (взаємозв'язок з нормативним змістом підготовки здобувачів вищої освіти, сформульованим у термінах результатів навчання у Стандарті).

Згідно з вимогами стандарту дисципліна забезпечує набуття студентами **компетентностей**:

Таблиця 2

<b>Інтегральна компетентність</b>	Здатність розв'язувати складні спеціалізовані задачі та практичні проблеми, які характеризуються комплексністю і невизначеністю умов у сфері розробки програмного забезпечення.
<b>Загальні компетентності</b>	ЗК 07. Здатність працювати в команді
<b>Спеціальні (фахові) компетентності</b>	СК1. Здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення. СК4. Здатність формулювати та забезпечувати вимоги щодо якості програмного забезпечення у відповідності з вимогами замовника, технічним завданням та стандартами.

Деталізація компетентностей відповідно до дескрипторів НРК у формі «Матриці компетентностей» наведено у табл. 3.

Таблиця 3

**Матриця компетентностей**

	<b>Компетентність</b>	<b>Знання</b>	<b>Уміння / навички</b>	<b>Комунікація</b>	<b>Автономія та відповідальність</b>
<b>Інтегральна компетентність</b>					
	Здатність розв'язувати складні спеціалізовані задачі та практичні проблеми, які характеризуються комплексністю і невизначеністю умов, у сфері розробки програмного забезпечення.	Різних методів тестування структуру звіту про дефекти	ідентифікувати дефекти та помилки в артефактах процесу розробки ПЗ	здатність ефективно формувати комунікаційну стратегію	користуватися системами обліку програмних помилок
<b>Загальні компетентності</b>					
	ЗК 07. Здатність працювати в команді			Здатність працювати в команді	
<b>Спеціальні (фахові) компетентності</b>					

<p>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.</p>	<p>основних міжнародних та національних стандартів з якості ПЗ та принципи побудови процесу тестування</p>	<p>створювати тестові плани та тестові приклади</p>	<p>здатність донести до фахівців інформації, ідеї, проблеми та їх вирішення</p>	<p>здатність визначити необхідні об'єми тестування ПЗ</p>
---	--	---	---	---

**Інтегративні кінцеві програмні результати навчання, формуванню яких сприяє навчальна дисципліна:**

<p><i><b>Програмні результати навчання</b></i></p>	<p>ПР09. Знати та вміти використовувати методи та засоби збору, формулювання та аналізу вимог до програмного забезпечення.  ПР10. Проводити передпроектне обстеження предметної області, системний аналіз об'єкта проектування.  ПР19. Знати та вміти застосовувати методи верифікації та валідації програмного забезпечення.  ПР20. Знати підходи щодо оцінки та забезпечення якості програмного забезпечення.</p>
--	---

**Результати навчання:**

Після опанування дисципліни студент повинен

**знати:**

- поняття якості;
- принципи управління якістю продукції;
- методи гарантування якості та верифікації;
- принципи культури якості, базові міжнародні стандарти якості;
- основні визначення процесу тестування, техніки та рівні тестування;
- інструменти тестування та методи оцінки розміру програмних систем;
- методи та засоби випробування ПЗ;

**уміти:**

- проектувати і реалізовувати плани з комплексного та модульного тестування;
- самостійно або в складі групи інспектувати процес розробки програмного забезпечення;
- проводити оцінки тестів, щільності дефектів та ймовірності відмови за допомогою метрик;

- визначати якісні показники ПЗ;
- розробляти методики оцінки якості, випробування та тестування ПЗ.

## 2. ІНФОРМАЦІЙНИХ ОБСЯГ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

На вивчення навчальної дисципліни «Якість програмного забезпечення та тестування» відводиться 90 годин 3 кредити ЄКТС.

### МОДУЛЬ I. ОСНОВИ ТЕСТУВАННЯ ПЗ

#### Змістовий модуль I. Якість програмного забезпечення

Тема 1. Тестування – спосіб забезпечення якості. Рівні і види тестування

Тема 2. UI/UX та Usability. Agile (Гнучка модель). Scrum. Selenium

#### Змістовий модуль II. Тестування програмного забезпечення

Тема 3. Типи тестування

Тема 4. Аналіз вимог (Requirement analysis)

Тема 5. Техніки тест дизайну (Test Design Techniques)

Тема 6. Test Case, Checklist and Bug Report. Fundamental Test Process

### 3. СТРУКТУРА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

№	Назви модулів і тем	Кількість годин (денна форма навчання)				
		Всього	у тому числі			
			Лекції	Практичні (семінарські)	Лабораторні	СРС
<b>МОДУЛЬ I. ОСНОВИ ТЕСТУВАННЯ ПЗ</b>						
<b>Змістовий модуль I. Якість програмного забезпечення</b>						
1.	<u>Тема 1.</u> Тестування – спосіб забезпечення якості. Рівні і види тестування	8	1	-	2	5
2.	<u>Тема 2.</u> UI/UX та Usability. Agile (Гнучка модель). Scrum. Selenium	10	1	-	4	5
<i>Всього за змістовий модуль I:</i>		<i>18</i>	<i>2</i>	<i>-</i>	<i>6</i>	<i>10</i>
<b>Змістовий модуль II. Тестування програмного забезпечення</b>						
3.	<u>Тема 3.</u> Типи тестування	16	2	-	4	10
4.	<u>Тема 4.</u> Аналіз вимог (Requirement analysis)	14	2	-	2	10
5.	<u>Тема 5.</u> Техніки тест дизайну (Test Design Techniques)	14	2	-	2	10
6.	<u>Тема 6.</u> Test Case, Checklist and Bug Report. Fundamental Test Process	28	2	-	8	18
<i>Всього за змістовий модуль II:</i>		<i>72</i>	<i>8</i>	<i>-</i>	<i>16</i>	<i>48</i>
<b><i>Всього годин за курс:</i></b>		<b><i>90</i></b>	<b><i>10</i></b>	<b><i>-</i></b>	<b><i>22</i></b>	<b><i>58</i></b>

## 4. ТЕМИ ЛЕКЦІЙ

№ з/п	Назва модулів, тем та їх зміст
<b>МОДУЛЬ I. ОСНОВИ ТЕСТУВАННЯ ПЗ</b>	
<b>Змістовний модуль I. Якість програмного забезпечення</b>	
1.1.	<p><u>Тема 1. Тестування – спосіб забезпечення якості. Рівні і види тестування</u></p> <p><i>Зміст теми:</i></p> <p>Поняття якості програмного продукту. Задачі тестування. Критерії коректності програми. Основні визначення. Тестування в життєвому циклі розробки програмного забезпечення. Еволюція поняття якості програмного забезпечення. Якість програмного забезпечення за Макколом. Якість програмного забезпечення за Боемом. Характеристика та атрибути якості програмного забезпечення. Фаза, стадії та види тестування. Модульне тестування. Інтеграційне тестування або тестування взаємодій (Integration Testing). Системне тестування або комплексне тестування (System Testing). Альфа і бета тестування. Тестування, що засноване на вимогах (Requirement Based Testing). Регресійне тестування (Regression Testing). Тестування чорного ящика (Black-Box Testing). Тестування за класами еквівалентності (Equivalence class Testing). Граничне тестування (Boundary Testing). Тестування крайніх представників (Best Representative Testing). Тестування білого ящика (White-Box Testing). Тестування логіки (Logic Testing). Тестування станів (State-based Testing). Тестування шляхів (Path testing). Техніки тестування. Аксиоми тестування. Місце тестування в циклі розробки програмного забезпечення. Принципи тестування програмного забезпечення.</p>
1.2.	<p><u>Тема 2. UI/UX та Usability. Agile (Гнучка модель). Scrum. Selenium</u></p> <p><i>Зміст теми:</i></p> <p>Гнучка модель (Agile). Життєвий цикл Agile. Agile особливості. Scrum. Scrum Framework: Roles. Scrum Master. Product Backlog. Product Owner. Team. Time boxes. Daily Scrum. Sprint Review. Sprint Retrospective. QC in Scrum. Scrum і Kanban – інструменти процесу. Selenium. Інтерфейс користувача (UI). Інтерфейс командного рядка (Command-line interface, CLI). Графічний інтерфейс користувача (Graphical user interface, GUI). SILK (Speech, image, language, knowledge). Сенсорний інтерфейс користувача (Touch user interface). Адаптивний інтерфейс (Hardware interface). Веб інтерфейс (WUI). Досвід взаємодії (UX). Usability. Usability Testing.</p>
<b>Змістовний модуль II. Тестування програмного забезпечення</b>	
1.3.	<p><u>Тема 3. Типи тестування</u></p> <p><i>Зміст теми:</i></p> <p>Рівні тестування. Компонентне або модульне тестування (Component Testing or Unit Testing). Інтеграційне тестування (Integration Testing). Системне тестування (System Testing). Приймальне тестування (Acceptance Testing). Функціональне тестування (Functional Testing) ПЗ. Тестування безпеки (Security and Access Control Testing). Тестування взаємодії (Interoperability Testing). Нефункціональне тестування (Non-functional testing) ПЗ. Тестування продуктивності (Performance Testing). Тестування навантаження (Load Testing). Стресове тестування (Stress Testing). Тестування стабільності або надійності (Stability/Reliability Testing). Об'ємне тестування (Volume Testing). Тестування установки (Installation Testing). Тестування зручності користування (Usability Testing). Тестування на відмову і відновлення (Failover and Recovery Testing). Конфігураційне тестування (Configuration Testing). Тестування, яке пов'язане зі змінами (Regression testing).</p>

	Димове тестування (Smoke Testing). Санітарне тестування (Sanity Testing). Тестування збірки (Build Verification Testing). Кросбраузерність (Cross-browser).
1.4.	<u>Тема 4. Аналіз вимог (Requirement analysis)</u> Зміст теми: Визначення і класифікація вимог. Рівні та типи вимог. Характеристики хороших вимог. Requirements Traceability. Traceability Matrix. Залежність (Dependency). Dependency Structure Matrix (DSM). Типи документів, через які вимоги можуть бути повідомлені. Аналіз вимог. Відстеження і залежні матриці
1.5.	<u>Тема 5. Техніки тест дизайну (Test Design Techniques)</u> Зміст теми: Процес тест дизайну. Тестові випадки (Test Case). Тестові умови (Test Items). Методи генерування тестів. Типи технік дизайну. Статичні техніки. Рецензування (Review). Статичний аналіз (Static Analysis). Динамічні техніки. Класи еквівалентності (Equivalence Class). Еквівалентне розбиття. Метод граничних значень. Таблиці рішень. Метод білого ящика або структурні методи. Покриття операторів (Statement Testing). Покриття рішень (Decision Testing). Критерії покриття шляхів. Вибір техніки тест дизайну
1.6.	<u>Тема 6. Test Cases, Checklist and Bug Report. Fundamental Test Process.</u> Зміст теми: Checklist – історія виникнення. Сфери застосування Checklist-ів. Методика створення Checklist. Функціональне тестування. Модульне тестування (Unit Testing). Що таке Test Case. Основні атрибути Test Case. Проблемні тестові випадки. Структура Test Case. Основні стани Test Case. Система відстеження помилок (Bugtracking System). Специфікація тест-кейса. Життєвий цикл дефекту. Типовий цикл дефекту (Defect Cycle). Інструменти управління тест-кейсом. Об'єкти модульного тестування. Переваги модульного тестування. Інтеграційне тестування. Об'єкти інтеграційного тестування. Система безперервної інтеграції. Системне тестування. Об'єкти системного тестування. Основні підходи системного тестування. Регресійне тестування. Види регресійного тестування. Прийнятне тестування. Методика створення Test Case. Тестовий набір (Test Suite). Інструменти створення Test Case. Bug. Bug Report. Цілі Bug Report. Логіка побудови Bug Report. Життєвий цикл Bug Report. Методика створення Bug Report. Bug Tracking Systems. Опис дефекту. Тестування мобільних додатків. Функціональне тестування мобільних додатків. Тестування сумісності мобільних додатків. Localization Testing. Laboratory Testing. Performance Testing. Stress Testing. Security Testing. Usability Testing. Документація з тестування. Test Policy. Test Strategy. Test Plan.

## 5. ТЕМИ ЛАБОРАТОРНО-ПРАКТИЧНИХ ЗАНЯТЬ

№ з/п	Теми лабораторних занять	К-сть годин
1	Тестування простого предмету	2
2	UI Testing	2
3	Методологія управління проектами Scrum. Роль QA та QC в Scrum	2
4	Типи тестування. Засоби навантажувального тестування (тест навантаження з Apache JMeter)	2
5	Тестування SOAP і REST запитів (тестування без інтерфейсу)	2
6	Аналіз вимог до ПЗ	2
7	Техніки тест дизайну при розробці тестових випадків	2
8	Створення контрольних списків (Checklist)	2

9	Функціональне тестування. Проектування та створення тестових випадків (Test Case). Створення тестових випадків (Test Case) за допомогою інструмента Testrail	2
10	Дефекти та Redmine. Написання Bug Report за допомогою інструмента Jira	2
11	Тестування мобільних додатків	2
<b>Всього:</b>		<b>22</b>

## 6. САМОСТІЙНА РОБОТА

№	Назва теми, що виноситься на самостійне опрацювання	К-сть годин
1.	Класифікація базових методів тестування ПС	3
2.	Термінологія тестування та ключові питання тестування	3
3.	Інструментальні платформи для автоматизованого тестування ПЗ ІС	3
4.	Методи тестування ПЗ, засновані на коді, на використанні та на типі програмного забезпечення ІС	3
5.	Моделі вимірювання та тестування програмного коду ІС	3
6.	Вимоги замовника і розробника до ПС та до якості ПС	3
7.	Комунікація вимог на етапах ЖЦ ПС	3
8.	Відображення вимог до ПС на характеристики якості	3
9.	Побудова цільової моделі якості ПЗ ІС відповідно до вимог	3
10.	Зовнішнє тестування програм та аналіз рівня якості ПС	3
11.	Статичне, динамічне та функціональне тестування	3
12.	Модель процесу тестування	3
13.	Класифікація дефектів ПЗ	3
14.	Тестування переходів між станами	3
15.	Методи "білого" та "чорного" ящиків	3
16.	Інженерія надійності. Моделі надійності програмних систем	3
17.	Прогнозуючі моделі. Вимірювальні моделі	3
18.	Моделі без підрахунку помилок.	3
19.	Модель Мотлі-Брукса і модель М.Холстеда	3
<b>Всього:</b>		<b>58</b>

## 7. МЕТОДИ НАВЧАННЯ

При викладанні навчальної дисципліни «Якість програмного забезпечення та тестування» застосовуються інформаційні та практичні методи навчання, а саме: виконання та захист лабораторних робіт, тестування та опитування, презентації результатів опрацювання лекційних занять, опанування термінологічними поняттями засобами створення тлумачного словника, а також консультування з виконання самостійної роботи студентів.

Методи навчально-пізнавальної діяльності: пояснювально-ілюстративний метод, репродуктивний метод, метод проблемного викладу, частково-пошуковий або евристичний метод, дослідницький метод.

Методи стимулювання й мотивації навчально-пізнавальної діяльності: індуктивні і дедуктивні методи навчання, методи стимулювання і мотивації навчання.

## 8. МЕТОДИ КОНТРОЛЮ

Відповідно до плану вивчення дисципліни «Якість програмного забезпечення та тестування» передбачається проведення поточного та підсумкового контролю.

Основними формами проведення поточного контролю є виконання та захист лабораторних робіт. Захист лабораторних робіт представляє собою усну відповідь на запитання викладача в межах теми лабораторної роботи та надання звіту про виконану роботу. При оцінці усної відповіді враховуються: знання теоретичного матеріалу з відповідної теми; цілісність та повнота відповіді на поставлені запитання; оперування науковими визначеннями та поняттями; термінологічна та технічна грамотність відповіді; логічність та лаконічність викладу матеріалу; уміння довести свою думку; уміння супроводжувати відповідь графічними засобами.

Перевірка результатів опрацювання теоретичних питань, проводиться у формі тестових завдань або формування реферату, презентації, розробки проекту.

Контроль самостійної роботи студентів спрямований на виявлення рівня розвитку пізнавальних здібностей та творчої ініціативи студентів, самостійності, відповідальності та організованості; рівня сформованості мислення, здібностей до саморозвитку, самовдосконалення та самореалізації та опанування студентами елементів методики наукових досліджень.

## 9. ФОРМА ПІДСУМКОВОГО КОНТРОЛЮ УСПІШНОСТІ НАВЧАННЯ

Формою підсумкового контролю є *залік*. Кожен модуль включає бал оцінки поточної роботи студента на лабораторних заняттях. Заходи з модульного контролю проводяться по завершенню вивчення навчального матеріалу даного модуля. При згоді студента, до відомості обліку успішності може бути проставлена залікова оцінка на підставі поточного рейтингового балу.

Виставлення результатів поточної роботи студентів протягом семестру проводиться наступним чином:

$$\boxed{\text{Поточна успішність (макс. 100 балів)}} = \boxed{\text{Підсумкова оцінка}}$$

Студенти, які бажають покращити свій рівень навчальних досягнень з дисципліни, можуть скласти залік додатково (форма проведення – тести).

$$\boxed{\text{Поточна успішність}} + \boxed{\text{Тести}} = \boxed{\text{Підсумкова оцінка}}$$

Студент має право підвищувати оцінку «добре», «задовільно», отриману за результатами поточного контролю, склавши семестровий залік.

## 10. СХЕМА НАРАХУВАННЯ ТА РОЗПОДІЛ БАЛІВ, ЯКІ ОТРИМУЮТЬ СТУДЕНТИ

### Накопичення балів протягом семестру

№ з/п	Вид діяльності	Кількість балів за дидактичну одиницю	Кількість	Загальна кількість балів
1	Практична частина (виконання практичних робіт із захистом звіту)	5	11	55
2	Теоретична частина (контроль на лекціях та тестування)	4	6	24
3	Самостійна робота (створення тлумачного словника)	21	1	21
<b>Максимальна кількість балів:</b>				<b>100</b>

### Загальна оцінка знань студентів за поточним контролем

Результати поточного контролю знань студентів в цілому оцінюються в діапазоні від **0** до **100** балів.

### Шкала оцінювання: національна та ECTS

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою	
		для екзамену, курсового проекту (роботи), практики	для заліку
90-100	A	відмінно	зараховано
82-89	B	добре	
74-81	C		
66-73	D	задовільно	
60-65	E		
30-59	FX	незадовільно з можливістю повторного складання	не зараховано з можливістю повторного складання
1-29	F	незадовільно з обов'язковим повторним вивченням дисципліни	не зараховано з обов'язковим повторним вивченням дисципліни

Загальна підсумкова оцінка в балах, за національною шкалою та за шкалою ECTS заноситься до заліково-екзаменаційної відомості, навчальної картки та залікової книжки студента.

## 11. МЕТОДИЧНЕ ЗАБЕЗПЕЧЕННЯ

- робоча навчальна програма дисципліни;
- електронний курс на платформі дистанційного навчання;
- плани лекцій, практичних занять та самостійної роботи студентів;
- тези лекцій з дисципліни;
- тестові завдання до тем лекцій;
- перелік питань до екзамену.

## 12. РЕКОМЕНДОВАНА ЛІТЕРАТУРА

### *Основна:*

1. Бандура В.В. Якість програмного забезпечення та тестування: Лабораторний практикум / В. В. Бандура. – Івано-Франківськ: ІФНТУНГ, 2018, - 69с. – електронний варіант.
2. Білас О. Якість програмного забезпечення та тестування: навч. посібник.- Львів: Видавництво Львівської політехніки, 2011.- 216с.
3. В. В. Бандура, В. І. Шекета, М. М. Піх. Якість програмного забезпечення та тестування: конспект лекцій. – Івано-Франківськ: ІФНТУНГ, 2022. – 199 с.
4. Інформаційні технології. Вимірювання програмного забезпечення. Вимірювання функційного розміру. Частина 3. Верифікація методів вимірювання функційного розміру (ISO/IEC TR 14143-3:2003, IDT): ДСТУ ISO/IEC TR 14143-3:2013 – [Чинний від 2014-07-01]. – К.: МІНЕКОНОМПРОЗВИТКУ України, 2014. – 16 с. – (Національний стандарт України).
5. Кузь М.В. Документування та кваліметрія програмних продуктів: Навчальний посібник / М.В. Кузь, С.І. Мельничук, Л.М. Заміховський // Видання друге перероблене і доповнене – Івано-Франківськ: ІФНТУНГ, 2014. – 116 с. (гриф МОН).
6. Програмна інженерія. Якість продукту. Частина 1. Модель якості (ISO/IEC 9126-1:2001, IDT): ДСТУ ISO/IEC 9126-1:2013 – [Чинний від 2014-07-01]. – К.: МІНЕКОНОМПРОЗВИТКУ України, 2014. – 20 с. – (Національний стандарт України).
7. Програмна інженерія. Якість продукту. Частина 2. Зовнішні метрики (ISO/IEC TR 9126-2:2003, IDT): ДСТУ ISO/IEC TR 9126-2:2008 – [Чинний від 2010-07-01]. – К.: Держспоживстандарт України, 2011. – 85 с. – (Національний стандарт України).
8. Програмна інженерія. Якість продукту. Частина 3. Внутрішні метрики (ISO/IEC TR 9126-3:2003, IDT): ДСТУ ISO/IEC TR 9126-3:2012 – [Чинний від 2013-05-01]. – К.: МІНЕКОНОМПРОЗВИТКУ України, 2013. – 46 с. – (Національний стандарт України).
9. Програмна інженерія. Якість продукту. Частина 4. Метрики якості під час використання (ISO/IEC TR 9126-4:2004, IDT): ДСТУ ISO/IEC TR 9126- 4:2012 – [Чинний від 2013-05-01]. – К.: МІНЕКОНОМПРОЗВИТКУ України, 2013. – 49 с. – (Національний стандарт України).
10. Системи управління якістю. Вимоги. (ISO 9001:2015, IDT): ДСТУ ISO 9001:2015 – [Чинний від 2016-07-01]. – К.: ДП «УкрНДНЦ», 2016. – 30 с. – (Національний стандарт України).
11. Системи управління якістю. Основні положення та словник термінів. (ISO 9000:2015, IDT): ДСТУ ISO 9000:2015 – [Чинний від 2016-07-01]. – К.: ДП «УкрНДНЦ», 2016. – 49 с. – (Національний стандарт України).
12. Якість програмного забезпечення та тестування: базовий курс. Навчальний посібник / За ред. Крепич С.Я., Співак І.Я. / для бакалаврів галузі знань 12

- «Інформаційні технології» спеціальності 121 «Інженерія програмного забезпечення». – Тернопіль: ФОП Паляниця В.А., 2020. – 478с.
13. Gojko Adzic. Fifty Quick Ideas to Improve Your Tests / Gojko Adzic, David Evans, Tom Roden// Neuri Consulting LLP. – 2015. – 198p.
  14. ISO/IEC 9126-1: 2001. Software engineering. Product quality. Part 1: Quality Model.
  15. Jonathan Rasmusson. The Way of the Web Tester. A Beginner's Guide to Automating Tests / J. Rasmusson// Pragmatic Bookshelf. – 2016. – 258p.

#### ***Додаткова:***

1. Бородкіна І.Л. Інженерія програмного забезпечення: навч. посіб. / І.Л. Бородкіна, Г.Л. Бородкін. – Київ. : ЦУЛ, 2019. – 204 С.
2. Дизайнери настрою. «Золотий перетин у веб-дизайні» [Електронний ресурс] – Режим доступу: <http://jarlex.com/article/zolo-toe-sechenie-v-veb-dizajne/> .
3. Життєвий цикл програмного забезпечення: навчальний посібник / Є.В. Левус, Т.А. Марусенкова, О.О. Нитребич. – Львів. : Видавництво Львівської політехніки, 2017. – 208 с.
4. Кучеров Д.П. Інженерія програмного забезпечення: навчальний посібник / Д.П. Кучеров, Є.Б. Артамонов. – Київ. : НАУ, 2017. – 386 с.
5. Лавріщева К. М. Програмна інженерія. Підручник. – К.: Академперіодика, 2008. – 319 с.
6. Левус, Є.В. Вступ до інженерії програмного забезпечення : навчальний посібник / Є.В. Левус, Н.Б. Мельник. - Львів. : Видавництво Львівської політехніки, 2018. - 246 С.
7. Об'єктно-орієнтоване моделювання при проектуванні вбудованих систем і систем часу: навчальний посібник: затверджено МОН України / В.В. Литвинов, С.В. Голуб, К.М. Григор'єв, В.Ю. Жигульська. – Київ-Черкаси. : ІнтролігаТОР, 2011. – 511 С.
8. Bugs Catcher. Thinking about high quality testing [Electronic resource] – Access mode: <http://bugscatcher.net/archives/3307>
9. Lars Lundberg. Software quality attributes and trade-offs. / Lars Lundberg, Michael Mattson, Claes Wohlin. – Blekinge Institute of Technology, 2005.

#### ***Інформаційні ресурси:***

1. Портал знань. [Електронний ресурс]. – Режим доступу: <http://www.znannya.org/?view=software-testing-testing> .
2. Система дистанційного навчання Міжнародного Європейського Університету. Курс «Якість програмного забезпечення та тестування». Режим доступу: <https://dist.ieu.edu.ua/course/view.php?id=631>.
3. Software Testing Help. “Entries Tagged 'Cookie Testing. Website Cookie Testing, Test cases for testing web application cookies?’”.- [Electronic resource]. – Режим доступу: <http://www.softwaretestinghelp.com/category/cookie-testing/>
4. Web-testing. [Electronic resource]. – Режим доступу: <http://www.edb.utexas.edu/minliu/multimedia/PDFfolder/WebTestingPadolina.pdf>