

**ДЕРЖАВНИЙ ТОРГОВЕЛЬНО-ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
СИСТЕМА УПРАВЛІННЯ ЯКІСТЮ**

Система забезпечення якості освітньої діяльності та якості вищої освіти
сертифікована на відповідність ДСТУ ISO 9001:2015 / ISO 9001:2015

Кафедра інженерії програмного забезпечення та кібербезпеки

ЗАТВЕРДЖЕНО

вченою радою Факультету інформаційних
технологій

(прот. № 14 від «26» 06 2024 р.)

Декан



Олександр ХАРЧЕНКО

**ТЕХНОЛОГІЇ РОЗРОБКИ ТА ТЕСТУВАННЯ
ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ/
TECHNOLOGIES OF DEVELOPMENT AND TESTING OF
SOFTWARE**

**РОБОЧА ПРОГРАМА /
COURSE OUTLINE**

освітній ступінь	магістр	/	master
галузь знань	12 Інформаційні технології	/	Information Technology
спеціальність	122 Комп'ютерні науки	/	Computer science
освітня програма	Комп'ютерні науки	/	Computer science

Київ 2024

**Розповсюдження і тиражування без офіційного дозволу ДТЕУ
заборонено**

Автори: Т.О. ЖИРОВА, кандидат педагогічних наук, доцент
кафедри інженерії програмного забезпечення та
кібербезпеки
Н.О. КОТЕНКО, кандидат педагогічних наук, доцент
кафедри інженерії програмного забезпечення та
кібербезпеки

Робочу програму розглянуто і затверджено на засіданні кафедри
інженерії програмного забезпечення та кібербезпеки «3» червня 2024р.,
протокол № 36.

Рецензенти: О.І. ПУРСЬКИЙ, доктор фізико-математичних наук,
професор, завідувач кафедри комп'ютерних наук та
інформаційних систем;
А.М. ДЕСЯТКО, PhD, доцент, доцент кафедри інженерії
програмного забезпечення та кібербезпеки;
О.О. РУДЕНКО, SFCC Front-End Team Lead, Raccoon
LLC

**ТЕХНОЛОГІЇ РОЗРОБКИ ТА ТЕСТУВАННЯ
ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ/
TECHNOLOGIES OF DEVELOPMENT AND TESTING OF
SOFTWARE**

**РОБОЧА ПРОГРАМА /
COURSE OUTLINE**

освітній ступінь	магістр	/	master
галузь знань	12 Інформаційні технології	/	Information Technology
спеціальність освітня програма	122 Комп'ютерні науки Комп'ютерні науки	/	Computer science Computer science

1. Структура дисципліни та розподіл годин за темами (тематичний план)

Назва теми	Кількість годин				Форми контролю*
	Всього годин / кредитів	з них			
		Лекції	Лабораторні заняття	Самостійна робота	
1	2	4	5	7	8
Тема 1. SDLC, SDM. Гнучкі методології розробки програмного забезпечення	13	4	4	5	ПСР, ПІЗ
Тема 2. Виявлення вимог до продукту. Тестування документації та вимог	18	4	4	10	ПСР, ПІЗ
Тема 3. Поняття та види тестування	12	2		10	КР, ПСР
Тема 4. Тестова документація	14	2	2	10	Т, ПСР
Тема 5. Основні аспекти тест-дизайну, техніки тест-дизайну	16	2	4	10	ПСР, ПІЗ, ЗП
Тема 6. Тестування вебдодатків	16	2	4	10	ПСР, ПІЗ, Т
Тема 7. Тестування безпеки вебдодатків	12	2		10	Т, ПСР, ПІЗ
Тема 8. Автоматизація тестування вебдодатків	16	2	4	10	Т, ПСР, ПІЗ
Тема 9. Тестування мобільних додатків	12	2		10	Т, ПСР, ПІЗ, ЗП
Тема 10. Автоматизація тестування мобільних додатків	16	2	4	10	Т, ПІЗ
Тема 11. Тестування комп'ютерних ігор	16	2	4	10	Т, ПСР, ПІЗ, ЗП
Тема 12. ШІ в управлінні проектами та тестуванні програмного забезпечення	12	2		10	Т, ПСР, ПІЗ
Тема 13. Метрики та методики оцінки. Оцінка трудовитрат, планування та звітність в тестуванні	7	2		5	МКР
Разом	180год/ 6кр	30	30	120	
Підсумковий контроль – Екзамен письмовий					

* ПСР – перевірка самостійної роботи; ПІЗ – перевірка індивідуальних завдань; ЗП – захист проєкту, Т – тестування, МКР – модульна контрольна робота.

2. Тематика та зміст лекцій, практичних занять, самостійної роботи студентів

Результати навчання	Навчальна діяльність	Робочий час студента (год.)
1	2	3
Тема 1. SDLC, SDM. Гнучкі методології розробки програмного забезпечення		
<p>Знати основні моделі життєвого циклу програмного забезпечення. Стандарти управління життєвим циклом ПЗ (ISO12207, ISO15504). Основи управління якістю розробки. Стандарти серії ISO9000.</p> <p>Етапи роботи відповідно до гнучкої моделі розробки ПЗ.</p> <p>Вміти застосовувати теоретичні знання для визначення оптимальної моделі життєвого циклу для визначеного ПЗ. Працювати в команді за моделлю Scrum; визначати потенційно необхідні людино-години для виконання поставленого завдання.</p>	<p>Лекція 1. SDLC, SDM. Гнучкі методології розробки програмного забезпечення</p> <p style="text-align: center;">План лекції</p> <ol style="list-style-type: none"> 1. Предмет вивчення і задачі дисципліни. 2. Поняття життєвого циклу програмного продукту. 3. Методології розробки програмного забезпечення. 4. Моделі розробки ПЗ: каскадна, еволюційна, покрокова, формальна, спіральна та ін. 5. Методологія Agile: визначення та коротка історія. 6. Маніфест Agile Software Development. 7. Концепція Scrum методології, основні терміни. 8. Requirements in Scrum. 9. Product Vision. Task, User Story, Epic, Theme. <p>Список рекомендованих джерел: Основний: 4 [с. 21-136] Додатковий: 9 [с. 3-24] Інтернет-ресурси: 14, 15, 17, 18</p>	4
	<p>Завдання для самостійної роботи: Вивчення матеріалів лекції на основі самостійного опрацювання основних літературних джерел, зазначених у списку та електронних матеріалів, які надаються викладачем. Проаналізувати зазначені стандарти та підготувати доповідь по основним положенням. Обґрунтувати свій вибір.</p> <ol style="list-style-type: none"> 1. Стандарти управління життєвим циклом ПЗ (ISO12207, ISO15504). 2. Основи управління якістю розробки. Стандарти серії ISO9000. <p>Список рекомендованих джерел: Основний: 4 [с. 21-136] Додатковий: 9 [с. 3-24] Інтернет-ресурси: 14, 15, 17, 18</p>	5
	<p>Лабораторна робота №1 «Розробка прототипу сайту за Scrum методологією»</p> <p><i>Мета: навчитися працювати в команді під час розробки прототипу сайту з дотриманням усіх етапів Scrum методології.</i></p> <p><i>Завдання:</i></p> <ol style="list-style-type: none"> 1. Визначитися з ролями в команді у відповідності до Scrum методології. 	4

1	2	3
	<p>2. В межах командної роботи визначити тематику сайту.</p> <p>3. В межах двох пар провести три спрінта у відповідності з Scrum методологією.</p> <p>4. Провести презентацію отриманого результату.</p>	
Тема 2. Тема 2. Виявлення вимог до продукту. Тестування документації та вимог		
<p>Знати основні принципи та алгоритми тестування документації; складати список вимог до ПЗ;</p> <p>Вміти застосовувати різні техніки тестування вимог до ПЗ.</p>	<p style="text-align: center;">Лекція 2. Виявлення вимог до продукту. Тестування документації та вимог</p> <p style="text-align: center;">План лекції</p> <p>1. <i>Поняття вимог до програмного забезпечення та їх значення.</i></p> <p>2. <i>Документування вимог до програмної продукції.</i></p> <p>3. <i>Рівні та типи вимог. Властивості якісних вимог.</i></p> <p>4. <i>Техніки тестування вимог. Приклад аналізу і тестування вимог. Типові помилки під час аналізу і тестування вимог.</i></p> <p>5. <i>Інструментальні засоби для командної роботи під час створення вимог до програмного забезпечення.</i></p> <p>6. <i>Trello, як система управління проектами в режимі онлайн в гнучких методологіях розробки програмного забезпечення.</i></p> <p>Список рекомендованих джерел: Основний: 4 [с. 136-182] Додатковий: 9 [с. 24-36] Інтернет-ресурси: 14, 15, 17, 18</p>	4
	<p>Завдання для самостійної роботи: Вивчення матеріалів лекції. Підготувати доповідь-презентацію на одну із запропонованих тем.</p> <p>Теми доповіді:</p> <p>1. Джерела та шляхи визначення вимог: спільний (мозковий штурм, вимоги до семінарів, фокус-групи, інтерв'ю, опитування/анкетування), дослідження (аналіз документів, аналіз інтерфейсу, моделювання процесів), експерименти (прототипування, спостереження).</p> <p>2. Моделювання вимог до продукту. Управління вимогами до продукту.</p> <p>3. Документування бізнес-вимог.</p> <p>4. Документування вимог користувачів та рішення.</p> <p>Пройти онлайн-курс «UA Business Analysis Lab» [27]</p> <p>Список рекомендованих джерел: Основний: 1, 2, 4 Додатковий: 5-7, 9, 12 Інтернет-ресурси: 14, 15, 17, 18</p>	10
	<p>Лабораторна робота №2 «Тестування вимог до програмного забезпечення»</p> <p><i>Мета: навчитися створювати та тестувати вимоги програмного забезпечення.</i></p> <p><i>Завдання:</i></p> <p>1. Для заданого програмного забезпечення розробити вимоги до його розробки.</p>	4

1	2	3
	2. Здійснити аналіз вимог до ПЗ іншої команди. 3. Здійснити уточнення вимог та написати звіт до отриманих результатів. 4. Здійснити тестування отриманих вимог.	
Тема 3. Поняття та види тестування		
<p>Знати загальну класифікацію методів тестування ПЗ, визначення кожного методу.</p> <p>Вміти здійснювати тестування в залежності від методу тестування</p>	<p style="text-align: center;">Лекція №3. Поняття та види тестування</p> <p style="text-align: center;">План лекції</p> <ol style="list-style-type: none"> 1. Принципи тестування. 2. Підходи до тестування. 3. Схема класифікації тестування. 4. Класифікація за запуском коду на виконання. 5. Класифікація за доступом до коду і архітектурі додатку. 6. Класифікація за ступенем автоматизації. 7. Класифікація за рівнем деталізації додатку (за рівнем тестування). 8. Класифікація за рівнем важливості функцій, які тестуються (за рівнем функціонального тестування). Класифікація за принципами роботи з додатком. Класифікація за природою додатку. 9. Класифікація за фокусом на рівні архітектури додатку. 10. Класифікація за рівнем залучення кінцевих користувачів. <p>Список рекомендованих джерел: Основний: 1 [с. 13-67], 2 [с. 18-65] Додатковий: 8 [с. 3-54] Інтернет-ресурси: 19, 20, 21</p>	2
	<p>Завдання для самостійної роботи: Вивчення матеріалів лекції. Підготувати презентацію на одну із запропонованих тем:</p> <ol style="list-style-type: none"> 1. Класифікація тестування за рівнем формалізації. Класифікація за цілями і задачами. Класифікація за техніками і підходами. Класифікація за моментом виконання (хронологією). 2. Альтернативні та додаткові класифікації тестування. <p>Використовуючи Мірго побудувати схему класифікації тестування.</p> <p>Список рекомендованих джерел: Основний: 1 [с. 13-67], 2 [с. 18-65] Додатковий: 8 [с. 3-54] Інтернет-ресурси: 19, 20, 21</p>	10
Тема 4. Тестова документація		
<p>Знати основну термінологію та атрибути щодо створення планування тестування.</p> <p>Вміти</p>	<p style="text-align: center;">Лекція №4. Тестова документація</p> <p style="text-align: center;">План лекції</p> <ol style="list-style-type: none"> 1. Поняття Check-list, правила його створення. 2. TestCase і його життєвий цикл. 3. Атрибути (поля) testcase. 4. Властивості якісних testcase. Набір testcase. 5. Логіка створення ефективних перевірок. 6. Типові помилки під час розробки check-list, testcase і 	2

1	2	3
<p>планування тестування, записувати відповідні атрибути для тест-кейсів.</p>	<p><i>наборів testcases.</i></p> <p>7. <i>Основна термінологія: помилки, дефекти, збої, відмови.</i></p> <p>8. <i>Звіт про дефект та його життєвий цикл.</i></p> <p>9. <i>Атрибути (поля) звіту про дефекти.</i></p> <p>10. <i>Властивості якісних дефектів.</i></p> <p>11. <i>Логіка створення ефективних звітів про дефекти.</i></p> <p>12. <i>Типові помилки під час написання звітів про дефекти.</i></p> <p>Список рекомендованих джерел: Основний: 4 [с. 200-231] Додатковий: 10 [с. 8-16] Інтернет-ресурси: 16, 19, 20, 21</p>	
	<p>Завдання для самостійної роботи: Вивчення матеріалів лекції.</p> <p>Здійснити порівняльний аналіз:</p> <p>1. <i>Інструментальних засобів управління тестуванням.</i></p> <p>2. <i>Інструментальних засобів керування звітами про дефекти.</i></p> <p>Оформити дане аналітичне дослідження у вигляді статті.</p> <p>Список рекомендованих джерел: Основний: 4 [с. 200-231] Додатковий: 10 [с. 8-16] Інтернет-ресурси: 16, 19, 20, 21</p>	10
	<p>Лабораторна робота №3 «Планування тестування, створення check-list, testcase і набору testcases»</p> <p><i>Мета: навчитися здійснювати планування та створювати check-list, testcase і набору testcases.</i></p> <p><i>Завдання:</i></p> <ol style="list-style-type: none"> 1. <i>Здійснити аналіз вимог до відповідного ПЗ.</i> 2. <i>Розробити стратегію тестування для заданих вимог.</i> 3. <i>Написати check-list, testcase і набору test cases для заданого ПЗ.</i> 	2
Тема 5. Основні аспекти тест-дизайну, техніки тест-дизайну		
<p>Знати техніки тестування, їх переваги та недоліки.</p> <p>Вміти обирати відповідну техніку тестування в залежності від поставленої задачі, вміти застосовувати різні техніки тестування.</p>	<p>Лекція 5. Основні аспекти тест-дизайну, техніки тест-дизайну</p> <p>План лекції</p> <ol style="list-style-type: none"> 1. <i>Поняття тест-дизайну. Важливість тест-дизайну та його рецензування.</i> 2. <i>Класифікація технік тест-дизайну за Лі Копландом.</i> 3. <i>Техніки тест-дизайну.</i> 4. <i>Тестування на основі класів еквівалентності.</i> 5. <i>Аналіз граничних значень.</i> 6. <i>Таблиця прийняття рішень</i> 7. <i>Метод попарного тестування.</i> 8. <i>Тестування на основі причинно-наслідкових зв'язків,</i> <p>Список рекомендованих джерел: Основний: 4 [с. 231-248] Додатковий: 10 [с. 18-24] Інтернет-ресурси: 16, 19, 20, 21</p>	2
	<p>Завдання для самостійної роботи: Вивчення матеріалів лекції.</p> <p>Написати тези доповіді, які досліджують такі вити тест-</p>	10

1	2	3
	<p>дизайну:</p> <ol style="list-style-type: none"> 1. Тестування на основі сценаріїв. 2. Тестування на основі діаграм станів. 3. Тестування на основі діаграм потоків даних. 4. Тестування на основі вимог. <p>Список рекомендованих джерел: Основний: 4 [с. 231-248] Додатковий: 10 [с. 18-24] Інтернет-ресурси: 16, 19, 20, 21</p>	
	<p>Лабораторна робота №4 «Техніки тест-дизайну для тестування ПЗ»</p> <p><i>Мета: навчитися застосовувати техніку класів еквівалентності та граничних умов для тестування ПЗ.</i></p> <p><i>Завдання:</i></p> <ol style="list-style-type: none"> 1. Проаналізувати вимоги до визначеного ПЗ. 2. Відповідно до вимог розробити тест-кейси. 3. Здійснити тестування за написаними тест-кейсами. 4. Оформити звіт по помилкам. 	4
Тема 6. Тестування вебдодатків		
<p>Знати архітектуру та особливості web-додатків. Етапи та особливості тестування web-додатків.</p> <p>Вміти здійснювати тестування web-додатків урахуванням їх особливостей, застосовуючи як ручне тестування, так і відповідні інструментальні засоби тестування.</p>	<p style="text-align: center;">Лекція №6. Тестування вебдодатків План лекції</p> <ol style="list-style-type: none"> 1. Основні види тестування вебдодатків: функціональне тестування; тестування юзабіліті (перевірка зручності користування); тестування сумісності (конфігураційне тестування); тест на продуктивність; перевірка безпеки. 2. Функціональне тестування вебдодатку: перевірка посилань (внутрішніх, вихідних, ізольованих і т. д.); перевірка форм; тест файлів cookie; валідація HTML/CSS; перевірка бази даних. 3. Перевірка зручності використання: навігаційне тестування; перевірка контенту; комфорт користування; тестування доступності. 4. Тестування сумісності: кросплатформений тестінг; перевірка на кросбраузерність; тестування на мобільних пристроях. 5. Тестування продуктивності: тестування навантаження; стрес-тестування; перевірка на швидкість з'єднання. <p>Список рекомендованих джерел: Основний: 4 [с. 248-260] Додатковий: 10 [с. 24-36] Інтернет-ресурси: 16, 19, 20, 21</p>	2
	<p>Завдання для самостійної роботи: Вивчення матеріалів лекції. Повторити тему «Архітектура та особливості вебдодатків» Дослідити, які сайти є доступними та який рівень доступності вони отримали. Дослідження оформити у вигляді презентації та зробити доповідь на парі.</p> <p>Список рекомендованих джерел: Основний: 4 [с. 248-260] Додатковий: 10 [с. 24-36] Інтернет-ресурси: 16, 19, 20, 21</p>	10

1	2	3
	<p>Лабораторна робота №5 «Дослідницьке тестування вебдодатків»</p> <p><i>Мета: ознайомитися з основними методами та інструментами для ручного тестування вебдодатків, навчитися проводити тестування функціоналу, продуктивності та безпеки вебдодатків.</i></p> <p>Завдання:</p> <ol style="list-style-type: none"> 1. Ознайомитися з теоретичними основами тестування вебдодатків, зокрема з поняттями: тестування функціональності, юзабіліті, продуктивності та безпеки. 2. Скласти план тестування: визначити функціональність, яку необхідно перевірити, та скласти сценарій тестування. 3. Провести ручне функціональне тестування вебдодатку, перевіривши основні функції: авторизацію, пошук, додавання товарів до кошика тощо. 4. Виконати тестування інтерфейсу (юзабіліті) для оцінки зручності користування додатком. 5. Скласти звіт про результати тестування, включаючи знайдені баги, їхнє критичне значення та рекомендації щодо їх усунення. <p><i>Звіт повинен містити: план тестування, сценарій тестування, результати тестування (знімки екрану з виявленими багами), висновки та рекомендації.</i></p>	4
Тема 7. Тестування безпеки вебдодатків		
<p>Знати архітектуру та особливості web-додатків та мобільних додатків. Етапи та особливості тестування web-додатків та мобільних додатків та комп'ютерних ігор.</p> <p>Вміти здійснювати тестування web-додатків та мобільних додатків та комп'ютерних ігор з урахуванням їх особливостей, застосовуючи як</p>	<p style="text-align: center;">Лекція №7. Тестування безпеки вебдодатків</p> <p style="text-align: center;">План лекції</p> <ol style="list-style-type: none"> 1. Проект по забезпеченню безпеки вебдодатків. 2. Проекти над якими працює OWASP: OWASP Security Knowledge Framework (SKF), OWASP Mobile Security Testing Guide (MSTG), OWASP Web Security Testing Guide (WSTG), OWASP Zed Attack Proxy (ZAP). 3. Принципи тестування, методологія тестування за OWASP. 4. Активне та пасивне тестування. 5. Розвідка і уразливості вебдодатків: відкриття вебсторінки/структури програми; збір інформації в вебдодатках; сканування вразливостей вебдодатків. <p>Список рекомендованих джерел: Основний: 4 [с. 261-273] Додатковий: 8 [с. 61-80], 10 [с. 36-41] Інтернет-ресурси: 20, 21, 22, 23</p>	2
	<p>Завдання для самостійної роботи. Вивчення матеріалів лекції.</p> <ol style="list-style-type: none"> 1. Ознайомитися з методологією тестування OWASP WSTG. 2. На основі вивченої інформації скласти план тестування для вебдодатку, обравши три категорії тестів із WSTG (наприклад, тестування автентифікації, тестування 	10

1	2	3
ручне тестування, так і відповідні інструментальні засоби тестування.	<p>управління сесією, тестування введення даних).</p> <p>3. Підготувати сценарії тестування для кожної категорії.</p> <p>Список рекомендованих джерел: Основний: 4 [с. 261-273] Додатковий: 8 [с. 61-80], 10 [с. 36-41] Інтернет-ресурси: 20, 21, 22, 23</p>	
Тема 8. Автоматизація тестування вебдодатків		
<p>Знати: переваги автоматизації тестування (швидкість, повторюваність, масштабованість), недоліки (початкові витрати, складність у налаштуванні та підтримці), а також основні галузі застосування автоматизованого тестування. основні інструменти автоматизації тестування вебдодатків (Selenium IDE, WebDriver) та їх можливості.</p> <p>Вміти: визначати випадки, коли автоматизація тестування є доцільною, та розробляти відповідні тест-кейси для автоматизації. розробляти автоматизовані сценарії для оцінки продуктивності під різними</p>	<p>Лекція 8. Автоматизація тестування вебдодатків План лекції</p> <ol style="list-style-type: none"> 1. Переваги і недоліки автоматизації. Галузь застосування автоматизації. Особливості тест-кейсів в автоматизації. Технології автоматизації тестування. 2. Використання продуктів Selenium IDE. Selenium WebDriver. Прийоми розробки тестів: пошук елементів (локаторів), дії з елементами, використання фрагментів JavaScript-коду. 3. Тестування продуктивності. Автоматизація тестів для оцінки продуктивності вебдодатка під різними навантаженнями. Інструменти: JMeter, Gatling, LoadRunner. 4. Тестування сумісності. Автоматизація тестів для перевірки роботи вебдодатка в різних браузерах та на різних платформах. Інструменти: BrowserStack, Sauce Labs. 5. API тестування. Автоматизація тестів для перевірки коректності роботи API. Інструменти: Postman, REST Assured, SoapUI. 6. Інфраструктура запуску тестів: запуск з консолі, запуск в різних браузерах, зберігання тестів на прикладі сервісу GitHub. Керування потоками виконання тестів. <p>Список рекомендованих джерел: Основний: 4 [с. 273-281] Додатковий: 7 [с. 3-320] Інтернет-ресурси: 11, 12, 13</p>	2
	<p>Завдання для самостійної роботи</p> <ol style="list-style-type: none"> 1. Вивчити матеріали лекції. 2. Підготувати порівняльний аналіз інструментів Selenium IDE та WebDriver. Включити основні функції, переваги та недоліки кожного інструмента. Пояснити, в яких випадках краще використовувати кожен з інструментів. 3. Написати тестовий сценарій на Selenium WebDriver для перевірки функціональності пошукової форми на вебсайті. <p>Список рекомендованих джерел: Основний: 4 [с. 273-281] Додатковий: 7 [с. 3-320] Інтернет-ресурси: 11, 12, 13</p>	10

1	2	3
навантаженнями	<p>Лабораторна робота №6 «Автоматизація тестування вебдодатків»</p> <p>Мета: ознайомитися з інструментами та методами автоматизації тестування вебдодатків, навчитися створювати та виконувати автоматизовані тести для функціонального, продуктивного, сумісного та API тестування вебдодатків.</p> <p>Завдання:</p> <ol style="list-style-type: none"> 1. Використовуючи Selenium WebDriver, створіть автоматизований тест для перевірки функціональності авторизації на сайті (перевірка введення правильного та неправильного логіна і пароля). 2. Створіть сценарій тестування продуктивності з використанням JMeter для навантажувального тестування вебдодатка. Налаштуйте тест для 50, 100, і 200 одночасних користувачів. 3. Використайте BrowserStack або Sauce Labs для перевірки сумісності вебдодатка в різних браузерах (наприклад, Chrome, Firefox, Safari) та на різних платформах (Windows, macOS, Android). 4. Використовуючи Postman або REST Assured, розробіть тест для перевірки функціональності API вебдодатка (наприклад, отримання списку товарів з онлайн-магазину). Додати перевірки на правильність відповіді сервера, коди статусів HTTP та вміст відповіді. 5. Налаштуйте середовище для автоматичного запуску тестів з консолі та зберігання проекту в репозиторії GitHub. Виконайте запуск тестів через командний рядок для декількох браузерів. <p>Вимоги до звіту:</p> <ol style="list-style-type: none"> 1. Опис ходу виконання кожного завдання. 2. Виведення результатів тестування та звіти про продуктивність, сумісність та API. 3. Код автоматизованих тестів, скріншоти та звіти з JMeter, BrowserStack/Sauce Labs та Postman/REST Assured. 4. Інструкція з налаштування та запуску тестів з консолі. 	4
Тема 9. Тестування мобільних додатків		
Знати: основи мобільних додатків. Особливості тестування мобільних додатків.	<p style="text-align: center;">Лекція 9. Тестування мобільних додатків</p> <p style="text-align: center;">План лекції</p> <ol style="list-style-type: none"> 1. Особливості тестування мобільних додатків: типи тестування мобільних додатків: тестування зручності використання, тестування сумісності, тестування інтерфейсу, тестування сервісів, тестування ресурсів низького рівня, тестування продуктивності, тестування 	2

1	2	3
<p>Типи тестування мобільних додатків. Стратегії тестування мобільних додатків. Інструменти для тестування мобільних додатків. Вміти: виконувати різні типи тестування мобільних додатків, включаючи тестування на зручність використання, сумісність, продуктивність та безпеку. Аналізувати результати тестування та ідентифікувати проблеми з безпекою, продуктивністю та інтерфейсом.</p>	<p><i>експлуатації, тестування встановлення, тестування безпеки.</i></p> <ol style="list-style-type: none"> 2. Стратегія тестування мобільних додатків. 3. Техніки тестування безпеки мобільних додатків. Автентифікація, авторизація, безпека даних, уразливості для злону, управління сесансами. 4. Інструменти тестування мобільних додатків: <i>ImmuniWeb MobileSuite; Zed Attack Proxy; Kiuwan; QARK; Micro Focus; Android Debug Bridge; CodifiedSecurity; Drozer; WhiteHat Security; Synopsys; Veracode; Mobile Security Framework.</i> <p>Список рекомендованих джерел: Основний: 2 [152-167] Додатковий: 9 [с. 80-90], 10 [с. 81-92] Інтернет-ресурси: 16, 18, 19</p> <p>Завдання для самостійної роботи</p> <ol style="list-style-type: none"> 1. Опрацювати матеріали лекції. 2. Підготувати короткий огляд різновидів мобільних додатків (нативні, гібридні, вебдодатки) та описати їх основні відмінності. 3. Підготувати презентацію з описом і порівнянням інструментів для тестування мобільних додатків (наприклад, ImmuniWeb MobileSuite, ZAP, QARK, Drozer). 4. Дослідити техніки тестування безпеки мобільних додатків та навести приклади реальних вразливостей, що були виявлені за допомогою інструментів для тестування безпеки. <p>Список рекомендованих джерел: Основний: 2 [152-167] Додатковий: 9 [с. 80-90], 10 [с. 81-92] Інтернет-ресурси: 16, 18, 19</p>	<p>3</p> <p>10</p>
Тема 10. Автоматизація тестування мобільних додатків		
<p>Знати: інструменти для тестування мобільних додатків. Особливості автоматизації тестування для мобільних вебдодатків та вміти використовувати CSS локатори для цього типу додатків. Основні</p>	<p style="text-align: center;">Лекція 10. Автоматизація тестування мобільних додатків</p> <p style="text-align: center;">План лекції</p> <ol style="list-style-type: none"> 1. Інструменти для автоматизації та їх дефінітивний аналіз. 2. Проблеми та особливості Appium та WebDriver. Середовище розробки. Запуск та налаштування Appium. Знайомство з UIAutomatorViewer. Appium Inspector. Підключення до програми, пошук елементів інтерфейсу, використання в коді. 3. Локатори: Xpath, Id, Class, Name, Css (mobile web only). 4. Підготовка тестового додатку, підготовка сценарію тест-кейсу, застосування вбудованих методів Click, Find_element, Send_keys та ін. Очікування Implicit vs Explicit, валідація та Asserts. 5. Паттерни та антипаттерни. Тестування жестів Swipe, Reporting, Listeners. 	<p>2</p>

1	2	3
<p>паттерни та антипаттерни тестування жестів (наприклад, Swipe) та розуміти, як уникати поширених помилок.</p>	<p>6. Особливості автоматизації тестування мобільних вебдодатків.</p> <p>Список рекомендованих джерел: Основний: 1 [с. 172-256], 4 [с. 280-296] Додатковий: 5 [с. 4-240] Інтернет-ресурси: 11, 12, 13, 16, 18, 19</p>	
<p>Вміти: аналізувати результати тестування та ідентифікувати проблеми з безпекою, продуктивністю та інтерфейсом. Виконувати тестування безпеки мобільних додатків, перевіряти автентифікацію, авторизацію, уразливості для атак, управління сесіями.</p>	<p>Завдання для самостійної роботи.</p> <ol style="list-style-type: none"> 1. Вивчити матеріал лекції. 2. Ознайомитися з інструментами Appium, UIAutomatorViewer та Appium Inspector. Підготувати короткий огляд кожного з них, описати їхні можливості та галузі застосування. 3. Дослідити основні паттерни та антипаттерни в автоматизації тестування мобільних додатків. Підготувати коротку доповідь з прикладами коду, що демонструють правильні та неправильні підходи до автоматизації. <p>Список рекомендованих джерел: Основний: 1 [с. 172-256], 4 [с. 280-296] Додатковий: 5 [с. 4-240] Інтернет-ресурси: 11, 12, 13, 16, 18, 19</p>	10
	<p>Лабораторна робота №7 «Тестування мобільних додатків»</p> <p>Мета: ознайомитися з інструментами для тестування мобільних додатків, виконати тестування безпеки, продуктивності та функціональності додатка, а також проаналізувати результати.</p> <p>Завдання:</p> <ol style="list-style-type: none"> 1. Тестування функціональності мобільного додатка за допомогою Android Debug Bridge (ADB). Виконати підключення до мобільного пристрою через ADB та запустити прості команди для перевірки основної функціональності додатка. 2. Тестування безпеки з використанням QARK або ImmuniWeb MobileSuite. Провести тестування безпеки мобільного додатка, визначивши потенційні уразливості (наприклад, перевірка безпеки зберігання даних або управління сесіями). 3. Тестування продуктивності з використанням Mobile Security Framework. Виконати тестування продуктивності мобільного додатка під навантаженням з використанням Mobile Security Framework або інших відповідних інструментів. 4. Аналіз сумісності за допомогою BrowserStack або Sauce Labs. Перевірити сумісність мобільного додатка на різних версіях операційних систем та пристроях за допомогою BrowserStack або Sauce Labs. <p>Вимоги до звіту:</p> <ol style="list-style-type: none"> 1. Докладний опис виконаних завдань та отриманих 	4

1	2	3
	<p>результатів.</p> <p>2. Виведення та аналіз результатів тестування.</p> <p>3. Рекомендації щодо покращення функціональності, безпеки та продуктивності мобільного додатка.</p>	
Тема 11. Тестування комп'ютерних ігор		
<p>Знати основні аспекти тестування геймдизайну, такі як ігровий баланс, інтуїтивність геймплея, і реакція гравців. Тестування гри на різних платформах та конфігураціях пристроїв. методи тестування сценаріїв і сюжетних ліній, перевірку інтерактивності гри.</p> <p>Вміти проводити тестування геймдизайну на основі балансування ігрових механік і зручності користувацького інтерфейсу. Використовувати спеціалізовані інструменти для автоматизованого тестування ігрових механік та продуктивності гри. Здійснювати тестування на кросплатформен</p>	<p>Лекція 11. Тестування комп'ютерних ігор План лекції</p> <ol style="list-style-type: none"> 1. Геймдизайн та його тестування. Основні аспекти тестування геймдизайну: ігровий баланс, інтуїтивність геймплея, реакція гравців. 2. Ігрові механізми, тестування ігрових механізмів: контроль і управління, бойові та інші системи, система винагород. 3. Тестування продуктивності та тестування FPS. 4. Тестування сумісності: тестування кросплатформенності, конфігураційне тестування. Тестування Геймплея. Тестування рівнів: перевірка складності, тестування логіки рівнів. 5. Тестування сценаріїв: тестування сюжетних ліній, перевірка інтерактивності. Тестування користувацького інтерфейсу (UI). 6. Особливості тестування ігрових платформ. Вимоги до сертифікації: дотримання вимог сертифікації кожної платформи (правила PlayStation Certification). 7. Тестування ігор на мобільних пристроях. 8. Інструменти для тестування комп'ютерних ігор: Unity Test Tools, Unreal Engine Automation Testing, Appium, Selenium, Fraps. <p>Список рекомендованих джерел: Основний: 1 [с. 256-308], 4 [с. 296-430] Додатковий: 6 [с. 126-153], 7 [23-40] Інтернет-ресурси: 11, 12, 13, 16, 18, 19</p>	2
	<p>Завдання для самостійної роботи</p> <ol style="list-style-type: none"> 1. Опрацювати матеріали лекції. 2. Дослідити ігрові механіки однієї обраної гри та підготувати аналіз їх тестування: основні виклики, які можуть виникнути під час тестування. 3. Вивчити вимоги до сертифікації для однієї з популярних ігрових платформ (PlayStation, Xbox, Nintendo) та підготувати презентацію про ключові моменти сертифікації. 4. Ознайомитися з Unity Test Tools та Unreal Engine Automation Testing. Підготувати огляд можливостей цих інструментів для тестування комп'ютерних ігор. <p>Список рекомендованих джерел: Основний: 1 [с. 256-308], 4 [с. 296-430] Додатковий: 6 [с. 126-153], 7 [23-40] Інтернет-ресурси: 11, 12, 13, 16, 18, 19</p>	10

1	2	3
<p>ність, включаючи тестування на різних пристроях та операційних системах. Проводити тестування рівнів гри та їх складності, а також перевіряти сценарії та сюжетні лінії на інтерактивність. Аналізувати продуктивність гри, вимірювати FPS та забезпечувати відповідність до стандартів продуктивності. Тестувати мобільні ігри та готувати їх до сертифікації на різних платформах.</p>	<p>Лабораторна робота №8 «Тестування комп'ютерних ігор» Мета: навчитися проводити автоматизоване та ручне тестування комп'ютерних ігор за допомогою спеціалізованих інструментів, аналізувати продуктивність гри та забезпечувати її сумісність на різних платформах. Завдання:</p> <ol style="list-style-type: none"> Провести тестування ігрового процесу (геймплея) для перевірки інтуїтивності управління та ігрового балансу. Використати Unity Test Tools для перевірки ігрових механік. Провести тестування продуктивності гри на різних налаштуваннях графіки та виміряти FPS. Підготувати звіт про вплив налаштувань на продуктивність гри. Використати BrowserStack або Sauce Labs для перевірки роботи гри на різних платформах (PC, Mac, Android). Оцінити кросплатформенну сумісність та підготувати звіт. Створити тест для перевірки коректності роботи сценарію гри та інтерактивності з гравцем. Використати Unreal Engine Automation Testing для автоматизації цього процесу. Провести тестування мобільної гри, використовуючи Appium для перевірки її продуктивності та функціональності на Android або iOS. <p>Звіт:</p> <ol style="list-style-type: none"> Опис ходу виконання кожного завдання. Результати тестування продуктивності та сумісності. Висновки щодо ігрового балансу, тестування сценаріїв та геймплея. Звіт про використання інструментів для автоматизації тестування. 	4
Тема 12. III в управлінні проєктами та тестуванні програмного забезпечення		
<p>Знати: методи автоматизації управлінських процесів на основі III. Генерацію тестових випадків та даних за допомогою III. Методи прогнозування дефектів та аналіз результатів тестування за</p>	<p style="text-align: center;">Лекція 12. III в управлінні проєктами та тестуванні програмного забезпечення План лекції</p> <ol style="list-style-type: none"> Управління проєктами за допомогою III. Оптимізація розподілу ресурсів. Аналіз ризиків. Підтримка прийняття рішень. Автоматизація управлінських процесів на основі III. Автоматизація тестування на основі III: автоматизація виконання тестових сценаріїв, інструменти III для автоматизації тестування. Генерація тестових випадків та даних на основі роботи III. Аналіз результатів тестування на основі III. Прогнозування дефектів. Оптимізація тестового покриття: аналіз програмного коду для визначення областей, які потребують тестування; інструменти III для оптимізації 	2

1	2	3
<p>допомогою ШІ. Вміти: використовувати інструменти ШІ для автоматизації управління проектами та тестування програмного забезпечення. Генерувати тестові сценарії та дані за допомогою ШІ. Аналізувати результати тестування за допомогою інструментів ШІ, проводити прогнозування дефектів. Оцінювати потенційні етичні ризики при використанні ШІ та враховувати питання приватності.</p>	<p><i>тестового покриття.</i></p> <p>5. <i>Інтеграція ШІ в процеси управління та тестування. Інтеграція в процес безперервної інтеграції/розгортання (CI/CD). Моніторинг та звітність.</i></p> <p>6. <i>Виклики та перспективи використання ШІ. Необхідність великих обсягів даних. Етичні питання та приватність. Підтримка та навчання моделей.</i></p> <p>Список рекомендованих джерел: Основний: 4 [с. 438-521] Додатковий: 8 [с. 90-94] Інтернет-ресурси: 24, 25</p> <p>Завдання для самостійної роботи</p> <p>1. <i>Вивчити матеріали лекції.</i></p> <p>2. <i>Використовуючи інструмент з підтримкою ШІ (наприклад, Monday.com або Trello з автоматизацією), створіть проєкт, у якому потрібно розподілити ресурси, спланувати етапи проєкту та провести аналіз ризиків. Оцінити, як ШІ допомагає у прийнятті рішень та аналізі ризиків.</i></p> <p>3. <i>Використовуючи інструмент Testim або AppliTools, створіть тестовий сценарій для вебдодатка. Налаштуйте ШІ для автоматичного визначення змін на сторінках та тестування функціональності. Проаналізуйте результати тестування та опишіть, як ШІ покращив процес автоматизації.</i></p> <p>Список рекомендованих джерел: Основний: 4 [с. 438-521] Додатковий: 8 [с. 90-94] Інтернет-ресурси: 24, 25</p>	<p>10</p>
<p>Тема 13. Метрики та методики оцінки. Оцінка трудовитрат, планування та звітність в тестуванні</p>		
<p>Знати основні типи метрик, які використовуються для оцінки якості вебдодатків. Знати, які метрики застосовуються для оцінки тестових випадків (кількість виконаних тестів, процент успішних/неуспі</p>	<p>Лекція 13. Метрики та методики оцінки. Оцінка трудовитрат, планування та звітність в тестуванні План лекції</p> <p>1. <i>Метрики якості.</i></p> <p>2. <i>Метрики за тестовими випадками.</i></p> <p>3. <i>Метрики за багами/дефектами. Метрики за задачами. Юзабіліті-метрики.</i></p> <p>4. <i>Метрики Kanban.</i></p> <p>5. <i>Метрики SCRUM.</i></p> <p>6. <i>Валідація і верифікація.</i></p> <p>7. <i>Планування і звітність.</i></p> <p>8. <i>Тест-план і звіт про результати тестування.</i></p> <p>9. <i>Оцінка трудовитрат.</i></p> <p>Список рекомендованих джерел: Основний: 4 [с. 613-682] Додатковий: 10 [с. 23-41]</p>	<p>2</p>

1	2	3
<p>шних тестів, покриття тестуванням тощо).</p> <p>Вміти використовувати різні метрики для оцінки якості та прогресу тестування (метрики якості, дефектів, задач, Kanban та SCRUM метрики). Вміти оцінювати необхідні трудовитрати на виконання тестування на основі метрик та попереднього досвіду.</p>	<p>Інтернет-ресурси: 11, 14, 16, 20, 21</p> <p>Завдання для самостійної роботи:</p> <ol style="list-style-type: none"> Дослідити основні метрики якості ПЗ. Ознайомтеся з різними метриками якості ПЗ (надійність, продуктивність, безпека) і підготуйте короткий опис їхньої важливості для тестування. Аналіз метрик дефектів та задач. Визначте, які метрики для аналізу дефектів використовуються у вашому проєкті. Порівняйте кілька типів метрик (пріоритет, кількість відкритих/закритих дефектів тощо) та проаналізуйте їх вплив на процес розробки. Використовуючи симуляцію SCRUM або Kanban процесу, зібрати метрики Velocity та Cycle Time для декількох спринтів/етапів. Підготуйте звіт із поясненням, як ці метрики допомагають у плануванні роботи команди. На основі вимог до вебдодатка розробіть тест-план, який містить опис запланованих тестів, їхні критерії успішності, а також оцінку трудовитрат. На прикладі завершеного проєкту складіть звіт про результати тестування, включаючи всі відповідні метрики та рекомендації для покращення якості. <p>Список рекомендованих джерел: Основний: 4 [с. 613-682] Додатковий: 10 [с. 23-41] Інтернет-ресурси: 11, 14, 16, 20, 21</p>	<p>5</p>
Разом		180 годин/ 6 кредитів
Підсумковий контроль		Письмовий екзамен

Всі лекційні та практичні заняття виконуються на основі інтерактивних методів навчання у комп'ютерному середовищі.

3. СПИСОК РЕКОМЕНДОВАНИХ ДЖЕРЕЛ¹

Основний

1. Jorgensen P. C., DeVries B. Software testing: a craftsman's approach, fifth edition. Auerbach Publishers, Incorporated, 2021. 528 p.
2. Jorgensen P., Vries B. D. Software testing. Taylor & Francis Group, 2021. 423p.
3. Niranjnamurthy M., Khaiyum S., Rakshitha K. P. Trends in software testing. Wiley & Sons, Incorporated, John, 2022. 341 p.
4. Ramonyai J. Software engineering: software development, architecture, design patterns, testing, product management, project lifecycle, TR5, programming, quality assurance, security systems, service and practic. Independently Published, 2022. 729 p.

Додатковий

5. Chandrasekara C. Hands-On Functional Test Automation: With Visual Studio 2017 and Selenium Paperback / Chaminda Chandrasekara, Pushpa Herath–Apress: May 2, 2019. – 252 p.
6. Gundecha M. Selenium Testing Tools Cookbook –Second Edition 2nd Edition / Unmesh Gundecha – Packt Publishing : November 23, 2019. – 326 p.
7. Horalek J., Urbanik P., Sobeslav V. Automated tests using selenium framework. Lecture notes of the institute for computer sciences, social informatics and telecommunications engineering. Cham, 2023. P. 23–40.
8. Вакалюк Т.А. Технології тестування програм. Навчально-методичний посібник для студентів напряму 6.040302 Інформатика. – Житомир: Вид-во ЖДУ, 2018. – 196 с.
9. Карпенко М. Ю. Технології створення програмних продуктів та інформаційних систем : навч. посібник / М. Ю. Карпенко, Н. О. Манакова, І. О. Гавриленко ; Харків. нац. ун-т міськ. госп-ва ім. О. М. Бекетова. – Харків : ХНУМГ ім. О. М. Бекетова, 2017. – 93 с.
10. Золотухіна О.А. Якість та тестування інформаційних систем. Навчальний посібник підготовлено до друку для самостійної роботи студентів вищих навчальних закладів. Золотухіна О.А., Негоденко О.В., Резник С.Ю., Разіна С.Я. Київ: ННІТ ДУТ, 2020. –128 с.

Інтернет-ресурси

11. Automated testing. Encyclopedia of computer graphics and games. Cham, 2024. P. 211. URL: https://doi.org/10.1007/978-3-031-23161-2_300154 (date of access: 10.06.2024).
12. Implementing Automated Software Testing – Continuously Track Progressand Adjust Accordingly – URL:

- <http://www.methodsandtools.com/archive/archive.php?id=94> (date of access: 22.05.24p.)
13. Parsa S. Software testing automation. Cham : Springer International Publishing, 2023. URL: <https://doi.org/10.1007/978-3-031-22057-9> (date of access: 10.06.2024).
14. The Agile System Development Life Cycle (SDLC) – URL: <http://www.ambysoft.com/essays/agileLifecycle.html> (date of access: 24.05.24p.)
15. The History of Software Testing – URL: <http://www.testingreferences.com/testinghistory.php> (date of access: 20.05.24p.)
16. What is documentation testing in software testing?– URL: <http://tryqa.com/what-is-documentation-testing/> (date of access: 22.05.24p.)
17. What is V-model - advantages, disadvantages and when to use it?– URL: <http://tryqa.com/what-is-v-model-advantages-disadvantages-and-when-to-use-it/> (date of access: 10.06.24p.)
18. Моделювання програмного забезпечення : навч.-метод. посіб. [Електронний ресурс] / С. Ю. Манаков та ін. Одеса : Фенікс, 2023. URL: <https://doi.org/10.32837/11300.25952> (дата звернення: 10.06.2024).
19. Родіонов П., Полупан Ю., Родіонова О. Теоретичні засади тестування програмного забезпечення. Наука і техніка сьогодні. 2024. № 3(31). URL: [https://doi.org/10.52058/2786-6025-2024-3\(31\)-941-952](https://doi.org/10.52058/2786-6025-2024-3(31)-941-952) (дата звернення: 10.06.2024).
20. O'Regan G. Software quality assurance. Undergraduate topics in computer science. Cham, 2022. P. 239–246. URL: https://doi.org/10.1007/978-3-031-07816-3_13 (date of access: 10.06.2024).
21. Технічні статті | Онлайн-курси від компанії QATestLab. Онлайн-курси від компанії QATestLab | Головна сторінка. URL: <https://training.qatestlab.com/blog/technical-articles/> (дата звернення: 10.06.2024).
22. Certified Tester Foundation Level Syllabus. International Software Testing Qualifications Board. URL: <https://www.istqb.org/> (date of access: 10.06.2024).
23. OWASP web security testing guide | OWASP foundation. OWASP Foundation, the Open Source Foundation for Application Security | OWASP Foundation. URL: <https://owasp.org/www-project-web-security-testing-guide/> (date of access: 10.06.2024).
24. Штучний інтелект у тестуванні ігор. Онлайн-курси від компанії QATestLab | Головна сторінка. URL: <https://training.qatestlab.com/blog/technical-articles/artificial-intelligence-in-game-testing/> (дата звернення: 10.06.2024).

25. Is AI the ultimate QA? / E. M. Weisberg et al. Journal of digital imaging. 2022. URL: <https://doi.org/10.1007/s10278-022-00598-8> (date of access: 10.06.2024).

26. UA Business Analysis Lab. URL: <https://elearn.epam.com/courses/course-v1:EPAM+UABAL+1121/courseware/35f775e92c27496ead4b6f3dbc97e06c/54076788345d45d9af1ca01afced74c4/> (date of access: 25.09.2024).

27.

**Курсивом зазначені джерела, що є в наявності в бібліотеці ДТЕУ*

¹*Курсивом виділені джерела, наявні в бібліотеці КНТЕУ*