Робоча програма дисципліни «Якість програмного забезпечення та тестування»

СПЕЦІАЛЬНІСТЬ 123 · КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

БАКАЛАВР

120

Годин

Загальний обсяг дисципліни

4

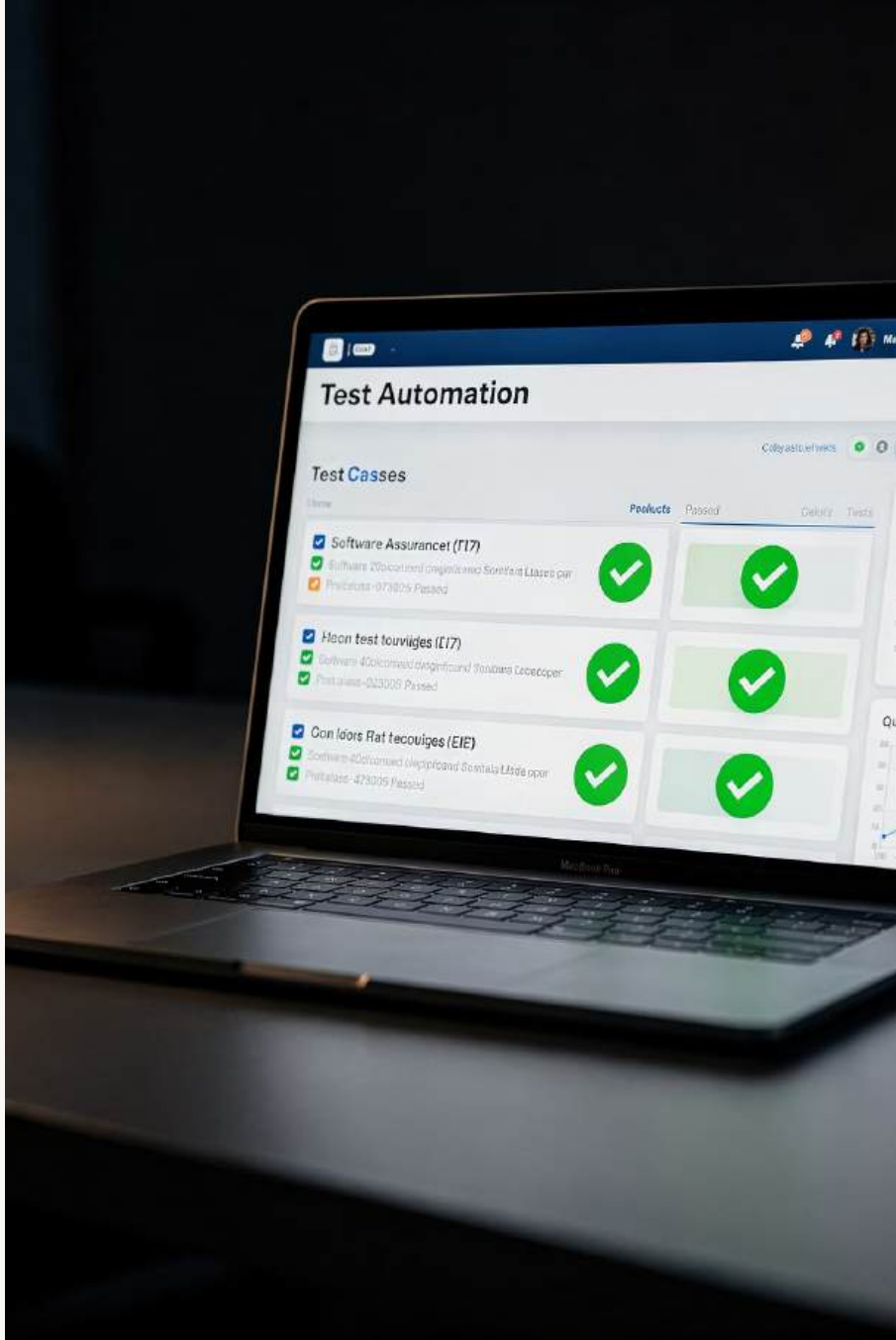
Кредити ЄКТС

Відповідно до стандарту

4

Модулі

Змістовних блоки курсу



Загальна інформація про дисципліну

Ідентифікація

Назва: Якість програмного забезпечення та тестування

Спеціальність: 123 Комп'ютерна інженерія

Ступінь: Бакалавр

Обсяг: 120 годин / 4 кредити ЄКТС

Модулів: 4 змістовних блоки

Призначення дисципліни

Дисципліна «Якість програмного забезпечення та тестування» формує у студентів системне розуміння принципів, методів та інструментів забезпечення якості програмних продуктів. Курс охоплює моделі якості ПЗ, стратегії тестування, процеси верифікації та валідації, автоматизацію тестування та відповідні міжнародні стандарти (ISO/IEC 25010, ISTQB).

Дисципліна є обов'язковою для підготовки бакалаврів зі спеціальності «Комп'ютерна інженерія» та забезпечує формування фахових компетентностей у сфері розробки надійного та якісного програмного забезпечення відповідно до освітньо-професійної програми.

Загальні компетентності

Аналітичне мислення

Здатність до аналізу, синтезу та критичного оцінювання інформації про якість ПЗ: виявлення дефектів, причинно-наслідкових зв'язків між помилками та їх наслідками, формування обґрунтованих висновків щодо якості продукту.

Комунікація

Здатність до усної та письмової комунікації: чітко описувати знайдені дефекти, оформлювати баг-репорти, тест-плани та аналітичні звіти, ефективно взаємодіяти з командою розробників і замовниками.

Самоорганізація

Уміння самостійно навчатися, планувати роботу та організовувати виконання тестових завдань у визначені строки, зокрема під час підготовки тест-кейсів, автоматизованих скриптів і звітів з тестування.

Командна робота

Відповідальність, ініціативність та готовність працювати в команді під час групового тестування програмних систем, спільного аналізу дефектів та розробки стратегій забезпечення якості ПЗ.

Студент повинен вміти: аналізувати вимоги до ПЗ та виявляти потенційні ризики; розробляти і виконувати тест-кейси; складати баг-репорти та звіти з тестування; планувати власну діяльність і дотримуватися дедлайнів; коректно взаємодіяти з розробниками; брати участь у командній роботі в QA-проектах.

Фахові компетентності

Моделі та стандарти якості ПЗ

Розуміння сутності та закономірностей забезпечення якості програмних систем відповідно до міжнародних стандартів (ISO/IEC 25010, ДСТУ ISO/IEC 25010:2025). Знання характеристик якості продукту: функціональна придатність, надійність, зручність використання, ефективність, безпека, супроводжуваність, переносимість.

Методи та техніки тестування

Здатність застосовувати методи тестування «чорного» та «білого» ящиків, функціональне та нефункціональне тестування, регресійне, навантажувальне, тестування безпеки; розробляти тестові сценарії та оцінювати покриття тестами.

Верифікація та валідація ПЗ

Знання процесів верифікації (перевірка відповідності специфікаціям) та валідації (підтвердження відповідності вимогам замовника). Уміння проводити інспекції коду, статичний аналіз та динамічне тестування на різних рівнях (модульний, інтеграційний, системний, приймальний).

Автоматизація тестування

Орієнтування в інструментах автоматизованого тестування (Selenium, JUnit, TestNG, Cypress); використання фреймворків для побудови ефективних тестових сценаріїв; формування обґрунтованих стратегій автоматизації відповідно до потреб проекту.

Результати навчання

01

Пояснення понять якості ПЗ

Пояснювати сутність ключових понять і явищ: якість ПЗ, дефект, тест-кейс, тестовий план, верифікація, валідація, регресія, покриття коду, надійність, метрика якості.

02

Застосування методів тестування

Застосовувати основні методи та техніки тестування ПЗ, розробляти тестові сценарії та тест-кейси, визначати рівні та види тестування відповідно до вимог проекту, оцінювати повноту тестового покриття.

03

Аналіз дефектів та звітність

Аналізувати знайдені дефекти, складати якісні баг-репорти з описом кроків відтворення, очікуваного та фактичного результату; формувати звіти про результати тестування для різних аудиторій.

04

Робота з інструментами та стандартами

Орієнтуватися в інструментальній базі QA-інженера; застосовувати міжнародні стандарти якості ПЗ (ISO/IEC 25010); використовувати системи управління тестуванням для організації процесу QA.

МОДУЛЬ 1

Основи якості програмного забезпечення

1

Тема 1.1

Поняття якості ПЗ. Моделі якості: ISO/IEC 9126, ISO/IEC 25010. Характеристики та метрики якості.

2

Тема 1.2

Процес забезпечення якості (QA). Тестування у життєвому циклі розробки ПЗ (SDLC). Моделі SDLC та місце тестування.

3

Тема 1.3

Рівні тестування: модульне, інтеграційне, системне, приймальне. Стратегії та підходи до тестування.

4

Тема 1.4

Тестова документація: тест-план, тест-кейс, чек-лист, баг-репорт. Стандарти тестової документації.

МОДУЛЬ 1 · ТЕМА 1.1

Поняття якості ПЗ. Моделі та стандарти якості

Лекційний матеріал (2 год.)

Лекція 1: Поняття «якість програмного забезпечення». Еволюція підходів до якості ПЗ. Стандарт ISO/IEC 9126: характеристики якості (функціональність, надійність, зручність використання, ефективність, супроводжуваність, переносимість). Стандарт ISO/IEC 25010: розширена модель якості — 8 характеристик та 31 підхарактеристика. ДСТУ ISO/IEC 25010:2025 — національний стандарт.

Метрики якості ПЗ: поняття метрики, внутрішні та зовнішні метрики, метрики якості у використанні. Інструменти оцінювання якості. Зв'язок між якістю ПЗ та вартістю розробки й супроводження.

Ключові питання лекції

- Еволюція поняття «якість ПЗ» від стандарту ISO 9126 до ISO/IEC 25010
- Характеристики якості за моделлю SQuaRE: функціональна придатність, надійність, безпека
- Метрики якості ПЗ: внутрішні, зовнішні, метрики якості у використанні
- Вплив якості ПЗ на вартість розробки та підтримки
- Роль стандартизації у забезпеченні якості програмних продуктів

Лабораторне заняття (2 год.)

Завдання 1. Аналіз характеристик якості програмного продукту за стандартом ISO/IEC 25010: ідентифікація характеристик та підхарактеристик на прикладі конкретного ПЗ.

Завдання 2. Кейс: оцінювання якості веб-застосунку за обраними характеристиками моделі ISO/IEC 25010. Формування висновків у вигляді звіту.

Завдання 3. Тестування за ключовими термінами теми (10 питань).

Самостійна робота (7 год.)

- Опрацювання посібника: Трофименко О. Г., Дика А. І. «Тестування та забезпечення якості програмних систем» — розділ «Показники якості ПЗ»
- Підготовка реферату: «Моделі якості ПЗ: ISO/IEC 9126 та ISO/IEC 25010 — порівняльний аналіз»
- Складання таблиці характеристик якості ПЗ за стандартом ISO/IEC 25010 з прикладами метрик

Очікуваний результат: студент вміє характеризувати моделі якості ПЗ, пояснювати характеристики стандарту ISO/IEC 25010 та застосовувати метрики якості.

МОДУЛЬ 1 · ТЕМА 1.2

QA у життєвому циклі розробки ПЗ (SDLC)

1

Планування

Визначення цілей якості, стратегії тестування, ресурсів і критеріїв завершення.

2

Проектування

Аналіз вимог, розробка тест-плану, тест-кейсів і тестового середовища.

3

Виконання

Запуск тестів, реєстрація дефектів, повторне тестування після виправлень.

4

Звітність

Аналіз результатів, формування звіту, оцінка якості та рекомендації щодо релізу.

Лекційний матеріал (2 год.)

Поняття SDLC та місце тестування у ньому. Моделі розробки ПЗ: каскадна, ітераційна, спіральна, Agile/Scrum, DevOps — особливості тестування в кожній моделі. V-модель тестування. QA vs QC vs тестування: відмінності понять. Принципи тестування (7 принципів ISTQB). Психологія тестування та незалежність тестувальника.

Лабораторна робота (2 год.) та СРС (7 год.)

Лабораторна: Аналіз V-моделі тестування; порівняльна характеристика підходів до тестування у Waterfall та Agile; дискусія «Тестування у DevOps: переваги та виклики CI/CD».

СРС: Опрацювання: Крепич С. Я., Співак І. Я. «Якість програмного забезпечення та тестування: базовий курс» (2020); підготовка аналітичної записки «7 принципів тестування ISTQB та їх практичне значення».

МОДУЛЬ 1 · ТЕМИ 1.3–1.4

Рівні тестування та тестова документація

Тема 1.3 — Рівні тестування (Лекція 3, 2 год.)

Рівні тестування: модульне (Unit Testing) — тестування окремих компонентів; інтеграційне — перевірка взаємодії модулів; системне — тестування системи в цілому; приймальне (UAT) — перевірка відповідності вимогам замовника. Стратегії «знизу вгору» та «зверху вниз». Тестування «чорного», «білого» та «сірого» ящиків. Регресійне тестування та його роль у CI/CD.

Лабораторна (2 год.): Написання модульних тестів мовою Python/Java із застосуванням фреймворку JUnit/pytest; аналіз покриття коду (code coverage); кейс «Рівні тестування: коли та що тестувати».

СРС (7 год.): Опрацювання: Трофименко О. Г., Дика А. І. «Тестування та забезпечення якості програмних систем» (Одеса: Фенікс, 2024) — розділ «Рівні та техніки тестування»; підготовка реферату «Регресійне тестування в умовах Agile-розробки».

Тема 1.4 — Тестова документація (Лекція 4, 2 год.)

Тест-план: структура, зміст, IEEE 829. Тест-кейс: складові (ідентифікатор, передумови, кроки, очікуваний результат), класифікація. Чек-лист як інструмент швидкої перевірки. Баг-репорт: структура, рівні пріоритету та серйозності дефекту. Системи управління тестуванням: Jira, TestRail, Zephyr. Тестова матриця відповідності вимогам (RTM). Метрики тестового процесу: кількість тест-кейсів, щільність дефектів, показник ефективності виявлення дефектів.

Лабораторна (2 год.): Розробка тест-плану для навчального проекту; складання тест-кейсів за вимогами; оформлення баг-репортів у Jira.

СРС (7 год.): Опрацювання: Крепич С. Я., Співак І. Я. «Якість програмного забезпечення та тестування: базовий курс»; підготовка порівняльної таблиці «Тест-кейс vs чек-лист: переваги та обмеження кожного підходу».

МОДУЛЬ 2

Функціональне та нефункціональне тестування

Тема 2.1

Функціональне тестування. Техніки тестування «чорного ящика»: еквівалентне розбиття, аналіз граничних значень, таблиці рішень, тестування переходів станів.

Тема 2.2

Нефункціональне тестування: навантажувальне, стресове, тестування продуктивності, безпеки, зручності використання (usability).

Тема 2.3

Тестування «білого ящика». Техніки структурного тестування: покриття операторів, гілок, умов. Інструменти статичного аналізу коду.

Тема 2.4

Управління дефектами. Класифікація, пріоритизація та відстеження дефектів. Показники процесу тестування. Метрики якості ПЗ.

МОДУЛЬ 2 · ТЕМА 2.1

Функціональне тестування та техніки «чорного ящика»

Лекційний матеріал (2 год.)

Функціональне тестування: визначення, цілі, область застосування. Техніки тестування «чорного ящика»: еквівалентне розбиття класів — виявлення репрезентативних вхідних значень; аналіз граничних значень — тестування на межах допустимих діапазонів; таблиці рішень — для складних логічних умов; тестування переходів станів — для систем зі станами. Техніка попарного тестування (Pairwise). Тестування варіантів використання (Use Case Testing). Дослідницьке тестування (Exploratory Testing).

Ключові питання

- Переваги та обмеження тестування «чорного ящика»
- Еквівалентне розбиття: як мінімізувати кількість тест-кейсів без втрати покриття
- Аналіз граничних значень: практичне застосування
- Таблиці рішень для тестування складної бізнес-логіки
- Дослідницьке тестування: коли і як застосовувати

Лабораторне заняття (2 год.)

Завдання 1. Розробка тест-кейсів методом еквівалентного розбиття та аналізу граничних значень для форми реєстрації користувача.

Завдання 2. Кейс: побудова таблиці рішень для тестування функції розрахунку знижок в інтернет-магазині. Групова робота та презентація результатів.

Завдання 3. Тестування за ключовими термінами теми (10 питань).

Самостійна робота (7 год.)

- Опрацювання: Крепич С. Я., Співак І. Я. «Якість програмного забезпечення та тестування: базовий курс» — розділ «Техніки функціонального тестування»
- Підготовка реферату: «Техніка Pairwise Testing: принципи та інструменти генерації тестів»
- Складання набору тест-кейсів для навчального веб-застосунку із застосуванням усіх вивчених технік

Очікуваний результат: студент вміє застосовувати техніки тестування «чорного ящика», розробляти тест-кейси на основі вимог до ПЗ та пояснювати переваги кожної техніки.

МОДУЛЬ 2 · ТЕМА 2.2

Нефункціональне тестування

Навантажувальне

Load Testing — перевірка поведінки системи під очікуваним навантаженням.

Продуктивність

Performance Testing — час відгуку, пропускна здатність, використання ресурсів.

Зручність

Usability Testing — оцінка зручності інтерфейсу та досвіду користувача (UX).

Стресове

Stress Testing — поведінка системи в умовах, що перевищують нормальний рівень навантаження.

Безпека

Security Testing — перевірка захисту від вразливостей: SQL ін'єкція, XSS, CSRF.

Лекційний матеріал (2 год.)

Нефункціональне тестування: визначення та класифікація. Тестування продуктивності: навантажувальне, стресове, об'ємне, тестування на витривалість (Soak Testing). Інструменти: Apache JMeter, Gatling. Тестування безпеки: основні вразливості (OWASP Top 10), техніки перевірки безпеки. Тестування зручності використання: евристики Нільсена, методи UX-тестування. Тестування сумісності, доступності та надійності.

Лабораторна (2 год.) та СРС (7 год.)

Лабораторна: Навантажувальне тестування веб-застосунку за допомогою Apache JMeter; аналіз показників продуктивності; дискусія «OWASP Top 10: найпоширеніші вразливості та як їх тестувати».

СРС: Опрацювання: Трофименко О. Г., Дика А. І., Лобода Ю. Г. «Аналіз інструментів тестування вебзастосунків» // Кібербезпека: освіта, наука, техніка. — 2023. — № 4(20). — С. 62–71; підготовка аналітичної записки «Тестування безпеки ПЗ: методи та інструменти».

МОДУЛЬ 2 · ТЕМИ 2.3–2.4

Тестування «білого ящика» та управління дефектами

Тема 2.3 — Тестування «білого ящика» (Лекція 7, 2 год.)

Структурне тестування («білий ящик»): доступ до вихідного коду, критерії покриття. Покриття операторів (Statement Coverage). Покриття гілок (Branch Coverage). Покриття умов (Condition Coverage). Покриття шляхів (Path Coverage). Метрика цикломатичної складності (McCabe). Статичний аналіз коду: інструменти SonarQube, ESLint, PMD. Динамічне тестування: профілювання, налагодження, аналіз пам'яті. Мутаційне тестування як метод оцінки якості тестів.

Лабораторна (2 год.): Аналіз покриття коду тестами за допомогою JaCoCo/pytest-cov; виявлення непокритих гілок; статичний аналіз коду в SonarQube; порівняння методів «чорного» та «білого» ящика.

СРС (7 год.): Опрацювання: Кузьмич О. М., Лахай В. Я., Сенів М. М. «Удосконалений підхід до автоматизованого інтеграційного тестування ПЗ» // Науковий вісник НЛТУ України. — 2023. — Т. 33, № 3. — С. 97–101; підготовка есе «Мутаційне тестування: ефективний метод оцінки якості тестового набору».

Тема 2.4 — Управління дефектами (Лекція 8, 2 год.)

Поняття дефекту (Bug/Defect): визначення, класифікація (функціональні, нефункціональні, UX). Рівні серйозності (Severity): Critical, Major, Minor, Trivial. Рівні пріоритету (Priority): High, Medium, Low. Життєвий цикл дефекту: New → Assigned → Open → Fixed → Retest → Closed / Reopened. Кореневий аналіз дефектів (Root Cause Analysis). Метрики якості тестового процесу: щільність дефектів, показник витоку дефектів (Defect Leakage), ефективність виявлення дефектів. Системи відстеження дефектів: Jira, Bugzilla, Mantis.

Лабораторна (2 год.): Оформлення баг-репортів різного рівня серйозності в Jira; побудова діаграми «відкриті/закриті дефекти» за спринт; аналіз причин дефектів методом «5 Чому».

СРС (7 год.): Підготовка аналітичної записки «Управління якістю в Agile: практики та метрики QA-процесу в Scrum-командах».

МОДУЛЬ 3

Автоматизація тестування

Тема 3.1

Основи автоматизації тестування. Підходи, стратегії та фреймворки автоматизованого тестування. «Піраміда тестування».

Тема 3.2

Автоматизоване модульне та інтеграційне тестування. Фреймворки JUnit, TestNG, pytest. Принципи TDD та BDD.

Тема 3.3

Автоматизація UI-тестування. Selenium WebDriver, Cypress. Page Object Model (POM). Запис та відтворення тестів.

Тема 3.4

Неперервна інтеграція (CI/CD) та автоматизоване тестування. Інструменти CI: Jenkins, GitHub Actions. Звітність у автоматизованому тестуванні.

МОДУЛЬ 3 · ТЕМА 3.1

Основи автоматизації тестування

Лекційний матеріал (2 год.)

Автоматизація тестування: поняття, переваги та обмеження. Коли автоматизувати, а коли проводити ручне тестування. «Піраміда тестування» Майка Кона: Unit → Integration → E2E. Стратегії автоматизації. Класифікація фреймворків автоматизованого тестування: data-driven, keyword-driven, BDD, hybrid. Критерії вибору інструментів автоматизації. ROI (повернення інвестицій) від автоматизації тестування. Обслуговуваність та підтримка автоматизованих тестів.

Ключові питання

- Переваги та обмеження автоматизованого тестування порівняно з ручним
- «Піраміда тестування»: чому більшість тестів мають бути на рівні Unit
- Критерії вибору фреймворку автоматизації для проекту
- Data-driven та BDD підходи: відмінності та застосування
- Розрахунок ROI від впровадження автоматизованого тестування

Лабораторне заняття (2 год.)

Завдання 1. Аналіз «піраміди тестування» для реального проекту: визначення оптимального співвідношення Unit/Integration/E2E тестів, обґрунтування рішень.

Завдання 2. Кейс: розробка стратегії автоматизації для e-commerce застосунку — вибір фреймворку, визначення пріоритетів автоматизації. Групова робота.

Завдання 3. Тестування за ключовими термінами теми (10 питань).

Самостійна робота (7 год.)

- Опрацювання: Крепич С. Я., Співак І. Я. «Якість програмного забезпечення та тестування: базовий курс» — розділ «Автоматизація тестування»
- Підготовка реферату: «BDD з Cucumber/Gherkin: принципи та практика написання специфікацій виконуваних вимог»
- Складання порівняльної таблиці фреймворків автоматизованого тестування: JUnit, TestNG, pytest, Cypress, Selenium

Очікуваний результат: студент вміє обґрунтовувати стратегію автоматизації тестування, класифікувати фреймворки та описувати переваги і обмеження автоматизованого тестування.

МОДУЛЬ 3 · ТЕМА 3.2

Модульне тестування та TDD/BDD

Лекційний матеріал (2 год.)

Модульне (Unit) тестування: ізоляція компонентів, заглушки (Stubs) та імітації (Mocks). Фреймворки: JUnit 5 (Java), pytest (Python), NUnit (C#). Структура тесту: Arrange–Act–Assert (AAA). Test-Driven Development (TDD): цикл Red–Green–Refactor, переваги для якості коду. Behavior-Driven Development (BDD): синтаксис Given–When–Then, Gherkin, Cucumber. Інтеграційне тестування: підходи Big Bang, Bottom-Up, Top-Down; інструменти Mockito, WireMock. Контрактне тестування (Contract Testing).

Лабораторна (2 год.) та СРС (7 год.)

Лабораторна: Написання модульних тестів для класу «Калькулятор» методом TDD (цикл Red–Green–Refactor); використання Mock-об'єктів для ізоляції залежностей; BDD-специфікація у форматі Given–When–Then з використанням Cucumber.

СРС: Опрацювання: Трофименко О. Г., Дика А. І. «Тестування та забезпечення якості програмних систем» (Одеса: Фенікс, 2024) — розділ «Автоматизоване тестування»; підготовка аналітичної записки «TDD vs BDD: порівняльний аналіз підходів до тест-орієнтованої розробки».

Очікуваний результат: студент вміє писати модульні тести у фреймворку JUnit/pytest, застосовувати TDD та BDD підходи, використовувати Mock-об'єкти.

МОДУЛЬ 3 · ТЕМА 3.3

Автоматизація UI-тестування. Selenium & Cypress

Локатори елементів

ID, Name, XPath, CSS Selector — способи ідентифікації UI-елементів для взаємодії в тестах.

Page Object Model

POM — патерн проектування: кожна сторінка = клас з методами, що описують дії та елементи.

Виконання сценаріїв

Запуск тестових сценаріїв, обробка очікувань (Explicit/Implicit Wait), скріншоти при падінні.

Лекційний матеріал (2 год.)

Selenium WebDriver: архітектура, підтримка браузерів, WebDriver API. Локатори елементів: ID, XPath, CSS. Обробка динамічних елементів: явні та неявні очікування. Page Object Model (POM) — шаблон проектування для тестів. Cypress: переваги над Selenium, архітектура, синтаксис. Playwright як сучасна альтернатива. Parallel Testing та хмарні платформи (BrowserStack, Sauce Labs). Генерація звітів: Allure Report, ExtentReports.

Лабораторна (2 год.) та СРС (7 год.)

Лабораторна: Написання UI-тесту для форми реєстрації за допомогою Selenium WebDriver з реалізацією патерну Page Object Model; налаштування очікувань елементів; генерація Allure-звіту.

СРС: Опрацювання: Трофименко О. Г., Дика А. І., Лобода Ю. Г. «Аналіз інструментів тестування вебзастосунків» // Кібербезпека: освіта, наука, техніка. — 2023. — № 4(20). — С. 62–71; підготовка аналітичної записки «Selenium vs Cypress vs Playwright: порівняльний аналіз інструментів UI-тестування».

МОДУЛЬ 3 · ТЕМА 3.4

CI/CD та автоматизоване тестування у DevOps

Лекційний матеріал (2 год.)

Концепція CI/CD (Continuous Integration / Continuous Delivery). Роль автоматизованого тестування у конвеєрі CI/CD. Інструменти CI: Jenkins, GitHub Actions, GitLab CI. Shift-Left Testing — зміщення тестування на ранні етапи розробки. API-тестування: REST, SOAP, інструменти Postman та RestAssured. Тестування мікросервісної архітектури. Контейнеризація та тестування Docker-контейнерів. Test Reporting в CI/CD: Allure, JUnit XML-звіти. Культура якості в DevOps: роль QA-інженера в сучасних командах.

Ключові питання

- Роль тестування в конвеєрі CI/CD та принципи Shift-Left
- API-тестування: методи HTTP, статус-коди, структура відповіді
- Тестування мікросервісів: переваги та виклики
- GitHub Actions: налаштування автоматичного запуску тестів
- DevOps-культура та роль QA: від тестувальника до інженера якості

Лабораторне заняття (2 год.)

Завдання 1. Налаштування GitHub Actions workflow для автоматичного запуску модульних тестів при кожному push у репозиторій; аналіз результатів у веб-інтерфейсі.

Завдання 2. API-тестування RESTful сервісу за допомогою Postman: написання колекції запитів з assertions; автоматизація через Newman CLI.

Завдання 3. Підготовка та презентація міні-проекту: «Повний CI/CD pipeline з автоматизованим тестуванням».

Самостійна робота (7 год.)

- Опрацювання: Крепич С. Я., Співак І. Я. «Якість програмного забезпечення та тестування: базовий курс» — розділ «Автоматизація та CI/CD»
- Підготовка аналітичної записки «Shift-Left Testing: впровадження культури якості на ранніх етапах розробки»
- Аналіз практики тестування мікросервісної архітектури на прикладі відкритих проектів

Очікуваний результат: студент вміє налаштовувати CI/CD конвеєр з автоматизованим тестуванням, проводити API-тестування та застосовувати принципи DevOps у QA.

МОДУЛЬ 4

Управління якістю та сучасні практики QA

Тема 4.1

Процеси та стандарти управління якістю ПЗ. CMMI, ISO 9001, ISTQB. Планування якості та аудит.

Тема 4.2

Верифікація та валідація ПЗ. Статичне тестування: інспекції, огляди, формальні перевірки коду. Метрики процесу.

Тема 4.3

Тестування в Agile та Scrum. Ролі та відповідальності. Регресійне та приймальне тестування у спринтах.

Тема 4.4

Сучасні тенденції QA: AI-assisted testing, тестування хмарних сервісів, тестування мобільних застосунків, безпека ПЗ.

МОДУЛЬ 4 · ТЕМА 4.1

Стандарти та процеси управління якістю ПЗ

Лекційний матеріал (2 год.)

Системи управління якістю: ISO 9001 — вимоги до системи менеджменту якості. CMMI (Capability Maturity Model Integration): 5 рівнів зрілості процесів розробки ПЗ. ISTQB (International Software Testing Qualifications Board): схема сертифікації тестувальників, Foundation Level. Планування якості: Quality Assurance Plan, критерії входу та виходу з тестування. Аудит якості ПЗ: внутрішній та зовнішній. Вимірювання та управління процесом забезпечення якості. Ризик-орієнтоване тестування.

Ключові питання

- CMMI: п'ять рівнів зрілості та їх ознаки
- ISO 9001 у контексті розробки ПЗ: вимоги та практика
- ISTQB Foundation Level: структура іспиту та ключові теми
- Ризик-орієнтоване тестування: пріоритизація тестів за рівнем ризику
- Критерії входу та виходу з тестування у практиці QA

Лабораторне заняття (2 год.)

Завдання 1. Аналіз рівнів зрілості CMMI: оцінювання гіпотетичної IT-компанії за ознаками кожного рівня; формування рекомендацій щодо покращення процесів.

Завдання 2. Дискусія: «Сертифікація ISTQB: чи варто інвестувати час і ресурси?» Аргументація на основі ринку праці та вимог роботодавців.

Завдання 3. Складання матриці ризиків для тестового проекту та планування ризик-орієнтованого тестування.

Самостійна робота (7 год.)

- Опрацювання: Трофименко О. Г., Дика А. І. «Тестування та забезпечення якості програмних систем» (Одеса: Фенікс, 2024) — розділ «Управління якістю»
- Підготовка реферату «CMMI в українській IT-індустрії: стан впровадження та перспективи»
- Аналіз вимог роботодавців до QA-інженерів на ринку праці України за відкритими вакансіями

Очікуваний результат: студент вміє характеризувати стандарти управління якістю ПЗ (ISO 9001, CMMI, ISTQB), планувати ризик-орієнтоване тестування.

МОДУЛЬ 4 · ТЕМА 4.2

Верифікація та валідація. Статичне тестування

Лекційний матеріал (2 год.)

Верифікація vs Валідація: «Чи правильно ми будуємо продукт?» vs «Чи правильний продукт ми будуємо?». Статичне тестування: аналіз артефактів без виконання коду. Техніки статичного аналізу: огляди (Reviews), інспекції (Inspections), наскрізні перегляди (Walkthroughs). IEEE 1028 — стандарт оглядів ПЗ. Статичний аналіз коду: метрики Halstead, метрика цикломатичної складності, дублювання коду. Аналіз вимог: якісні характеристики вимог (SMART), перевірка повноти та несуперечності. Роль статичного тестування у зниженні вартості дефектів.

Лабораторна (2 год.) та СРС (7 год.)

Лабораторна: Проведення формальної інспекції коду (Code Review) за чек-листом; виявлення дефектів вимог у тестовому документі; статичний аналіз у SonarQube — інтерпретація метрик якості.

СРС: Опрацювання: Крепич С. Я., Співак І. Я. «Якість програмного забезпечення та тестування: базовий курс» — розділ «Верифікація та валідація»; підготовка аналітичної записки «Вартість виправлення дефектів на різних фазах SDLC: правило «1–10–100»».

Очікуваний результат: студент вміє проводити статичний аналіз коду та вимог, застосовувати техніки інспекцій та оглядів, розрізняти верифікацію та валідацію.

МОДУЛЬ 4 · ТЕМА 4.3

Тестування в Agile та Scrum

Sprint Planning

Визначення критеріїв приймання (AC), стратегії тестування спринту, оцінка тестових задач.

Sprint Review

Приймальне тестування (UAT), демонстрація замовнику, перевірка критеріїв готовності (DoD).

1 — 2 — 3 — 4

Sprint Execution

Щоденне тестування нових функцій, регресійне тестування, написання автотестів паралельно з розробкою.

Sprint Retrospective

Аналіз якості процесу тестування, виявлення вузьких місць, покращення QA-практик.

Лекційний матеріал (2 год.)

Agile Manifesto та принципи гнучкої розробки. Scrum-модель: ролі (Product Owner, Scrum Master, Dev Team), артефакти (Product Backlog, Sprint Backlog), події. Роль QA в Agile-команді: «Whole Team Approach». User Stories та критерії приймання (Acceptance Criteria). Definition of Done (DoD) і роль тестування. Kanban та тестування. SAFe (Scaled Agile Framework) та тестування у великих командах.

Лабораторна (2 год.) та СРС (7 год.)

Лабораторна: Аналіз User Stories та написання Acceptance Criteria; складання Definition of Done для навчального проекту; симуляція Sprint Planning з плануванням тестових задач; дискусія «Scrum vs Kanban: що краще підходить для QA-процесів?».

СРС: Опрацювання: Трофименко О. Г., Дика А. І. «Тестування та забезпечення якості програмних систем» (Одеса: Фенікс, 2024) — розділ «Agile-тестування»; підготовка аналітичної записки «Роль QA-інженера в Scrum-команді: практика українських IT-компаній».

МОДУЛЬ 4 · ТЕМА 4.4

Сучасні тенденції QA: AI, хмара, мобільні застосунки

Лекційний матеріал (2 год.)

AI-assisted Testing: використання штучного інтелекту для генерації тест-кейсів, аналізу дефектів, самовідновлення тестів (Self-Healing Tests). Інструменти: Testim, Mabl, Functionize. Тестування хмарних сервісів (Cloud Testing): особливості, платформи AWS, Azure, GCP. Тестування мобільних застосунків: Android vs iOS, Appium як кросплатформний інструмент. Тестування безпеки (Security Testing): OWASP Top 10, пентестинг, SAST/DAST-інструменти. Accessibility Testing: стандарт WCAG. Тестування з використанням великих мовних моделей (LLM).

Ключові питання

- AI у тестуванні: можливості та ризики «самовідновлення» тестів
- Тестування хмарних мікросервісів: специфіка та інструменти
- Тестування мобільних застосунків: Appium та платформна специфіка
- SAST vs DAST: статичний та динамічний аналіз безпеки
- Майбутнє QA: як змінюється роль тестувальника в епоху AI

Лабораторне заняття (2 год.)

Завдання 1. Тестування мобільного застосунку за допомогою Appium або BrowserStack: написання базових автоматизованих сценаріїв для Android/iOS.

Завдання 2. Дискусія: «AI-тестування: революція чи еволюція? Чи замінить штучний інтелект QA-інженера?» Аргументація з посиланням на актуальні тренди ринку.

Завдання 3. Підготовка та презентація міні-проекту: «Стратегія тестування для реального застосунку з використанням сучасних інструментів QA».

Самостійна робота (7 год.)

- Опрацювання: Руденко О. А., Руденко З. М., Ревуцька Н. М. «Аналіз моделей надійності програмного забезпечення» (International Scientific Unity, 2024)
- Підготовка аналітичної записки «Перспективи AI-assisted Testing в українській IT-індустрії»
- Аналіз міжнародних стандартів безпеки ПЗ (OWASP) та практики їх застосування у вітчизняних компаніях

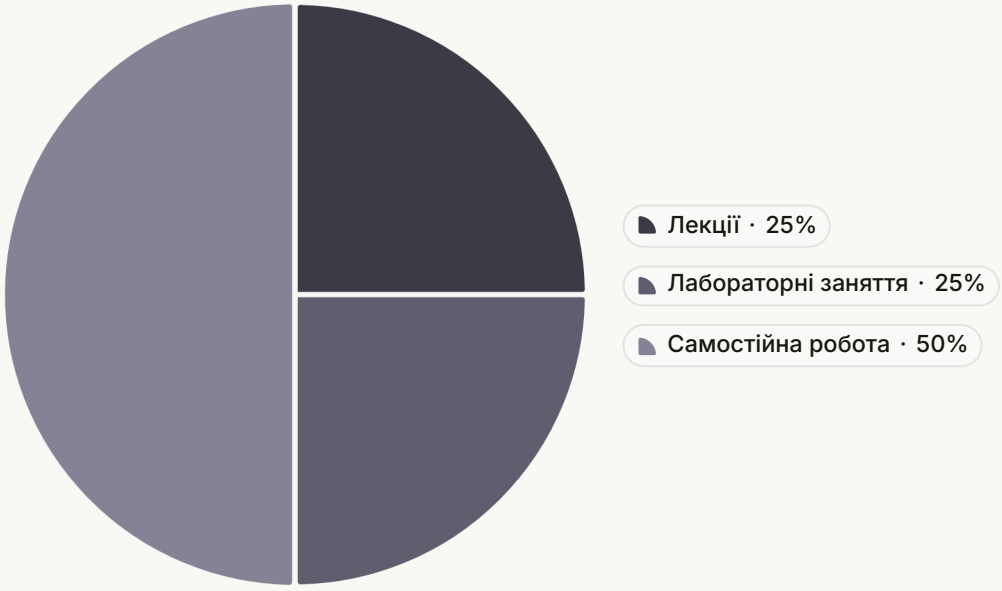
Очікуваний результат: студент вміє аналізувати сучасні тенденції QA, оцінювати можливості AI-assisted Testing та застосовувати інструменти тестування мобільних і хмарних застосунків.

Розподіл навчального часу — 120 годин

Модуль	Тема	Лекції (год.)	Лабораторні (год.)	СРС (год.)
Модуль 1	Тема 1.1 — Моделі та стандарти якості ПЗ	2	2	7
	Тема 1.2 — QA у життєвому циклі SDLC	2	2	7
	Тема 1.3 — Рівні тестування	2	2	7
	Тема 1.4 — Тестова документація	2	2	7
Модуль 2	Тема 2.1 — Функціональне тестування («чорний ящик»)	2	2	7
	Тема 2.2 — Нефункціональне тестування	2	2	7
	Тема 2.3 — Тестування «білого ящика»	2	2	7
	Тема 2.4 — Управління дефектами	2	2	7
Модуль 3	Тема 3.1 — Основи автоматизації тестування	2	2	7
	Тема 3.2 — Модульне тестування, TDD/BDD	2	2	7
	Тема 3.3 — UI-тестування (Selenium, Cypress)	2	2	7
	Тема 3.4 — CI/CD та тестування у DevOps	2	2	7
Модуль 4	Тема 4.1 — Стандарти управління якістю ПЗ	2	4	7
	Тема 4.2 — Верифікація, валідація, статичне тестування	2	2	7
	Тема 4.3 — Тестування в Agile та Scrum	2	2	7
	Тема 4.4 — Сучасні тенденції QA (AI, хмара, мобайл)	2	2	7
Разом	**16 тем**	**30**	**30**	**60**

Загальний обсяг: 120 годин = 4 кредити ЄКТС. Розподіл є орієнтовним і може коригуватися відповідно до специфіки навчального процесу.

Структура навчального часу



Коментар до розподілу

Загальний обсяг — **120 годин (4 кредити ЄКТС)** — розподілено між трьома формами роботи для забезпечення балансу між теорією та практикою.

Лекції — 30 год. (25%)

Системне подання ключового теоретичного матеріалу: моделі якості ПЗ, методи та техніки тестування, стандарти і процеси QA, сучасні інструменти та підходи.

Лабораторні — 30 год. (25%)

Відпрацювання практичних навичок тестування: написання тест-кейсів, автоматизація тестів, налаштування CI/CD, робота з інструментами QA. Рівна частка з лекціями підкреслює практичну спрямованість курсу.

Самостійна робота — 60 год. (50%)

Поглиблене вивчення матеріалу, підготовка завдань, робота з фаховою літературою та документацією інструментів. Самостійна робота становить **50% обсягу дисципліни**.

Самостійна робота студентів — Детальний опис

Опрацювання літератури

Поглиблене вивчення теоретичного матеріалу за підручниками та посібниками з якості ПЗ та тестування (Трофименко, Крепич, Співак та ін.), стандартами ISO/IEC, документацією ISTQB, науковими статтями у фахових виданнях. Студент конспектує ключові положення та порівнює підходи різних авторів.

Індивідуальні завдання

Виконання аналітичних завдань (порівняльний аналіз методів тестування, характеристика інструментів QA, аналіз стандартів); творчих завдань — есе, аналітичні записки, міні-дослідження з актуальних питань якості ПЗ та практик тестування.

Робота з інформаційними ресурсами

Опрацювання офіційних ресурсів: ISTQB (**istqb.org**), OWASP (**owasp.org**), документація Selenium (**selenium.dev**), Cypress (**cypress.io**), SonarQube (**sonarqube.org**). Аналіз наукових статей у фахових виданнях, пошук актуальної інформації про тренди QA.

Підготовка до занять

Підготовка до лабораторних занять (налаштування середовища, вивчення API інструментів, повторення теорії); підготовка до тестувань (ключові терміни, техніки); підготовка до контрольних робіт за кожним модулем (систематизація знань).

Форми контролю знань

1	2	3
Поточний контроль	Проміжний контроль	Підсумковий контроль
Завдання: аналітичні та тестові вправи за темами змістовного модуля. Оцінюються регулярність виконання лабораторних робіт, якість написаних тест-кейсів, баг-репортів та автотестів. Поточний контроль формує накопичувальну оцінку протягом семестру. Проводиться після кожної теми у формі тестування (10–15 питань) або захисту лабораторної роботи.	Самостійна робота — комплексне практичне завдання або реферат, які охоплюють теми змістовних модулів (розробка тест-плану, написання тест-кейсів, автоматизація тестів, аналіз стандартів якості). Оцінюються повнота відповіді, якість артефактів тестування та обґрунтованість висновків. Проводиться після завершення кожного змістовного модуля.	Іспит відповідно до навчального плану. Форма проведення — тестування та практичні завдання (написання тест-кейсів, аналіз вимог). Підсумкова оцінка враховує результати поточного, проміжного та підсумкового контролю відповідно до встановленої шкали: **100-бальна система** з переведенням у національну шкалу та ЄКТС (A, B, C, D, E, FX, F).

Методи викладання

Викладання дисципліни «Якість програмного забезпечення та тестування» поєднує класичні академічні та інтерактивні методи, що забезпечує формування не лише міцної теоретичної бази, а й практичних навичок тестування, необхідних для майбутньої професійної діяльності у сфері комп'ютерної інженерії та IT-індустрії.

Лекції

Системне подання теоретичного матеріалу з використанням наочних схем, діаграм, таблиць та прикладів із реальної практики QA; студенти ведуть конспекти та задають уточнювальні запитання.

Лабораторні роботи

Практична робота з реальними інструментами QA (Selenium, JUnit, JMeter, Jira, SonarQube); написання тест-кейсів, автоматизованих тестів та баг-репортів у реалістичних умовах.

Кейс-стаді

Аналіз реальних ситуацій забезпечення якості ПЗ; студенти виявляють проблему, пропонують стратегію тестування та обґрунтовують вибір інструментів і методів.

Дискусії

Обговорення дискусійних питань QA-практики (ручне vs автоматизоване, Agile vs Waterfall); розвиток критичного мислення, аргументації та здатності відстоювати технічні рішення.

Самостійне дослідження

Підготовка аналітичних записок, рефератів і презентацій за обраною темою QA; розвиток навичок самостійного пошуку та систематизації технічної інформації.

Міні-проекти

Розробка комплексної стратегії тестування для реального застосунку; командна робота, розподіл ролей (QA Lead, QA Engineer, автоматизатор), презентація результатів.

Усі методи спрямовані на формування компетентностей, передбачених освітньо-професійною програмою спеціальності 123 «Комп'ютерна інженерія».

Рекомендована література — Підручники та посібники (2022–2024 рр.)

1. Трофименко О. Г., Дика А. І.
Тестування та забезпечення якості програмних систем : навч. посіб. для здобувачів вищої освіти галузі знань 12 «Інформаційні технології». — Одеса : Фенікс, 2024. — 195 с. DOI: 10.32837/11300.27717

2. Трофименко О. Г., Дика А. І., Лобода Ю. Г.
Аналіз інструментів тестування вебзастосунків // Кібербезпека: освіта, наука, техніка. — 2023. — № 4(20). — С. 62–71. DOI: 10.28925/2663-4023.2023.20.6271

3. Кузьмич О. М., Лахай В. Я., Сенів М. М.
Удосконалений підхід до автоматизованого інтеграційного тестування ПЗ за умов застосування безсерверної архітектури // Науковий вісник НЛТУ України. — 2023. — Т. 33, № 3. — С. 97–101.

4. Руденко О. А., Руденко З. М., Ревуцька Н. М.
Аналіз моделей надійності на основі різнорідного процесу Пуассона з метою врахування неповного налагодження програмного забезпечення. — International Scientific Unity, 2024.

5. Крепич С. Я., Співак І. Я.
Якість програмного забезпечення та тестування: базовий курс : навч. посіб. для бакалаврів галузі знань 12. — Київ : Центр навчальної літератури, 2020. — 479 с. *(Базовий підручник курсу, широко використовується у ЗВО України)*

6. Демиденко М. А.
Управління проектами цифрової економіки : навч. посіб. — Дніпро : НТУ «ДП», 2022. — 187 с. *Рекомендується для тем 4.1–4.3 (управління якістю в Agile).*

7. Трофименко О. Г., Пастернак Ю. Ю., Манаков С. Ю., Лобода Ю. Г.
Автоматизація тестування вебсайтів електронної комерції // Сучасна спеціальна техніка. — 2021. — № 2(65). — С. 46–59. DOI: 10.36486/mst2411-3816.2021.2(65).5. *Рекомендується для тем 3.3–3.4.*

8. ДСТУ ISO/IEC 25010:2025
Інженерія систем і програмних засобів. Вимоги до якості систем і програмних засобів та її оцінювання (SQuaRE). Модель якості продукту. — ДП «УкрНДНЦ», 2025. *Чинний національний стандарт якості ПЗ.*

Додаткова література та наукові статті

9. Єршова О. О., Гончаренко І. М.

Сучасні моделі управління розвитком бізнесу в умовах цифрової трансформації // Журнал стратегічних економічних досліджень. — 2022. — № 2(7). — С. 75–84. *Рекомендується для тем 4.3–4.4 (Agile та управління якістю).*

10. Яковина В. С.

Методи та засоби аналізу надійності функціонування програмного забезпечення з урахуванням етапів життєвого циклу. — Львів : Національний університет «Львівська політехніка». *Рекомендується для теми 1.2 (SDLC та місце тестування).*

11. Федасюк Д. В., Яковина В. С., Сердюк П. В., Нитребич О. О.

Метод побудови сценаріїв тестування програмного забезпечення на основі аналізу його змінних // Вісник НУ «Львівська політехніка». — № 771. — С. 209–213. *Рекомендується для теми 2.1 (функціональне тестування).*

12. Чабанюк Я. М., Кукурба В., Гнатів Л.

Критерій достатності процесу тестування програмного забезпечення // Вісник НУ «Львівська політехніка». Комп'ютерні науки та інформаційні технології. — № 672. — С. 346–358. *Рекомендується для модулів 2–3 (критерії завершення тестування).*

Нормативно-правова база

→ ДСТУ ISO/IEC 25010:2025

Інженерія систем і програмних засобів. Вимоги до якості систем і програмних засобів та її оцінювання (SQuaRE). Модель якості продукту. Чинний з 01.12.2025. **zakon.rada.gov.ua**

→ Закон «Про освіту» від 05.09.2017 № 2145-VIII

Визначає засади освітньої діяльності в Україні, у т. ч. підготовку фахівців у сфері комп'ютерних наук та інженерії. **zakon.rada.gov.ua**

→ Закон «Про вищу освіту» від 01.07.2014 № 1556-VII

Визначає стандарти вищої освіти та вимоги до освітньо-професійних програм підготовки бакалаврів. **zakon.rada.gov.ua**

→ Стандарт вищої освіти спеціальності 123 «Комп'ютерна інженерія»

Затверджений МОН України. Визначає компетентності та результати навчання для бакалаврів за спеціальністю 123. **mon.gov.ua**

→ ISO/IEC 29119 — Стандарт тестування програмного забезпечення

Міжнародний стандарт, що визначає концепції, процеси, документацію та техніки тестування ПЗ. **iso.org**

Офіційні ресурси та фахові видання

ISTQB — International Software Testing Qualifications Board

istqb.org — схема міжнародної сертифікації тестувальників, глосарій термінів, навчальні матеріали Foundation Level.

OWASP — Open Web Application Security Project

owasp.org — відкритий ресурс з безпеки веб-застосунків, OWASP Top 10, чек-листи та методології тестування безпеки.

Верховна Рада України — Законодавча база

zakon.rada.gov.ua — нормативно-правова база, стандарти у сфері IT та освіти, закони та підзаконні акти.

Національна бібліотека України ім. В. І. Вернадського

nbuv.gov.ua — електронні ресурси, наукові статті, монографії та дисертації з інформаційних технологій.

DOU.ua — Спільнота розробників України

dou.ua — аналітика ринку праці QA в Україні, статті практиків, огляди інструментів тестування.

Журнал «Кібербезпека: освіта, наука, техніка»

csecurity.kubg.edu.ua — фахове видання з кібербезпеки та тестування ПЗ; DOI-індексовані статті.

Selenium Documentation

selenium.dev — офіційна документація Selenium WebDriver, API-довідник, навчальні матеріали.

Міністерство освіти і науки України

mon.gov.ua — освітні стандарти, навчальні програми, стандарти вищої освіти за спеціальністю 123.

Зв'язок дисципліни зі спеціальністю 123 «Комп'ютерна інженерія»

Чому «Якість ПЗ та тестування» для інженерів?

Дисципліна «Якість програмного забезпечення та тестування» є невід'ємною складовою підготовки бакалаврів зі спеціальністю 123 «Комп'ютерна інженерія». Розуміння методів QA формує у майбутніх інженерів культуру написання надійного коду, критичне мислення та здатність будувати якісні програмні системи.

Сучасний фахівець у сфері комп'ютерної інженерії працює у командах, де забезпечення якості є невід'ємною частиною процесу розробки. Розуміння тестування, CI/CD та стандартів якості є конкурентною перевагою на ринку праці як в Україні, так і за кордоном.

Практичне значення для майбутньої кар'єри

- Навички QA та тестування є обов'язковими для роботи в провідних українських IT-компаніях
- Розуміння CI/CD і DevOps — ключова вимога роботодавців для Junior-розробників
- Вміння писати Unit-тести підвищує якість власного коду та полегшує Code Review
- Знання стандартів ISO/IEC 25010 та ISTQB дає перевагу при роботі з міжнародними замовниками
- Навички автоматизованого тестування (Selenium, Cypress) відкривають кар'єрний шлях QA-автоматизатора
- Культура якості (TDD, BDD) підвищує надійність та супроводжуваність програмних систем

Ключові компетентності курсу — Підсумок

Курс охоплює всі ключові аспекти забезпечення якості програмного забезпечення — від фундаментальних моделей якості до сучасних практик автоматизованого тестування у DevOps-середовищі — і формує у студентів цілісне розуміння ролі QA-інженера в сучасній IT-індустрії.

Бажаємо успіхів у вивченні якості ПЗ та тестування!

120

Годин

Загальний обсяг дисципліни

4

Кредити ЄКТС

Відповідно до стандарту

16

Тем

У 4 змістовних модулях

8+

Джерел

Рекомендованої літератури 2022–2024 рр.

> «Якість — це не те, що ви вкладаєте у продукт. Це те, що клієнт отримує від нього.» — Пітер Друкер

Спеціальність 123 · Комп'ютерна інженерія · Бакалавр · 4 кредити ЄКТС