



ДЕРЖАВНИЙ ТОРГОВЕЛЬНО-ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій
Кафедра інженерії програмного забезпечення та
кібербезпеки

СИЛАБУС (SYLLABUS)

Дисципліна «Технології розробки та тестування програмного
забезпечення»/
«Technologies Of Development And Testing Of Software»

ІНФОРМАЦІЯ ПРО ВИКЛАДАЧА

Викладач	Жирова Тетяна Олександрівна
Науковий ступінь	Кандидат педагогічних наук
Вчене звання	Доцент
Посада	Доцент кафедри інженерії програмного забезпечення та кібербезпеки
Адреса кафедри	м.Київ, вул. Кіото 19, каб. Б-305
E-mail	zhyrova@knteu.edu.ua
Консультації	Відповідно до графіку індивідуальних консультацій на сайті кафедри

ПОЛІТИКА АКАДЕМІЧНОЇ ДОБРОЧЕСНОСТІ

<https://knteu.edu.ua/file/MzEyMQ==/c12a9f74e87d9154696ca0f761da2e5c.pdf>

Дотримання академічної доброчесності передбачає:

- самостійне виконання навчальних завдань, завдань поточного та підсумкового контролю результатів навчання (для осіб з особливими освітніми потребами ця вимога застосовується з урахуванням їхніх індивідуальних потреб і можливостей);
- посилання на джерела інформації у разі використання не авторських ідей, розробок, тверджень, відомостей і т.п.;
- дотримання норм законодавства про авторське право і суміжні права;
- надання достовірної інформації про результати власної наукової діяльності, використанні методики досліджень і джерела інформації.

Порушенням академічної доброчесності вважається:

- академічний плагіат – оприлюднення (частково або повністю) наукових (творчих) результатів, отриманих іншими особами, як результатів власного дослідження (творчості) та/або відтворення опублікованих текстів (оприлюднених творів мистецтва) інших авторів без зазначення авторства;
- самоплагіат – оприлюднення (частково або повністю) власних раніше опублікованих наукових результатів як нових наукових результатів;
- фабрикація – вигадкування даних чи фактів, що використовуються в наукових дослідженнях;
- фальсифікація – свідомо зміна чи модифікація вже наявних даних, що стосуються наукових досліджень.

За порушення академічної доброчесності здобувачі освіти можуть бути притягнені до академічної відповідальності:

- повторне проходження оцінювання (модульний контроль, іспит, залік тощо);
- повторне проходження відповідного освітнього компонента освітньо-професійної програми;

- відрахування з Університету;
- позбавлення наданих університетом пільг;
- відмова у присудженні відповідного ступеня вищої освіти;

ПОЛІТИКА ЩОДО ВІДВІДУВАННЯ ЗАНЯТЬ

- відвідування занять є обов'язковим;
- студент, який пропустив практичне заняття, самостійно вивчає матеріал (при виникненні питань може звертатися за консультацією згідно розкладу консультацій викладачів оприлюдненого на сайті кафедри) за наведеними джерелами, виконує завдання і здає його викладачу.
- за об'єктивних причин (наприклад, хвороба, міжнародне стажування та ін.) навчання може відбуватись в он-лайн формі за погодженням із викладачем дисципліни.

ПОЛОЖЕННЯ ПРО АПЕЛЯЦІЮ РЕЗУЛЬТАТІВ ЕКЗАМЕНІВ У ДТЕУ

<https://knute.edu.ua/file/MjkwNQ==/cf2f392763bdbe0447eed3c254854ec5.pdf>

ВРЕГУЛЮВАННЯ КОНФЛІКТНИХ СИТУАЦІЙ

Учасники освітнього процесу повинні дотримуватися принципів гідності, взаємоповаги, толерантності, доброчесності. Адміністрація ДТЕУ забезпечує попередження, запобігання, своєчасне виявлення та врегулювання конфліктних ситуацій, пов'язаних із цькуванням, дискримінацією, сексуальними домаганнями (див. Положення про врегулювання конфліктних ситуацій ДТЕУ (<https://knute.edu.ua/file/MjkwMjQ=/b91ca19cb0c629d8b9938ba46ccc41f5.pdf>)).

УЗГОДЖЕННЯ ЗМІСТУ ДИСЦИПЛІНИ З ЦІЛЯМИ СТАЛОГО РОЗВИТКУ (ЦСР)

Зміст дисципліни «Технології розробки та тестування програмного забезпечення» узгоджується з кількома ключовими Цілями сталого розвитку (ЦСР), зокрема:

1. **ЦСР 4: Забезпечення всеохоплюючої і справедливої якісної освіти та заохочення можливості навчання впродовж усього життя для всіх.** Дисципліна сприяє розвитку якісної освіти шляхом забезпечення студентів сучасними знаннями та навичками у сфері розробки і тестування програмного забезпечення. Використання інноваційних технологій та інструментів навчання підвищує конкурентоспроможність студентів на ринку праці та готує їх до викликів цифрового майбутнього.
2. **ЦСР 8: Сприяння поступальному, всеохоплюючому та сталому економічному зростанню, повній і продуктивній зайнятості та гідній праці для всіх.** Викладання принципів тестування та розробки ПЗ сприяє підвищенню продуктивності праці у сфері ІТ, що стимулює інновації та сприяє економічному зростанню. Студенти отримують навички, необхідні для роботи в умовах глобальної економіки та створення безпечних і ефективних цифрових продуктів.
3. **ЦСР 9: Створення стійкої інфраструктури, сприяння всеохоплюючій і сталій індустріалізації та інноваціям.** Тема дисципліни безпосередньо пов'язана з розвитком інфраструктури інформаційних технологій та інновацій у сфері розробки програмного забезпечення. Студенти вивчають технології, які можуть забезпечити інноваційний розвиток та підтримувати стійку інфраструктуру ІТ-систем.
4. **ЦСР 11: Забезпечення відкритості, безпеки, життєстійкості й екологічної стійкості міст, інших населених пунктів.** Завдяки вивченню дисципліни, студенти здобувають знання для розробки екологічно та соціально орієнтованих програмних продуктів, що сприяють розвитку «розумних» міст та стійких спільнот, підвищуючи безпеку та якість життя громадян.
5. **ЦСР 12: Забезпечення переходу до раціональних моделей споживання і виробництва;** Студенти вивчають принципи ефективного управління проектами у сфері розробки ПЗ, що включає відповідальне використання ресурсів, мінімізацію

відходів та оптимізацію процесів тестування. Важливий акцент робиться на створенні якісного програмного забезпечення, яке відповідає стандартам сталого розвитку.

ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Назва дисципліни / тип дисципліни	Технології розробки та тестування програмного забезпечення/ обов'язкова
Навчальний рік	2024-2025
Факультет	Факультет інформаційних технологій
Курс	2
Семестр	4
Освітній ступінь	Магістр
Галузь знань	12 «Інформаційні технології»
Спеціальність	122 «Комп'ютерні науки»
Загальна характеристика	Кількість годин –180 Кількість кредитів – 6 Види занять: лекції, практичні, самостійна робота. Співвідношення аудиторних годин і годин самостійної роботи - 60/120 Мова викладання – українська Форма викладання – очна
Підсумковий контроль	Екзамен
Програмне забезпечення	Trello, IntelliJ IDEA, Git, BrowserStack та ін.
Обладнання	Проектор, комп'ютерна техніка із встановленим програмним забезпеченням та доступом до мережі Інтернет.
Необхідні попередні дисципліни	Впевнений користувач ПК, знання офісних програм, ООП
Методика вивчення	Методика вивчення дисципліни полягає у набутті студентами знань теоретичного та практично-прикладного характеру через інтерактивні лекції, лабораторні роботи, самостійну роботу, групові проекти та дослідження. Студенти використовують сучасні інструменти та технології для розробки та тестування програмного забезпечення. Особлива увага приділяється практичному застосуванню отриманих знань через виконання реальних кейсів.
Мета і завдання	Метою вивчення навчальної дисципліни «Технології розробки та тестування програмного забезпечення» є засвоєння необхідних знань щодо сучасних методологій, технологій розробки якісних програмних продуктів, а також набуття практичних навичок з використання сучасних методів та засобів тестування програмного забезпечення. Завданням дисципліни є теоретична та практична підготовка майбутніх фахівців з таких питань: технології розробки програмного забезпечення та їх дефінітивний аналіз; тестування документації та вимог до програмного забезпечення; ручне тестування та його види; техніки тестдизайну; автоматизація тестування програмного забезпечення.
Місце дисципліни в освітньо-професійній програмі	
Загальні компетентності	ЗК2. Здатність застосовувати знання у практичних ситуаціях. ЗК5. Здатність вчитися й оволодівати сучасними знаннями.
Фахова компетентності	СК6 Здатність застосовувати існуючі і розробляти нові алгоритми розв'язування задач у галузі комп'ютерних наук. СК7 Здатність розробляти програмне забезпечення відповідно до

	<p>сформульованих вимог з урахуванням наявних ресурсів та обмежень.</p> <p>СК8 Здатність розробляти і реалізовувати проекти зі створення програмного забезпечення, у тому числі в непередбачуваних умовах, за нечітких вимог та необхідності застосовувати нові стратегічні підходи, використовувати програмні інструменти для організації командної роботи над проектом.</p> <p>СК10 Здатність оцінювати та забезпечувати якість ІТ-проектів, інформаційних та комп'ютерних систем різного призначення, застосовувати міжнародні стандарти оцінки якості програмного забезпечення інформаційних та комп'ютерних систем, моделі оцінки зрілості процесів розробки інформаційних та комп'ютерних систем.</p> <p>СК11 Здатність ініціювати, планувати та реалізовувати процеси розробки інформаційних та комп'ютерних систем та програмного забезпечення, включно з його розробкою, аналізом, тестуванням, системною інтеграцією, впровадженням і супроводом.</p>
<p>Програмні результати навчання</p>	<p>РН11 Створювати нові алгоритми розв'язування задач у сфері комп'ютерних наук, оцінювати їх ефективність та обмеження на їх застосування.</p> <p>РН13 Оцінювати та забезпечувати якість інформаційних та комп'ютерних систем різного призначення.</p> <p>РН14 Тестувати програмне забезпечення.</p> <p>РН15 Виявляти потреби потенційних замовників щодо автоматизації обробки інформації.</p> <p>РН17 Виявляти та усувати проблемні ситуації в процесі експлуатації програмного забезпечення, формулювати завдання для його модифікації або реінжинірингу.</p> <p>РН18 Збирати, формалізувати, систематизувати і аналізувати потреби та вимоги до інформаційної або комп'ютерної системи, що розробляється, експлуатується чи супроводжується</p>

ТЕМАТИКА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Тема 1. SDLC, SDM. Гнучкі методології розробки програмного забезпечення

Предмет вивчення і задачі дисципліни. Поняття життєвого циклу програмного продукту. Методології розробки програмного забезпечення. Моделі розробки ПЗ: каскадна, еволюційна, покрокова, формальна, спіральна та ін. Стандарти управління життєвим циклом ПЗ (ISO12207, ISO15504). Основи управління якістю розробки. Стандарти серії ISO9000.

Методологія Agile: визначення та коротка історія. Маніфест Agile Software Development. Концепція Scrum методології, основні терміни. Ролі та їх обов'язки в Scrum. Основні практики в Scrum: Daily Scrum Meeting, Sprint Review Meeting, Sprint Abnormal Termination. Артефакти в Scrum: Product Backlog, Sprint Backlog, Burndown Chart. Requirements in Scrum. Product Vision. Task, User Story, Epic, Theme.

Тема 2. Виявлення вимог до продукту. Тестування документації та вимог

Поняття вимог до програмного забезпечення та їх значення. 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).

Документування вимог до програмної продукції. Документування бізнес-вимог. Документування вимог користувачів та рішення.

Моделювання вимог до продукту. Управління вимогами до продукту.

Рівні та типи вимог. Властивості якісних вимог. Техніки тестування вимог. Приклад аналізу і тестування вимог. Типові помилки під час аналізу і тестування вимог.

Інструментальні засоби для командної роботи під час створення вимог до програмного забезпечення. Trello, як система управління проектами в режимі онлайн в гнучких методологіях розробки програмного забезпечення.

Тема 3. Поняття та види тестування

Принципи тестування. Підходи до тестування. Схема класифікації тестування. Класифікація за запуском коду на виконання. Класифікація за доступом до коду і архітектурі додатку. Класифікація за ступенем автоматизації. Класифікація за рівнем деталізації додатку (за рівнем тестування). Класифікація за рівнем важливості функцій, які тестуються (за рівнем функціонального тестування). Класифікація за принципами роботи з додатком. Класифікація за природою додатку. Класифікація за фокусом на рівні архітектури додатку. Класифікація за рівнем залучення кінцевих користувачів. Класифікація за рівнем формалізації. Класифікація за цілями і задачами. Класифікація за техніками і підходами. Класифікація за моментом виконання (хронологією).

Альтернативні та додаткові класифікації тестування. Класифікація за належністю до тестування методом білої та чорної скриньки.

Тема 4. Тестова документація

Поняття Check-list, правила його створення. TestCase і його життєвий цикл. Атрибути (поля) testcase. Інструментальні засоби управління тестуванням. Властивості якісних testcase. Набори testcase. Логіка створення ефективних перевірок. Типові помилки під час розробки check-list, testcase і наборів testcases.

Основна термінологія: помилки, дефекти, збої, відмови тощо. Звіт про дефект та його життєвий цикл. Атрибути (поля) звіту про дефекти. Інструментальні засоби керування звітами про дефекти. Властивості якісних дефектів. Логіка створення ефективних звітів про дефекти. Типові помилки під час написання звітів про дефекти.

Тема 5. Основні аспекти тест-дизайну, техніки тест-дизайну

Поняття тест-дизайну. Важливість тест-дизайну та його рецензування. Класифікація технік тест-дизайну за Лі Копландом. Техніки тест-дизайну: тестування на основі класів еквівалентності, аналіз граничних значень, таблиця прийняття рішень, метод попарного тестування, тестування на основі причинно-наслідкових зв'язків, тестування на основі сценаріїв, тестування на основі діаграм станів, тестування на основі діаграм потоків даних, тестування на основі вимог.

Тема 6. Тестування вебдодатків

Архітектура та особливості вебдодатків. Основні види тестування вебдодатків: функціональне тестування; тестування юзабіліті (перевірка зручності користування); тестування сумісності (конфігураційне тестування); тест на продуктивність; перевірка безпеки.

Функціональне тестування вебдодатку: перевірка посилань (внутрішніх, вихідних, ізольованих і т. д.); перевірка форм; тест файлів cookie; валідація HTML/CSS; перевірка бази даних.

Перевірка зручності використання: навігаційне тестування; перевірка контенту; комфорт користування; тестування доступності.

Тестування сумісності: кросплатформений тестінг; перевірка на кросбраузерність; тестування на мобільних пристроях.

Тестування продуктивності: тестування навантаження; стрес-тестування; перевірка на швидкість з'єднання.

Тема 7. Тестування безпеки вебдодатків

Проект по забезпеченню безпеки вебдодатків. 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).

Принципи тестування, методологія тестування за OWASP. Активне та пасивне тестування. Розвідка і уразливості вебдодатків: відкриття вебсторінки/структури програми; збір інформації в вебдодатках; сканування вразливостей вебдодатків.

Тема 8. Автоматизація тестування вебдодатків

Переваги і недоліки автоматизації. Галузь застосування автоматизації. Особливості тест-кейсів в автоматизації. Технології автоматизації тестування.

Використання продуктів Selenium IDE, Selenium WebDriver. Прийоми розробки тестів: пошук елементів (локаторів), дії з елементами, використання фрагментів JavaScript-коду.

Тестування продуктивності. Автоматизація тестів для оцінки продуктивності вебдодатка під різними навантаженнями. Інструменти: JMeter, Gatling, LoadRunner.

Тестування сумісності. Автоматизація тестів для перевірки роботи вебдодатка в різних браузерах та на різних платформах. Інструменти: BrowserStack, Sauce Labs.

API тестування. Автоматизація тестів для перевірки коректності роботи API. Інструменти: Postman, REST Assured, SoapUI.

Інфраструктура запуску тестів: запуск з консолі, запуск в різних браузерах, зберігання тестів на прикладі сервісу GitHub. Керування потоками виконання тестів.

Тема 9. Тестування мобільних додатків

Виникнення мобільних додатків. Різновиди мобільних додатків. Особливості тестування мобільних додатків. Типи тестування мобільних додатків: тестування зручності використання, тестування сумісності, тестування інтерфейсу, тестування сервісів, тестування ресурсів низького рівня, тестування продуктивності, тестування експлуатації, тестування встановлення, тестування безпеки.

Стратегія тестування мобільних додатків. Техніки тестування безпеки мобільних додатків. Автентифікація, авторизація, безпека даних, уразливості для злому, управління сесіями.

Інструменти тестування мобільних додатків: ImmuniWeb MobileSuite; Zed Attack Proxy; Kiwan; QARK; Micro Focus; Android Debug Bridge; CodifiedSecurity; Drozer; WhiteHat Security; Synopsys; Veracode; Mobile Security Framework.

Тема 10. Автоматизація тестування мобільних додатків

Знайомство з інструментами розробки та тестування мобільних додатків. Інструменти для автоматизації та їх дефінітивний аналіз. 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.

Локатори: XPath, Id, Class, Name, Css (mobile web only).

Підготовка тестового додатку, підготовка сценарію тест-кейсу, застосування вбудованих методів Click, Find_element, Send_keys та ін. Очікування Implicit vs Explicit, валідація та Asserts.

Паттерни та антипаттерни. Тестування жестів Swipe, Reporting, Listeners. Особливості автоматизації тестування мобільних вебдодатків.

Тема 11. Тестування комп'ютерних ігор

Геймдизайн та його тестування. Основні аспекти тестування геймдизайну: ігровий баланс, інтуїтивність геймплея, реакція гравців.

Ігрові механізми, тестування ігрових механізмів: контроль і управління, бойові та інші системи, система винагород.

Тестування продуктивності та тестування FPS.

Тестування сумісності: тестування кросплатформності, конфігураційне тестування. Тестування Геймплея. Тестування рівнів: перевірка складності, тестування логіки рівнів.

Тестування сценаріїв: тестування сюжетних ліній, перевірка інтерактивності. Тестування користувацького інтерфейсу (UI).

Особливості тестування ігрових платформ. Вимоги до сертифікації: дотримання вимог сертифікації кожної платформи (правила PlayStation Certification).

Тестування ігор на мобільних пристроях.

Інструменти для тестування комп'ютерних ігор: Unity Test Tools, Unreal Engine Automation Testing, Appium, Selenium, Fraps.

Тема 12. ШІ в управлінні проектами та тестуванні програмного забезпечення

Управління проектами за допомогою ШІ. Оптимізація розподілу ресурсів. Аналіз ризиків. Підтримка прийняття рішень. Автоматизація управлінських процесів на основі ШІ.

Автоматизація тестування на основі ШІ: автоматизація виконання тестових сценаріїв, інструменти ШІ для автоматизації тестування. Генерація тестових випадків та даних на основі роботи ШІ. Аналіз результатів тестування на основі ШІ. Прогнозування дефектів. Оптимізація тестового покриття: аналіз програмного коду для визначення областей, які потребують тестування; інструменти ШІ для оптимізації тестового покриття.

Інтеграція ШІ в процеси управління та тестування. Інтеграція в процес безперервної інтеграції/розгортання (CI/CD). Моніторинг та звітність.

Виклики та перспективи використання ШІ. Необхідність великих обсягів даних. Етичні питання та приватність. Підтримка та навчання моделей.

Тема 13. Метрики та методики оцінки. Оцінка трудовитрат, планування та звітність в тестуванні

Метрики якості. Метрики за тестовими випадками. Метрики за багами/дефектами. Метрики за задачами. Юзабіліті-метрики. Метрики Kanban. Метрики SCRUM. Валідація і верифікація.

Планування і звітність. Тест-план і звіт про результати тестування. Оцінка трудовитрат.

Перелік навчальних робіт студентів та оцінки їх у балах з дисципліни «Технології розробки та тестування програмного забезпечення»

Види робіт	К-сть балів
Лабораторна робота №1 «Розробка прототипу сайту за Scrum методологією»	5
Лабораторна робота №2 «Тестування вимог до програмного забезпечення»	5
Лабораторна робота №3 «Планування тестування, створення check-list, testcase і набори testcases»	5
Лабораторна робота №4 «Техніки тест-дизайну для тестування ПЗ»	10
Лабораторна робота №5 «Дослідницьке тестування вебдодатків»	10
Лабораторна робота №6 «Автоматизація тестування вебдодатків»	10
Лабораторна робота №7 «Тестування мобільних додатків»	10
Лабораторна робота №8 «Тестування комп'ютерних ігор»	5
Модульний контроль	10
Разом: Аудиторна робота	70
Самостійна робота (СР)	30
Всього:	100

КОНТРОЛЬ ТА КРИТЕРІЇ ОЦІНЮВАННЯ ЗНАНЬ СТУДЕНТІВ

При вивченні дисципліни використовуються наступні форми контролю знань студентів: поточний; модульний; підсумковий.

Поточний контроль передбачає перевірку теоретичних питань, самостійної роботи, практичних робіт та усне опитування по кожній практичній роботі. По даному виду контролю оцінювання знань здійснюється у відповідності до бального розподілу наведеного в попередній таблиці.

Модульний контроль передбачає виконання модульної контрольної роботи. Всі завдання оцінюються в 10 балів. Проводиться у вигляді тестування (20 питань – 10 балів).

Формою підсумкового контролю є екзамен. Екзаменаційна оцінка (100 балів) є результатом виконання тестування (50 питань = 50 балів) та практичного завдання (50 балів).

Результуюча оцінка з дисципліни визначається як середня від балів набраних протягом семестру та отриманих на іспиті.

СПИСОК РЕКОМЕНДОВАНИХ ДЖЕРЕЛ

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.