

**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**

**АВТОМАТИЗОВАНЕ ТЕСТУВАННЯ ПРОГРАМНИХ
ПРОДУКТІВ/
TEST AUTOMATION SOFTWARE PRODUCTS**

**СИЛАБУС/
SILABUS**

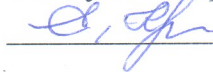
ЗАТВЕРДЖЕНО

засіданням кафедри

(протокол №. 1)



від «04» серпня 2024 р.)

завідувач кафедри



Олена КРИВОРУЧКО

Київ 2024

Назва освітньої компоненти	АВТОМАТИЗОВАНЕ ТЕСТУВАННЯ ПРОГРАМНИХ ПРОДУКТІВ/ TEST AUTOMATION SOFTWARE PRODUCTS
Спеціальність	121 «Інженерія програмного забезпечення»
Освітній ступінь	Другий (магістерський)
Освітньо-професійна програма	УПРАВЛІННЯ ПРОЄКТАМИ ПРОГРАМНИХ ПРОДУКТІВ
	Лектор: Жирова Тетяна -доцент -кандидат педагогічних наук -доцент Резюме викладача: http://knute.edu.ua/blog/read/?pid=39727&uk Науковий профіль: https://knute.edu.ua/blog/read/?pid=46722 е-пошта: zhyrova@knute.edu.ua
	Асистент лектора: Котенко Наталія -доцент, гарант освітньої програми «Інженерія програмного забезпечення» -кандидат педагогічних наук -доцент Резюме викладача: https://knute.edu.ua/blog/read/?pid=39712&uk Науковий профіль: https://knute.edu.ua/blog/read/?pid=46719 е-пошта: kotenkono@knute.edu.ua
Консультації	https://knute.edu.ua/blog/read/?pid=47103&uk
Програма освітньої компоненти	https://knute.edu.ua/blog/read/?pid=48215
ЗМІСТ ОСВІТНЬОЇ КОМПОНЕНТИ	
Тема 1. Вступ до автоматизованого тестування програмних продуктів	Основні поняття та принципи автоматизації тестування. Тестові сценарії (Test Scripts). Інструменти автоматизації (Automation Tools), особливості та області застосування. Принципи автоматизації тестування програмних продуктів: модульність (Modularity), повторне використання (Reusability), незалежність даних (Data Independence), прозорість та чіткість (Transparency and Clarity). Переваги та недоліки автоматизації тестування. Виклики автоматизації тестування програмних продуктів.
Тема 2. Тестові	Поняття Check-list, правила його створення. TestCase і його

артефакти	<p>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 test cases. Особливості test cases в автоматизації.</p> <p>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.</p>
Тема 3. Технології автоматизації тестування програмних продуктів	<p>Піраміда тестування. Рівні тестування: Unit tests; Integration tests; GUI tests; Manual tests. Складові частини автоматизації тестування. Обмеження автоматизації тестування. Підходи до автоматизації тестування: тестування на рівні коду і GUI-тестування.</p> <p>Аналітичний огляд еволюції високорівневих технологій тестування. Підходи автоматизації тестування: частинні рішення, тестування під керуванням даними (DDT), тестування під керуванням ключовими словами (KDT), використання фреймворків, запис та відтворення, тестування під керуванням поведінкою.</p>
Тема 4. Інструментальні засоби роботи з проектом	<p>Робота з системами контролю версіями Git. Коротка історія Git. Установка та налаштування Git. Основи Git: створення Git-репозиторію, запис змін в репозиторії, перегляд історії комітів, робота з віддаленим репозиторієм. Розгалуження в Git. Git на сервері. Розподілений Git. Налаштування і конфігурація облікового запису GitHub, супровід проекту. Інструменти та налаштування Git.</p> <p>Вступ в Enterprise-розробку. Основи Maven. Файл POM. Життєвий цикл збірки проектів. Профілі збірки. Репозиторії. Плагіни. Налаштування Maven Environment. Засоби обробки проектів з використанням Maven. Збірка і тестування проекту. Системи підтримки безперервної інтеграції TeamCity, Hudson (Jenkins), TFS, Atlassian Bamboo та їх порівняння. Jenkins – призначення та налаштування Tomcat. Налаштування Jenkins і Maven, конфігурація Maven. Налаштування збірки. Модульне тестування, автоматизація тестування, звітність. GitLab CI/CD: система безперервної інтеграції в репозиторії. Azure DevOps: безперервна інтеграція для Microsoft-стеку.</p>
Тема 5. Планування та стратегія автоматизації тестування програмних	<p>Розробка стратегії автоматизації. Визначення цілей автоматизації: чітке формулювання цілей, вимірювані цілі, аналіз поточних процесів тестування, оцінка ручних процесів, виявлення слабких місць.</p> <p>Вибір інструментів для автоматизації: огляд ринку, оцінка</p>

<p>продуктів</p>	<p>інструментів. Розробка архітектури автоматизації: структура тестового середовища, інтеграція з існуючими системами. Планування ресурсів: людські ресурси, технічні ресурси. Створення документації: стратегія автоматизації, план тестування. Вибір тестів для автоматизації. Визначення критеріїв для вибору тестів: частота виконання, критичність функціональності, час та ресурси. Класифікація тестових сценаріїв: функціональні тести, нефункціональні тести. Пріоритезація тестів. Аналіз повернення інвестицій (ROI): вартість автоматизації, економія часу, якість продукту. Виконання пілотних тестів та аналіз результатів, розширення покриття.</p>
<p>Тема 6. Основи написання автоматизованих тестів. Модульне тестування та розробка через тестування</p>	<p>Основи автономного тестування. Вибір мови програмування. Структура автоматизованих тестів. Розробка через тестування. Каркаси автономного тестування, каркаси сімейства xUnit. Фреймворк NUnit/JUnit. Установка та налаштування NUnit/JUnit, використання їхніх атрибутів. Властивості NUnit/JUnit. Клас Assert. Стилестичне оформлення тестового коду. Рефакторинг – параметризовані тести. Перевірка очікуваних виключень. Визначення категорій тестів. Використання заглушок для розриву залежностей. Тестування взаємодії за допомогою підставних об'єктів. Ієрархія і організація тестів.</p>
<p>Тема 7. Автоматизація тестування веб-додатків</p>	<p>Використання Selenium для автоматизації тестування веб-додатків. Огляд Selenium: коротка історія, компоненти (Selenium WebDriver, Selenium Grid, Selenium IDE). Переваги використання Selenium: Відкритий вихідний код, підтримка багатьох мов програмування (Java, Python, C#, Ruby, JavaScript), сумісність з різними браузерами та операційними системами. Налаштування середовища. Встановлення Selenium WebDriver: Завантаження та налаштування необхідних бібліотек та драйверів для різних браузерів (ChromeDriver, GeckoDriver для Firefox). Налаштування проекту: Створення проекту в IDE (IntelliJ IDEA, Eclipse, Visual Studio Code), підключення необхідних бібліотек через Maven або Gradle. Основи написання тестів на Selenium. Локатори в Selenium: використання різних стратегій для знаходження елементів на веб-сторінці (By.id, By.name, By.className, By.xpath, By.cssSelector). Автоматизація основних сценаріїв. Навігація та взаємодія з елементами, чекачі та затримки. Розширені можливості Selenium. Обробка спливаючих вікон</p>

	<p>та діалогових вікон. Робота з фреймами та вкладками. Знімки екрану.</p> <p>Інтеграція з тестовими фреймворками: JUnit/TestNG та Page Object Model (POM).</p> <p>Тестування кроссбраузерної сумісності. Інструменти для кроссбраузерного тестування: Selenium Grid, Хмарні сервіси.</p> <p>Налаштування середовища для кроссбраузерного тестування.</p> <p>Створення та виконання кроссбраузерних тестів. Аналіз результатів та усунення несправностей.</p>
<p>Тема 8.</p> <p>Автоматизація тестування мобільних додатків</p>	<p>Інструменти для автоматизації тестування мобільних додатків. Appium, його архітектура, переваги та недоліки. Espresso, його архітектура, переваги та недоліки. Особливості тестування на різних платформах: Android. Емулятори та реальні пристрої. Розробка кросплатформених тестів. Контейнеризація та віртуалізація. Паралельне тестування. Безперервна інтеграція.</p>
<p>Тема 9. Тестування API та мікросервісів</p>	<p>Види тестів в тестуванні API: оглядове тестування, юзабіліті тестування, тестування безпеки, тестування безпеки, автоматизація тестування, тестування документації.</p> <p>Дефекти, які виявляються під час API тестування: збій обробки помилкових умов; невикористані прапори; відсутній або дублюється функціонал; питання надійності: труднощі при підключенні і отриманні відповіді від API; проблеми з безпекою; питання багатопоточності; проблеми з продуктивністю: час відгуку API дуже високо; помилкові дефекти; некоректна обробка валідних значень; дані відповіді некоректно структуровані (JSON або XML).</p> <p>Введення в архітектуру мікросервісів. Визначення та характеристики: розуміння того, що таке мікросервіси, чим вони відрізняються від монолітних архітектур, а також їхні ключові характеристики, такі як децентралізоване керування даними, незалежне розгортання та масштабованість. Стратегії тестування мікросервісів.</p> <p>Інструменти для роботи з API: Postman, jMeter, Fiddler, SoapUI, Runscope, Advanced REST Client.</p>
<p>Тема 10. Тестування продуктивності та навантаження</p>	<p>Тестування продуктивності. Види тестування продуктивності: Load Testing (тестування навантаження), тестування стабільності, тестування відмовостійкості, тестування відновлення, стресове тестування, тестування масштабованості, тестування конфігурації.</p> <p>Етапи тестування продуктивності: аналіз системи і вимог, підготовка стратегії, налаштування генератора навантаження, моніторинг серверів та генератора навантаження, підготовка тестових даних, розробка скрипів навантаження, тестування, аналіз результатів і підготовка звітів.</p>

	Інструменти тестування продуктивності: JMeter, LoadRunner, Gatling, Taurus, Java Microbenchmark Harness, BenchmarkDotNet, Google Lighthouse
Тема 11. Розширені техніки автоматизації тестування	<p>Вступ до ШІ та машинного навчання в тестуванні. Основні концепції штучного інтелекту (ШІ) та машинного навчання (МН). Види задач, які можуть вирішувати ШІ та МН у контексті автоматизації тестування.</p> <p>Застосування ШІ та МН для генерації тестових випадків. Використання алгоритмів машинного навчання для аналізу вихідного коду і автоматичного створення тестових випадків. Приклади інструментів та фреймворків, що використовують ШІ для генерації тестів.</p> <p>Автоматичне визначення та виправлення дефектів. Використання ШІ для аналізу результатів тестування та визначення потенційних дефектів. Техніки самовиправлення коду за допомогою машинного навчання.</p> <p>Прогнозування та пріоритизація тестових випадків. Використання моделей машинного навчання для прогнозування ризиків та визначення пріоритету тестування на основі історичних даних. Інтеграція прогнозування ризиків у процес тестування.</p> <p>Аналіз тестового покриття та оптимізація тестових наборів. Використання ШІ для аналізу тестового покриття і виявлення пробілів у тестуванні. Оптимізація тестових наборів для забезпечення максимальної ефективності тестування.</p> <p>Розпізнавання та автоматизація тестування інтерфейсів користувача (UI). Використання технологій комп'ютерного зору для автоматизації тестування інтерфейсів користувача. Інструменти та методи для автоматичного розпізнавання та взаємодії з елементами UI.</p> <p>Автоматичне оновлення та підтримка тестових сценаріїв. Використання ШІ для автоматичного оновлення тестових сценаріїв при змінах у програмному продукті. Підтримка актуальності тестових сценаріїв та мінімізація ручної роботи.</p>
Тема 12. Гнучке тестування в Agile	<p>Переваги гнучкого тестування. Принципи гнучкого тестування. Роль тестувальника в Scrum. Гнучка тестова діяльність в Scrum. Відмова від традиційних способів тестування на користь Agile. Маніфест Agile-тестування. Відмінності традиційного тестування від тестування в Agile. Піраміда тестування. Shift-left testing: розподіл тестових активностей в команді; практики, «змінюють» ролі в команді: тестування вимог (Specification by Example), тестування в розробці (TDD, ATDD, BDD), планування / ретроспектива в тестуванні, парні сесії (парне програмування). Тестування безпеки, продуктивності та інших атрибутів якості. Стратегія</p>

	автоматизації тестування: фреймворки автоматизації тестування, паттерни і антипаттерн в автоматизації. Тестова документація.
СПИСОК ОСНОВНИХ РЕКОМЕНДОВАНИХ ДЖЕРЕЛ	
<ol style="list-style-type: none"> 1. Gregory J. Agile Testing Condensed: A Brief Introduction Paperback /Janet Gregory,Lisa Crispin – September 3, 2019. – 113 p. 2. Jorgensen P. C., DeVries B. Software testing: a craftsman's approach, fifth edition. Auerbach Publishers, Incorporated, 2021. 528 p. 3. Khorikov V. Unit Testing Principles, Practices, and Patterns: Effective testing styles, patterns, and reliable automation for unit testing, mocking, and integration testing with examples in C# 1st Edition./ Vladimir Khorikov – Apress ,January 14, 2020. – 304 p. 4. Westerveld D. API Testing and Development with Postman: A practical guide to creating, testing, and managing APIs for automated software testing. Packt Publishing, 2021. 340 c. 	
РЕЗУЛЬТАТИ ВИВЧЕННЯ ОСВІТНЬОЇ КОМПОНЕНТИ	
Дисципліна забезпечує оволодіння здобувачами вищої освіти загальними та фаховими компетентностями і досягнення ними програмних результатів навчання:	
ЗК01.	Здатність до абстрактного мислення, аналізу та синтезу.
ЗК02.	Здатність спілкуватися іноземною мовою як усно, так і письмово.
ЗК03.	Здатність проводити дослідження на відповідному рівні.
ЗК05.	Здатність генерувати нові ідеї (креативність).
СК01.	Здатність аналізувати предметні області, формувати, класифікувати вимоги до програмного забезпечення.
СК07.	Здатність критично осмислювати проблеми у галузі інформаційних технологій та на межі галузей знань, інтегрувати відповідні знання та розв'язувати складні задачі у широких або мультидисциплінарних контекстах.
СК08.	Здатність розробляти і координувати процеси, етапи та ітерації життєвого циклу програмного забезпечення на основі застосування сучасних моделей, методів та технологій розроблення програмного забезпечення.
СК09.	Здатність забезпечувати якість програмного забезпечення.
РН01	Знати і застосовувати сучасні професійні стандарти і інші нормативно-правові документи з інженерії програмного забезпечення
РН02	Оцінювати і вибирати ефективні методи і моделі розроблення, впровадження, супроводу програмного забезпечення та управління відповідними процесами на всіх етапах життєвого циклу.
РН11	Забезпечувати якість на всіх стадіях життєвого циклу програмного забезпечення, у тому числі з використанням релевантних моделей та методів оцінювання, а також засобів автоматизованого тестування і верифікації програмного забезпечення.
РН16	Планувати, організувати та здійснювати тестування, верифікацію та

валідацію програмного забезпечення.

ОЦІНЮВАННЯ ЗНАНЬ ЗДОБУВАЧІВ ОСВІТИ

Сума балів, накопичених здобувачем вищої освіти за виконання всіх видів поточних навчальних завдань (робіт) на лабораторних/практичних заняттях, свідчить про ступінь оволодіння ним програмою освітньої компоненти на конкретному етапі її вивчення. Протягом семестру здобувачі освіти можуть набрати від 0 до 100 балів, що переводяться у національну шкалу оцінювання і відповідно у шкалу ЄКТС. Кількість балів відповідає певному рівню засвоєння дисципліни

Довідник з розподілу оцінок ДТЕУ (Шкала ЄКТС):

Бали ДТЕУ	Відсоток балів відносно загальної кількості одержаних прохідних балів	Кумулятивний відсоток отриманих прохідних балів
90-100	20	20
82-89	10	30
75-81	20	50
69-74	10	60
60-68	40	100

Роподіл балів за видами робіт:

Вид роботи	Бали	Вид роботи	Бали
Лабораторна робота 1	10	Самостійна робота 1	2
Лабораторна робота 2	10	Самостійна робота 2	2
Лабораторна робота 3	10	Самостійна робота 3	2
Лабораторна робота 4	10	Самостійна робота 4	2
Лабораторна робота 5	10	Самостійна робота 5	2
		Самостійна робота 6	2
		Самостійна робота 7	2
		Самостійна робота 8	2
		Самостійна робота 9	2
		Самостійна робота 10	2
		Самостійна робота 11	2
		Самостійна робота 12	2
Додаткові бали	16	Наукова робота	10

Вимоги до критеріїв оцінювання самостійної роботи студента (оцінювання одного завдання у відсотковому еквіваленті)

40%	Детальний розгляд сутності та вмісту основних джерел. Подання фактів, ідей і результатів досліджень у логічній послідовності. Правильно проаналізовано поточний стан дослідження проблеми та зроблено огляд перспектив подальшого розвитку даного питання.
40%	Обґрунтованість аргументів, підтвердження особистого ставлення, пропозиції стосовно вирішення завдання, встановлення напрямків аналізу.
20%	Оформлення звіту у відповідності вимог

Критерії оцінювання самостійної роботи студента (оцінювання одного завдання у відсотковому еквіваленті)	
100%	В повному обсязі володіє навчальним матеріалом, вільно самостійно та аргументовано його викладає під час усних виступів та письмових відповідей, глибоко та всебічно розкриває зміст теоретичних питань та лабораторних завдань, використовуючи при цьому обов'язкову та додаткову літературу. Правильно вирішив усі тестові завдання.
80%	Достатньо повно володіє навчальним матеріалом, обґрунтовано його викладає під час усних виступів та письмових відповідей, в основному розкриває зміст теоретичних питань та лабораторних завдань, використовуючи при цьому обов'язкову літературу. Але при викладанні деяких питань не вистачає достатньої глибини та аргументації, допускаються при цьому окремі несуттєві неточності та незначні помилки. Правильно вирішив більшість тестових завдань
60%	В цілому володіє навчальним матеріалом викладає його основний зміст під час усних виступів та письмових відповідей, але без глибокого всебічного аналізу, обґрунтування та аргументації, без використання необхідної літератури допускаючи при цьому окремі суттєві неточності та помилки. Правильно вирішив половину тестових завдань.
40%	Не в повному обсязі володіє навчальним матеріалом. Фрагментарно, поверхово (без аргументації та обґрунтування) викладає його під час усних виступів та письмових відповідей, недостатньо розкриває зміст теоретичних питань та лабораторних завдань, допускаючи при цьому суттєві неточності, правильно вирішив меншість тестових завдань.
20%	Частково володіє навчальним матеріалом не в змозі викласти зміст більшості питань теми під час усних виступів та письмових відповідей, допускаючи при цьому суттєві помилки. Правильно вирішив окремі тестові завдання.
0%	Не володіє навчальним матеріалом та не в змозі його викласти, не розуміє змісту теоретичних питань та практичних завдань. Не вирішив жодного тестового завдання.
ОСНОВНІ ПОЛОЖЕННЯ, ЩО РЕГЛАМЕНТУЮТЬ ОСВІТНІЙ ПРОЦЕС	
діючі положення	https://knute.edu.ua/blog/read/?pid=44402
нормативно-правова база організації освітнього процесу	https://knute.edu.ua/blog/read/?pid=7330&uk
студенту	https://knute.edu.ua/#forstudent
НЕФОРМАЛЬНА ОСВІТА	
Рекомендовані сертифікаційні програми, курси, посібники користувача	

Основи тестування програмного забезпечення	https://courses.prometheus.org.ua/courses/course-v1:LITS+115+2017_T4/about
Курс Automated Testing Self-Paced	https://training.epam.ua/ua/training/3483
ПОЛІТИКА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ:	
Відвідування лекційних та лабораторних занять: відвідування	Відвідування лекційних та лабораторних занять є обов'язковим. Допускаються пропуски занять з таких поважних причин, як хвороба (викладачу надається копія довідки від медичного закладу), участь в олімпіаді, творчому конкурсі тощо за попередньою домовленістю та згодою викладача за умови дозволу деканату (надаються документи чи інші матеріали, які підтверджують заявлену участь у діяльності студента).
Відпрацювання пропущених занять:	Відпрацювання пропущених занять є обов'язковим незалежно від причини пропущеного заняття. Лекційне заняття має бути відпрацьоване до наступної лекції на консультації викладача з використанням ПЗ 365 Office Teams. Відпрацювання лекційного матеріалу передбачає вивчення пропущеного теоретичного матеріалу та складання тесту за цим матеріалом. Лабораторне заняття відпрацьовується під час консультації викладача (розклад консультацій на сайті).
Правила поведінки під час занять	обов'язковим є дотримання техніки безпеки в комп'ютерних лабораторіях. Студенти повинні приймати активну участь в обговоренні навчального матеріалу ознайомившись з ним напередодні (навчальний матеріал надається викладачем). Мобільні пристрої дозволяється використовувати лише під час он-лайн тестування та підготовки практичних завдань в процесі заняття. Задля зручності, дозволяється використання ноутбуків та інших електронних пристроїв під час навчання в комп'ютерних аудиторіях (за взаємною згодою всіх учасників освітнього процесу)
Політика академічної доброчесності ДТЕУ	https://knute.edu.ua/blog/read/?pid=38987&uk