



**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ТОРГОВЕЛЬНО-
ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ**
Факультет інформаційних технологій
Кафедра інженерії програмного забезпечення
та кібербезпеки

СИЛАБУС (SYLLABUS)

**Дисципліна «Технології розробки та тестування програмного
забезпечення» /**
Technology of development and testing of software

ІНФОРМАЦІЯ ПРО ВИКЛАДАЧА

Викладач	Жирова Тетяна Олександрівна
Науковий ступінь	Кандидат педагогічних наук
Вчене звання	-
Посада	Старший викладач кафедри інженерії програмного забезпечення та кібербезпеки
Адреса кафедри	м.Київ, вул. Кіото 19, каб. Б-524, Б-531
E-mail	progen@ukr.net
Консультації	Відповідно до графіку індивідуальних консультацій на сайті кафедри

ПОЛІТИКА АКАДЕМІЧНОЇ ДОБРОЧЕСНОСТІ

<https://knute.edu.ua/file/NjY4NQ==/bf27ad9293fa2bb6f9b2c3031d4b6e4a.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;
- відраховання 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.

ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Назва дисципліни / тип дисципліни	Технології розробки та тестування програмного забезпечення / вибіркова
Навчальний рік	2020-2021
Факультет	Факультет інформаційних технологій
Курс	4
Семестр	7-8
Освітній ступінь	Бакалавр
Галузь знань	12 «Інформаційні технології»
Спеціальність	122 «Комп'ютерні науки»
Загальна характеристика	Кількість годин –180 Кількість кредитів – 6 Види занять: лекції, лабораторні, самостійна робота. Співвідношення аудиторних годин і годин самостійної роботи - 68/112 Мова викладання – українська Форма викладання – очна
Підсумковий контроль	Екзамен
Програмне забезпечення	Open Web Application Security Project
Обладнання	Проектор, комп'ютерна техніка із встановленим програмним забезпеченням та доступом до мережі Інтернет.
Необхідні попередні дисципліни	Основи програмування мовами C#/Java, основи HTML та CSS.
Методика вивчення	Методика вивчення дисципліни полягає у набутті студентами знань теоретичного і практично-прикладного характеру під час лекцій, практичних занять, самостійної роботи та вивчення першоджерел і навчально-методичної літератури.
Мета і завдання	Мета дисципліни «Технології розробки та тестування програмного забезпечення»: 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. Завдання дисципліни: теоретична та практична підготовка майбутніх фахівців з таких питань: <ul style="list-style-type: none"> • технології розробки програмного забезпечення та їх дефінітивний аналіз; • Agile, як методологія розробки програмного забезпечення;

	<ul style="list-style-type: none"> • класифікація тестування; • тестування документації та вимог до програмного забезпечення; • ручне тестування та його види; • особливості тестування мобільних, web-додатків та комп'ютерних ігор; • тестування безпеки; • автоматизація тестування.
Місце дисципліни в освітньо-професійній програмі	
Фахові компетентності (результати навчання)	<p>СК 8 Здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління.</p> <p>СК 12 Здатність забезпечити організацію обчислювальних процесів в інформаційних системах різного призначення з урахуванням архітектури, конфігурування, показників результативності функціонування операційних систем і системного програмного забезпечення.</p> <p>СК 13 Здатність до розробки мережевого програмного забезпечення, що функціонує на основі різних топологій структурованих кабельних систем, використовує комп'ютерні системи і мережі передачі даних та аналізує якість роботи комп'ютерних мереж.</p> <p>СК 14 Здатність застосовувати методи та засоби забезпечення інформаційної безпеки, розробляти й експлуатувати спеціальне програмне забезпечення захисту інформаційних ресурсів об'єктів критичної інформаційної інфраструктури.</p>
Програмні результати навчання	<p>ПР 9 Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв'язання задач в галузі комп'ютерних наук.</p> <p>ПР 13 Володіти мовами системного програмування та методами розробки програм, що взаємодіють з компонентами комп'ютерних систем, знати мережні технології, архітектури комп'ютерних мереж, мати практичні навички технології адміністрування комп'ютерних мереж та їх програмного забезпечення.</p> <p>ПР 14 Володіти мовами системного програмування та методами розробки програм, що взаємодіють з компонентами комп'ютерних систем, знати мережні технології, архітектури комп'ютерних мереж, мати практичні навички технології адміністрування комп'ютерних мереж та їх програмного забезпечення.</p>

ТЕМАТИКА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Тема 1. Вступ до навчальної дисципліни. Життєвий цикл програмного забезпечення. Життєвий цикл тестування програмного забезпечення.

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

Тема 2. Гнучка методологія розробки програмного забезпечення.

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 3. Тестування документації та вимог.

Поняття вимог до програмного забезпечення та їх значення. Джерела та шляхи визначення вимог. 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 4. Класифікація 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 5. Check-list, TestCase, набори TestCases.

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.

Тема 6. Тестове покриття. Звіт про дефекти.

Оцінка трудовитрат, планування та звітність.

Тестове покриття: покриття вимог, покриття коду, тестове покриття на базі аналізу потоку керування. 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. Планування і звітність. Тест-план і звіт про результати тестування. Оцінка трудовитрат.

Тема 7. Техніки тестування.

Техніки тест дизайну. Верифікація та валідація. Позитивні та негативні тест-кейси. Класи еквівалентності та граничні умови. Доменне тестування та комбінації параметрів. Попарне тестування та пошук комбінацій. Вичерпне тестування. Матриця відповідності вимог. Діаграма переходів станів. Дослідницьке тестування. Пошук причин виникнення дефектів.

Тема 8. Тестування мобільних та web-додатків. Тестування комп'ютерних ігор.

Архітектура та особливості web-додатків. Валідація сайту, HTML структури. Сервіси перевірки CSS від W3C. Протокол HTTP: взаємодія web-клієнта та web-сервера. Технології тестування клієнтської частини: HTML, CSS, JavaScript. Технології тестування серверної частини: бізнес-логіки та бази даних. Функціональне тестування web-додатків: перевірка зовнішніх та внутрішніх посилань, тестування інтерфейсу, тестування бізнес-логіки, тестування навігації, кросбраузерне тестування. Нефункціональне тестування: тестування безпеки, тестування навантаження, тестування зручності використання. Архітектура та особливості мобільних додатків. Типи мобільних додатків: нативні, web-додатків, гібридні. Етапи тестування мобільних додатків: тестування документації, функціональне тестування (тестування полів, функціональності, переривань, відгуків користувача, оновлення, ресурсів), юзабіліті-тестування, тестування користувацького інтерфейсу, тестування продуктивності, тестування безпеки. Особливості тестування комп'ютерних ігор. Геймдизайн та його тестування. Ігрові механізми. Технічне тестування та тестування геймплея. Особливості тестування ігрових платформ. Інструментальні засоби тестування мобільних й web-додатків та комп'ютерних ігор.

Тема 9. Тестування безпеки, види вразливостей

Поняття тестування безпеки та його призначення. Принципи безпеки програмного забезпечення: конфіденційність, цілісність, доступність, аутентифікація, авторизація, безвідмовність. Види уразливостей: Cross-Site Scripting, Request Forgery, Code injections, Server-Side Includes (SSI) Injection, Authorization Bypass. Стандарт оцінювання відповідності безпеки додатків. Методика тестування безпеки ПЗ, Open Web Application Security Project. Програмне забезпечення для тестування безпеки ПЗ і виявлення вразливостей системного та прикладного програмного забезпечення. Тестування безпеки мобільних та web-додатків.

Тема 10. Особливості автоматизованого тестування. Підходи до автоматизації тестування, побудова фреймворків.

Переваги і недоліки автоматизації. Галузь застосування автоматизації. Особливості тест-кейсів в автоматизації. Технології автоматизації тестування. Зовнішні джерела даних. XML і його застосування. JSON і його застосування. Приклади реалізації тестування під керуванням даними. Тестування під керуванням ключових слів: загальні принципи та приклад реалізації. Тестування під керуванням бізнес домена: загальні принципи, специфікація BDD, приклади використання. Функціональна декомпозиція і патерни проектування. Принципи створення фреймворків.

Тема 11. Автоматизація тестування веб-орієнтованих додатків.

Принципи побудови і технології веб-орієнтованих додатків. Джерела багів у веб-орієнтованих додатках. Використання продуктів Selenium IDE. Прийоми розробки тестів: пошук елементів (локаторів), дії з елементами, використання фрагментів JavaScript-коду. Інфраструктура запуску тестів: запуск з консолі, запуск в різних браузерах, зберігання тестів на прикладі сервісу GitHub. Керування потоками виконання тестів.

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

Види робіт	К-сть балів
Лабораторне заняття №1. Тема: «Розробка прототипу сайту за Scrum методологією».	5
Лабораторне заняття №2. Тема: «Тестування вимог до програмного забезпечення».	5
Лабораторне заняття №3. Тема: «Планування та організація процесу тестування відповідним методом».	5
Лабораторне заняття №4. Тема: «Планування тестування, створення check-list, testcase і набори testcases».	5
Лабораторне заняття №5. Тема: «Тестування ПЗ. Звіт про дефекти».	5
Лабораторне заняття №6. Тема: «Тестування ПЗ за технікою «Класи еквівалентності та граничні умови».	5
Лабораторне заняття №7. Тема: «Тестування ПЗ за технікою «Класи еквівалентності та граничні умови».	5
Лабораторне заняття №8. Тема: «Тестування ПЗ за технікою «Класи еквівалентності та граничні умови».	5
Лабораторне заняття №9. Тема: «Вивчення основ роботи інструментальних засобів для створення автотестів»	5
Лабораторне заняття №10. Тема: «Автоматизація тестування веб-орієнтованих додатків».	5
Модульний контроль	20

Виконання індивідуального завдання (СР)	30
Разом: Аудиторна робота	70
Самостійна робота (СР)	30
Всього:	100

КОНТРОЛЬ ТА КРИТЕРІЇ ОЦІНЮВАННЯ ЗНАТЬ 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 20 балів. Перше завдання (теоретичне) – 4 бали, друге завдання (практичне) – 8 балів, третє завдання (практичне) – 8 балів.

Формою підсумкового контролю є екзамен. Екзаменаційна оцінка (100 балів) є результатом виконання двох теоретичних питань (2 x 20 балів = 40 балів) та практичного завдання (60 балів).

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

СПИСОК РЕКОМЕНДОВАНИХ ДЖЕРЕЛ

Основний:

1. Авраменко А.С. Тестування програмного забезпечення. Навчальний посібник. / Авраменко А.С., Авраменко В.С., Косенюк Г.В. – Черкаси: ЧНУ імені Богдана Хмельницького, 2017. – 284 с.
2. Вакалюк Т.А. Технології тестування програм. Навчально-методичний посібник для студентів напряму 6.040302 Інформатика. – Житомир: Вид-во ЖДУ, 2018. – 96 с.
3. 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.
4. Коцовський В.М. Технологія програмування та створення програмних продуктів: Методичний посібник для студентів спеціальності «Інженерія програмного забезпечення», «Комп'ютерні науки та інформаційні технології» / В. М. Коцовський. - Ужгород: Видавництво УжНУ «Говерла», 2016. - 83 с.