Національний технічний університет України
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Кафедра інформатики та програмної інженерії

КОМПОНЕНТИ ПРОГРАМНОЇ ІНЖЕНЕРІЇ

Якість та тестування програмного забезпечення

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни	
Рівень вищої освіти	***Перший (бакалаврський)***
Галузь знань	*12 Інформаційні технології*
Спеціальність	*121 Інженерія програмного забезпечення*
Освітня програма	*Інженерія програмного забезпечення інформаційних систем*
Статус дисципліни	*Нормативна (обов'язкова)*
Форма навчання	*очна (денна)*
Рік підготовки, семестр	*III курс, осінній семестр*
Обсяг дисципліни	*6 кредитів (180 год)*
Семестровий контроль/ контрольні заходи	*Іспит*
Розклад занять	
Мова викладання	*Українська*
Інформація про керівника курсу	**Лектор*: к.т.н., доцент, Сватко Віталій Володимирович, vsvatko83@gmail.com,*** **Лабораторні: *к.т.н., доцент, Сватко Віталій Володимирович, vsvatko83@gmail.com,***
Розміщення курсу	*https://softserve.academy/course/view.php?id=28* *https://classroom.google.com/c/NzEwNDc4MDkwODUy*

Програма навчальної дисципліни

1. Опис навчальної дисципліни, її мета, предмет вивчення та результати навчання

Мета вивчення дисципліни – сформувати у студента системи знань та вмінь з використання сучасних методів, засобів, технологій тестування програмного забезпечення, вивчення сучасних парадигм та технологій забезпечення якості програмного забезпечення при його розробці.

Предметом вивчення дисципліни є сучасні методи, засоби та технології тестування програмного забезпечення.

Завдання вивчення дисципліни:

- оволодіння студентами методами тестування, верифікації і валідації;
- вивчення студентами підходів до створення звітності по проблемах при розробці програмного забезпечення;
- ознайомлення студентів з сучасними методами та інструментальними засобами контролю якості.

Вивчення дисципліни сприяє формуванню творчого мислення студента, проведенню аналізу

та вибору потрібних джерел та літератури за обраною темою, вмінню формулювати висновки.

В результаті освоєння дисципліни повинні бути сформовані такі компетентності:

ФК 1	Здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення
ФК 2	Здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування
ФК 3	Здатність розробляти архітектури, модулі та компоненти програмних систем
ФК 4	Здатність формулювати та забезпечувати вимоги щодо якості програмного забезпечення у відповідності з вимогами замовника, технічним завданням та стандартами
ФК 7	Володіння знаннями про інформаційні моделі даних, здатність створювати програмне забезпечення для зберігання, видобування та опрацювання даних
ФК 8	Здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв’язання завдань інженерії програмного забезпечення
ФК 10	Здатність накопичувати, обробляти та систематизувати професійні знання щодо створення і супроводження програмного забезпечення та визнання важливості навчання протягом усього життя
ФК 11	Здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення
ФК 12	Здатність здійснювати процес інтеграції системи, застосовувати стандарти і процедури управління змінами для підтримки цілісності, загальної функціональності і надійності програмного забезпечення
ФК 13	Здатність до алгоритмічного та логічного мислення

Після засвоєння дисципліни студенти мають продемонструвати такі результати навчання:

ПРН 1	Аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідкові ресурси і знання з урахуванням сучасних досягнень науки і техніки
ПРН 2	Знати кодекс професійної етики, розуміти соціальну значимість та культурні аспекти інженерії програмного забезпечення і дотримуватись їх в професійній діяльності
ПРН 3	Знати основні процеси, фази та ітерації життєвого циклу програмного забезпечення
ПРН 4	Знати і застосовувати професійні стандарти і інші нормативно-правові документи в галузі інженерії програмного забезпечення
ПРН 6	Вміння вибирати та використовувати відповідну задачі методологію створення програмного забезпечення
ПРН 7	Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення
ПРН 8	Вміти розробляти людино-машинний інтерфейс
ПРН 9	Знати та вміти використовувати методи та засоби збору, формулювання та аналізу вимог до програмного забезпечення
ПРН 10	Проводити передпроектне обстеження предметної області, системний аналіз об'єкта проектування
ПРН 11	Вибирати вихідні дані для проектування, керуючись формальними методами опису вимог та моделювання
ПРН 13	Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань

ПРН 14	Застосовувати на практиці інструментальні програмні засоби доменного аналізу, проектування, тестування, візуалізації, вимірювань та документування програмного забезпечення
ПРН 15	Мотивовано обирати мови програмування та технології розробки для розв'язання завдань створення і супроводження програмного забезпечення
ПРН 16	Мати навички командної розробки, погодження, оформлення і випуску всіх видів програмної документації
ПРН 17	Вміти застосовувати методи компонентної розробки програмного забезпечення
ПРН 18	Знати та вміти застосовувати інформаційні технології обробки, зберігання та передачі даних
ПРН 19	Знати та вміти застосовувати методи верифікації та валідації програмного забезпечення
ПРН 20	Знати підходи щодо оцінки та забезпечення якості програмного забезпечення.
ПРН 23	Вміти документувати та презентувати результати розробки програмного забезпечення

2. Пререквізити та постреквізити дисципліни (місце в структурно-логічній схемі навчання за відповідною освітньою програмою)

Вивчення дисципліни спирається на знання, отримані студентами при вивченні дисциплін «Об'єктно-орієнтоване програмування», «Веб-орієнтована розробка програмного забезпечення», мови програмування.

3. Зміст навчальної дисципліни

Тема 1	Тестування, як засіб підвищення надійності програмного забезпечення.
Тема 2	Фундаментальний процес тестування.
Тема 3	Аналіз вимог до ПЗ. Тестування UI/UX дизайну.
Тема 4	Життєвий цикл дефекту.
Тема 5	Виконання тестів. Документування дефектів в JIRA та Confluence.
Тема 6	Типи, види та рівні тестування.
Тема 7	Техніки тест-дизайну.
Тема 8	Тестова документація.
Тема 9	Типи програмного забезпечення. Тестування мобільних пристроїв. Специфіка тестування Android додатків.
Тема 10	Автоматизація тестування: JMeter, REST.Postman

4. Навчальні матеріали та ресурси

Базова література

1. Авраменко А. С. Тестування програмного забезпечення : навч. посіб. / А. С. Авраменко, В.С. Авраменко, Г. В. Косенюк. – Черкаси : ЧНУ імені Богдана Хмельницького, 2018. – 284с.
2. Крепич С. Я. Якість програмного забезпечення та тестування: базовий курс. Навчальний посібник для бакалаврів галузі знань 12 "Інформаційні технології" спеціальності 121 "Інженерія програмного забезпечення"/ С. Я. Крепич, І. Я. Співак. – Тернопіль : ФОП Паляниця В. А., – 2020. – 478 с.
3. Skorin Y. Introduction of electronic test tools with the aim of increasing the effectiveness of the knowledge test / Y. Skorin, O. Shcherbakov // Development Management. – Volume 4. – 2018. – Issue 4. – Pp. 14–25.
4. Дідковська М. В. Тестування. Основні визначення, аксіоми та принципи. Текст лекцій. Частина І [Електронний ресурс] / М. В. Дідковська, Ю. О. Тимошенко. – Режим доступу : https://www.quality-assurancegroup.com/book/testuvannya-osnovni-viznachennya-aksiomi-ta-printsipi-tekstlektsij-chastina-i.

5. Дідковська М. В. Текст лекцій до курсу "Технології розробки і тестування програм" [Електронний ресурс] / М. В. Дідковська. – Режим доступу : http://mmsa.kpi.ua/disciplines/sac/14c.
6. Дідковська М. В. Тестування: Критерії та методи. Текст лекцій. Частина II – К., 2010. //Електронний ресурс. Режим доступу: http://mmsa.kpi.ua/studentam/study-materials-ua/testing/testuvannya-kriter456- ta-metodi-tekst-lekc456i-chastina-456456.

Додаткова література

1. Лавріщева К. М. Програмна інженерія. Підручник. – К.: 87 Академперіодика, 2008. – 319с.
2. Burnstein I. Practical Software Testing. A process-oriented approach. Springer-Verlag, New York, 2003, – 732 p.
3. Myers G.J. The Art Of Software Testing [Text] / G.J. Myers – New York: John Wiley & Sons, Inc., 2004. – 254 p. – ISBN 0- 471-46912-2.
4. Вакалюк Т.А. Технології тестування програм. Навчально-методичний посібник для студентів напряму 6.040302 Інформатика. – Житомир: Вид-во ЖДУ,2013. – 96с.

Політика та контроль

5. Політика навчальної дисципліни (освітнього компонента)

Як викладач, так і студент зобов'язані дотримуватись Кодексу честі Національного технічного університету України «Київський політехнічний інститут».

Основні положення політики:

1. Матеріали для вивчення дисципліни розміщені викладачем в електронному вигляді на платформі https://softserve.academy/course/view.php?id=28, до якого надано доступ групам студентів.
2. Лекції з дисципліни проводяться викладачем із використанням сучасних мультимедійних презентаційних технологій.
3. В умовах дистанційного навчання 2022-2023 н.р. усі види занять, у тому числі контрольні заходи, проводяться в Zoom.

5.1. Тематика лекцій

Лекція 1	*Тестування, як засіб підвищення надійності програмного забезпечення*	1. *Основні визначення* 2. *Рівні і види тестування* 3. *Техніки тестування* 4. *Аксіоми тестування*
Лекція 2	*Фундаментальний процес тестування*	1. *Основні принципи тестування* 2. *Ключові етапи процесу фундаментального тестування відповідно до стандарту ISTQB*
Лекція 3	*Аналіз вимог до ПЗ*	1. *Визначення та класифікація вимог* 2. *Характеристики хороших вимог* 3. *Типи документів, за допомогою яких можна повідомити вимоги* 4. *Аналіз вимог* 5. *Матриця простежуваності* 6. *Матриця залежностей*
Лекція 4	*Тестування UI/UX дизайну*	1. *Інтерфейс користувача* 2. *Елементи керування інтерфейсу користувача* 3. *Досвід користувача* 4. *Юзабіліті* 5. *Правила розробки інтерфейсу*

Лекція 5	*Життєвий цикл дефекту*	1. *Життєвий цикл дефекту* 2. *Звіт про дефект* 3. *Поради щодо звітування про дефекти* 4. *Інструменти відстеження дефектів. Jira*
Лекція 6	*Виконання тестів*	1. *Що таке збірка?* 2. *Контроль версій, системи контролю версій* 3. *Процес виконання тесту* 4. *Тестове виконання в Zephyr для Jira*
Лекція 7	*Документування дефектів в JIRA та Confluence*	1. *Jira як інструмент управління проектами* 2. *Confluence як інструмент командної співпраці*
Лекція 8	*Типи, види та рівні тестування*	1. *Тестові підходи* 2. *Типи тестів за рівнями тестів* 3. *Типи тестів за цілями тесту* 4. *Порядок тестування*
Лекція 9	*Техніки тест-дизайну*	1. *Статичне та динамічне тестування* 2. *Методики тестування* 3. *Чорний ящик* 4. *Білий ящик* 5. *На основі досвіду* 6. *Вибір техніки розробки тесту*
Лекція 10	*Розробка та реалізація тесту*	1. *Процес розробки та впровадження тесту* 2. *приклад* 3. *Інструменти керування тестовими випадками*
Лекція 11	*Тестова документація*	• *Test Policy* • *Test Strategy* • *Test Plan* • *Test Design Specification* • *Test Case, Test Scenario, Checklist* • *Test Case Specification* • *Test Procedure Specification* • *Test Incident Report* • *Test Summary Report* • *Level of formality for Test Documentation*
Лекція 12	*Типи програмного забезпечення*	1. *Типи програмного забезпечення* 2. *Типи додатків* 3. *Basic Application Architecture* 4. *Web Application* 5. *Cloud/SaaS* 6. *Mobile Application*
Лекція 13	*Тестування мобільних пристроїв*	1. *Принципи тестування мобільних пристроїв* 2. *Функціональне тестування* 3. *Тестування Usability* 4. *Тестування інсталяції* 5. *Performance тестування*
Лекція 14	*Специфіка тестування Android додатків*	• *Android user experience (guides)* • *interruption (incoming call, alarm, etc)*

		• *rotation* • *notification* • *internet connection (wi-fi/3g)* • *GPS, Biometric Scanners, Gyroscopes, Accelerometers* • *Installation/upgrade testing* • *Processing Power and Battery Life* • *crash types*
Лекція 15	*Автоматизація тестування: JMeter*	1. *Тестування продуктивності* 2. *Про JMeter* 3. *Простий тест* 4. *Тестування кількох сторінок* 5. *Запис сценарію користувача* 6. *Логування за допомогою JMeter* 7. *Читання тестових даних із файлу*
Лекція 16	*Автоматизація тестування: REST.Postman*	1. *Веб-архітектура* 2. *Коди стану HTTP* 3. *Що таке API?* 4. *Методи запиту HTTP (s).*

5.2. Тематика комп'ютерних практикумів/лабораторних робіт

1. *Тестування UI/UX дизайну*
2. *User Story (аналіз вимог до ПЗ)*
3. *Розробка Test Plan*
4. *Побудова чек-листа та тестового покриття*
5. *Розробка позитивних та негативних тест-кейсів*
6. *Види та типи тестування*
7. *Техніки тестування*

6. Політика та контроль

6.1. Політика навчальної дисципліни (освітнього компонента)

При зарахування та оцінювання лабораторних робіт беруться до уваги наступні чинники:

- *Повнота виконання завдання на лабораторну роботу за індивідуальним варіантом;*
- *Своєчасність виконання лабораторної роботи згідно графіку;*
- *Самостійність виконання лабораторної роботи та відсутність ознак плагіату;*
- *Відповіді на питання щодо змісту лабораторної роботи під час її захисту.*

При оцінюванні контрольних робіт до уваги приймаються:

- *Правильність та повнота виконання завдань;*
- *Кількість виконаних завдань в умовах обмеженого часу;*
- *Самостійність виконання завдань та відсутність ознак плагіату;*
- *Кількість спроб виконання контрольних, які передують тій, що оцінюється.*

Для підготовки до контрольних студенти отримують перелік теоретичних питань та зміст типових задач, які будуть у завданнях на контрольних.
При першої та другої атестації до уваги приймається кількість лабораторних робіт та контрольних робіт зарахованих на час проведення атестації

6.2. Види контролю та рейтингова система оцінювання результатів навчання (РСО)

Система оцінювання успішності студентів з дисципліни «Компоненти програмної інженерії. Якість та тестування програмного забезпечення» ґрунтується на «Положенні про систему

оцінювання результатів навчання в КПІ ім. Ігоря Сікорського» (https://document.kpi.ua/files/2020_1-273.pdf), а саме Рейтингової системі оцінювання (РСО) другого типу (РСО-2).

РСО-2 складається з двох складових:

- *стартової (RS);*
- *екзаменаційної (RE).*

Стартові бали формуються як сума балів, що отримані за результатами заходів поточного контролю (лабораторних робіт (RL) і контрольних (RK)), заохочувальних (RЗ) та штрафних (RШ) балів:

$$RS = RL + RK + RЗ + RШ,$$

де RL за 8 лабораторних робіт складає 8 X 5 = 40 балів,

RK за дві контрольні складає 2 X 10 = 20 балів.

Таким чином максимальна основна сума стартових балів складає 40 + 20 = 60 балів.

Поза основною шкалою оцінювання передбачені заохочувальні та штрафні бали, які враховуються у загальній сумі балів, але не входять до основної шкали РСО. Заохочувальні бали враховують відповідь на питання та виконання завдань на лекційних

заняттях, якість конспекту. Штрафні бали передбачені за несвоєчасне виконання лабораторних робіт, тобто зі запізненням відносно графіку.

Екзаменаційні бали (RE), таким чином складають максимально 40 балів, але для допуску до екзамену студент повинен мати не менше ніж 60% від максимальної суми стартових балів, що

складає

$$60 \times 0.6 = 36 \text{ балів.}$$

Після складання екзамену стартові бали RS складаються зі екзаменаційними RE.

Оцінювання результатів навчання здійснюється за 100-бальною шкалою з подальшим переведенням оцінок за університетською шкалою згідно таблиці:

Оцінка ECTS RD	Традиційна оцінка
95…100	відмінно
85 … 94	дуже добре
75 … 84	добре
65 … 74	задовільно
60 … 64	достатньо
30 ≤ RD <60	незадовільно
RD <30	не допущений

Робочу програму навчальної дисципліни (силабус):

Складено доцент, к.т.н., Сватко Віталій Володимирович

Ухвалено кафедрою ІПІ (протокол № 16 від 29.05.2024 р.)

Погоджено Методичною комісією факультету (протокол № 10 від 21.06.2024 р.)