КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

БУДІВНИЦТВА І АРХІТЕКТУРИ

МАГІСТР
(освітній ступінь)

Кафедра ***інформаційних технологій***

«Затверджую»
Голова НМР факультету автоматизації і інформаційних технологій
_____ Олександр ТЕРЕНТЬЄВ/
«___» _____ 2024 р.

РОБОЧА ПРОГРАМА ОСВІТНЬОЇ КОМПОНЕНТИ

ОК 5 «ЗАСОБИ КЕРУВАННЯ ЯКІСТЮ ПРОЦЕСУ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ»

(шифр та назва освітньої компоненти)

Шифр	назва спеціальності, освітньої програми
122	***Комп'ютерні науки,*** ***Комп'ютерні науки***

Мова викладання: українська
Розробники:
Ігор АЧКАСОВ, доктор технічних наук, професор, професор кафедри ІТ _____
(ім'я та прізвище, науковий ступінь, звання) (підпис)
Олексій МАЦІЄВСЬКИЙ, асистент кафедри ІТ _____
(ім'я та прізвище, науковий ступінь, звання) (підпис)

Робоча програма затверджена на засіданні кафедри інформаційних технологій
Протокол № ***13*** від ***«25» червня 2024 року***

Завідувачка кафедри ІТ _____ Тетяна ГОНЧАРЕНКО
(підпис)

Схвалено гарантом освітньої програми

Гарант ОП _____ Тетяна ГОНЧАРЕНКО
(підпис)

Розглянуто на засіданні науково-методичної комісії спеціальності
122 «Комп'ютерні науки».
Протокол № ***3*** від ***«28» червня 2024*** року.

ВИТЯГ З РОБОЧОГО НАВЧАЛЬНОГО ПЛАНУ НА 2024-2025 НАВЧАЛЬНИЙ РІК

Шифр	Назва спеціальності, освітньої програми	Форма здобуття ВО: **денна**											Форма контролю	Семестр	Погодження заступником декана факультету
		Кількість кредитів	Кількість годин						Кількість індивідуальних робіт						
			Всього	Аудиторних				Самостійна робота	КП	КР	РГР	Контрольна робота			
				Разом	у тому числі										
					Лекції	Лабораторні	Практичні								
122	Комп'ютерні науки, Комп'ютерні науки	***5,0***	***150***	***60***	***30***	***30***		***90***			***1***		***Залік***	***1***	

Анотація. Мета навчальної дисципліни

Пререквізити: Програмування та алгоритмічні мови; проектування інформаційних систем; розробка програмного забезпечення.

Посилання на сторінку електронного навчально-методичного комплексу дисципліни: *https://org2.knuba.edu.ua/course/section.php?id=17882*

Мета дисципліни - теоретична підготовка здобувачів вищої освіти у напрямку вивчення принципів забезпечення якості програмних продуктів. Дисципліна охоплює сучасні моделі, методи та засоби керування якістю на всіх етапах життєвого циклу програмного забезпечення, включаючи аналіз вимог, проектування, кодування, тестування, впровадження та супроводження. Студенти набудуть практичних навичок у застосуванні інструментів і методів забезпечення якості, оцінювання та вдосконалення процесів розробки програмного забезпечення, що дозволить їм ефективно розв'язувати як теоретичні, так і прикладні задачі у галузі програмної інженерії.

Завдання дисципліни полягає у вивченні теоретичних основ і практичних методів забезпечення якості програмного забезпечення на всіх етапах його життєвого циклу. Дисципліна охоплює питання впровадження стандартів і методологій управління якістю, використання сучасних інструментів для автоматизації процесів контролю, аналізу та оцінювання якості з використанням метрик та показників. Особлива увага приділяється підвищенню ефективності процесів розробки, забезпеченню відповідності програмного забезпечення встановленим вимогам, мінімізації дефектів та покращенню його надійності, продуктивності й зручності для користувачів.

Вивчення дисципліни «Засоби керування якістю процесу розробки програмного забезпечення» сприяє формуванню у здобувачів **наступних компетентностей**.

Компетенції здобувачів освітньої програми, що формуються в результаті засвоєння освітньої компоненти

Код	Зміст
Інтегральна компетентність	
ІК	Здатність розв'язувати задачі дослідницького та/або інноваційного характеру у сфері комп'ютерних наук.
Загальні компетентності	
ЗК02	Здатність застосовувати знання у практичних ситуаціях
ЗК05	Здатність вчитися й оволодівати сучасними знаннями.
ЗК06	Здатність бути критичним і самокритичним.
СПЕЦІАЛЬНІ (ФАХОВІ) КОМПЕТЕНТНОСТІ	
СК04	Здатність збирати і аналізувати дані (включно з великими), для забезпечення якості прийняття проєктних рішень.
СК07	Здатність розробляти програмне забезпечення відповідно до сформульованих вимог з урахуванням наявних ресурсів та обмежень.
СК08	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 проєктом.
СК09	Здатність розробляти та адмініструвати бази даних та знань.
СК010	Здатність оцінювати та забезпечувати якість ІТ-проєктів, 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.
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.

Це забезпечує досягнення програмних результатів навчання, згідно з якими **студент повинен мати знання з питань**:

Програмні результати здобувачів освітньої програми, що формуються в результаті засвоєння освітньої компоненти

Код	Програмні результати
РН1	Мати спеціалізовані концептуальні знання, що включають сучасні наукові здобутки у сфері комп'ютерних наук і є основою для оригінального мислення та проведення досліджень, критичне осмислення проблем у сфері комп'ютерних наук та на межі галузей знань.
РН3	Зрозуміло і недвозначно доносити власні знання, висновки та аргументацію у сфері комп'ютерних наук до фахівців і нефахівців, зокрема до осіб, які навчаються.
РН5	Оцінювати результати діяльності команд та колективів у сфері інформаційних технологій, забезпечувати ефективність їх діяльності.
РН12	Проектувати та супроводжувати бази даних та знань.
РН13	Оцінювати та забезпечувати якість інформаційних та комп'ютерних систем різного призначення.
РН14	Тестувати програмне забезпечення.
РН17	Виявляти та усувати проблемні ситуації в процесі експлуатації програмного забезпечення, формулювати завдання для його модифікації або реінжинірингу.

Структура навчальної дисципліни

Назва теми	Кількість годин	
	Лекції	**Лабораторні роботи**
Змістовний модуль 1. Основи забезпечення якості програмного забезпечення		
Лекція 1. Вступ до забезпечення якості програмного забезпечення *Тема 1. Основні поняття забезпечення якості.* *Тема 2. Історія розвитку забезпечення якості.* *Тема 3. Роль якості у розробці програмного забезпечення.* *Тема 4. Розбір основних термінів.*	**2**	**2**
Лекція 2. Моделі якості програмного забезпечення *Тема 1. Стандарти якості програмного забезпечення.* *Тема 2. Огляд моделі ISO/IEC 25010.* *Тема 3. Атрибути та показники якості.* *Тема 4. Використання моделей якості в практиці.*	**2**	**2**
Лекція 3. Інструменти та методи тестування програмного забезпечення *Тема 1. Типи тестування.* *Тема 2. Ручне та автоматизоване тестування.* *Тема 3. Інструменти для забезпечення якості.* *Тема 4. Аналіз популярних інструментів тестування.*	**2**	**2**
Лекція 4. Верифікація та валідація програмного забезпечення *Тема 1. Основи верифікації та валідації.* *Тема 2. Методи верифікації.* *Тема 3. Методи валідації.* *Тема 4. Впровадження та приклади використання.*	**2**	**2**
Змістовий модуль 2. Практичні аспекти забезпечення якості програмного забезпечення		
Лекція 5. Планування та організація тестування *Тема 1. Розробка тестових сценаріїв.* *Тема 2. Управління тестовим процесом.*	**4**	**4**

Тема 3. Стратегії тестування. *Тема 4. Ефективне планування тестування.*		
Лекція 6. Автоматизація тестування програмного забезпечення *Тема 1. Основи автоматизації тестування.* *Тема 2. Інструменти для автоматизації.* *Тема 3. Створення тестового середовища.* *Тема 4. Приклади автоматизованого тестування.*	**4**	**4**
Лекція 7. Управління дефектами та контроль якості *Тема 1. Життєвий цикл дефекту.* *Тема 2. Інструменти для управління дефектами.* *Тема 3. Контроль якості програмного забезпечення.* *Тема 4. Методи усунення дефектів.*	**4**	**4**
Лекція 8. Метрики якості та оцінка ефективності процесу забезпечення якості *Тема 1. Основи метрик якості.* *Тема 2. Вибір показників для оцінки якості.* *Тема 3. Аналіз ефективності процесів.* *Тема 4. Використання метрик у реальних проєктах.*	**4**	**4**
Змістовний модуль 3. Методи та інструменти забезпечення якості програмного забезпечення: тестування та автоматизація REST API		
Лекція 9. Забезпечення якості REST API: тестування, валідація та автоматизація *Тема 1. Стандарти якості для REST API.* *Тема 2. Методи тестування REST API.* *Тема 3. Планування тестування REST API.* *Тема 4. Інструменти для автоматизації REST API.* *Тема 5. Аналіз результатів тестування та підвищення якості REST API.*	**6**	**6**
РАЗОМ	**30**	**30**

Теми лабораторних робіт

№	Назва теми	Формат виконання	К-сть годин	К-сть балів
	Змістовий модуль 1. Основні положення забезпечення якості програмного забезпечення			
1	Лабораторна робота №1. Вступ до забезпечення якості програмного забезпечення.	Індивідуально	2	5
2	Лабораторна робота №2. Моделі якості програмного забезпечення	Індивідуально	2	5
3	Лабораторна робота №3. Інструменти та методи тестування програмного забезпечення	Індивідуально	2	6
4	Лабораторна робота №4. Верифікація та валідація програмного забезпечення	Індивідуально	2	6
	Змістовий модуль 2. Практичні аспекти забезпечення якості програмного забезпечення			
5	Лабораторна робота №5. Планування та організація тестування	Індивідуально	4	7
6	Лабораторна робота №6. Автоматизація тестування програмного забезпечення	Команда	4	7
7	Лабораторна робота №7. Управління дефектами та контроль якості	Індивідуально	4	7
9	Лабораторна робота №8. Метрики якості та оцінка ефективності процесу забезпечення якості	Індивідуально	4	7
	Змістовий модуль 3. Міжнародні стандарти та сертифікація якості програмного забезпечення			
9	Лабораторна робота №9. Забезпечення якості REST API: тестування, валідація та автоматизація	Команда	6	10
	Разом		30	60

Розподіл годин самостійної роботи здобувачів

№	Назва теми	Кількість годин
	Змістовий модуль 1.	**22**
1	Розгляд ключових принципів забезпечення якості, таких як орієнтація на користувача, попередження дефектів, забезпечення прозорості процесів та постійного вдосконалення. Аналіз взаємозв'язку між якістю продукту та процесів розробки. Ознайомлення з циклами якості: PDCA (Plan-Do-Check-Act) та SPI (Software Process Improvement). Вивчення ролі стандартів у побудові процесу забезпечення якості.	4
2	Дослідження основних характеристик та підхарактеристик моделі ISO/IEC 25010: функціональність, продуктивність, надійність, зручність використання, сумісність, безпека, переносність та обслуговуваність. Аналіз практичного застосування моделі для оцінки якості програмного забезпечення. Вивчення прикладів використання моделі ISO/IEC 25010 для оцінки реальних продуктів.	4
3	Вивчення підходу чорного ящика для тестування зовнішньої поведінки ПЗ без знання внутрішньої структури. Ознайомлення з методом білого ящика для тестування логіки та коду. Дослідження регресійного тестування: його цілей, завдань і значення для стабільності ПЗ після внесення змін. Порівняння ефективності кожного методу для різних типів задач.	6
4	Практичне вивчення принципів створення чек-листів: визначення ключових критеріїв перевірки, структуризація завдань, перевірка покриття вимог. Розробка тест-кейсів для забезпечення чіткості та послідовності тестування. Вивчення технік опису передумов, кроків виконання та очікуваних результатів для кожного тест-кейсу. Ознайомлення з прикладами добре структурованих чек-листів і тест-кейсів для різних типів ПЗ.	8
	Змістовий модуль 2.	**26**
4	Вивчення основних компонентів плану тестування, включаючи визначення обсягу тестування, методів, ресурсів та термінів. Ознайомлення з техніками аналізу ризиків для формування пріоритетів тестування. Розробка тест-стратегій на основі типів тестування: функціонального, навантажувального, регресійного. Дослідження прикладів реальних планів тестування, їх структури та особливостей для різних типів проектів.	6
5	Ознайомлення з основними можливостями Selenium для автоматизації тестування веб-інтерфейсів: запис сценаріїв, інтеграція з мовами програмування, налаштування тестових середовищ. Вивчення JUnit для створення автоматизованих тестів Java-додатків: структура тестів, використання анотацій, перевірка результатів. Аналіз переваг та недоліків кожного інструменту для різних задач автоматизації. Практичне виконання простих тестових сценаріїв за допомогою Selenium та JUnit.	7
6	Ознайомлення з основними принципами управління дефектами: їх реєстрація, класифікація, пріоритизація та моніторинг. Дослідження можливостей JIRA для управління дефектами: створення завдань, використання робочих дошок, інтеграція з процесами Agile. Аналіз функцій Redmine для роботи з дефектами: налаштування трекерів, створення звітів. Практичне використання обох інструментів для ведення дефектів у тестовому проекті.	6
	Дослідження метрик покриття тестами: рівень покриття коду, функціональності	7

	та вимог. Ознайомлення зі способами розрахунку та інструментами для аналізу (наприклад, Jacoco, SonarQube). Вивчення метрики часу на відновлення системи (MTTR): методи вимірювання, значення для оцінки стабільності системи. Аналіз впливу цих метрик на процес ухвалення рішень щодо покращення якості. Розгляд прикладів використання метрик на реальних проектах.	
	Змістовий модуль 3.	**24**
7	Ознайомлення зі структурою тест-плану, включаючи його основні компоненти: цілі, обсяг тестування, ризики, критерії виходу та завершення тестування. Вивчення технік визначення тестових пріоритетів на основі аналізу ризиків. Дослідження підходів до розподілу ресурсів і формування графіків тестування. Практика створення тест-плану для невеликого програмного продукту, враховуючи різні типи тестування (функціональне, навантажувальне, регресійне).	8
8	Вивчення інструмента Selenium для автоматизації веб-тестування: налаштування середовища, написання простих скриптів для взаємодії з веб-сторінками, використання елементів DOM. Ознайомлення з JUnit як фреймворком для тестування Java-додатків: написання юніт-тестів, використання анотацій, запуск тестових наборів. Дослідження інтеграції автоматизованих тестів у CI/CD-процеси. Практичне виконання тестових сценаріїв для веб-додатків або Java-програм.	4
9	Розгляд основ управління дефектами: етапи життєвого циклу дефекту, класифікація за пріоритетом та критичністю. Вивчення можливостей JIRA: створення завдань, налаштування робочих процесів, створення звітів про дефекти. Ознайомлення з Redmine: створення трекерів, управління проєктами, формування звітів. Практичне виконання завдань у JIRA та Redmine для ведення дефектів у тестовому проєкті.	4
10	Вивчення метрики покриття тестами: аналіз покриття коду, функцій та вимог; способи вимірювання з використанням інструментів, таких як Jacoco. Дослідження метрики часу на відновлення системи (MTTR): методи вимірювання, значення для оцінки стабільності системи. Розгляд методів підвищення якості системи на основі отриманих метрик. Практичне застосування інструментів аналізу покриття тестами та оцінки MTTR у навчальних проєктах.	8
	Виконання РГР за варіантом	12
	Підготовка до заліку	6
	Всього	**90**

Структура розрахунково-графічної роботи, розподіл часу та балів

Завдання на розрахунково-графічну роботу: розробити план забезпечення якості програмного забезпечення для гіпотетичного проекту, застосовуючи сучасні моделі, методи та інструменти. Робота повинна включати теоретичну частину з описом вибраних методів та практичну частину з прикладами застосування.

	Структура розрахунково-графічної роботи	Години	Бали
1	Завдання повинно включати: 1. Аналіз предметної області та постановка задачі 2. Розробка та тестування програмних модулів 3. Написання пояснювальної записки 4. Підготовка презентації 5. Публічний захист	12	20
	Разом	**12**	**20**

Політика щодо академічної доброчесності

Тексти індивідуальних завдань (в т.ч. у разі, коли вони виконуються у формі презентацій або в інших формах) можуть перевірятись на плагіат. Для цілей захисту індивідуального завдання оригінальність тексту має складати не менше 70%. Виключення становлять випадки зарахування публікацій Здобувачів у матеріалах наукових конференціях та інших наукових збірниках, які вже пройшли перевірку на плагіат.

Списування під час тестування та інших опитувань, які проводяться у письмовій формі, заборонені (в т.ч. із використанням мобільних девайсів). У разі виявлення фактів списування з боку здобувача він отримує інше завдання. У разі повторного виявлення призначається додаткове заняття для проходження тестування.

Політика щодо відвідування

Навчальний процес з курсу «Засоби керування якістю процесу розробки програмного забезпечення» організовано з використанням платформи Microsoft Teams, що забезпечує гнучкість у форматі навчання.

Особливості організації навчального процесу:

- Усі лекційні заняття записуються та зберігаються у відповідному каналі Teams протягом семестру.
- Лабораторні роботи представлені в електронному вигляді з докладними інструкціями виконання.
- Матеріали курсу (презентації, додаткові ресурси) доступні в Teams.
- Консультації можливі як в очному форматі, так і через Teams.

Виконання та захист лабораторних робіт:

- Лабораторні роботи можуть виконуватися дистанційно.
- Захист робіт можливий протягом усього семестру.
- Командні проєкти можуть виконуватися розподілено з використанням систем контролю версій.

У разі пропуску занять здобувач має:

- Надати до деканату та продемонструвати викладачу документи, що підтверджують поважність причини пропуску (медичні довідки, документи про участь у конференціях, стажуваннях тощо).
- Переглянути відеозапис пропущеної лекції в Teams.
- Виконати всі практичні завдання, передбачені за темою пропущеного заняття.

Можливість онлайн-навчання надається за таких умов:

- Хвороба (за наявності медичної довідки).
- Участь у міжнародному стажуванні.
- Участь у наукових конференціях.
- Інші об’єктивні обставини за погодженням з керівником курсу.

Методи контролю

Основні форми участі Здобувачів у навчальному процесі, що підлягають поточному контролю: доповнення, опанування до виступу, рецензія на виступ; участь у дискусіях; аналіз першоджерел; письмові завдання (тестові, індивідуальні роботи у формі рефератів); та інші письмові

роботи, оформлені відповідно до вимог. Кожна тема курсу, що винесена на лекційні та практичні заняття, відпрацьовується.

Під час оцінювання рівня знань Здобувача аналізу підлягають:

- характеристики відповіді: цілісність, повнота, логічність, обґрунтованість, правильність;

- якість знань (ступінь засвоєння фактичного матеріалу): осмисленість, глибина, гнучкість, дієвість, системність, узагальненість, міцність;

- ступінь сформованості уміння поєднувати теорію і практику під час розгляду ситуацій, практичних завдань;

- рівень володіння розумовими операціями: вміння аналізувати, синтезувати, порівнювати, абстрагувати, узагальнювати, робити висновки з проблем, що розглядаються;

- досвід творчої діяльності: уміння виявляти проблеми, розв’язувати їх, формувати гіпотези;

- самостійна робота: робота з навчально-методичною, науковою, допоміжною вітчизняною та зарубіжною літературою з питань, що розглядаються, уміння отримувати інформацію з різноманітних джерел (традиційних; спеціальних періодичних видань, ЗМІ, Internet тощо).

Тестове опитування може проводитись за одним або кількома змістовими модулями. В останньому випадку бали, які нараховуються Здобувачу за відповіді на тестові питання, поділяються між змістовими модулями.

Індивідуальне завдання підлягає у виконанні РГР згідно індивідуального варіанту для закріплення теоретичних знань і практичних навичок з QA тестування за 3 семестр.

Результати **поточного контролю** заносяться до журналу обліку роботи. Позитивна оцінка поточної успішності Здобувачів за відсутності пропущених та невідпрацьованих практичних занять та позитивні оцінки за індивідуальну роботу є підставою для допуску до підсумкової форми контролю. Бали за аудиторну роботу відпрацьовуються у разі пропусків.

Підсумковий контроль здійснюється під час проведення залікової сесії з урахуванням підсумків поточного та модульного контроля. Під час семестрового контролю враховуються результати здачі усіх видів навчальної роботи згідно зі структурою кредитів.

Оцінювання проводиться за 100-бальною шкалою.

Розподіл балів для дисципліни з формою контролю залік

Поточне оцінювання				Підсумковий контроль	Сума балів
Змістовий модуль 1	Змістовий модуль 2	Змістовий модуль 3	Індивідуальне завдання		
22	28	10	20	20	100

Шкала оцінювання: національна та ECTS

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою
90-100	**A**	відмінно
82-89	**B**	добре
74-81	**C**	
64-73	**D**	задовільно
60-63	**E**	
35-59	**FX**	Не зараховано з можливістю повторного складання
0-34	**F**	Не зараховано з обов'язковим повторним вивченням дисципліни

Умови допуску до підсумкового контролю

Здобувачу, який має підсумкову оцінку за дисципліну від 35 до 59 балів, призначається додаткова залікова сесія. В цьому разі він повинен виконати додаткові завдання, визначені викладачем.

Здобувач, який не виконав вимог робочої програми за змістовими модулями, не допускається до складання підсумкового контролю. В цьому разі він повинен виконати визначене викладачем додаткове завдання за темами відповідних змістових модулів в період між основною та додатковою сесіями.

Здобувач має право на опротестування результатів контроля (апеляцію). Правила подання та розгляду апеляції визначені внутрішніми документами КНУБА, які розміщені на сайті КНУБА та зміст яких доводиться Здобувачам до початку вивчення дисципліни.

Методичне забезпечення дисципліни

Навчальні посібники:

1. Авраменко А.С., Авраменко В.С. Косенюк Г.В.“ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ”: навч. посіб. Черкаський національний університет імені Богдана Хмельницького. – Д. : [Електронний ресурс] Доступно: https://eprints.cdu.edu.ua/1482/1/testyvan.pdf
2. Крепич С.Я., Співак І.Я. “Якість програмного забезпечення та тестування: базовий курс”: навч. посіб. Західноукраїнський національний універиситет [Електронний ресурс] Доступно: bit.ly/3DCKmHn

Методичні вказівки:

1. Засоби керування якістю процесу розробки програмного забезпечення : методичні вказівки та завдання до виконання лабораторних робіт №1-7 для підготовки здобувачів другого магістерського
рівня вищої освіти спеціальності 122 «Комп’ютерні науки» Київ. нац. ун-т буд-ва і архіт. ; укладачі : Т. А. Гончаренко, та О. О. Мацієвський. – Київ : КНУБА, 2024. – 19 с. : іл. - Бібліогр. : с. 18.
https://org2.knuba.edu.ua/course/section.php?id=17883

Посилання на програмне забезпечення та інструментальні засоби:

1. https://pypi.org/project/pycheck/0.1dev
2. https://www.atlassian.com/software/jira
3. https://www.testrail.com/
4. https://github.com/Reshift-Security
5. https://scan.coverity.com/

Додаткові джерела

1. V. Levytskyi, P. Kruk, O. Lopuha, D. Sereda, V. Sapaiev, O. Matsiievskyi. Use of Deep Learning Methodologies in Combination with Reinforcement Techniques within Autonomous Mobile Cyber-Physical Systems. 2024 IEEE 4th International Conference on Smart Information Systems and Technologies, Kazakhstan, Astana 2024, pp. 455-460 (Scopus). https://doi.org/10.1109/SIST61555.2024.10629589

2. O. Matsiievskyi, T. Honcharenko, O. Solovei, T. Liashchenko, I. Achkasov, V. Golenkov. Using Artificial Intelligence to Convert Code to Another Programming Language. 2024 SIST 2024 - 2024 IEEE 4th International Conference on Smart Information Systems and Technologies, Proceedings, 2024, Kazakhstan, Astana pp. 379–385 (Scopus).
https://doi.org/10.1109/SIST61555.2024.10629305

3. Bushuyev, S., Sukach, S., Kontsevyi, V., ...Achkasov, I., Murovanskiy, G. Inspirational project governance into the holacracy environment. CEUR Workshop Proceedings This link is disabled., 2023, 3453, pp. 47–58 (ceur-ws).
https://ceur-ws.org/Vol-3453/paper5.pdf

Інформаційні ресурси:

1. http://library.knuba.edu.ua
2. http://org2.knuba.edu.ua
3. DOU.UA [Електронний ресурс]. Доступно: https://dou.ua/forums/topic/40666/
4. Test Matrix [Електронний ресурс]. Доступно: https://testmatick.com/uk/ponyattya-informaczijnoyi-systemy-zgidn/