

ІВАНО-ФРАНКІВСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ НАФТИ І  
ГАЗУ  
ІНСТИТУТ ЕКОНОМІКИ ТА МЕНЕДЖМЕНТУ  
Кафедра менеджменту та адміністрування

**ЗВІТ  
ПРО ПРОХОДЖЕННЯ ПЕДАГОГІЧНОГО ПРАКТИКУМУ**

Аспіранта \_\_\_\_\_ Смаги Любомира Романовича \_\_\_\_\_  
(прізвище, ім'я, по батькові)

Освітня програма третього (освітньо-наукового) рівня вищої освіти ступеня «доктор філософії»

Спеціальність \_\_\_\_\_ 073 «Менеджмент» \_\_\_\_\_  
(назва)

2022 рік навчання, група А-073-22

База практики Кафедра менеджменту та адміністрування Інституту економіки та менеджменту Івано-Франківського національного технічного університету нафти і газу \_\_\_\_\_  
(назва підприємства, установи)

**Терміни практики**

з « 1 » жовтня 2023 р. по « 30 » листопада 2023 р.

**Науковий керівник аспіранта**

к.е.н., доцент Станьковська Ірина Мирославівна \_\_\_\_\_  
(вчене звання, ПІБ, посада) (підпис)

**Керівник практики від кафедри:**

д.е.н., професор Запужляк Іванна Богданівна \_\_\_\_\_  
(вчене звання, ПІБ, посада) (підпис)

**Івано-Франківськ – 2023 р.**

## Зміст

Вступ.....	6
Цілі викладацької практики.....	6
Завдання викладацької практики.....	6
Лекційне заняття.....	8
Навчальна дисципліна.....	8
Тема.....	8
Мета.....	8
Завдання.....	8
План заняття.....	9
Лекційне заняття: конспект.....	10
Вступ.....	10
Ключові компоненти управління якістю програмного забезпечення.....	10
Проектні ролі через призму управління проектами.....	12
Метрики vs. ключові показники ефективності (KPI).....	16
Огляд застосування метрик та KPI управління якістю на прикладі пропонованого ІТ проекту.....	17
Підсумок основних пунктів лекції та обговорення ключових ідей.....	20
Самоаналіз.....	21
Висновок.....	21
Практичне заняття.....	23
Навчальна дисципліна.....	23
Тема.....	23
Мета.....	23
Завдання.....	23
План заняття.....	23
Питання для перевірки засвоєння лекційного матеріалу.....	23
Відеоматеріал.....	24
Тести.....	24
Рекомендована література для самостійного навчання.....	27
Рекомендовані відеоматеріали.....	27
Список використаних джерел.....	28
Додатки.....	29

## **ЩОДЕННИК ПЕДАГОГІЧНОГО ПРАКТИКУМУ**

аспіранта Смаги Любомира Романович за період з «  1  »  жовтня  2023 р.  
по «  30  »  листопада  2023 р.

Місце практики Кафедра менеджменту та адміністрування Інституту економіки та менеджменту Івано-Франківського національного технічного університету нафти і газу

Викладач-керівник к.е.н., доцент Станьковська Ірина Мирославівна

Дата «  1  »  жовтня  2023 р

Вид виконаної роботи \_\_\_\_\_

Відмітка про виконання \_\_\_\_\_

Підпис аспіранта \_\_\_\_\_

Підпис викладача-керівника \_\_\_\_\_

## Заходи, пов'язані з проходженням активної і пасивної практики

Дата	Вид виконаної роботи	Дисципліна, тема та зміст заняття	Відмітка про виконання, оцінка
<b>Пасивна практика</b>			
02.09.23	Опрацювання матеріалів	Опрацювання методичних вказівок за предметом «Управління якістю». Підбір теми лекційного та практичного заняття.	
06.10.23	Консультації	Консультації з викладачем-керівником	
08.10.23	Опрацювання матеріалів	Опрацювання теоретичних матеріалів для підготовки та проведення лекційних та практичних занять	
10.11.23	Підготовка матеріалів лекції	Підготовка матеріалів лекції	
15.11.23	Підготовка матеріалів практичного заняття	Підготовка матеріалів практичного заняття	
		Підпис аспіранта	Підпис викладача
<b>Активна практика</b>			
16.11.23	Проведення лекційного заняття	<p>Тема: «Управління якістю в ІТ проєктах: критерії оцінки та метрики».</p> <ol style="list-style-type: none"> <li>1. Ключові компоненти управління якістю програмного забезпечення.</li> <li>2. Проєктні ролі через призму управління якості.</li> <li>3. Метрики управління якості з перспективи різних проєктних ролей.</li> <li>4. Метрики vs. Ключові показники ефективності (KPI).</li> <li>5. Приклад критеріїв оцінки якості впровадження та методів їхнього обрахунку.</li> </ol> <p>Навчальна мета: Ознайомлення слухачів із сутністю та важливістю управління якістю в ІТ проєктах, розкриття</p>	

		критеріїв оцінки та метрик, які допомагають забезпечити ефективний контроль та підвищити якість проектів у сфері інформаційних технологій.	
16.11.23	Проведене практичне заняття	Тема: «Управління якістю в ІТ проєктах: критерії оцінки та метрики». Завдання: <ol style="list-style-type: none"> <li>1. Питання для перевірки засвоєння лекційного матеріалу.</li> <li>2. Перегляд відеоматеріалу</li> <li>3. Тести</li> </ol>	
		Підпис аспіранта _____	Підпис викладача _____

## Вступ

### *ЦІЛІ ВИКЛАДАЦЬКОЇ ПРАКТИКИ*

1. Оволодіння педагогічним мистецтвом:
  - a. Розвиток навичок планування та проведення занять.
  - b. Вивчення методик викладання.
  - c. Вдосконалення навичок роботи з різними категоріями студентів.
2. Набуття Інтерактивних Методів Роботи:
  - a. Оволодіння та використання на практиці інтерактивних методів навчання.
3. Розвиток здатності до самоаналізу та вдосконалення навиків:
  - a. Аналіз власного викладацького досвіду.
  - b. Впровадження власних корекційних заходів.
4. Розвиток професійної самосвідомості:
  - a. Самооцінка та визначення напрямків власного професійного розвитку.
  - b. Усвідомлення власного педагогічного стилю.
5. Розвиток навиків міжособистісного спілкування:
  - a. Вдосконалення навиків комунікації зі студентами.
  - b. Розвиток навиків роботи у колективі та управління конфліктами.
6. Вивчення нормативно-правової бази освіти та методів її впровадження.

### *ЗАВДАННЯ ВИКЛАДАЦЬКОЇ ПРАКТИКИ*

1. Планування та проведення занять:
  - a. Розробка плану лекційних та практичних занять.
  - b. Підготовка до та проведення лекційних та практичних занять.
2. Використання інтерактивних методів:
  - a. Створити інтерактивні матеріали для використання на заняттях.
3. Аналізувати власний викладацький досвід та фіксувати можливі покращення.
4. Впроваджувати корекції на основі аналізу результатів занять.

5. Здійснювати рефлексію щодо власного педагогічного стилю та працювати над його удосконаленням.
6. Ознайомитись з та слідувати елементам нормативно-правової бази освіти.

## Лекційне заняття

### НАВЧАЛЬНА ДИСЦИПЛІНА

Навчальна дисципліна «Управління якістю».

#### ТЕМА

Темою заняття є «Управління якістю в ІТ проектах: критерії оцінки та метрики».

#### МЕТА

Ознайомлення слухачів із сутністю та важливістю управління якістю в ІТ проектах, розкриття критеріїв оцінки та метрик, які допомагають забезпечити ефективний контроль та підвищити якість проектів у сфері інформаційних технологій.

#### ЗАВДАННЯ

1. Ознайомлення зі змістом лекційного заняття.
2. Огляд ключових компонентів управління якістю програмного забезпечення.
3. Розкриття важливості управління якістю.
4. Розкриття важливості залучення різних проектних ролей до процесів управління якістю.
5. Розкриття перспектив різних проектних ролей на якість ІТ проекту.
6. Огляд критеріїв управління якістю з перспективи різних проектних ролей, що є важливими для слідування процесам управління якістю.
7. Приклади використання метрик якості для покращення процесів розробки та управління проектами на прикладі програми проектів.
8. Взаємодія з учасниками:
  - a. Створення можливостей для питань та обговорення важливих аспектів управління якістю в ІТ проектах.
  - b. Заохочення учасників поділитися власними досвідом та викликами у цій області.



## *ПЛАН ЗАНЯТТЯ*

<b>Розклад</b>	<b>Аудиторія</b>	<b>Тривалість</b>	<b>Тема</b>	<b>Деталі теми</b>
Частина 1	Уся	5 хв	Вступ. Представлення. Зміст.	<ol style="list-style-type: none"> <li>1. Привітання та введення в тему.</li> <li>2. Пояснення мети та важливості управління якістю в ІТ проектах.</li> <li>3. Огляд змісту та структури лекції.</li> </ol>
Частина 2	Уся	35 хвилин	Теоретичні аспекти управління якістю в ІТ проектах	<ol style="list-style-type: none"> <li>1. Ключові компоненти управління якістю програмного забезпечення.</li> <li>2. Проектні ролі через призму управління проектами.</li> <li>3. Метрики vs. ключові показники ефективності (KPI).</li> </ol>
Частина 3	Уся	30 хвилин	Практичний приклад	Огляд застосування метрик та KPI управління якістю на прикладі пропонуваного ІТ проєкту.
Частина 4	Усі	10 хвилин	Висновки та обговорення у форматі «питання- відповідь»	<ol style="list-style-type: none"> <li>1. Підсумок основних пунктів лекції та обговорення ключових ідей.</li> <li>2. Відповіді на питання студентів та обговорення додаткових тем.</li> </ol>

## Лекційне заняття: конспект

### *Вступ*

1. Представлення лектора, представлення теми лекції, вступ.
2. Огляд змісту, пояснення суті кожного пункту поточної лекції.

### *КЛЮЧОВІ КОМПОНЕНТИ УПРАВЛІННЯ ЯКІСТЮ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.*

1. Управління якістю програмного забезпечення (УЯПЗ) - це комплекс дій і процесів, спрямованих на забезпечення того, що програмні продукти відповідають визначеним вимогам та мають високу якість. УЯПЗ включає систематичне управління процесами розробки та тестування програм, щоб гарантувати, що кінцевий продукт задовольняє очікування клієнта і відповідає відповідним стандартам. Ось деякі ключові компоненти управління якістю програмного забезпечення:

- a. Планування якості:

- i. Встановлення якісних цілей та стандартів для проекту.
- ii. Визначення процесів, інструментів та технік, які будуть використовуватися для забезпечення якості.

- b. Забезпечення якості:

- i. Впровадження систематичних заходів для забезпечення того, що процеси, використовувані для розробки та постачання програм, є ефективними та ефективними.
- ii. Проведення регулярних аудитів та оглядів для перевірки відповідності встановленим процесам та стандартам.
- iii. Визначення та впровадження поліпшень в процеси розробки програм.

- c. Контроль якості:

- i. Моніторинг та оцінка фактичного продукту для забезпечення відповідності визначеним вимогам.

- ii. Проведення тестувальних заходів для виявлення та виправлення дефектів в програмному забезпеченні.
    - iii. Впровадження коригувальних заходів при виявленні дефектів.
  - d. Управління процесами:
    - i. Визначення та управління процесами розробки програм для забезпечення їхньої послідовності та повторюваності.
    - ii. Постійне вдосконалення процесів на основі зворотного зв'язку та метрик продуктивності.
  - e. Управління конфігурацією:
    - i. Управління змінами в програмному забезпеченні та пов'язаній документації протягом усього життєвого циклу розробки.
    - ii. Контроль та відстеження змін для забезпечення цілісності та стабільності програмного продукту.
  - f. Метрики та вимірювання:
    - i. Визначення та збір метрик для оцінки ефективності процесів розробки програм.
    - ii. Використання метрик для визначення областей для поліпшення та відстеження прогресу з часом.
  - g. Навчання та освіта:
    - i. Надання програм навчання для підвищення навичок і знань команди розробки.
    - ii. Забезпечення того, що члени команди знайомі з встановленими процесами та стандартами якості.
  - h. Управління ризиками:
    - i. Визначення потенційних ризиків для якості програмного продукту та розробка стратегій для їхнього зменшення чи управління.

- ii. Моніторинг та оновлення оцінок ризиків протягом усього життєвого циклу розробки програм.
- i. Документація:
  - i. Створення та підтримка документації, яка описує процеси розробки програм, стандарти та дії забезпечення якості.
  - j. Постійне вдосконалення:
    - i. Встановлення культури постійного вдосконалення, де вивчені уроки з попередніх проектів використовуються для поліпшення майбутніх проектів.
    - ii. Регулярне оглядання та оновлення процесів управління якістю на основі зворотного зв'язку та змін в середовищі розробки.

Інтеграцією цих компонентів у життєвий цикл розробки програм організації можуть створити комплексну систему управління якістю програмного забезпечення для поставки високоякісних програмних продуктів.

### ***ПРОЕКТНІ РОЛІ ЧЕРЕЗ ПРИЗМУ УПРАВЛІННЯ ПРОЕКТАМИ***

Представлення ролей проектного менеджера, бізнес-аналітика, аналітика з якості (тестувальника), архітектора/технічного ліда, технічного комунікатора та відділів підтримки ведення бізнесу. Концептуальне роз'яснення їхніх ролей в контексті управління якістю.

Огляд ключових показників управління якістю, які є важливими для аналітика з якості (тестувальника). Пояснення суті наступних показників:

1. Показник виправлення дефектів (Defect Kill Rate).
2. Час виправлення дефектів першого пріоритету (Defect Resolution Time (P1))
3. Прогноз виправлення дефектів, що наявні у беклозі (Defect Backlog Resolution Forecast)

#### 4. Показник ефективності ідентифікації дефектів (Defect Detection Efficiency)

Огляд ключових показників управління якістю, які є важливими для бізнес-аналітика. Пояснення суті різниці між метриками на ключовими показниками ефективності (KPI) на прикладах.

Заглиблення в пропоновані метрики для вимірювання якості бізнес-аналітиком. Пояснення суті наступних показників:

1. Показник повноти вимог (Requirements completeness) – вказує на якість пропрацьованого беклогу і готовність до старту наступної ітерації.
2. Показник повноти та готовності переліку вимог для наступних ітерації розробки (Sprint Scope Readiness Completeness) – дозволяє зрозуміти наскільки вимоги пропрацьовані наперед та чи зможе команда працювати за умови, що особа, яка відповідальна за специфікацію вимог випадає з робочого процесу.
3. Показник наявності оцінок для робіт, що передбачені в ітерації розробки (Sprint Estimates Completeness Variance) – вказує наскільки вимоги, які плануються впроваджувати в майбутньому є пропрацьовані та оцінені з точки зору затрат часу.
4. Показник наявності оцінок для робіт, що передбачені в релізі розробки (Release Estimates Completeness Variance) – вказує наскільки вимоги, які плануються впроваджувати в рамках релізу є пропрацьовані та оцінені з точки зору затрат часу
5. Об'єм робіт, заблокованих зовнішніми факторами (Release Readiness Variance) – показник рівня вимог, які не можуть бути впроваджені через зовнішні фактори, на які команда не має впливу.
6. Показник «завеликих» вимог (Scope Granularity Variance) – даний показник дозволяє зрозуміти відсоток вимог, які ж не достатньо гранулярними, щоб їх можна було взяти в роботу.
7. Показник чистоти беклогу (User Story Release Allocation) – дозволяє зрозуміти кількість вимог, які не пов'язані з будь-яким релізом.

8. Простій команди через відсутність вимог (Team Delays due to Requirements) – показник дозволяє зрозуміти розмір непродуктивно витраченого часу через недостатнє пропрацювання вимог.

Огляд метрик, які пропонуються технічним лідам/архітекторам для визначення розуміння технічної якості програмного забезпечення.

На слайді представлені наступні приклади обов'язкових та опціональних метрик.

Обов'язкові:

1. Рейтинг надійності створеного програмного забезпечення (Reliability Rating)
2. Оцінка безпеки (Security Rating)
3. Рівень зручності обслуговування (Maintainability Rating)
4. Рівень покриття юніт тестами (Coverage)
5. Рівень дублюваності коду (Duplication)

Опціональні:

1. Цикломатична складність (Cyclomatic Complexity)
2. Когнітивна складність (Cognitive Complexity)
3. Технічний борг (Technical Debt)

Рейтинг надійності створеного програмного забезпечення (Reliability Rating) – дана метрика дозволяє зрозуміти співвідношення між розміром створеного коду та кількістю дефектів, що припадають на цей код. Використовується для розуміння якості написання коду.

Оцінка безпеки (Security Rating) – це числове або рейтингове значення, що відображає рівень безпеки програмного продукту або системи. Ця оцінка може враховувати різні аспекти безпеки, такі як наявність вразливостей, якість реалізації механізмів захисту, ефективність заходів забезпечення безпеки, інші фактори, що впливають на стійкість до атак і конфіденційність даних. «Світлофор» вказує на рівень якості даної метрики.

«Maintenance Rating» або «Maintainability Rating» – це метрика, що визначає рівень зручності та вартості обслуговування програмного продукту

протягом його життєвого циклу. Основні аспекти цієї метрики можуть включати:

1. Читабельність коду: Оцінка легкості зрозуміння та редагування вихідного коду продукту.
2. Модульність: Розбиття програми на модулі або компоненти для полегшення розширення та змін.
3. Документація: Якість та повнота документації, яка допомагає розробникам розуміти і підтримувати систему.
4. Тестованість: Наявність та якість тестових наборів, що допомагають виявляти помилки під час розробки та змін.
5. Легкість виправлення помилок: Здатність легко виправляти помилки без додаткових проблем або побічних ефектів.

«Unit Test Coverage» – це метрика, яка визначає співвідношення між кількістю коду програми, що покрита юніт-тестами, і загальною кількістю коду в програмі. Вона вказує на те, яка частина програмного коду пройшла тестування на рівні юніт-тестів, які перевіряють коректність роботи окремих компонентів або модулів програми.

Ця метрика допомагає визначити рівень впевненості в тому, що різні частини програми ведуть себе так, як очікується, і дозволяє виявити ті частини коду, які можуть залишитися непокритими тестами. Зазвичай виражається у відсотках, наприклад, 80% unit test coverage вказує на те, що тести покривають 80% коду програми. Висока unit test coverage може свідчити про високу якість тестування та допомагати в попередженні виникнення помилок під час розробки.

«Code Duplication» (рівень дублювання коду) – це метрика, яка визначає кількість та обсяг повторюваного коду в програмному продукті. Вона вказує на те, скільки фрагментів коду в програмі ідентичні або мають схожу структуру. Велика кількість дубльованого коду може призводити до проблем у розробці та обслуговуванні програми, таких як ускладнення внесення змін,

збільшення ймовірності помилок та важкість виявлення та виправлення дефектів.

Метрика дублювання коду зазвичай виражається у відсотках або конкретних числах, вказуючи на рівень дублювання в кодовій базі. Керування дублюванням коду може сприяти поліпшенню читабельності, зменшенню зусиль для внесення змін та підтриманню програми.

Технічний борг (Technical Debt) – це метрика, яка вказує на суму компромісів у якості програмного коду, зроблених на ранніх етапах розробки програмного продукту на користь швидкості чи інших короткострокових вигід. Технічний борг може виникати в результаті вибору швидшого, але менш якісного рішення або недоліків у дизайні, які будуть важко виправляти в майбутньому.

Визначення технічного боргу включає в себе розгляд необхідності подальшого вдосконалення та оптимізації коду для покращення якості та стабільності продукту. Аналіз технічного боргу може допомагати команді розробників приймати рішення щодо вкладення ресурсів у виправлення дефектів, оптимізацію та покращення архітектури.

### ***МЕТРИКИ VS. КЛЮЧОВІ ПОКАЗНИКИ ЕФЕКТИВНОСТІ (KPI)***

Метрики та ключові показники ефективності (KPI) використовуються для вимірювання та оцінки результатів бізнесу, проекту або окремої особи, але вони служать трошки різним цілям.

Метрика - це кількісний показник, який використовується для відстеження та оцінки стану конкретного процесу чи діяльності. Метрики можна застосовувати до різних аспектів бізнесу, таких як продажі, маркетинг, обслуговування клієнтів та операції. Це, по суті, точки даних чи вимірювання, які надають інформацію щодо певного аспекту ефективності.

Приклади метрик:

1. Кількість відвідувань веб-сайту: Ця метрика може служити для оцінки результатів інтернет-маркетингової кампанії.



2. Оцінка задоволення клієнтів: Ця метрика може вимірювати ступінь задоволення клієнтів продуктом чи послугою.
3. Вартість приведення ліда: Ця метрика може допомогти оцінити ефективність маркетингової кампанії у генерації потенційних клієнтів.

Ключові показники ефективності (КРІ) – це конкретна підмножина метрик, які визначені як критичні для успіху організації чи конкретного проекту. КРІ - це високорівневі, стратегічні показники, які надають інформацію щодо загальної ефективності організації у досягненні своїх цілей та завдань. КРІ вибираються на основі їхньої прямої відносності до стратегічних пріоритетів організації.

Приклади КРІ:

1. Темп зростання доходів: Цей КРІ може вказувати на те, наскільки добре компанія впоралася з генерацією доходу з часом.
2. Життєвий цикл клієнта: Цей КРІ може надавати відомості про довгострокову цінність клієнта для бізнесу.
3. Чистий показник рекомендацій (NPS): Цей КРІ вимірює ймовірність того, що клієнти порекомендують продукти чи послуги компанії іншим.

Узагальнюючи, хоча всі КРІ є метриками, не всі метрики є КРІ. Метрики - це сирий матеріал у вигляді точок даних чи вимірювань, тоді як КРІ - це конкретні метрики, які визначені як важливі для успіху організації чи проекту. КРІ часто використовуються на вищому рівні для відстеження загальної ефективності та прогресу до стратегічних цілей.

### ***ОГЛЯД ЗАСТОСУВАННЯ МЕТРИК ТА КРІ УПРАВЛІННЯ ЯКІСТЮ НА ПРИКЛАДІ ПРОПОНОВАНОГО ІТ ПРОЕКТУ***

1. Перелік КРІв, що використовуються у пропонованому прикладі.
2. Наголос на «світлофорі», який використовується для розуміння якості як результату обрахунку певного КРІ.
3. Наголос на періоді оновлення метрики.

4. Пояснення щодо «тегів», які використовуються для групування КРІв за певними критеріями.

Огляд КРІ «Прогноз закриття усіх дефектів беклогу» (Defect Backlog Resolution Forecast), його порогових значень (помаранчева та червона зони). КРІ дозволяє зрозуміти кількість ітерацій, що є необхідними для закриття усіх «відкритих» дефектів, беручи до уваги скільки в середньому дефектів закривається командою за 1 ітерацію часу.

Графік КРІ «Прогноз закриття усіх дефектів беклогу» відображає порівняння кількості дефектів знайдених в ітерації та кількості закритих дефектів в рамках тієї ж ітерації протягом останніх 3-х ітерацій.

Огляд КРІ «Ефективність ідентифікації дефекту» (Defect detection efficiency), його порогових значень (помаранчева та червона зони). КРІ дозволяє виміряти % дефектів, які ідентифікуються на різних середовищах розробки при перенесенні коду з одного середовища на інше.

Графік КРІ «Ефективність ідентифікації дефекту» дозволяє зрозуміти КРІ у вигляді графіку.

Огляд КРІ «Ефективність виправлення дефекту» (Defect kill rate), його порогових значень (помаранчева та червона зони). КРІ дозволяє виміряти співвідношення «закритих» до «ідентифікованих» дефектів за ітерацію часу.

Графік КРІ «Ефективність виправлення дефекту» відображає порівняння кількості дефектів знайдених в ітерації та кількості закритих дефектів в рамках тієї ж ітерації протягом останніх 3-х ітерацій, при цьому подаючи лінію тренду за 3 ітерації та порогові значення.

Огляд КРІ «Період виправлення дефектів 1го пріоритету» (Defect resolution time (P1)), його порогових значень (помаранчева та червона зони). КРІ дозволяє зрозуміти затрати часу командою в «днях» на виправлення критичних дефектів для клієнта.

Графік КРІ «Період виправлення дефектів 1го пріоритету» дозволяє зрозуміти час, який було затрачено в рамках ітерації на виправлення дефектів 1го пріоритету протягом останніх 3-х ітерацій.

Огляд KPI «Критична точка досягнення проміжної цілі» (Forecasted Milestone Completion), його порогових значень (помаранчева та червона зони). KPI дозволяє зрозуміти наскільки близько команда знаходиться від критичної часової межі по досягненню проміжної цілі.

Графік KPI «Критична точка досягнення проміжної цілі» візуально показує наскільки далеко від сьогодні знаходиться критична точка.

Огляд KPI «Покриття вимог тест-кейсами» (Requirements Test Coverage), його порогових значень (помаранчева та червона зони). KPI дозволяє зрозуміти який відсоток вимог покрито тест-кейсами.

Графік KPI «Покриття вимог тест-кейсами» дозволяє зрозуміти відсоток вимог, які покрито тест-кейсами протягом останніх 3 ітерацій та порівняти значення між собою.

Огляд KPI «Якість програмного забезпечення» (Software Quality), його порогових значень (помаранчева та червона зони). KPI акумулює значення декількох різних метрик та в залежності від їх значень дозволяє зрозуміти загальний рівень якості програмного забезпечення.

Огляд KPI «Рівень виконання зобов'язань за ітерацію часу» (Sprint Commitment Success), його порогових значень (помаранчева та червона зони). KPI дозволяє зрозуміти який % зобов'язань було виконано командою за ітерацію часу.

Графік KPI «Рівень виконання зобов'язань за ітерацію часу» дозволяє порівняти рівень зобов'язань та рівень виконання цих зобов'язань протягом ітерації часу, побачити тенденцію за 3 останні ітерації, середні значення тощо.

Огляд KPI «Рівень покриття коду автоматичними тестами» (Test Automation Coverage), його порогових значень (помаранчева та червона зони). KPI вимірює якість коду та показує яка його частина покрита автоматичними тестами.

Графік КРІ «Рівень покриття коду автоматичними тестами» дозволяє побачити % покриття коду автоматичними тестами протягом останніх 3 ітерацій.

Огляд КРІ «Рівень покриття коду юніт тестами» (Unit Test Coverage), його порогових значень (помаранчева та червона зони). КРІ вимірює якість коду та показує яка його частина покрита юніт тестами.

Графік КРІ «Рівень покриття коду юніт тестами» дозволяє побачити % покриття коду юніт тестами протягом останніх 3 ітерацій.

### *ПІДСУМОК ОСНОВНИХ ПУНКТИВ ЛЕКЦІЇ ТА ОБГОВОРЕННЯ КЛЮЧОВИХ ІДЕЙ*

Ключові аспекти Управління Якістю: Управління якістю програмного забезпечення є комплексним процесом, що включає планування, забезпечення якості, контроль якості та постійне вдосконалення.

Роль Планування: Ефективне планування визначає якісні цілі проекту та встановлює процеси, необхідні для досягнення цих цілей.

Забезпечення Якості і Контроль Якості: Забезпечення якості спрямоване на забезпечення ефективності процесів, тоді як контроль якості фокусується на моніторингу та виправленні дефектів у програмному продукті.

Управління Процесами: Управління процесами розробки програм допомагає забезпечити послідовність та повторюваність, а також визначити області для покращення.

Метрики та Вимірювання: Використання метрик і вимірювань дозволяє об'єктивно оцінювати ефективність процесів та визначати області для оптимізації.

Культура Постійного Вдосконалення: Впровадження культури постійного вдосконалення дозволяє організації вивчати уроки з попередніх проектів та постійно удосконалювати свої процеси.

Управління Ризиками: Ефективне управління ризиками допомагає визначати та зменшувати потенційні загрози для якості програмного продукту.

Роль Навчання та Освіти: Навчання та освіта є ключовими елементами для забезпечення того, що члени команди розробки розуміють та дотримуються встановлених процесів та стандартів якості.

Загальна мета управління якістю програмного забезпечення полягає в створенні продуктів, які відповідають вимогам замовника, є ефективними та надійними. Послідовна реалізація цих принципів допомагає забезпечити успішне впровадження та підтримку програмних продуктів в індустрії.

### *САМОАНАЛІЗ*

Привітався зі студентами, представився, розказав про досвід у сфері, про яку розказував. Ознайомив студентів зі змістом лекційного заняття. Озвучив мету та завдання лекційного заняття. Згідно змісту, слідував плану лекційного заняття, знайомлячи студентів з теоретичним матеріалом. Надав практичні приклади для теоретичного матеріалу, про який розказував. Намагався відповісти на усі питання, які виникали в процесі лекційного заняття. В кінці лекції відвів час для відповідей на запитання, які сформувались у студентів в результаті прослуханої лекції. Лекція пройшла успішно, слідував графіку лекції відповідно до плану та робочої програми дисципліни.

### *ВИСНОВОК*

Усі цілі лекційного заняття досягнуто. Завдання, що планувалось реалізувати в рамках лекційного заняття повністю виконані. Студенти проявили зацікавленість в темі, хоча відчувалось, що їхні знання в сфері інформаційних технологій, яка лежала в основі лекційного заняття були обмеженими. В рамках лекційного матеріалу, студентів також було ознайомлено зі звичною термінологією для сфери інформаційних технологій,

було зроблено наголос на основних категоріях, що впливають на тему лекції. Було наголошено на важливості застосування методів та механізмів управління якістю при впровадженні проектів у сфері інформаційних технологій, наголошено на перевагах, до яких це може призвести та ризиках, що наступають за умови відсутності таких практик на підприємствах.

## Практичне заняття

### НАВЧАЛЬНА ДИСЦИПЛІНА

Навчальна дисципліна «Управління якістю».

#### ТЕМА

Темою заняття є «Управління якістю в ІТ проектах: критерії оцінки та метрики».

#### МЕТА

Перевірити засвоєння знань студентами за результатами проведеного лекційного заняття.

#### ЗАВДАННЯ

1. Ознайомлення зі змістом практичного заняття.
2. Перевірка засвоєння студентами матеріалу, який обговорювався на лекційному завданні.
3. Обговорення відкритих запитань, які виникли в студентів за результатами проведеного лекційного заняття та самостійного опрацювання теми лекції.
4. Проходження тестування за засвоєним матеріалом.

#### ПЛАН ЗАНЯТТЯ

Розклад	Аудиторія	Тривалість	Тема	Деталі теми
Частина 1	Уся	40 хв	Питання для перевірки засвоєння лекційного матеріалу	Перелік питань для перевірки надається нижче.
Частина 2	Уся	20 хв	Перегляд відеоматеріалу	Перегляд відеоматеріалу та його обговорення
Частина 3	Уся	20 хв	Тести	Перевірка засвоєння лекційного матеріалу методом тестування.

#### ПИТАННЯ ДЛЯ ПЕРЕВІРКИ ЗАСВОЄННЯ ЛЕКЦІЙНОГО МАТЕРІАЛУ

1. Що входить до поняття «критерії оцінки» у контексті управління якістю в ІТ проектах?

2. Наведіть приклади загальних критеріїв успішності проекту. Як вони можуть відрізнятися для ІТ проектів порівняно з іншими видами проектів?
3. Які основні аспекти ІТ проектів можуть бути вимірювані за допомогою метрик якості?
4. Які ключові показники ефективності (KPI) можна використовувати для вимірювання успіху ІТ проекту?
5. Як метрики якості допомагають в процесі управління якістю програмного забезпечення на різних етапах його життєвого циклу?
6. Як вибрати правильні метрики для конкретного ІТ проекту?
7. Які виклики можуть виникнути при визначенні метрик якості для гнучких (Agile) проектів? Як їх можна вирішити?
8. Як можна використовувати метрики для постійного покращення якості в ІТ проектах?

### ***ВІДЕОМАТЕРІАЛ***

Як створити план управління якістю на проекті (How to Create a Project Quality Management Plan) [YouTube](#) відео. Дане відео переглядається разом зі студентами під час практичного заняття, обговорюються особливості створення такого плану.

### ***ТЕСТИ***

1. Яка основна мета використання критеріїв оцінки в управлінні якістю в ІТ проектах?
  - a. Забезпечення безпеки проекту
  - b. Визначення термінів виконання проекту
  - c. **Оцінка та контроль якості робіт**
  - d. Максимізація прибутку від проекту
2. Що визначають ключові показники ефективності (KPI) в контексті ІТ проектів?
  - a. Дозволяють спробувати покористуватись продуктом



- b. **Важливі аспекти проекту, які визначають його успішність**
  - c. Кількість розробників у команді
  - d. Час, витрачений на вирішення проблем
3. Які зазначені нижче метрики відносяться до процесу розробки програмного забезпечення?
- a. Кількість виявлених помилок під час тестування
  - b. Частота виходження нових версій продукту
  - c. **Кількість задач, виконаних за день розробниками**
  - d. Участь команди у тренінгах і підвищенні кваліфікації
4. Що визначає поняття «системне мислення» у контексті управління якістю в ІТ проектах?
- a. Зосередження на окремих елементах проекту
  - b. **Розгляд проекту як інтегрованої системи**
  - c. Максимізація індивідуальних досягнень
  - d. Використання принципів Agile методології
5. Які виклики можуть виникнути при визначенні метрик якості для гнучких (Agile) проектів?
- a. Відсутність необхідності у використанні метрик у гнучких проектах
  - b. **Суперечливість між цілями команди та вимогами замовника**
  - c. Стабільність усіх параметрів проекту протягом всього життєвого циклу
  - d. Застосування традиційних метрик без будь-яких змін
6. Що включає в себе поняття «метрики якості» в контексті управління якістю в ІТ проектах?
- a. **Сукупність кількісних показників, які вимірюють різні аспекти якості програмного забезпечення**
  - b. Всі документаційні вимоги проекту
  - c. Інструкції з тестування продукту
  - d. Список задач для розробників

7. Яке питання допомагає визначити правильність обраної метрики для конкретного ІТ проекту?
- a. «Яка є цільова аудиторія проекту?»
  - b. «Скільки грошей буде витрачено на проект?»
  - c. «Хто є керівником проекту?»
  - d. «Які конкретні завдання вирішує ця метрика та як вона відображає цілі проекту?»**
8. Які метрики можуть бути використані для оцінки ефективності тестування в ІТ проекті?
- a. Кількість виявлених помилок
  - b. Кількість витраченого часу на тестування
  - c. Спроби виявлення помилок, які не призвели до виправлення
  - d. Усі відповіді вірні**
9. Що розуміється під терміном «запити на зміни» у контексті управління якістю в ІТ проектах?
- a. Зміни в керівництві проектом
  - b. Зміни у вимогах до проекту, які виникають під час його реалізації**
  - c. Зміни в команді розробників
  - d. Зміни в графіку проекту
10. Як визначити успішність ІТ проекту за допомогою ключових показників ефективності (KPI)?
- a. За кількістю виявлених помилок
  - b. За відповідністю проекту графіку виконання та бюджету**
  - c. За тривалістю етапів розробки
  - d. За змінами у складі команди розробників

### *РЕКОМЕНДОВАНА ЛІТЕРАТУРА ДЛЯ САМОСТІЙНОГО НАВЧАННЯ*

1. «Agile Estimating and Planning» by Mike Cohn. Ця книга розглядає аспекти оцінювання та планування в гнучких (Agile) проектах, включаючи критерії оцінки та метрики.
2. «Quality Software Management: Systems Thinking» by Gerald M. Weinberg. Автор досліджує системне мислення та як воно може бути застосоване до управління якістю в ІТ проектах.
3. «Software Engineering: A Practitioner's Approach» by Roger S. Pressman. Розділ книги, присвячений якості програмного забезпечення, містить корисні інформацію про метрики та оцінювання.
4. «Measuring Software Quality and Testing» by Charles W. Rich. Книга детально розглядає метрики, пов'язані з якістю програмного забезпечення та процесом тестування.

### *РЕКОМЕНДОВАНІ ВІДЕОМАТЕРІАЛИ*

1. «Learn agile estimation in 10 minutes» ([YouTube](#)). Коротке відео, яке введе у поняття оцінювання та планування в Agile проектах.
2. «The Complete 2023 Software Testing Bootcamp» ([Udemy](#)). Курс, який охоплює основи тестування та забезпечення якості в програмному забезпеченні.
3. «The difference between Metrics, KPIs & Key Results» ([YouTube](#)). Відео, яке розкриває сутність та застосування метрик, ключових показників ефективності (КРІ) та ключових результатів та пояснює різницю між ними.
4. «Quality Management Systems, Tools and Strategies» ([Udemy](#)). Онлайн-курс, який орієнтований на студентів, які хочуть освоїти дисципліну в цілому, не шукаючи різну інформацію в різних місцях.

## Список використаних джерел

1. Скрипник М. І. Професія – викладач: практикум із дидактики вищої школи: навчально-методичний посібник. 2018.
2. Зайченко І.В., Теслюк В.М., Каленський А.Н. Основи педагогічної майстерності та етика викладача вищої школи : підручник. За ред. проф. І. В. Зайченка. Київ : Видавництво Ліра-К, 2017. 484 с.
3. Нормативна база освітнього процесу в ІФНТУНГ  
<https://nung.edu.ua/department/navchalnyy-viddil/05-polozhennya>.

## Додатки

Додаток 1

Мультимедійна презентація, що використовувалась на лекції ([посилання](#))