

# **СУЧАСНІ ЗАСОБИ ІНФОРМАТИКИ ТА КОМП'ЮТЕРНИЙ РИНОК**

**121 «Інженерія програмного  
забезпечення»**

**ІПЗм-21**

**2021 / 2021 навчальний рік**

# МЕТРИКИ / METRICS (1)

# МЕТРИКИ ПРОДКУТУ, КОМПАНІЙ, РИКУ

- Визначення метрик
- Побудова метрик
- Метрики програмного продукту (soft)

# МЕТРИКИ. ОСНОВНІ ВИЗНАЧЕННЯ

**Metric / метрика:** певний метод і шкала вимірювання.

**Класифікація:**

- **внутрішні (internal)** – метрики, що прямо або непрямо вимірюють власне **продукт**, або його характеристики.
- **зовнішні (external)** – метрики, що віддзеркалюють ступінь, в якій продукт задовольняє **встановленим потребам** при використанні в заданих умовах.

# МЕТРИКИ. ОСНОВНІ ВИЗНАЧЕННЯ

**Metric / метрика:** певний метод і шкала вимірювання.

**Класифікація:**

- **прямі (direct)** - міра атрибута (параметра), яка не залежить від міри будь-якого іншого атрибута.
- **непрямі (indirect)** - міра атрибута, яка отримана із вимірів одного або більшого числа інших атрибутів.

# МЕТРИКИ. ОСНОВНІ ВИЗНАЧЕННЯ

**Scale / шкала:** набір значень з певними властивостями. Типи шкал (кількісні):

**Інтервальна** - відповідає впорядкованій шкалою з рівновіддаленими розподілами; зазвичай містить чималу кількість поділів з кількісними значеннями (наприклад шкала з розподілами 0, 1, 2, ..., 10);

**Відносна** - відповідає впорядкованій шкалою з рівновіддаленими діленням, в відносних одиницях щодо деякої абсолютної величини (зазвичай в діапазоні від 0 до 1).

# МЕТРИКИ. ОСНОВНІ ВИЗНАЧЕННЯ

**Scale / шкала:** набір значень з певними властивостями. Типи шкал (якісні):

**Номінальна** - відповідає набору категорій; класифікує програми за ознакою наявності або відсутності деякого властивості без урахування градацій (наприклад, «так», «ні»);

**Впорядкована** - відповідає впорядкованого набору розподілів шкали; дозволяє рангувати властивості шляхом порівняння з опорними значеннями; має невелику кількість поділів (наприклад, чотири градації: «відмінно», «добре», «задовільно», «незадовільно»);

# МЕТРИКИ. ОСНОВНІ ВИЗНАЧЕННЯ

Теорія метрик включає також **методи** для **категоризації** якісних даних (даних, які не можна виміряти кількісно).

**МЕТРИКА** - це **якісний** або **кількісний** вимір, що дозволяє отримати чисельне значення деякої властивості *програмного продукту*, або його специфікацій.



# ЖИТТЯ ПРОГРАМНОГО ПРОДКТУ

**ПЛАНУВАННЯ  
ПРОЕКТУВАННЯ**

**ВИРОБНИЦТВО**  
- проектування  
- тестування

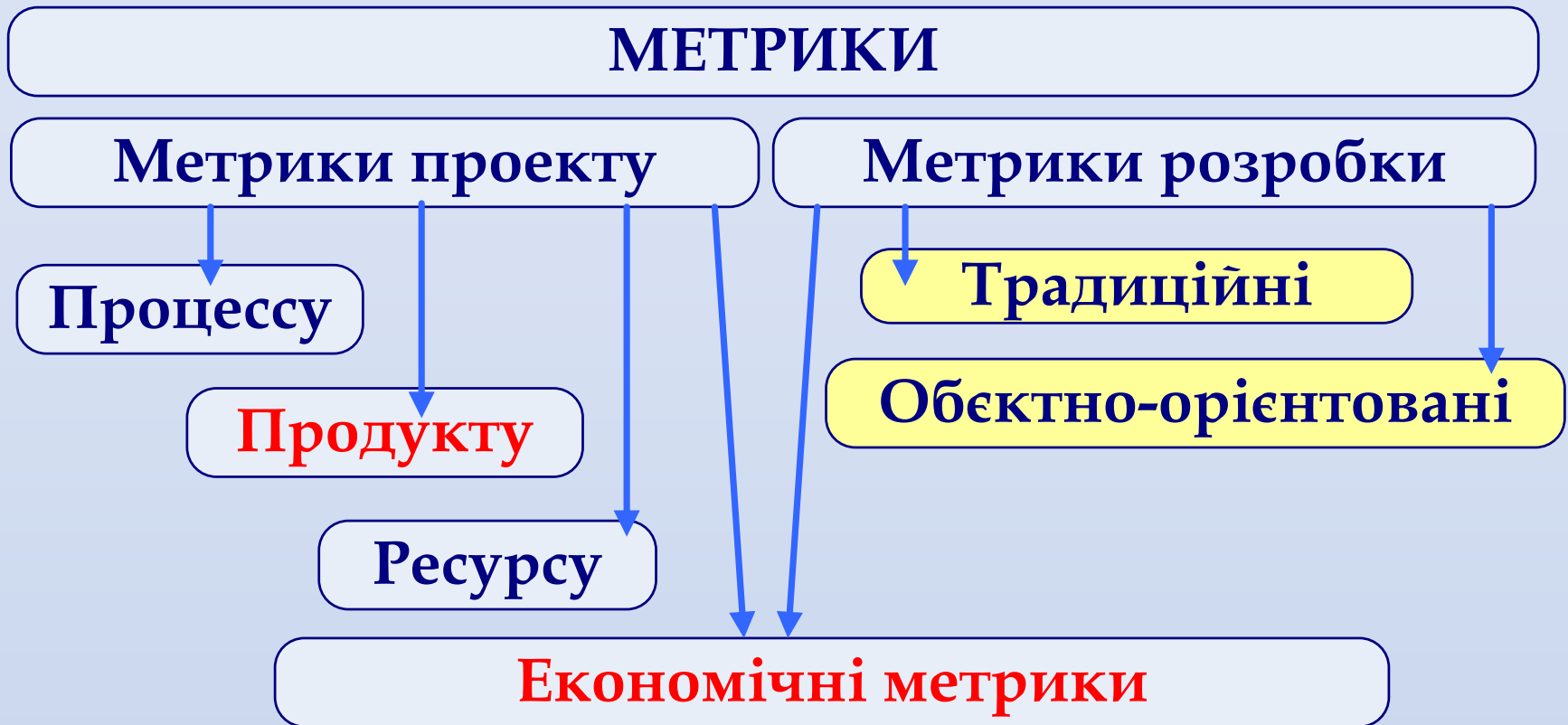
**ВПРОВАДЖЕННЯ**  
- розгортання  
- експлуатація  
- удосконалення

**МЕТРИКИ Програмного Продукту**

**БІЗНЕС МЕТРИКИ**

**ФІНАНСУВАННЯ / МОНЕТИЗАЦІЯ**

# МЕТРИКИ



!!! Більше 100 метрик відносно програмного продукту.

# МЕТРИКИ ПРОЕКТУ

Три категорії:

**Метрики продукту** - описує характеристики продукту, такі як **розмір**, **складність**, особливості дизайну, продуктивність і рівень якості.

**Метрики ресурсу** - описують характеристики і виконання проекту (число розробників програмного забезпечення, штатний розклад протягом життєвого циклу програмного забезпечення, вартість, графік і продуктивність).

**Метрики процесу** – характеристики, що використовуються для поліпшення діяльності по розробці, впровадження, супроводу програмного забезпечення.

# МЕТРИКИ СКЛАДНОСТІ ПРОГРАМНОГО ПРОДУКТУ

Існує більше 20 метрик відносно **складності** програмного продукту.

Приклади:

<https://www.youtube.com/watch?v=yd981JSD0tk>

Detailed report on module AuthenticationHandler

<https://blog.cles.jp/media/1/winstone-src-0.9.10-cccc/AuthenticationHandler.html>

Більш детально див. :

<https://habr.com/ru/company/intel/blog/106082/>

# МЕТРИКИ СКЛАДНОСТІ ПРОГРАМНОГО ПРОДУКТУ

## 1. МЕТРИКИ РОЗМІРУ (ОБСЯГУ)

- **ЛОС - обсяг програми (в тисячах рядках вихідного коду);**
- загальні трудовитрати (в людино-місяцях, людино-годинах);
- вартість розробки;
- обсяг документації;
- помилки, виявлені протягом року експлуатації;
- кількість людей, які працювали над виробом;
- термін розробки.

# МЕТРИКИ СКЛАДНОСТІ ПРГРАМНОГО ПРОДУКТУ

## LOC

- кількість «фізичних» рядків коду - SLOC (KLOC, KSLOC, DSLOC) - визначається як загальне число рядків вихідного коду, включаючи коментарі і порожні рядки (при вимірюванні показника на кількість порожніх рядків, як правило, вводиться обмеження - при підрахунку враховується число порожніх рядків, що не перевищує 25% загального числа рядків в вимірюваному блоці коду).

# МЕТРИКИ СКЛАДНОСТІ ПРОГРАМНОГО ПРОДУКТУ

## LOC

- кількість **«логічних»** рядків коду – SLOC (LSI, DSI, KDSI, «SI» - source instructions) - визначається як кількість команд і залежить від використовуваної мови програмування. У тому випадку, якщо мова не допускає розміщення кількох команд на одному рядку, то кількість «логічних» SLOC буде відповідати числу «фізичних», за винятком числа порожніх рядків і рядків коментарів. У тому випадку, якщо мова програмування підтримує розміщення кількох команд на одному рядку, то одна фізична рядок повинна бути врахована як кілька логічних, якщо вона містить більше однієї команди мови.

# МЕТРИКИ СКЛАДНОСТІ ПРОГРАМНОГО ПРОДУКТУ

**ЛОС** – недоліки:

- не враховує досвід співробітників і їх інші якості;
- процес вимірювання може бути спотворений за рахунок того, що співробітники знають про вимірювані показники і прагнуть оптимізувати ці показники, а не свою роботу;
- Неточність: немає метрик, які були б одночасно і значимі і досить точні. Кількість рядків коду - це просто кількість рядків, цей показник не дає уявлення про складність вирішуваної проблеми.

**метрика ЛОС не відображає трудомісткості по  
створенню програми.**



# МЕТРИКИ СКЛАДНОСТІ ПРОГРАМНОГО ПРОДУКТУ

## 2. МЕТРИКИ СКЛАДНОСТІ

### Об'єктно - орієнтовані метрики

- **WMC1 (Weighted Methods Per Class) /**  
Зважена насиченість класу 1 - відбиває відносну міру складності класу на основі цикломатическая (топологічна) складності кожного його методу. Клас з більш складними методами і великою кількістю методів вважається більш складним. При обчисленні метрики не враховуються батьківські класи.

# МЕТРИКИ СКЛАДНОСТІ ПРОГРАМНОГО ПРОДУКТУ

## 2. МЕТРИКИ СКЛАДНОСТІ

### Об'єктно - орієнтовані метрики

- **WMC2 (Weighted Methods Per Class) / Зважена насиченість класу 2** - міра складності класу, заснована на урахуванні того, що клас з великим числом методів, є більш складним, і що метод з великою кількістю параметрів також є більш складним. При обчисленні метрики батьківські класи не враховуються.

# МЕТРИКИ СКЛАДНОСТІ ПРОГРАМНОГО ПРОДУКТУ

## 2. МЕТРИКИ СКЛАДНОСТІ

### Об'єктно - орієнтовані метрики

- **DIT (Depth of inheritance tree) / Глибина дерева спадкування** - довжина найдовшого шляху успадкування, що закінчується на даному модулі. Чим глибше дерево успадкування модуля, тим може виявитися складніше передбачити його поведінку. З іншого боку, збільшення глибини дає більший потенціал повторного використання даним модулем поведінки, певного для класів-предків.

# МЕТРИКИ СКЛАДНОСТІ ПРОГРАМНОГО ПРОДУКТУ

## 2. МЕТРИКИ СКЛАДНОСТІ

### Об'єктно - орієнтовані метрики

- **Number of children / Кількість дітей** - число модулів, що безпосередньо успадковують даний модуль. Більше значення цієї метрики вказують на широкі можливості повторного використання; при цьому занадто велике значення може свідчити про погано обраної абстракції.

# МЕТРИКИ СКЛАДНОСТІ ПРОГРАМНОГО ПРОДУКТУ

## 2. МЕТРИКИ СКЛАДНОСТІ

### Об'єктно - орієнтовані метрики

- **Coupling between objects / Можливості підключення об'єктів** - кількість модулів, пов'язаних з даним модулем в ролі клієнта або постачальника. Надмірна зв'язність говорить про слабкість модульної інкапсуляції, може перешкоджати повторному використанню.
- **Response For Class / Відгук на клас** - кількість методів, які можуть викликатися екземплярами класу; обчислюється як сума кількості локальних методів, так і кількості вилучених.

# МЕТРИКИ СКЛАДНОСТІ ПРОГРАМНОГО ПРОДУКТУ

## 2. МЕТРИКИ СКЛАДНОСТІ

### Метрики Холстеда

- **NUOprtr (Number of Unique Operators)** - число унікальних операторів програми, включаючи символи-роздільники, імена процедур і знаки операцій (словник операторів);
- **NUOprnd (Number of Unique Operands)** - число унікальних операндів програми (словник операндів);
- **Noprtr (Number of Operators)** - загальне число операторів в програмі;
- **Noprnd (Number of Operands)** - загальне число операндів в програмі.

# МЕТРИКИ СКЛАДНОСТІ ПРОГРАМНОГО ПРОДУКТУ

## 2. МЕТРИКИ СКЛАДНОСТІ

Метрики Холстеда – розраховуються оцінки

- Словник програми (Halstead Program Vocabulary, HPVoc):  $HPVoc = NUOprtr + NUOprnd$ ;
- Довжина програми (Halstead Program Length, HPLen):  $HPLen = Noprtr + Noprnd$ ;
- Обсяг програми (Halstead Program Volume, HPVol):  $HPVol = HPLen \log_2 HPVoc$ ;
- Складність програми (Halstead Difficulty, HDiff):  $HDiff = (NUOprtr / 2) \times (NOprnd / NUOprnd)$ ;

Heff (Halstead Effort) оцінка зусиль програміста:

$$HEff = HDiff \times HPVol.$$

# МЕТРИКИ СКЛАДНОСТІ ПРОГРАМНОГО ПРОДУКТУ

## 2. МЕТРИКИ СКЛАДНОСТІ

Цикломатична складність за Мак-Кейбом - обчислюється на основі графа керуючої логіки програми (control flow graph), будується в вигляді орієнтованого графа, в якому обчислювальні оператори або вирази представляються у вигляді вузлів, а передача управління між вузлами - у вигляді дуг.

$$C = e - n + 2$$

$E$  - кількість дуг,  $N$  - кількість вузлів.

Метрика дозволяє: провести оцінку трудомісткості реалізації окремих елементів програмного проекту, скорегувати загальні показники оцінки тривалості і вартості проекту, оцінити пов'язані ризики і прийняти необхідні управлінські рішення.



# ЯКІСТЬ ПРОГРАМНОГО ПРОДУКТУ

## ХАРАКТЕРИСТИКИ ЯКОСТІ

**Functionality / Функціональність** - сукупність властивостей ПР, що визначається наявністю і конкретними особливостями набору функцій, здатних задовольняти задані або побічні потреби користувача.

**Reliability / Надійність** - сукупність властивостей, що характеризує здатність ПП зберігати заданий рівень придатності в заданих умовах напротязі заданого інтервалу часу.

# ЯКІСТЬ ПРОГРАМНОГО ПРОДУКТУ

## ХАРАКТЕРИСТИКИ ЯКОСТІ

**Usability / Практичність** - сукупність властивостей ПП, що характеризує зусилля, необхідні для його використання, і індивідуальну оцінку результатів його використання заданим або імовірною колом користувачів.

**Efficiency / Ефективність** - сукупність властивостей програмного засобу, що характеризує ті аспекти його рівня придатності, які пов'язані з характером і часом використання ресурсів, необхідних при заданих умовах функціонування.

# ЯКІСТЬ ПРОГРАМНОГО ПРОДУКТУ

## ХАРАКТЕРИСТИКИ ЯКОСТІ

**Maintainability / Супроводженість** - сукупність властивостей програмного засобу, що характеризує зусилля, які необхідні для його модифікації.

**Portability / Мобільність** - сукупність властивостей програмного засобу, що характеризує пристосованість для перенесення з одного середовища функціонування в інші.

# МЕТРИКИ ЯКОСТІ ПРОГРАМНОГО ПРОДУКТУ

## ВНУТРІШНІ МЕТРИКИ

**Portability / Переносимість** - кількість люд.-год., необхідний для перенесення програми з платформи X на платформу Y. Допустимий поріг: 1 люд.-год. на 1К рядків вихідного коду

**Testability / Тестуємість** - кількість люд.-год., необхідний для повного тестування 90% всіх модулів. Допустимий поріг 10 люд.-год. на 1К рядків вихідного коду.

# МЕТРИКИ ЯКОСТІ ПРОГРАМНОГО ПРОДУКТУ

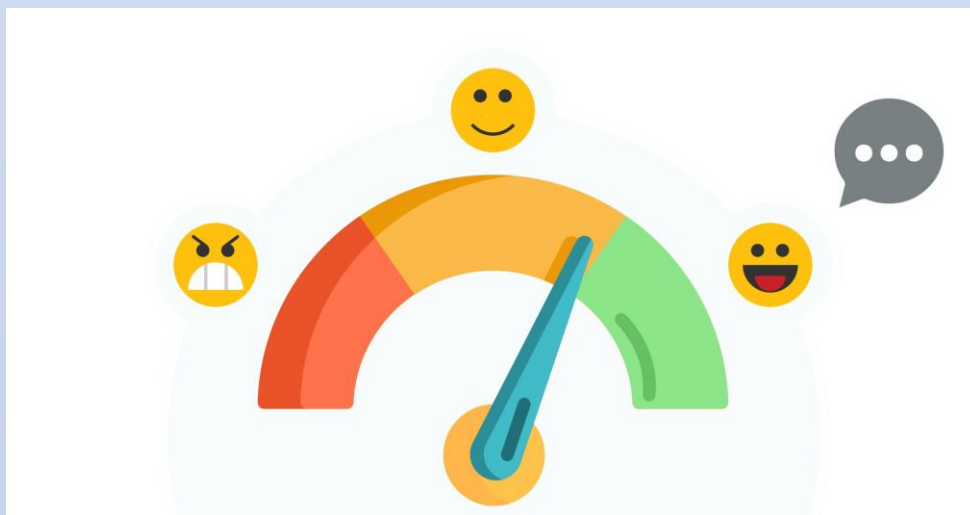
## ВНУТРІШНІ МЕТРИКИ

Reliability / Надійність - середній час  
напрацювання на відмову. Допустимий поріг:  
120 операційних днів.

# МЕТРИКИ ЯКОСТІ ПРОГРАМНОГО ПРОДУКТУ

## ЗОВНІШНІ МЕТРИКИ

**Customer satisfaction / Ступінь задоволення потреб замовника - показник, що визначає, наскільки задоволені споживачі продуктом, послугами та можливостями компанії.  
(опитування, рейтинги).**

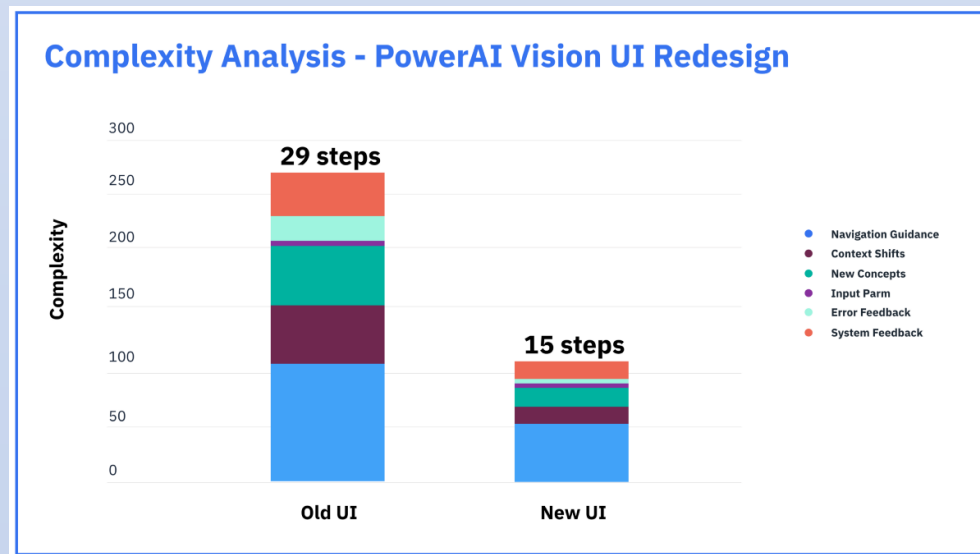


# МЕТРИКИ ЯКОСТІ ПРОГРАМНОГО ПРОДУКТУ

## ЗОВНІШНІ МЕТРИКИ

Interface complexity / Складність інтерфейсу – оцінка юзабіліті, що відображає наскільки інтерфейс простий для виконання конкретних завдань користувачем.

- Зміна контексту
- Навігація
- Вибір параметрів
- Зворотній зв'язок
- Реакція на помилки



# МЕТРИКИ ЯКОСТІ ПРОГРАМНОГО ПРОДУКТУ

## МЕТРИКИ ЯКОСТІ В ВИКОРИСТАННІ

**Результативність** – оцінює, чи досягає користувач заданих цілей з точністю і повнотою в заданому контексті використання.

**Продуктивність** - оцінює ресурси, які витрачають користувачі відповідно до досягнутої результативністю в заданому контексті використання.

**Безпека** - оцінюють рівень ризику заподіяння шкоди людям, бізнесу, програмного забезпечення, майну або навколишньому середовищу в заданому контексті використання



# Контрольні запитання

- Надайте визначення метрики. Визначте базові класи метрик.
- Наведіть визначення шкали та характеристики типів шкал для кількісних і якісних метрик.
- Наведіть категорії метрик програмного продукту.
- Охарактеризуйте метрику LOC.
- Охарактеризуйте метрики WMC1, WMC2.
- Охарактеризуйте метрику Холстеда.
- Надайте визначення цикломатичної складності за Мак-Кйбом.
- Надайте визначення характеристик якості ПП.
- Надайте визначення внутрішніх метрик якості ПП.
- Надайте визначення зовнішніх метрик якості ПП.
- Надайте визначення метрик якості в використанні ПП.

# **Рекомендована ЛІТЕРАТУРА**

- . ISO/IEC 9126–1–4:2001–2004**
- . ISO/IEC 14598–1:1999**
- . Меркулов А., Савохин К. Монетизация сайта. Секреты больших денег в Интернете. Питер, 2013.- 224 с.**

**The END**  
**Mod 1. Lec 3.**