

Практична робота 2

Метрики розміру. Метрика Lines of Code

Мета роботи: Ознайомитися с загальними поняттями щодо вимірювань та метрикою розміру Lines of Code. Напрацювати вміння застосування засобів вимірювання метрики. Отримати загальні вміння щодо застосування метрики в економіці програмного забезпечення.

Теоретичні відомості

Вимірювання це загально науковий, емпіричний метод пізнання, який визначається як процес, за допомогою якого числа або символи присвоюються атрибутам сутностей в реальному світі, таким чином, щоб описати їх чітко, відповідно до певних правил.

Сутність може бути об'єктом, наприклад, людина, або специфікація програмного забезпечення (ПЗ), або подія, наприклад, тестування програмного проекту.

Атрибут - властивість сутності, така як довжина або функціональність (в специфікації), вартість, або тривалість (етапу тестування).

Величина - іменований атрибут сутності. Атрибути іменують тому, що сутність звичайно має декілька атрибутів (властивостей).

Міра - кількісна одиниця для виміру деякого атрибута продукту або процесу, наприклад, лексема або стрічка для довжини програми, функціональна крапка для розміру, людино/місяць для продуктивності процесу.

Метрика – кількісний вираз ступеня, в якій система, компонент або процес має відповідний атрибут.

Наприклад, довжина програми в лексемах. Таким чином, довжина програми це властивість, лексема – міра, довжина програми в лексемах - метрика.

Метрика ПЗ - чітка міра, що дозволяє оцінити певні властивості конкретного учасника програмного коду. Для кожної метрики зазвичай існують її еталонні показники, що вказують, при яких крайніх значеннях варто звернути увагу на дану ділянку коду. Метрики коду розділяються на категорії та можуть оцінювати досконало різні аспекти програмної системи: складність та структурованість програмного коду, зв'язковість компонентів, відносний обсяг програмних компонентів та ін. Найбільш проста для розуміння метрика - кількість строк коду в програмній системі в сукупності з іншими метриками може слугувати для отримання формалізованих даних для оцінки коду. Наприклад, можна побудувати співвідношення між кількістю коду класу та кількістю методів/власності в класі, отримавши характеристику, показуючи, наскільки методи оцінки класу є обсягами. Крім того, такі оцінки можна використовувати в сукупності з метриками складності (наприклад, цикломатичної складності Мак-Кейба) для визначення найбільш складних частин у програмному коді та прийняття відповідних мер.

Існують два основних типи вимірювань і метрик: прямі і непрямі.

Прямі вимірювання (метрики) є вимірювання (метрики), яке не залежить від вимірювання будь-якого іншого атрибута. Наприклад, кількість операндів, розмір документації (в сторінках).

Непрямі вимірювання (метрики) є вимірювання (метрики), яке включає вимір одного або більше інших атрибутів. Наприклад, рівень коментарів в програмі $F = N_{com} / N_{loc}$.

Шкала вимірювань, це асоційована з атрибутом множина значень.

Метрика Lines of Code, це пряма, розмір-орієнтована метрика продукту (тексту ПЗ), яка визначає розмір продукту в кількості логічних або фізичних строк коду.

Економічні розрахунки

Зусилля (людина/місяць) $Effort = ab * size^{bb}$,

де ab , bb коефіцієнти,

$size$ – розмір продукту в KLOC.

Вартість (грн.) $Cost = Effort * salary$,

де $salary$ – заробітна платня.

Час на розробку $Schedule = cb * Effort^{db}$,

де cb , db – коефіцієнти.

Таблиця 1.

Тип проекту	ab	bb	cb	db
Organic	2.4	1.05	2.5	0.38
Semi-detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

Tun «Organic» являє собою відносно невеликий (до 25 тис. рядків коду) та простий проект, який виконується невеликою командою.

Tun «Semi-detached» являє собою середній за розміром (до 75 тис. рядків коду) та складністю проект, в якому команда має змішаний рівень досвіду і відносно жорсткі вимоги.

Tun «Embedded» являє собою проект, який виконується в умовах жорстких технічних, програмних та експлуатаційних обмежень.

Визначення рівня мови програмування

Для визначення рівня мови потрібно скористатись наступною формулою:

$$Level = \frac{BytecodeLOC}{LOC},$$

де **BytecodeLOC** – кількість рядків коду байткоду, а **LOC** – кількість рядків програмного коду на мові програмування (Java або C#).

Наприклад порівняємо рівні мов C# та Java, узявши однакові реалізації сортування.

У Java кількість LOC у сирцевому коді становить 16, а в скомпільованому – 82. Рівень мови становить $16:82 = 1:5.125$

У C# кількість LOC у сирцевому коді становить 23, а в скомпільованому – 83. Рівень мови становить $23:83 = 1:3.95$

Результати показують, що Java має більший рівень мови.

Завдання

1. З відкритих джерел вибрати три проекти (відповідно типів наведених в таблиці1). Можна використовувати і власні проекти якщо вони відповідають заданим параметрам.
2. Застосовуючи вимірювачі у відповідних середовищах програмування (наприклад Visual Studio, CodeCounter for Java, CodeCounter, та інші), визначити розмір кожного проекту.
3. Здійснити відповідні економічні розрахунки наведені в роботі.
4. Провести визначення мов програмування C# та Java (або на вибір студента).
5. Підготувати звіт
6. Скласти викладачу

Контрольні запитання

1. Типи метрик за типами складових життєвого циклу
2. Типи метрик розміру продукту
3. Поняття щодо вимірювань – вимірювання, властивість, атрибут, величина, сутність
4. Типи вимірювань та метрик
5. Поняття шкали вимірювань, типи шкал
6. Метрика Lines of Code, сутність, розвиток, проблеми
7. Рівень мови програмування, зв'язок з метрикою Lines of Code
8. Стан розвитку індустрії ІТ України
9. Розвиток економіки програмного забезпечення (1960 – 1979)
10. Розвиток економіки програмного забезпечення (1980 – 2004)
11. Поняття економіки, інженерної економіки, економіки програмного забезпечення
12. Попит і пропозиція, дефіцит ресурсів, зв'язок
13. Закони ринку