

ОСОБЛИВОСТІ ОТРИМАННЯ МЕТРИК З JAVA ТА KOTLIN ПРОГРАМНИХ ПРОЕКТІВ НАТИВНИХ МОБІЛЬНИХ ANDROID ЗАСТОСУНКІВ

Макарова Л.М., к.т.н, доц.¹; Татаренко М.А.²

^{1,2} Національний університет кораблебудування імені адмірала Макарова

^{1,2} Україна, Миколаїв

¹ lidia.makarova@nuos.edu.ua; ² multi3volume@gmail.com

Анотація. Кількісні значення метрик програмного забезпечення використовують для вимірювання властивостей програмних проектів або розробки програмного забезпечення, побудови регресійних моделей для оцінювання його розміру або якості. Такі значення метрик отримують за допомогою спеціальних інструментів-аналізаторів. Спектр вибору таких інструментів залежить від мови програмування. У роботі показано що для одного і того ж програмного проекту, однієї і тієї ж метрики, отримані кількісні значення цієї метрики сильно залежать від конкретного інструменту. Зокрема, це важливо при побудові регресійних моделей для оцінювання розміру або якості нативних мобільних застосунків для ОС Android.

Ключові слова: метрики програмного забезпечення; вимірювання метрик програмного забезпечення; регресійні моделі; проектування мобільних застосунків; Java; Kotlin, Android.

Вступна частина. Процес отримання кількісних значень метрик програмного забезпечення займає одне з ключових місць при побудові якісної регресійної моделі. Цей процес є не тільки важливим, але й трудомістким. Чим більше кількість проектів, з яких будуть отримані кількісні значення метрик, тим більша трудомісткість цього процесу.

Для підвищення якості та точності регресійної моделі, що буде будуватися, потрібно враховувати різні складові, серед яких мова програмування, якою реалізовані програмні проекти та тип цих проектів. Врахування даних складових - наступний крок у підвищенні точності регресійної моделі.

Запропонований підхід застосовується до будь-яких програмних проектів, у тому числі і до нативних мобільних застосунків для ОС Android, які до того ж відрізняються від звичайних десктопних або інших програм [1]. Для реалізації таких нативних мобільних застосунків використовуються дві основні мови програмування: Java та Kotlin.

Мета роботи. Аналіз існуючих інструментів для отримання кількісних значень метрик із програмних проектів нативних мобільних Android застосунків, реалізованих мовами програмування Java і Kotlin, для побудови регресійних моделей для оцінювання розміру або якості.

Основна частина. Перед побудовою регресійної моделі відбувається процес збору програмних проектів у набір даних (або кілька наборів даних), потім йде процес отримання кількісних значень з метрик ПЗ. Наступний процес - обробка даних з метрик ПЗ, наприклад для оцінювання розміру або якості об'єктно-орієнтованого проектування програмних проектів. Зазвичай у таких випадках застосовуються наступні метрики: LOC (Lines Of Code) - кількість рядків вихідного коду, NC (Number of Classes) - кількість класів. Також застосовуються метрики Чідамбера та Кемерера (Chidamber & Kemerer, CK metrics): WMC (Weighted methods per class) – зважені методи на клас, DIT (Depth of Inheritance Tree) –

глибина дерева успадкування, NOC (Number of Children) – кількість спадкоємців (підкласів) для класу, CBO (Coupling Between Object classes) – зв'язок між класами об'єктів, RFC (Response for a Class) - відповідь для класу, LCOM (Lack of Cohesion in Methods) - відсутність зв'язності методів. Можуть застосовуватися і інші метрики. Також слід зазначити, що для оцінювання розміру ПЗ на ранніх етапах його розробки використовуються метрики UML діаграм, в першу чергу діаграми класів.

Для отримання кількісних значень метрик із програмних проектів реалізованих мовою програмування Java існує велика кількість інструментів [2], наприклад: ck, ckjm, CCCC, Metrics Reloaded, CodeMR, Metrics Tree, DR-Tools Code Health. У той же час для проектів, реалізованих мовою Kotlin таких інструментів значно менше.

Для побудови регресійних моделей для оцінювання розміру або якості об'єктно-орієнтованого проектування нативних мобільних застосунків для ОС Android, які враховують мову програмування, необхідно аналізувати програмні проекти, реалізовані мовами програмування Java та Kotlin. Звичайно, можливе застосування двох різних аналізаторів окремо для кожної мови. Однак існує проблема, що для однієї і тієї ж програмної системи і однієї і тієї ж метрики отримані значення цієї метрики сильно залежать від конкретного інструменту, що використовується. Різні інструменти по-різному розраховують одну і ту ж метрику навіть для проектів, реалізованих однією і тією ж мовою програмування.

В якості прикладу, за допомогою трьох інструментів таких як: ck (1), Metrics Reloaded (2), CodeMR (3), було проаналізовано три програмні проекти, реалізовані мовою програмування Java:

- реалізація шаблону проектування Observer (5 класів, 1 інтерфейс);
- реалізація шаблону проектування MVC (6 класів, 1 інтерфейс);
- застосунок File renamer (16 класів, 2 інтерфейси).

Були зібрані такі метрики: WMC, DIT, CBO, RFC. Результати наведено у таблиці 1.

Таблиця 1 - Порівняння кількісних значень метрик, отриманих різними інструментами для кількох програмних проектів

Метрик а ПЗ	Observer			MVC			File renamer		
	(1)	(2)	(3)	(1)	(2)	(3)	(1)	(2)	(3)
WMC	16	10	16	55	53	44	171	156	132
DIT	6	4	6	19	17	9	26	44	44
CBO	9	10	5	18	20	9	117	61	42
RFC	14	29	32	43	95	65	292	417	259

З таблиці 1 видно, що кожен інструмент показує різні кількісні значення з однієї й тієї ж метрики. При аналізі інших проектів було виявлено, що невідповідність може становити від 10% до кількох разів. Таку різницю в результатах можна пояснити різною інтерпретацією і, відповідно, розрахунком однієї й тієї ж метрики кожним інструментом. Як приклад можна навести метрику WMC. Ця метрика надає вагу кожному методу класу і складає ці ваги.

Найбільш поширені ваги:

- присвоїти значення, рівне одиниці кожному методу (тим самим підрахувавши кількість методів);
- обчислити складність кожного метода за допомогою метрики цикломатичної складності Маккейба (але може застосовуватися і будь яка інша метрика складності).

Застосування інструменту, який поєднує розрахунок метрик для кількох мов програмування є найкращим варіантом на противагу застосуванню окремих інструментів-аналізаторів для кожної мови програмування. Це можна пояснити припущенням, що закладені в такому інструменті алгоритми розраховують одну і ту ж метрику однаково.

Прикладом такого інструменту є CodeMR [3]. Це плагін для інтегрованих середовищ розробки IntelliJ та Eclipse. Існують безкоштовна обмежена та повнофункціональна платна версія даного інструменту. CodeMR дозволяє розраховувати метрики LOC, NC, Чидамбера і Кемерера та інші для програм, написаних на Java, Kotlin, Scala. Плагін підтримує набори метрик на рівні проекту, пакета, класу та методу. У документації наведена інтерпретація та опис усіх метрик, які він дозволяє отримати. В CodeMR основний наголос в представленні результатів розрахунку метрик зроблений на різні варіанти візуалізації. Але в повнофункціональній версії значення метрик доступні у вигляді таблиці. Даний плагін дозволяє генерувати HTML-звіти для всього проекту або кожного робочого набору (Working set). Робочий набір дозволяє розділити весь програмний проект на дрібні частини для більш детального аналізу.

Висновки. При виборі інструменту для отримання метрик з програмних проектів слід виходити з повноти документації, в якій має бути зазначено, як інтерпретуються всі метрики, які дає можливість отримувати конкретний інструмент. Під час аналізу програмних проектів, написаних різними мовами програмування, рекомендується вибирати однаковий аналізатор. Зокрема, це важливо при побудові регресійних моделей для оцінювання розміру або якості нативних мобільних застосунків для ОС Android, для написання яких застосовуються дві основні мови: Java і Kotlin. У роботах з побудови регресійних моделей рекомендується вказувати інструмент-аналізатор, за допомогою якого було отримано кількісні значення тих чи інших метрик.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- [1] Shahwaiz, S.A., Malik, A.A., & Sabahat, N. (2016). A parametric effort estimation model for mobile apps. *2016 19th International Multi-Topic Conference (INMIC)*, 1-6. DOI: 10.1109/INMIC.2016.7840114
- [2] Макарова, Л.М., & Татаренко, М.А. (2024). Аналіз інструментів для отримання метрик Чидамбера і Кемерера з програмних проектів, реалізованих мовою Java. *Інновації в суднобудуванні та океанотехніці: зб. матеріалів XIV Міжнародної науково-технічної конференції*. (Миколаїв: НУК імені адмірала Макарова, 26–27 вересня 2024 рік). Миколаїв: НУК імені адмірала Макарова, 2024. 610-612.
- [3] CodeMR is a software quality and static code analysis tool for Java, Kotlin and Scala projects. (2024). Retrieved October 10, 2024, from <https://plugins.jetbrains.com/plugin/10811-codemr>

Makarova L.M., Tatarenko M.A.

Features of obtaining metrics from Java and Kotlin software projects of native mobile Android applications

Abstract. Quantitative values of software metrics are used to measure the properties of software projects or software development, build regression models to assess its size or quality. Such metrics values are obtained using special analyzer tools. The range of such tools depends on the programming language. This work shows that for the same software project, the same metric, the obtained quantitative values of this metric strongly depend on a specific analyzer tool. In particular, this is important when building regression models for estimating the size or quality of native mobile applications for the Android OS.

Keywords: software metrics; measuring software metrics; regression models; mobile application design; Java; Kotlin; Android.