

ПІДБІР МЕТРИК ДЛЯ ВЛАСТИВОСТЕЙ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

О.П. Дишлевий

Національний авіаційний університет, 03680, проспект Космонавта Комарова, 1, корп. 6, тел. 406 7098,
oleksiy.dyshlevy@livenau.net

Робота присвячена рішенням задачі підбору метрик для властивостей програмного забезпечення. Аналізуються існуючі методи рішення задачі, які є експертними, та пропонується метод формалізованого підбору метрик. Метод базується на застосуванні статистичного аналізу. Наводиться приклад використання запропонованого методу для властивості «тестованість».

The article is dedicated to metrics selection for the software attributes. It is analyzed expert methods of solving this task and it is proposed a method of formalized metrics selection. Method is based upon statistic analysis. It is making an example of using of formalized metrics selection for the software property «testability».

Вступ

Емпіричні дослідження програмного забезпечення – важливий напрям інженерії програмного забезпечення [1]. У цих дослідженнях часто постає задача оцінки властивостей програмного забезпечення. Властивості виступають характеристиками сутностей програмного забезпечення. Оцінка властивостей дозволяє отримати деякі значення для цих характеристик. У свою чергу це допомагає аналізувати складові розробки програмного забезпечення (процеси та продукти).

Ключовим методом досліджень програмного забезпечення є вимірювання [2]. Головні показники, які тут застосовуються, – метрики програмного забезпечення [3]. Метрики використовуються для оцінки властивостей програмного забезпечення. Остання проводиться шляхом підбору метрик для цих властивостей [2]. Головна задача, яка тут постає, як з великого набору підібрати метрики для властивостей програмного забезпечення, що найкраще слугують оцінці властивостей.

Огляд останніх досліджень

Існує три методи підбору метрик для властивостей програмного забезпечення: ціль–питання–метрика [4], модель відповідальної особи [5] та стандартизовані метрики [6].

Перший метод передбачає три етапи підбору метрик для властивості [4]:

- формулювання цілі, яка буде досягнута шляхом вимірювання програмного забезпечення;
- формулювання питань на які потрібно отримати відповідь щоб зрозуміти, чи досягається ціль;
- визначаються метрики, які можна використати для відповіді на запитання, що визначені на другому етапі.

Другий метод передбачає підбір необхідних метрик безпосередньо відповідальною особою [5]. Підбір метрик у даному випадку пов'язується з задачами, які повинні бути вирішені та рішеннями, які потрібно приймати при вирішенні даних задач. Рішення про вимірювання метрики приймаються на основі досвіду даної особи або за рекомендаціями експертів.

Як видно, перші два методи цілком не формалізовані. Їх використання покладається на високу кваліфікацію та досвід експертів.

Третій метод пов'язаний з використанням рекомендацій або стандартів деяких організацій, які займаються дослідженнями в потрібній сфері [6]. Цей метод базується на проведених емпіричних дослідженнях, його можна вважати напівформалізованим. Проте використання його дуже обмежується сферою проведених досліджень. Рекомендації з підбору метрик носять загальний характер.

Усі три методи дають можливість вирішувати поставлену задачу, тобто підбирати метрики для властивості програмного забезпечення. Проте вони не є формалізованими, що накладає певну долю ймовірності отримання неточних значень. Крім цього, в результаті використання цих методів можна отримати лише набір метрик, які можуть слугувати для оцінки властивості. Але вони не дозволяють визначити метрики, які найкраще підходять для оцінки властивості.

Тому, для вирішення задачі підбору метрик для властивості програмного забезпечення пропонується формалізувати поставлену задачу за допомогою використання предметно-орієнтованого методу побудови залежностей між метриками програмного забезпечення [7].

Мета роботи

Метою цієї роботи є наступне:

- описання методу підбору метрик для властивостей програмного забезпечення;
- демонстрація застосування методу підбору метрик для властивості «тестованість».

Предметно-орієнтований метод побудови залежностей між метриками

Метод був розроблений для побудови залежностей між метриками програмного забезпечення при аналітичному виведенні функціональної залежності однієї метрики від іншої. Для реалізації методу розроблений засіб [8], який буде використаний для підбору метрик.

Сутність предметно-орієнтованого методу побудови залежностей між метриками програмного забезпечення полягає в тому, що побудова залежностей відбувається статистичними методами з урахуванням високої точності вимірювань програмного забезпечення без похибок та гіперболічною спадною залежністю між значеннями метрик та кількістю виміряних програм [7].

Метод складається з трьох етапів, які певною мірою відповідають статистичному аналізу [7, 9]:

- первинний аналіз – розрахунок кількості необхідних вимірювань та статистичних характеристик метрик;
- кореляційний аналіз – визначення наявності залежностей між метриками та сила залежностей;
- регресійний аналіз – побудова найоптимальніших функцій залежності між метриками.

Особливості методу є наступними [7]:

- обов'язкова велика кількість вимірювань пов'язана з необхідністю великого обсягу даних. Так як метрики не мають нормального закону розподілу, потрібно бути впевненим у достатності вибірки;
- відсутність первинного статистичного аналізу через визначену при дослідженнях відсутність нормального закону розподілу метрик. Тому непотрібно визначати закон розподілу, що виконується на етапі первинного статистичного аналізу;
- використання в кореляційному аналізі тільки парної рангової кореляції пов'язана з відсутністю нормального закону розподілу метрик;
- відсутність перевірок точності коефіцієнтів через точність вимірювань (відсутність систематичних та несистематичних похибок через пасивну участь дослідника у вимірюваннях та точністю вимірювання використаних засобів вимірювання);
- відсутність перевірки спільного закону розподілу метрик через відсутність нормального закону розподілу метрик. Так як метрики не мають нормального закону розподілу, що впливає на вибір способу розрахунків, всі розрахунки проводяться для значень, які не мають нормального закону;
- наявність тільки нелінійної функціональної залежності між метриками через відсутність нормального закону розподілу;
- побудова регресії методом лінеаризації через точність отриманих значень у порівнянні з іншими методами.

Формалізований підбір метрик для властивостей програмного забезпечення

Сутність формалізованого підбору метрик для властивостей програмного забезпечення полягає в призначенні метрикам ваги, яка відображає зв'язок метрики з властивістю. Вага – це коефіцієнт, який визначає до якої міри метрика оцінює властивість. Розрахунок ваги проводиться за допомогою методу побудови залежностей між метриками програмного забезпечення на етапі кореляційного аналізу.

Використання методу передбачає попереднє отримання оцінок властивостей програмного забезпечення, підбір та вимірювання метрик, які характеризують обрану властивість.

В залежності від поставленої задачі вибір метрик для вимірювання можна провести двома способами:

- використати один із методів, описаних у [4–6];
- поміряти всі доступні метрики.

Перший спосіб варто використати у випадку коли задачею являється визначення найбільш вагомих із обраних метрик. Другий спосіб – коли необхідно перевірити правильність обраних метрик чи важко підібрати метрики. Для цього до обраних додаються інші метрики. Розрахована далі вагомість підтверджує або спростовує попередній вибір.

Вимірювання вибраних метрик можна здійснювати будь-яким засобом, який дозволяє отримати всі обрані метрики.

Попереднє отримання оцінок властивостей необхідне для визначення тенденцій у взаємозв'язках між метриками та властивостями. Важливо визначити, які метрики демонстрували гарні характеристики властивостей, а які – погані. Тому пропонується провести експертне оцінювання властивостей для того ж програмного забезпечення, що вимірювалося [10].

Далі необхідно застосувати метод побудови залежностей між метриками таким чином. Спочатку на першому етапі проводиться перевірка обсягу вхідних даних і розрахунок статистичних характеристик для метрик та експертних оцінок. Важливою передумовою є великий обсяг вхідних даних. Тому для дослідження необхідно використати велику кількість програм.

Найважливішим етапом для формалізованого підбору метрик є другий. На цьому етапі безпосередньо розраховується вагомість. Вага отримується як коефіцієнт кореляції, розрахований для метрики та експертної оцінки. Найбільше значення коефіцієнту серед вагомих метрик говорить про те, що ця метрика є найбільш вагомою для властивості. Крім цього у випадку використання всіх доступних метрик розраховується значущість, яка показує чи є вагомою метрика для властивості.

Використання регресійного аналізу не є необхідним. Регресійний аналіз у даному випадку показує вид функції залежності між окремою метрикою та властивістю. При цьому вид не дає можливості побачити взаємозв'язок між вагомими метриками для оцінки властивості. Тому отримані графіки можна використовувати тільки як демонстрацію загальних закономірностей між метрикою та властивістю.

Застосування методу для властивості «тестованість»

Отримання вхідних даних для дослідження. Вхідними даними дослідження є властивість та метрики. Застосування методу можна продемонструвати на прикладі властивості «тестованість» та метриках. Властивість «тестованість» показує ступінь легкості тестування коду, функціоналу, вимог у програмному забезпеченні. Дана властивість є ключовою при тестуванні програмного забезпечення на етапі верифікації та атестації [11]. Попередня оцінка цієї властивості дозволяє запобігти трудношам тестування на попередніх етапах життєвого циклу.

Для проведення дослідження було вибрано близько 50 об'єктно-орієнтованих відкритих проектів з загальною кількістю класів більше 8000. Як показали попередні дослідження [7] такого обсягу даних достатньо для вирішення поставленої задачі.

Вхідними даними при підборі метрик для властивості «тестованість» є виміряні за допомогою засобу iPlasma [12] метрики та експертні оцінки «тестованості». Даний засіб був обраний з декількох причин: можливість поміряти значну кількість важливих об'єктно-орієнтованих метрик [12] (чого не дають інші засоби), при цьому метрики можна отримати як для проектів в цілому, так і для окремих класів та методів; крім того засіб розповсюджується безкоштовно.

Спочатку були виміряні всі доступні метрики для проектів та класів та був перевірений необхідний обсяг вибірки, як описано в [7]. Даного обсягу вибірки виявилось достатньо для подальших розрахунків.

Для характеристики властивості було проведено експертне оцінювання, як описано в [10]. За оцінкою «тестованості» враховувалися наступні фактори: чіткість, послідовність, зрозумілість, повнота сформульованих вимог; наскільки просто вибрати технології тестування та покрити тестами всі частини проекту; наскільки просто писати тести для тестованого програмного забезпечення. Експертне оцінювання проводилося для того ж програмного забезпечення. Оцінки виставлялися за десятибальною шкалою – від «1» до «10». Для оцінювання були залучені експерти різного рівня кваліфікації.

Попередні розрахунки. Наступним етапом було отримання основних статистичних характеристик для досліджуваних величин:

- математичне сподівання

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i ; \quad (1)$$

- середнє квадратичне відхилення

$$S^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2 ; \quad (2)$$

- коефіцієнт асиметрії

$$\hat{A} = \frac{1}{n\hat{\sigma}^3} \sum_{i=1}^n (x_i - \bar{x})^3 ; \quad (3)$$

- коефіцієнт ексцесу

$$\hat{E} = \frac{1}{n\hat{\sigma}^4} \sum_{i=1}^n (x_i - \bar{x})^4 , \quad (4)$$

де σ – дисперсія.

У табл. 1 наведені статистичні характеристики для деяких метрик, розраховані за формулами (1) – (4) з [13].

Таблиця 1

Статистичні характеристики

| Позначення метрики | Математичне сподівання | Середнє квадратичне відхилення | Коефіцієнт асиметрії | Коефіцієнт ексцесу | Пояснення метрики |
|--------------------|------------------------|--------------------------------|----------------------|--------------------|---|
| AVRPF | 58.8417 | 17.9308 | -0.429 | 2.4917 | Середнє значення виключень у модулі |
| TM | 2.5140 | 2.4356 | 1.7780 | 6.0558 | Загальна кількість функцій (модулів компонента) |
| TLOC | 170.9636 | 142.7548 | 1.2357 | 3.8945 | Загальне число непорожніх рядків, включаючи коментарі та рядки заголовків |
| TNCSS | 121.5091 | 111.8934 | 1.3023 | 4.0540 | Загальне число вихідних конструкцій. Не включає порожні рядки, рядки коментаря чи заготовочні рядки |
| CYC | 0.7387 | 1.8768 | 8.1986 | 12.3719 | Цикломатична складність МакКейба |
| NOM | 0.2264 | 1.1670 | 8.5597 | 14.8171 | Кількість методів класу |
| FOUT | 1.4133 | 1.3537 | 3.4646 | 25.9815 | Число викликаних класів |
| TVG | 16.1000 | 19.7032 | 2.1608 | 7.5764 | Загальна цикломатична складність у всіх модулях (кількість ключових слів розгалужень в кожному модулі). |
| MFO | 8.9912 | 9.3785 | 1.5897 | 5.7260 | Максимальне число викликів інших функцій у модулі |
| MFI | 1.0000 | 1.7008 | 2.4493 | 10.2992 | Максимальне число викликів модуля |
| AVGFI | 0.3838 | 0.5715 | 1.5512 | 5.0540 | Загальне число викликів інших функцій у модулях |
| 'TA' | 7.6562 | 1.2077 | 0.0182 | -1.2201 | «тестованість» |

Підбір вагомих метрик. На основі статистичних характеристик, розрахованих на попередньому етапі, після наступних розрахунків були вибрані вагомі метрики для властивості «тестованість». Вибір метрик здійснювався на основі розрахунків коефіцієнтів парної рангової кореляції Спірмена та Кендала за наступними формулами [13]:

- коефіцієнт Спірмена

$$\hat{t}_k = \frac{2S}{n(n-1)}; \quad (5)$$

- коефіцієнт Кендала

$$\hat{t}_c = \frac{1}{n(n^2-1)} \sum_{i=1}^n (r_{xi} - r_{yi}), \quad (6)$$

де r_{xi}, r_{yi} – порядкові номери варіант метрик та експертних оцінок,

S – алгебраїчна сума кількості найвищих рангів відносно кожного більш вищого рангу:

$$S = \sum_{i=1}^{n-1} \sum_{j=i+1}^n v_{ij} \cdot v_{ij} = \begin{cases} +1, & r_{yi} < r_{yj} \\ -1, & r_{yi} > r_{yj} \end{cases}, \quad (7)$$

Розраховані коефіцієнти для деяких метрик наведені в табл. 2. Крім коефіцієнтів кореляції була визначена значущість отриманих коефіцієнтів. Вона показує чи є вагомою метрика для оцінки властивості. Перевірка значущості проводилася з використанням статистичної характеристики t (8):

$$t = \frac{\hat{t}_c \sqrt{n-2}}{\sqrt{1-\hat{t}_c^2}}. \quad (8)$$

Якщо $|t| > t_{\alpha/2}$ (де $t_{\alpha/2}$ було отримано з таблиці розподілів t-Стюдента), то робився висновок, що метрика вагома; якщо умова не виконувалася, то робився висновок, що метрика не вагома.

Таблиця 2

Коефіцієнти кореляції для властивості «тестованість» та метрик

| Назва метрики | Коефіцієнт кореляції Кендала | Коефіцієнт кореляції Спірмена | Значущість |
|---------------|------------------------------|-------------------------------|------------|
| AVRPF | 0.0912 | 0.1368 | + |
| TM | 0.3986 | 0.5979 | + |
| TLOC | 0.0693 | 0.1040 | + |
| TNCSS | 0.0263 | 0.0395 | - |
| TVG | -0.2482 | -0.3724 | + |

| Назва метрики | Коефіцієнт кореляції Кендала | Коефіцієнт кореляції Спірмена | Значущість |
|---------------|---------------------------------|----------------------------------|------------|
| CYC | 0.2002 | 0.2574 | + |
| NOM | 0.1059 | 0.1311 | + |
| FOUT | 0.0636 | 0.0796 | - |
| MFO | 0.0161 | 0.0241 | - |
| MFI | 0.1941 | 0.2912 | + |
| AVGFI | 0.1327 | 0.1991 | + |

Як виявилось, властивість «тестованість» характеризують такі метрики: AVRPF, TM, TLOC, TVG, CYC, NOM, MFI, AVGFI, AMW, HIT, WMC, TCC, MAXNESTING. Отримані результати дозволяють зробити висновок що на «тестованість» впливає обсяг коду та його складність.

Визначення вагомості метрик. Вагомість метрик можна спостерігати на основі розрахованих коефіцієнтів кореляції. Чим більше значення коефіцієнта, тим більшою мірою метрика характеризує властивість. При цьому найвищу вагу має цикломатична складність МакКейба (CYC). На основі проведеного дослідження можна сказати що «тестованість» більше характеризують метрики, які відображають складність програмного забезпечення. При цьому вплив у більшості метрик (за винятком TNCSS) має зростаючий характер, на що вказує знак коефіцієнтів кореляції. Це говорить про те, що більшому значенню метрики відповідатиме більше значення властивості.

Висновки

Розглянуто особливості застосування предметно-орієнтованого методу побудови залежностей між метриками програмного забезпечення при формалізованому підборі метрик для властивості програмного забезпечення. Виявилось, що необхідно мати експертні оцінки властивості або провести експертне оцінювання власноруч; вибір метрик можна робити на основі [4–6] або брати для дослідження всі метрики, з яких будуть вибрані впливові; можна вибрати метрики, які найбільше характеризують властивість та показати загальні тенденції. Проте даний метод не дає можливості отримати із властивості метрику, яка б відображала функціональну залежність від вимірних метрик.

Підбір метрик для властивості програмного забезпечення було продемонстровано на прикладі властивості «тестованість» та об'єктно-орієнтованих метриках. Виявилось, що «тестованість» характеризують обсяг коду та складність, причому складність – у більшій мірі. Найвищу вагу має цикломатична складність МакКейба (CYC).

1. *Koji Torii, Ken-ichi Matsumoto, Kumiyo Nakakoji, Yoshiro Takada, Kaduyuki Shima*, Ginger 2: An Enviroment for Computer-Aided Empirical Software Engineering // IEEE Transactions on Software Engineering, vol 25. N 4. – 1999. – P. 475–486.
2. *Norman E. Fenton, Shari Lawrence Pfleeger* Software Metrics: A Rigorous and Practical Approach. – Cambridge University Press. – 1996.– 638p.
3. *Forrest Shull, Janice Singer, Dag I.K. Sjoberg* Guide to Advanced Empirical Software Engineering. – Springer-Verlag London Limited 2008. – 394 p.
4. *Basili V.R., Weiss D.M.* A method for collection valid software engineering data // IEEE Transaction on Software Engineering, 10(6), 1984. – P. 728–38.
5. *J. McGary, D. Card, C. Jones, B. Layman, W. Clark, J. Dean, and F. Hall.* Practical Software Measurement. Objective Information for Decision Makers, Addison-Wesley, Boston. – 2002. – 294 p.
6. *Linda M. Laird, M. Carol Breman* Software Measurement and Estimation: a practical approach. John Wiley & Sons, Inc., Hoboken, New Jersey 2006. – 257 p.
7. *Душлевий О.П.* Предметно-орієнтований метод побудови залежностей між метриками програмного забезпечення // Вісник НАУ. – 2009. – № 3. – С. 206–212.
8. *Душлевий О.П.* Пакет статистичного аналізу для емпіричної інженерії програмного забезпечення // Наука і молодь. Прикладна серія. Зб. наук. праць. – 2009. – № 9. – С. 104–108.
9. *Вентцель Е.С.* Теория вероятностей: Учеб. для вузов. – 7–е изд. стер. – М.: Высш. шк., 2001. – 575 с.
10. *Хоменко В.А., Душлевий А.П.* Метод експертного оцінювання свойств повторно используемых компонентов програмного обеспечения // Том 1: Матер. VIII міжнар. наук.-техн. конф. «Авіа-2007». – Т. 1. – К.: НАУ, 2007. – С. 13.169–13.172.
11. *Соммервил Иан.* Инженерия програмного обеспечения, 6-е издание. : Пер. с англ. – М.: Издательский дом «Вильямс», 2002. – 624 с.
12. *Michele Lanza, Radu Marinescu.* Object-oriented metrics in practice: Using software metrics to characterize, evaluate, and improve the design of object-oriented systems. – Springer. – Verlag Berlin Limited. – 2006. – 205 p.
13. *Бабак В.П., Білецький А.Я., Приставка О.П., Приставка П.О.* Статистична обробка даних // Монографія. – К.: «МІВВЦ», 2001. – 388 с.