Аналіз та опрацювання метрик оцінки якості програмного забезпечення на етапі проектування

Тетяна Говорущенко

Кафедра системного програмування, Хмельницький національний університет, УКРАЇНА, м.Хмельницький, вул. Інститутська, 11, E-mail: tat_yana@ukr.net

Abstract *– In article software quality and complexity valuation on the design stage, software metrics in the view of their exactness on the design stage and software capability maturity model are described. Problem of metric analysis results processing was viewed. The conclusions about life cycle model, acceptable for software quality evaluation on the design stage, and about quality coefficients measuring procedures on the design stage were made. Defects of known software system quality valuation techniques were specified.*

Ключові слова – software quality, software life cycle, software metrics, software capability maturity model.

I. Вступ

Згідно визначення ISO [1], якість - це ступінь відповідності пристуніх характеристик вимогам. Згідно [2], якість - це повнота властивостей і характеристик продукту, процесу або послуги, які забезпечують здатність задовольняти оголошеним або передбачуваним потребам. Згідно [3], якість програмного забезпечення (ПЗ) - це ступінь, в якій воно володіє потрібною комбінацією властивостей. Іншими словами, якість ПЗ - це характеристика ПЗ, яка відображає ступінь його відповідності вимогам.

Проблема полягає в тому, щоб забезпечити потрібну якість функціювання ПЗ з врахуванням того, що деяка невідома кількість помилок та дефектів завжди залишається в складних комплексах програм, і повинна бути блокована або скорочена їх негативна дія до допустимого рівня. В зв'язку з цим стратегічна задача в життєвому циклі сучасного ПЗ - забезпечення якості програмних продуктів [4].

II. Аналіз моделей життєвого циклу ПЗ. Вибір методу одержання оцінки значень показників якості на етапі проектування

Якість ПЗ визначається якістю методів та інструментальних засобів, які застосовувались для забезпечення всього їх життєвого циклу. На практиці важливо оцінювати якість програм не лише в завершеному вигляді, але й в процесі їх проектування і розробки. Оціночна або прогнозована якість програмного продукту - характеристики, оцінені або передбачені для ПЗ на кожній стадії життєвого циклу (ЖЦ), які базуються на якості процесів та технології його забезпечення [4].

ЖЦ є моделлю створення і використання ПЗ, яка відображає його різні стани, починаючи з моменту виникнення необхідності в даному програмному продукті і закінчуючи моментом його повного виходу із вжитку у всіх користувачів. Існує багато моделей життєвого циклу ПЗ, але в міжнародних стандартах як фундаментальні класифікуються три з них [4]: каскадна (водоспадна), інкрементна (поетапна), еволюційна (спіральна).

На етапі проектування важливо закласти цілу низку вимог по якості: вимоги до структури програмної системи (ПС); вимоги до навігації по ПС; вимоги до дизайну інтерфейсів користувача; вимоги до мультимедіа-компонентів ПС; вимоги по зручності (usability); технічні вимоги. На етапі проектування формується відповідь на питання "Яким чином програмна система буде реалізовувати висунуті до неї вимоги?". Інформаційні потоки етапу проектування ПЗ [5]: вимоги до ПЗ, представлені інформаційною, функційною та поведінковою моделями аналізу. Інформаційна модель описує інформацію, які повинна обробляти ПЗ на думку замовника. Функційна модель визначає перелік функцій обробки інформації та перелік модулів програмної системи. Поведінкова модель фіксує бажану динаміку системи (режими її роботи). На виході етапу проектування - розробка даних, розробка архітектури та процедурна розробка ПЗ.

Для одержання оцінки значень показників якості за стандартом [6] використовуються наступні методи: вимірювальний, реєстраційний, розрахунковий та експертний, а також комбінації цих методів. Вимірювальний метод базується на використанні вимірювальних та спеціальних програмних засобів для одержання інформації про характеристики ПЗ (обсяг ПЗ, кількість рядків коду, кількість операторів, кількість гілок в програмі, кількість точок входу/виходу та ін). Реєстраційний метод використовується при підрахунку часу, кількості збоїв або відмов, початку і кінця роботи ПЗ в процесі його виконання. Розрахунковий метод базується на статистичних даних, зібраних при проведенні випробувань, експлуатації та супроводженні ПЗ. Розрахунковими методами оцінюються показники надійності, точності, стійкості та ін. Експертний метод здійснюється групою експертів. Їх оцінка базується на досвіді та інтуїції, а не на безпосередніх результатах розрахунків або експериментів. Цей метод реалізується шляхом перегляду програм, кодів, супровідних документів, описів вимог до ПЗ групою експертів [7]. Метод застосовується при оцінці таких показників, як аналізованість, документованість, структурованість ПЗ та ін.

Отже, вже на етапі проектування спіральна модель життєвого циклу дає можливість визначати якість ПЗ із застосуванням розрахункових та експертних методів вимірювання.

III.Метрики оцінки якості ПЗ на етапі проектування

В якості метрик процесу проектування використовуються:

- очікувана LOC-оцінка - за кожною функцією експерти надають краще, гірше та імовірне значення, тоді очікувана LOC-оцінка обчислюється за формулою:

$LOC_{оч_i} = (LOC_{кращ_i} + LOC_{гірш_i} + 4 \times LOC_{імов_i}) / 6$;

ця метрика на етапі проектування має *прогнозоване значення*;

- метрики складності ПЗ - метрики Холстеда та Маккейба; з метрик Холстеда на етапі проектування використовуються [5]: міра довжини модуля ($N = n_1 \log_2(n_1) + n_2 \log_2(n_2)$, де n_1 - кількість різних операторів, n_2 - кількість різних операндів) та обсяг модуля як кількість символів для запису всіх операторів і операндів тексту програми ($V = N \times \log_2(n_1 + n_2)$) ; для оцінки складності ПС, виходячи з топології внутрішніх зв'язків, Маккейб розробив метрику цикломатичної складності $V(G) = E - N + 2$, де E - кількість дуг, а N - кількість вершин в керуючому графі ПЗ; ці метрики на етапі проектування мають *прогнозоване значення*;

- метрики Чепіна - полягають в оцінці інформаційної міцності окремо взятого програмного модуля за допомогою аналізу характеру використання змінних зі списку введення-виведення; ці метрики вже на етапі проектування мають *точне значення*;

- метрики зв'язності (зчеплення) - чим вище зв'язність модуля з іншими модулями, тим вища чутливість до внесення змін, тобто високий ступінь зв'язності (зчеплення) модуля з іншими модулями - це суттєвий недолік; ці метрики вже на етапі проектування мають *точне значення*;

- метрика Джилба - кількість операторів циклу, кількість операторів умови, кількість модулів або підсистем, відношення кількості зв'язків між модулями до кількості модулів; на етапі проектування можна тільки підрахувати кількість модулів та відношення кількості зв'язків між модулями до кількості модулів; ці складові метрики вже на етапі проектування мають *точне значення*;

- метрика Хансена - пара (цикломатична кількість, кількість операторів); ця метрика на етапі проектування має *прогнозоване значення*;

- метрика Мак-Клура - міра складності, заснована на кількості можливих шляхів виконання програми, кількості керуючих конструкцій та змінних; метрика, спрямована на вимірювання архітектури системи; ця метрика вже на етапі проеткування має *точне значення*;

- метрика Кафура - метрика, заснована на врахуванні потоку даних; оскільки на етапі проектування вже є відомості про оброблювану інформацію, то й метрика має *точне значення* вже на етапі проектування;

- метрика Зольновського, Сіммонса, Тейєра - складова метрики (структура, взаємодія, обсяг, дані) обчислюється вже на етапі проектування і має *точне значення*;

- метрика звертання до глобальних змінних - пара (модуль, глобальна змінна); ця метрика вже на етапі проектування має *точне значення*;

- метрика Тая - складність програми визначається вимірюванням інформаційного потоку даних; вже на етапі проектування метрика має *точне значення*;

- метрика Кокола - комплексна метрика, яка, як правило, враховує значення метрик Холстеда, Маккейба і LOC, отже, на етапі проектування має *прогнозоване значення*;

- метрика Вітворфа, Зулевського - складова метрики "міра складності потоку даних" має *точне значення* вже на етапі проектування;

- загальний час розробки і окремий час для кожної стадії - метрика має *прогнозоване значення* на етапі проектування;

- час модифікації моделей - метрика має *точне значення* на етапі проектування;

- час виконання робіт в процесі - метрика має *прогнозоване значення* на етапі проектування;

- кількість знайдених помилок при інспектуванні - метрика має *точне значення* на етапі проектування;

- прогнозована кількість операторів програми - $N_{прогн} = NF * N_{од}$, де NF — кількість функцій або вимог в специфікації вимог до розроблюваної програми; $N_{од}$ - одиничне значення кількості операторів (середня кількість операторів, які доводяться на одну середню функцію або вимогу);

- прогнозована оцінка складності програми - $C = \frac{NI}{NF * NI_{од} * K_{скл}}$, де NI - загальна кількість змінних, які передаються по інтерфейсах між компонентами програми (статистична); $NI_{од}$ – одиничне значення кількості змінних, які передаються по інтерфейсах між компонентами (середня кількість змінних, які передаються по інтерфейсах, що доводяться на одну середню функцію або вимогу); $K_{скл}$ — коефіцієнт складності розроблюваної програми, враховує ріст одиничної складності програми (складності, що доводиться на одну функцію або вимогу специфікації вимог до програми) для великих та складних програм в порівнянні з середнім показником складності;

- очікувана вартість розробки кожної функції:

$$ВАРТІСТЬ_i = LOC_{оч_i} \times ВАРТІСТЬ_{рядка_i},$$

вартість рядка є константою і не змінюється від реалізації до реалізації;

- прогнозована продуктивність розробки кожної функції (на основі продуктивності аналогічних функцій в анлогічних програмних продуктах):

$$ПРОДУКТ_i = ПРОДУКТ_{ан_i} \times (LOC_{ан_i} / LOC_{оч_i})$$

- прогнозовані витрати на розробку кожної функції: $ВИТРАТИ_i = (LOC_{оч_i} / ПРОДУКТ_i)$;

- прогнозований функційний розмір FP - використовується як відносна метрика для порівняння з попередніми проектами, за його допомогою можна обчислити кількість рядків коду, що дозволяє визначити загальну трудомісткість та терміни проекту;

- прогнозована оцінка трудовитрат та тривалості проекту за моделлю Боема [9, 10];

- прогнозована вартість перевірки якості;
- прогнозована вартість процесу розробки.

Різні метрики відображають різні аспекти ПЗ. Для всебічного врахування даних аспектів при оцінці ПЗ застосовується не одна метрика, а їх сукупність. Якщо було одержано декілька метрик, то кожне значення метрики множиться на відповідний ваговий коефіцієнт, встановлений експертним шляхом з огляду на домінуючі критерії якості відповідно до принципових задач, особливостей, функціонального призначення та властивостей ПЗ, а потім додаються всі показники для одержання комплексної оцінки рівня якості ПЗ або якості процесу проектування ПЗ. Найбільш актуально застосовувати досить великі сукупності метрик складності на етапі проектування, а в наступних етапах значення метрик фактично уточнюються.

Процес розроблення ПЗ можна виміряти лише в порівнянні з даними по організації або по галузі, а самі по собі одержані числа ні про що не говорять.

В результаті аналізу метрик ПЗ можна визначити *основні проблеми метрології ПЗ*: відсутність загальноприйнятої номенклатури показників якості; неможливість проведення натурних випробувань програм на всій множині початкових даних; низька достовірність та недостатність інформації для одержання оцінок показників якості; недостатність засобів вимірювання метрик програми; відсутність обгрунтованих вимог до метричної інформації, виражених в числовому вигляді, які могли б підлягати перевірці; відсутність можливості інтерпретації одержуваних метрик і оцінок показників якості програм.

Отже, методи оцінки якості ПС, особливо на етапі проектування, на сьогодні є суб'єктивно залежними, оскільки використовуються експертні вагові коефіцієнти для метрик; порівняння значень метрик поточних проектів з попередніми (постає проблема, що робити, якщо проект принципово новий); немає загальноприйнятої номенклатури метрик; відсутні точні значення метрик, з якими можна було б порівняти поточні одержані значення.

IV.Модель зрілості можливостей ПЗ

Загальне покращення процесу розроблення ПЗ в першу чергу вимагає класифікації типів робіт і процесів, яка цілком залежить від компанії.

Для визначення загального рівня розвитку технологічних процесів в програмних організаціях розробили спеціальну систему оцінки зрілості технологічних процесів софтверних організацій - *модель Capability Maturity Model (CMM)*, засновану на так званих рівнях зрілості (maturity levels) [11]. Таких рівнів зрілості в CMM п'ять, і кожен з них характеризує певний ступінь якості програмних продуктів [11]:

- початковий (initial) - відсутнє стабільне середовище розробки і супроводження; не витримуються терміни випуску продуктів; всі сили спрямовано на кодування і тестування програми (75% софтверних організацій);
- повторюваний (repeatable) - жорстке керування, планування і контроль; акцент робиться на початкові вимоги, методи оцінки і конфігураційний менеджмент (15% софтверних організацій);
- фіксований (defined) - процеси повністю документовані, стандартизовані і інтегровані в єдиний технологічний потік (8% софтверних організацій);
- керований (managed) - намагання оцінити якість процесів і готового продукту кількісно; для контролю над процесами використовуються метрики (1,5% софтверних організацій);
- оптимізований (optimizable) - намагання покращення роботи, керуючись кількісними критеріями якості; основна мета - випуск бездефектних продуктів, в яких помилки усунені ще на стадії внутрішнього тестування (0,5% софтверних організацій).

V. Опрацювання метрик оцінки якості ПЗ етапу проектування

На основі аналізу метрик можна зробити висновки про рівень якості вихідного коду. Однією з проблем аналізу метрик вихідного коду є складність інтерпретації обчислених величин.

Статистичний аналіз метричної інформації передбачає:

1) аналіз метрик за релізами: накопичення статистичної інформації метрик складності і якості ПЗ служить основою для управління складністю і якістю ПЗ в наступних проектах. Так, метрики довжини і обсягу програми дають інформацію про збільшення чи зменшення обсягу програми в часі. Метрика цикломатичної складності показує, чи зростає складність від релізу до релізу. Метрика кількості рядків на реалізацію вимоги попереджає про виявлення збільшення кількості рядків під час виконання типового запиту чи під час реалізації типової вимоги і дозволяє програмісту спрогнозувати кількість рядків коду при використанні типових вимог і функцій. Вони використовуються для прогнозу складності на ранніх етапах на основі статистики.

2) аналіз метрик за замовниками: якщо виділяються окремі гілки або функції програми для конкретного замовника. Потрібно зберігати інформацію про всі зміни коду для всіх замовників.

3) вибірка проектних даних за певними критеріями - проектами, програмами, замовниками.

Для визначення задовільності чи незадовільності досягнутого рівня складності ПС використовується відношення розрахункового значення метрики до базового (статистичного) значення метрики, взятого зі статистичних даних попередніх проектів. Якщо таке відношення близьке до 1 або менше 1, то можна зробити висновок про задовільний рівень складності ПС за аналізованою метрикою.

При статистичному аналізі метричної інформації основною проблемою є неможливість опрацювання метрик для принципово нового проекту.

Важливість якості ПЗ збільшує інтерес до нових методів, які використовуються в побудові моделей якості ПЗ для прогнозування атрибутів якості. Одним з таких методів є метод, що базується на штучній нейронній мережі (ШНМ). В [12] показано використ-

тання ШНМ для прогнозу якості ПЗ на основі об'єктно-орієнтованих метрик. Тому можна зробити висновок, що ШНМ можуть бути корисні в побудові моделі якості ПЗ. В [13] доведено застосовність байєсових мереж в якості інструменту підтримки для числової оцінки ПЗ в реальному масштабі часу, особливо для додатків з особливими вимогами щодо безпеки.

Очевидно, що проблема оцінювання результатів метричного аналізу програмних продуктів з використанням інтелектуальних методів (зокрема з використанням ШНМ) досліджувалась мало, причому лише для конкретних видів метрик (об'єктно-орієнтовані метрики) і для конкретних типів програмного забезпечення (об'єктно-орієнтоване ПЗ та ПЗ з особливими вимогами щодо безпеки). Отже, актуальними є подальші дослідження на предмет можливості використання ШНМ для прогнозу якості ПЗ на основі аналізу інших типів метричної інформації для різних типів ПЗ.

Висновки

Для визначення якості ПЗ на етапі проектування, найбільш прийнятною є спіральна модель життєвого циклу ПЗ. З опису методів вимірювання показників (метрик) якості зрозуміло, що на етапі проектування ПЗ можуть бути задіяні лише розрахункові та експертні методи вимірювання, оскільки неможливо виміряти жодної характеристики ще не розробленого ПЗ і неможливо реєструвати моменти процесу виконання ще не існуючого ПЗ.

Незважаючи на численні дослідження програмних метрик, в цій галузі залишається *багато невирішених питань*: 1) лише 1,5% софтверних організацій намагаються оцінити якість процесів і готового продукту кількісно, за допомогою метрик і лише 0,5% софтверних організацій намагаються покращити роботу, керуючись кількісними критеріями якості з метою випуску бездефектних продуктів; 2) технологія вимірювання якості ще не досягла зрілості, оскільки лише 0,5% софтверних організацій знаходяться на оптимізованому, зрілому рівні моделі CMM; 3) відсутні єдині стандарти на метрики, створено більше тисячі метрик, тому кожен постачальник "вимірювальної" системи пропонує власні способи оцінки якості і відповідно метрики; 4) існує проблема складності інтерпретації величин метрик. Саме через невирішеність цих питань поки що неможливо створити бездефектне високоякісне ПЗ.

Приймаючи до уваги результати аналізу методів та засобів оцінки складності ПС, можна зробити висновок, що перспективним напрямком досліджень є розроблення інтелектуальних систем, які:

1) обчислюватимуть розрахунковими та експертними методами точні або прогнозовані значення метрик програмного забезпечення вже на етапі проектування;

2) не лише будуватимуть метрики, але й на основі одержаних значень метрик надаватимуть рекомендації і висновки про розроблюване програмне забезпечення, аналізуватимуть і опрацьовуватимуть результати метричних оцінок, оскільки для більшості користувачів як метрики, так і їх конкретні значення ні про що не говорять.

[1] ISO 9001:1994 Quality systems - Model for quality assurance in design, development, production, installation and servicing

[2] ISO 8402:1994 Quality management and quality assurance

[3] 1061-1998 IEEE Standard for Software Quality Metrics Methodology

[4] Липаев В.В. Программная инженерия. Методологические основы. - М.: ТЕИС, 2006. - 608 с.

[5] Орлов С.А. Технологии разработки программного обеспечения. Разработка сложных программных систем: Учебник для ВУЗов - СПб.: Питер, 2004. - 527 с.

[6] ДСТУ 3230–1995. Управление качеством и обеспечение качества. Термины и определения.

[7] Петрухин В.А., Лаврищева Е.М. Методы и средства инженерии программного обеспечения // http://www.intuit.ru/department/se/swebok/10/1.html

[8] http://www.construx.com

[9] Брауде Э. Технология разработки программного обеспечения - СПб.: Питер, 2004. - 655 с.

[10] Boehm B. Software Engineering Economics - NJ: Prentice Hall, 1981. - 392 p.

[11] http://www.rol.ru/news/it/press/cwm/25_96/teh.htm

[12] K. K. Aggarwal, Yogesh Singh, Arvinder Kaur, and Ruchika Malhotra. Application of Artificial Neural Network for Predicting Maintainability using Object-Oriented Metrics // PROCEEDINGS OF WORLD ACADEMY OF SCIENCE, ENGINEERING AND TECHNOLOGY VOLUME 15 OCTOBER 2006 ISSN 1307-6884 // http://www.waset.org/pwaset/v15/v15-52.pdf

[13] Janusz Zalewski , Andrew J. Kornecki, Henry L. Pfister. Numerical Assessment of Software Development Tools in RealTime Safety Critical Systems Using Bayesian Belief Networks // http://www.proceedings2006.imcsit.org/pliks/194.pdf