

УДК 331; 004.42

В. В. Колдовский,
кандидат економічних наук, доцент кафедри економічної кібернетики,
Українська академія банківської справи НБУ, м. Суми

АКТУАЛЬНІ ПРОБЛЕМИ ОЦІНКИ ЯКОСТІ В УПРАВЛІННІ ПРОГРАМНИМИ ПРОЕКТАМИ З АВТОМАТИЗАЦІЇ ЕКОНОМІКИ

Досліджуються проблеми оцінки якості програмного коду при реалізації проектів зі створення інформаційних систем, які використовуються для вирішення економічних задач. Запропоновано деякі перспективні підходи вирішення існуючих проблем у цій галузі.

Ключові слова: управління програмними проектами, інформаційна система, програмний код, метрики, візуалізація коду

The problems of assessing the quality of code in projects to create information systems that are used to solve economic problems. Proposed some promising approaches to solve existing problems in this area.

Keywords: Software Project Management, information system, code, metrics, code visualization

Вступ. Протягом двох останніх десятиліть робота організацій практично будь-якої сфери діяльності значним чином трансформувалася у напрямку суттєвого зростання кількості і глибини проникнення комп'ютеризованих інформаційних систем (ІС). Розширюючи область автоматизації від централізованих обчислюваних і комунікаційних систем до автоматизації робочих місць персоналу, деякі компанії досягли 100-відсоткового показника оснащення співробітників обчислювальною технікою. В окремих випадках на одного співробітника приходить понад одного ПК.

Постановка задачі. Одночасно із збільшенням обсягів використання апаратного, зростає роль і програмного забезпечення (ПЗ). Конкурентний тиск, мінливе зовнішнє середовище і необхідність постійно впроваджувати нові продукти та послуги для клієнтів спонукають компанії пришвидшувати темпи автоматизації та використовувати різноманітне за походженням і якістю виконання ПЗ. Водночас все актуальнішою постає проблема ризиків, які виникають унаслідок неякісного ПЗ. Мова йде як про цілеспрямовані атаки злочинного походження, так і збоїв в роботі ІС, які виникають внаслідок ненавмисних помилок у програмному коді чи є наслідком інших недоліків при розробці ПЗ. Таким чином, існує актуальна задача оцінки якості програмного коду ІС для подальшого його вдосконалення чи заміни альтернативними рішеннями вищої якості.

Аналіз джерел і публікацій. Науково-методологічною основою дослідження проблеми оцінки якості програмного коду є наукові праці таких вітчизняних і зарубіжних вчених як Баррі Боема (Barry Boehm), Говард Холстеда (Howard Halsted), Томаса МакКейба (Thomas McCabe) та інших [1,3,4]. Першим кількісним показником як обсягу, так і складності програмного коду історично визначився показник кількості рядків коду, який активно почав застосовуватися з часів перших ЕОМ у 1950-х рр. Лише в кінці 1970-х рр. відбулися певні істотні зрушення у цьому напрямку. Зокрема, Томас МакКейб у 1976 запропонував одну з перших комплексних метрик програмного коду – метрику цикломатичної складності, яка досить точно відображала залежність між складністю конструкцій програмного коду і здатності до подальшої його підтримки, а також наявності прихованих дефектів у ньому. У подальшому Говард Холстед запропонував власний набір метрик, які отримали назву «Метрики Холстеда» і за допомогою яких визначалося п'ять окремих характеристик програмного коду (довжина програми, словник програми, обсяг, складність та зусилля). Подальші праці вчених у цьому напрямку були тісно пов'язані з розвитком об'єктно-орієнтованого програмування, яке потребувало своїх підходів до оцінки. Водночас однозначність і достовірність оцінок, отриманих у результаті використання різноманітних метрик нерідко ставиться під сумнів. Зокрема, у 2008 році були опубліковані результати дослідження, згідно з яким достовірність оцінок метрики цикломатичної складності не є вищою за найпримітивніші оцінки на основі кількості рядків коду [5].

Невирішені раніше частини проблеми. Незважаючи на активну роботу, яка проводиться вченими у напрямку вирішення проблеми оцінки якості програмного коду, задача в цілому не є вирішеною. Існуючі методики часто не дають цілісної картини про реальний стан коду, різні показники нерідко показують протилежні та неузгоджені результати. Використання більшості існуючих підходів у певних галузях економіки, наприклад, у банківській сфері, ускладнене з причин неврахування специфіки її діяльності.

Все зазначене вище обумовлює актуальність удосконалення підходів до оцінки якості програмного коду, формування цілісної картини по певному проекту, його придатності для використання практичних задач, а також напрямків для підвищення якості.

Метою статті є визначення актуальних проблем оцінки якості в управлінні програмними проектами з автоматизації економіки і формування рекомендацій для їх вирішення.

Виклад основного матеріалу. Поняття якості інформаційної системи є достатньо широким. З точки зору користувача (власника) системи якість визначається у придатності системи до вирішення його прикладних задач і відповідності його вимогам. Така характеристика якості ІС є зовнішньою. З точки зору розробника якість системи характеризується такими факторами як зручність її розвитку, супроводження, розуміння коду та ін. Така характеристика є внутрішньою. Оскільки сучасні ІС в своїй основі мають, як правило, стандартизоване апаратне забезпечення, ОС та відрізняються програмним кодом, то в рамках даної роботи увага буде акцентуватися на якісних характеристиках саме ПЗ, які за своєю природою відповідають наведеним характеристикам для ІС взагалі.

Очевидно, що якість програмного коду як основи ПЗ і, відповідно, ІС визначає як зовнішні так і внутрішні якісні характеристики системи в цілому. Таким чином, за умов використання надійних методів оцінки якості програмного коду, вдається сформувати надійний механізм, який хоч і не дозволить отримати всеохоплюючу характеристику якості ІС у широкому розумінні (мова йде, насамперед, про такі показники як здатність системи коректно вирішувати поставлені задачі, відповідати очікуванням замовника та ін.), однак дозволить виявити прогалини у системі, до усунення яких використання певної ІС буде являтися джерелом підвищеного ризику.

Традиційно у програмній інженерії кількісною характеристикою якості програмного коду є метрика, як певна чисельна величина, яка дозволяє отримати значення деякої властивості ПЗ. Метрики є поширеним і достатньо надійним механізмом оцінки внутрішньої будови програм, класифікація метрик за різними ознаками наведена на рис. 1.



Рис. 1. Класифікація метрик програмного коду

Незважаючи на те, що розрахунок метрик може проводитися у повністю автоматизованому режимі, питання ефективного аналізу результату залишається невизначеним. Значні обсяги даних, отримані у результатах розрахунків, протилежні значення результатів певних метрик, складність і неоднозначність трактування та суб'єктивність обчислень деяких з них є серйозною перепоною на шляху до широкого практичного застосування метрик у якості надійного універсального інструменту оцінки якості програмного коду.

В таблиці 1 наведені лише деякі з існуючих метрик, які використовуються для оцінки програмних проектів, однак навіть вони можуть повертати неузгоджені між собою результати. Наприклад, низьке абсолютне значення трудомісткості за метрикою Холстеда (E), яке характеризує проект як відносно простий, може бути отримано одночасно із високим абсолютним значенням цикломатичної складності для окремих фрагментів проекту (M), що, навпаки свідчить про його високу складність. Водночас залежність обох наведених показників від загальної кількості рядків коду (SLOC) не є прямою, хоча як правило складність проекту позитивно корелює із його обсягом. Враховуючи, що загальна кількість метрик, які активно використовуються в сучасному менеджменті програмних проектів, становить декілька десятків, то інтерпретація їх результатів перетворюється на надзвичайно трудомістку задачу, яка із-за своєї високої складності часто залишається невирішеною, а самі проекти залишаються фактично некерованими, оскільки управління неможливе без кількісних показників, на які мають опиратися рішення.

Таблиця 1.
Формули розрахунку деяких поширених метрик програмного коду

| № п/п | Назва метрики | Формула розрахунку | Пояснення |
|-------|-----------------------------|---|--|
| 1 | Кількість рядків коду, SLOC | $SLOC = \sum_{i=1}^N S_i$ | N – кількість структурних блоків у кодї, S _i – обсяг i-го блоку |
| 2 | Цикломатична складність, M | $M = E - N + 2P$ | Обраховується на основі графу потоку управління, де E – кількість граней графу, N – кількість вузлів графу, P – кількість зв'язаних компонентів |
| 3 | Метрики Холстеда | $N = N_1 + N_2$ $n = n_1 + n_2$ $V = N \times \log_2 n$ $D = (n_1/2) \times (N_2/n_2)$ $E = D \times V$ | n ₁ – кількість унікальних операторів, n ₂ – кількість унікальних операндів, N ₁ – загальна кількість операторів, N ₂ – загальна кількість операндів N – довжина програми, n – словник програми, V – обсяг програми, D – складність програми, E – трудомісткість |

Таким чином, незважаючи на достатньо розвинені механізми кількісної оцінки якості програмного коду на основі метрик, їх використання пов'язане із певними труднощами, що спонукає до пошуку нових, більш ефективних у практичному використанні підходів.

На наш погляд, одним із способів покращення ефективності використання оцінки якості програмного коду може слугувати візуалізація показників проекту таким чином, щоб інформація подавалася достатньо наглядно, а її розуміння не було б особливо складною задачею.

Розробка інтегрального показника якості пов'язана із проблемою його подальшої інтерпретації у разі протилежних значень показників, які входять до його складу (відповідно до наведеного раніше прикладу). Використання декількох агрегованих показників ускладнене особливостями сприйняття людиною багатовимірної інформації. У даному разі, на наш погляд, доцільно використання показників, які можна наглядно представити у дво- чи тривимірному просторі.

Прикладом науково-дослідницьких проектів, які вдало працюють у цьому напрямку, слід назвати CodeCity [2]. На рисунку 2 наведена візуалізація проекту у CodeCity. Вдало проведена аналогія між внутрішньою будовою проекту і зовнішнім виглядом міста, поділеного на блоки і квартали, природно сприймається людиною.

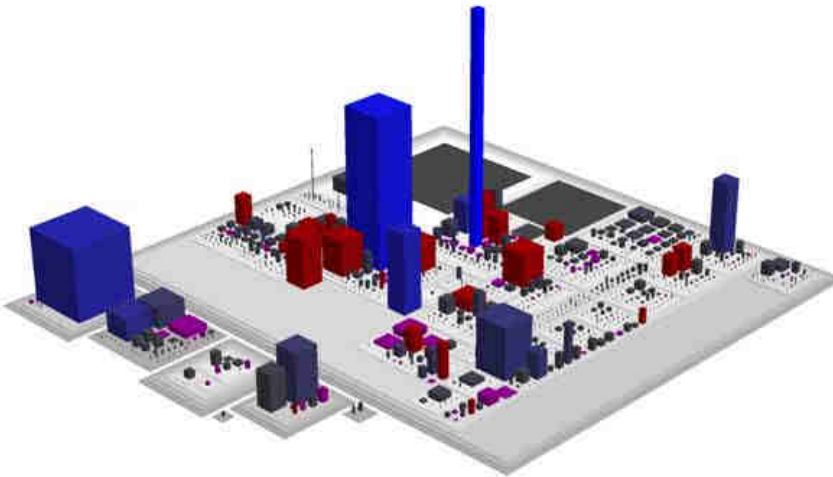


Рис. 2 Приклад візуалізації проекту у CodeCity

Відповідно до рис. 2 кожна «будівля» на отриманому тривимірному зображенні представляє собою програмний клас, висота «будівлі» визначається кількістю його методів (обсягом обчислень), а площа «фундаменту» – кількістю атрибутів (обсягом даних). Саме «місто» розділене на «квартали», які визначають приналежність класів до пакетів (модулів), а оскільки останні можуть бути вкладеними, то і структура міста може бути непростюю. Колір «будівель» обирається від темно-сірого до світло-блакитного виходячи в залежності від їх обсягу у кількості рядків коду, а «квартали» - від темно-сірого до світло-сірого в залежності від рівня вкладеності.

Незважаючи на очевидну наглядність результатів, підходи до оцінки якості програмного коду і його візуалізації на принципах, схожих до тих, які використовуються у CodeCity, поки що не набули достатнього поширення і на даній стадії розвитку становлять більше академічний інтерес, ніж надають можливості для практичного застосування у реальних проектах. Основна причина цього полягає, на наш погляд, у тому, що показники оцінки якості, які використовуються у реальних проектах, погано піддаються візуалізації за допомогою простих аналогій, обраних авторами проекту. Подати у наглядному вигляді вдалося лише достатньо обмежений обсяг показників, прийняття управлінських рішень лише на їх основі далеко не завжди вдається можливим.

Висновки. Таким чином, задача оцінки якості програмних проектів поки що не отримала прийнятного розв'язку у сучасній науці. Існуючі поширені підходи на основі метрик нерідко призводять до результатів, які погано узгоджуються між собою та є достатньо складними для інтерпретації і прийняття рішень на їх основі. Академічні проекти аналогічні до згаданого вище CodeCity дозволяють по-новому підійти до вирішення проблеми, водночас поки що не придатні для масового практичного застосування в основному з причин як своєї новизни, так і незавершеності концепцій, використаних для представлення показників якості по проектам.

Подальші дослідження у цьому напрямку, на наш погляд, мають бути зосереджені на тому, щоб привести існуючу базу метрик програмного коду до такого вигляду, у якому вони могли б слугувати реальною опорою для прийняття рішень керівниками програмних проектів, використовуючи комплексні і узгоджені між собою показники. Використання природно зрозумілих людині аналогій (структура міста, вигляд дерева та ін.) допомогло б подати інформацію більш наглядно, розширити коло людей, які могли нею оперувати при реалізації проекту.

Література

1. Аналіз методів та засобів оцінки якості програмних систем / Поморова О.В., Говорушенко Т.О. // Радіоелектронні і комп'ютерні системи, 2009. – № 6(40). – С. 148-158
2. Визуализация программного кода: увидеть незримое [Текст] / Колдовский В.В. // Компьютерное обозрение, 2010. – № 20. – С. 42–44
3. Определение информативности метрик объектно-ориентированного программного кода [Текст] / Винников М. А., Панекин Н. Н. // Вісник Донбаської державної машинобудівної академії, 2006. – № 1Е(6). – С.14-18
4. Предметно-орієнтований метод побудови залежностей між метриками програмного забезпечення [Текст] / Дишлевий О.П. // Вісник НАУ, 2009. – № 3. – С. 206-212
5. The role of empiricism in improving the reliability of future software [Електронний ресурс] / Les Hatton // Testing: Academic and Industrial Conference 2008. Режим доступу: <http://www.leshatton.org/Documents/TAIC2008-29-08-2008.pdf>

Стаття надійшла до редакції 8 травня 2011 року.



ТОВ "ДКС Центр"