УДК 004.891.3: 004.3

О.В. ПОМОРОВА, Т.О. ГОВОРУЩЕНКО

Хмельницький національний університет, Хмельницький, Україна

СУЧАСНІ ПРОБЛЕМИ ОЦІНЮВАННЯ ЯКОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Проведено аналіз впливу якості програмного забезпечення (ПЗ) на роботу складних апаратно-програмних комплексів, стандартів якості ПЗ та моделей, методів і засобів оцінювання якості ПЗ, який дозволив виявити важливу тенденцію - необхідність оцінювання та прогнозування якості ПЗ на ранніх етапах життєвого циклу. Доведено суб'єктивну залежність та штучну пристосовуваність використовуваних концепцій, а також відсутність необхідних математичних моделей, теорії та методології в галузі оцінювання та забезпечення якості ПЗ.

Ключові слова: *програмне забезпечення (ПЗ), якість ПЗ, стандарти якості ПЗ, характеристики якості ПЗ, показники якості ПЗ, оцінювання якості ПЗ.*

Вступ

Виробництво ПЗ сьогодні — найбільша галузь світової економіки, в якій зайнято близько 3 млн фахівців (програмістів, розробників ПЗ та ін.). Ще кілька мільйонів осіб безпосередньо залежать від успішної діяльності виробників ПЗ.

Програмне забезпечення розробляють вже понад п’ятдесяти років, і за цей період задачі, які воно може вирішувати, рівень їх складності та форми представлення отриманих результатів кардинально змінилися. Але й дотепер розробка якісних програмних продуктів не стала нормою. Також потребують розвитку та вдосконалення методології розроблення надійного ПЗ з відповідними витратами та в межах заданого часу.

Джерела несправностей сучасного ПЗ вкрай різноманітні, і це лише ускладнює проблему, а також збільшує її масштаб та вартість.

На думку Д.Паттерсона, світова гонитва за продуктивністю призвела до залежності людини від технологій, і людству пора створювати такі інформаційні технології, на які світ дійсно може покластися, повністю довіряючи їм [1].

Звісно, хаотичний період розвитку ПЗ, коли значно більше уваги приділялось саме програмному коду, а не його якості, став відходити у минуле. За останні роки програмна індустрія досягла такого рівня розвитку, при якому вимоги до забезпечення якості стали обов’язковим пунктом договорів на предмет розроблення програмних систем, оскільки саме якість ПЗ є його найважливішою характеристикою з точки зору користувача.

Гарантування якості ПЗ – проблема, вирішення якої потребує комплексного дослідження за наступними напрямками:

1) розроблення засобів аналізу і оцінювання якості ПЗ на різних етапах його життєвого циклу (ЖЦ);

2) визначення і управління параметрами, які впливають на якість ПЗ на всіх етапах його ЖЦ.

1. Аналіз впливу якості ПЗ на функціонування складних апаратно-програмних комплексів

Криза у галузі забезпечення якості ПЗ була помітною ще більше 50 років тому - великі проекти стали виконуватися з відставанням від графіка або з перевищенням кошторису витрат, розроблений продукт не мав необхідних функціональних можливостей, продуктивність його була низька, якість програмного забезпечення не влаштовувала споживачів. При наявності ряду методів та засобів, залученні кращих фахівців для розроблення технологій та стандартів забезпечення якості програмних комплексів, якість ПЗ, як і раніше, залежить від знань та досвіду розробників.

Сучасне ПЗ характеризується: складністю опису; наявністю сукупності тісно взаємодіючих компонентів з локальними завданнями та цілями; відсутністю повних аналогів, а відтак і відсутністю можливості використання будь-яких типових проектних рішень; необхідністю інтеграції наявного і розроблюваного ПЗ; функціонуванням у неоднорідному середовищі на різних апаратних платформах; відокремленістю та різнорідністю окремих груп розробників із різним рівнем кваліфікації; великими термінами виконання проекту [2]. Отже, наразі істотною і невід’ємною властивістю програмних систем є їх складність. Постійне зростання складності функцій ПЗ неминуче призводить до збільшення їх обсягу та трудомісткості створення.

Щільність помилок, поява яких очікується в ПЗ різного розміру, представлена у табл. 1 [2].

Таблиця 1

Типова щільність помилок для ПЗ різного розміру

Розмір ПЗ	Типова щільність помилок
Менше 2 К	0-25 помилок на 1000 рядків коду
2К-16К	0-40 помилок на 1000 рядків
16К-64К	0,5-50 помилок на 1000 рядків
64К-512К	2-70 помилок на 1000 рядків
Більше 512К	4-100 помилок на 1000 рядків

З табл. 1 слідує, що сучасне ПЗ обсягом в мільйони рядків коду в принципі не може бути безпомилковим.

Проблема полягає в тому, щоб забезпечити потрібну якість ПЗ з врахуванням того, що деяка невідома кількість помилок та дефектів у ПЗ завжди залишається, а їх негативна дія повинна бути блокована або скорочена до допустимого рівня.

Отже, *актуальною* є проблема оцінювання існуючого рівня якості та створення методів і засобів досягнення необхідного рівня якості для проекту та розроблюваного ПЗ.

За наближеними оцінками витрати на розроблення ПЗ складають близько 275 мільярдів доларів, але лише 72% програмних проектів досягають етапу впровадження і всього 26% програмних проектів завершуються успіхом [3]. Програмні проекти часто зазнають невдач через неадекватне формулювання вимог, невдале проектування або неефективне планування, невірне розуміння або недостатній аналіз специфікації та проекту, тобто через помилки на ранніх етапах життєвого циклу ПЗ [2].

Проаналізуємо помилки *вбудованого ПЗ*, що призвели до відомих катастроф та інцидентів, які були внесені на ранніх етапах життєвого циклу. Такий аналіз наведено у табл. 2.

Таблиця 2

Аналіз помилок вбудованого ПЗ та їхніх наслідків

Подія	Причина	Наслідки
Етап формулювання вимог		
У 1971 р. "Космос-419" не стартував до Марсу [4]	Невірність специфікації	Втрата апарату
У 1971 р. станція "Марс 2" не змогла відстикуватись від корабля [4]	Невірність специфікації	Невиконання завдання
Аварія гелікоптера Chinook у 1994 році [4]	Невірність специфікації	Загинуло 29 чоловік
"Смертельні" сеанси радіаційної терапії із застосуванням Therac-25 [4-6]	Неповнота специфікації	6 пацієнтів одержали смертельну дозу опромінення
Вибух ракети-носія Ariane 5 у1996 році [4-6]	Протиріччя вимог щодо необхідності забезпечення надійності та максимально допустимого навантаження	Вартість обладнання та розроблення - 7,5 млрд. доларів, "упущена вигода" – 2 млрд. доларів
Етап проектування		
"Смертельні" сеанси радіаційної терапії із застосуванням Therac-25 [4-6]	Помилки у розробленні та постановці проекту; некоректні процедури оцінки та прогнозування ризиків	6 пацієнтів одержали смертельну дозу опромінення
Помилка процесора Intel Pentium у 1994 році [5]	Відхилення від специфікації	Втрати компанії – 475 млн. доларів
Аварія станції Mars Climate Orbiter та зникнення зв'язку із Mars Polar Lander у 1999 році [4, 6]	Помилка в проекті через використання різних одиниць вимірювання	327,6 млн. доларів - апарати та 91,7 млн. доларів - запуски
Аварійне падіння ракетоносія "Рокот" у 2005 році [7]	Логічна помилка в алгоритмі системи керування	Втрата європейського наукового супутнику CryoSat
Аварія аеробуса А330 у жовтні 2008 року [8]	Помилка у системному алгоритмі обробки даних	110 пасажирів та 9 членів екіпажу були поранені
Падіння ракети-носія "Зенит-3SL" у 2013 році [9]	Невірний проект системи керування	Втрата супутника "Intelsat-27"

Проаналізуємо тепер, які помилки *прикладного ПЗ* з катастрофічними наслідками були внесені на ранніх етапах життєвого циклу (табл. 3).

До помилок вбудованого ПЗ із катастрофічними наслідками також належить і відмова системи енергопостачання супутника "Коронас-Фотон" внаслідок помилкового проекту та неповноти тестування, яка призвела до втрати зв'язку із ним (2009 рік). Варто згадати також наступні помилки у прикладному ПЗ, які призвели до катастрофічних наслідків: 1) невірний розрахунок навантаження на ПЗ на етапі формулювання вимог призвів до падіння рейтингу та втрати 500 мільярдів доларів компанією Dow Jones Industrial Average на біржевих торгах (1987 р.) [6]; 2) невірно спроектована система розпізнавання Patriot призвела до помилкової ідентифікації британського бомбардувальника Tornado як ворожої ракети (2003 р.) - загинули 2 пілоти [5,6].

Аналіз таблиць 2 і 3 показує, що чимало помилок вбудованого і прикладного ПЗ, які спричинили катастрофи та інциденти, були внесені на ранніх етапах життєвого циклу.

Таблиця 3

Аналіз помилок прикладного ПЗ та їхніх наслідків

Подія	Причина	Наслідки
Етап формулювання вимог		
Збій у системі Нью-Йоркського банку [10]	Недостатність пам'яті через невірні вимоги	32 млрд. доларів
У 1990 р. на AT&T відбулась 9-годинна аварія [5, 6]	Проблеми з граничними умовами у специфікації	75 млн. нездійснених дзвінків, втрата 60 млн. доларів
У 1998 р. на AT&T відбулась 26-годинна аварія [5, 6]	Проблеми з прихованими граничними умовами у специфікації	Непрацездатність служб, пов’язаних з передачею даних
Помилка Y2K - помилка двоцифрового збереження року у даті (1999 р.) [5,6]	Невірність або неповнота специфікації	Втрати - 500 млрд. доларів
Етап проектування		
У 1983 р. на станції "Серпухов-15" спрацювала система виявлення [6]	Невірно спроектована система розпізнавання	Світ був на межі ядерної війни
У 1991 р. система протиракетної оборони Patriot не перехопила іракську ракету [4-6]	Помилка заокруглення - некоректний розрахунок місцезнаходження ракети, що наближалась	Загинули 28 американських солдат та близько 100 чоловік одержали поранення
Помилка Y2K - помилка двоцифрового збереження року у даті [5,6]	Невірний проект	Втрати 500 мільярдів доларів
"Смертельна" терапія у Національно-му онкологіч-ному інституті у Панама-Сіті в 2001 році [5]	Некоректне обчисення доз радіації у ПЗ компанії Multidata Systems International	28 пацієнтів зазнали надмірного опромінення, декілька хворих померли
Аварія на заводі по переробці урану у Західній Австралії в 2001 році [4]	Логічна помилка у алгоритмі	Викид радіоактивної речовини

Табл. 4 відображає відсоткову кількість помилок, які виникають на етапах формулювання вимог, проектування та реалізації ПЗ різного обсягу [2].

Таблиця 4

Розподіл помилок, припущених на різних етапах життєвого циклу

Етап ЖЦ	Обсяг ПЗ				
	2К	8К	32К	128К	512К
Форму-лювання вимог	До 10%	До 15%	До 20%	До 22%	До 23%
Проекту-вання	До 15%	До 19%	До 25%	До 28%	До 32%
Конст-руювання	До 75%	До 66%	До 55%	До 50%	До 45%

З табл. 4 видно, що помилки формулювання вимог та проектування складають 25-55% всіх помилок, причому чим більший обсяг ПЗ, тим більше помилок вноситься саме на ранніх етапах.

Слід врахувати також і факт, що вартість виправлення помилки проектування в 2-4 рази вища вартості виправлення помилки конструювання [2]. Залежність вартості виправлення помилок від етапу ЖЦ ПЗ наведено на рис. 1 [11].

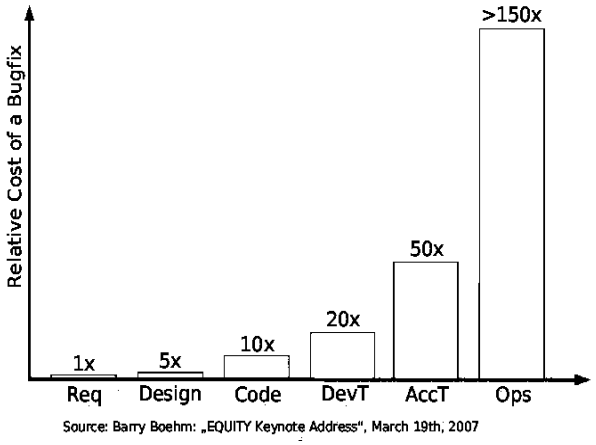

Рис. 1. Залежність вартості виправлення помилок від етапу життєвого циклу програмного забезпечення

Дослідження останніх 25 років переконливо довели високу вартість внесення змін, яких можна було уникнути. Так вчені з компаній Hewlett-Packard, IBM, Hughes, Aircraft, TRW виявили, що виправлення помилки до початку конструювання (реалізації) в 10-100 разів дешевше, ніж її усунення в кінці роботи над проектом, під час тестування або після його випуску [2], рис. 2.

Із збільшенням інтервалу між моментами внесення та виявлення дефекту вартість його виправлення сильно зростає. Чим довше помилка зберігається у ланцюгу розроблення ПЗ, тим сильніше вона проникає в інші частини ПЗ, тим більше шкоди завдає на наступних етапах і тим більше коштів доведеться витратити на її усунення. Наприклад, один дефект у вимогах може призвести до одного або декількох дефектів у проекті, які, в свою чергу, можуть призвести до появи множини дефектів у коді [2].

Отже, забезпечення можливості раннього виявлення помилок та оцінювання якості проекту і прогнозування рівня якості розроблюваного за проектом ПЗ на етапі проектування дали б можливість зменшити витрати на розроблення ПЗ, а то й уникнути ряду катастроф та інцидентів, причини яких були внесені на етапах формулювання вимог та проектування.

	Час виявлення дефекту				
Час внесення дефекту	Формулювання вимог	Проектування архітектури	Конструювання (кодування)	Тестування	Після випуску ПЗ
Формулювання вимог	1	3	5-10	10	10-100
Проектування архітектури	-	1	10	15	25-100
Конструювання (кодування)	-	-	1	10	10-25

Рис. 2. Середня вартість виправлення дефектів в залежності від часу їх внесення та виправлення (в умовних одиницях)

Наразі в більшості програмних проектів основна частина зусиль з виявлення та виправлення помилок все ще виконується на етапі тестування, а то й після випуску ПЗ, отже, на відлагодження та переробку йде близько 50% часу типового циклу розроблення ПЗ. В десятках компаній було виявлено, що політика раннього виправлення дефектів може в кілька разів знизити фінансові та часові витрати на розроблення ПЗ, що є вагомим аргументом на користь якомога більш раннього виявлення та усунення дефектів. Дослідження 50 проектів, на які витрачено понад 400 людинороків і які включили майже 3000000 рядків коду, проведені у Лабораторії проектування ПЗ NASA, показали, що підвищена увага до раннього контролю якості дозволяє істотно знизити рівень помилок, але не підвищує загальних витрат на розроблення [2].

В основу процесу розроблення ПЗ покладено фундаментальну ідею: проектування ПЗ є формальним процесом, який можна вивчати і вдосконалювати. Завдяки засвоєнню і правильному застосуванню методів і засобів створення ПЗ можна підвищити його якість, забезпечити керованість його процесу проектування і збільшити термін його життя. Але при тому, що складність деяких програмних розробок на сьогодні вже перевищує складність багатьох маши-

нобудівних або інфраструктурних проектів, а наслідки помилок є катастрофічними, процес розроблення ПЗ залишається не забезпеченим фундаментальною теорією та методологією надбання знань і навичок. Всі дослідження у галузі оцінювання якості на ранніх етапах життєвого циклу мають хаотичний, несистематизований характер, хоча, як доведено вище, саме в кінці етапу проектування можна й варто виявляти та усувати до 55% всіх помилок майбутнього ПЗ. Безумовно, є ряд фундаментальних досліджень (роботи Боема, Дейкстри, Мейєра), але відсутня завершена, протестована та апробована теорія та методологія розроблення якісного ПЗ, а також оцінювання та прогнозування якості ПЗ на ранніх етапах життєвого циклу. Отже, індустрія оцінювання якості програм потребує кардинальних змін, інакше світ і надалі очікуватимуть техногенні катастрофи та інциденти, викликані помилками програмного забезпечення, які проникає в усі сфери людської діяльності.

2. Аналіз стандартів якості програмного забезпечення

При розробленні ПЗ софтверні організації зобов'язані керуватись стандартами як щодо процесів розроблення, так і щодо процесів оцінювання та забезпечення якості. Саме стандарти забезпечують зв'язок між розробниками та користувачами та повинні містити у собі попередній досвід розробників. Але тривалий термін розроблення стандартів, їх несвоєчасне оновлення та вдосконалення призводять до гальмування розвитку технологій, відставання від практичних потреб та перешкоджання впровадженню новацій.

Згідно [12], можна виділити наступні основні критерії для оцінювання стандартів в галузі розроблення та оцінювання якості ПЗ: 1) довільність стандартної інтерпретації для оцінки відповідності; 2) значущість аналізу вимог до продукції; 3) можливість багаторазового використання процесу; 4) значущість керування; 5) можливість зміни та розвитку технологій; 6) відповідність теорії систем та системної інженерії; 7) відповідність теорії безпеки; 8) відповідність людському фактору; 9) розділення рівнів абстракції у стандарті; 10) цілісність та критичність рівнів абстракції; 11) визначення цілей та норм для зацікавлених сторін; 12) властивості (функції) компонентів та систем; 13) підтримка незалежної оцінки та сертифікації; 14) значущість етапу експлуатації. За кожним з наведених критеріїв з використанням 5-бальної шкали (від 0 до 4 балів) у [12] оцінено стандарти [14-16] і виявлено, що стандарт [13] має 25 балів, стандарт [14] - 32 бали, стандарт [15] - 10 балів. Даний аналіз показав, що стандарт [13], виданий у 2008 році, гірший за основними критеріями, ніж стандарт [14], виданий у 2003 році, тобто не завжди оновлені стандарти приймають попередній досвід розробників та найбільш інноваційні технології.

Деякі з сьогоднішніх стандартів не повністю відповідають вимогам до сучасного програмного забезпечення, а також всім потребам користувача. Наразі створено велику кількість стандартів, які порізному стандартизують та регламентують одні й ті ж процеси (як правило, рутинні масові процеси), внаслідок чого виникає неповне покриття об'єктів стандартизації, несумісність нормативних документів різних організацій, лобіювання інтересів окремих софтверних організацій та пристосування стандартів розробниками до своїх потреб.

Своєрідним виходом з такої ситуації є використання нормативної документації від одного-двух розробників, зокрема для оцінювання та забезпечення якості ПЗ розглядатимемо стандарти ISO та IEEE, оскільки саме ці організації мають найбільш розвинуту систему стандартів в галузі розроблення та оцінювання якості ПЗ [16].

Найчастіше *якість ПЗ* - це характеристика ПЗ, яка відображає ступінь його відповідності вимогам, тобто придатність ПЗ задовольняти певні потреби відповідно до призначення. При цьому існує ряд незалежних визначень поняття якості у різних стандартах. Згідно [17], якість - це повнота властивостей і характеристик продукту, процесу або послуги, які забезпечують здатність задовольняти оголошеним або передбачуваним потребам. Згідно [18], якість ПЗ - це ступінь, в якій воно володіє потрібною комбінацією властивостей. Згідно [19], якість програмного засобу - це сукупність властивостей програмного засобу, які обумовлюють його придатність задовільняти задані або передбачувані потреби у відповідності до його призначення. Стандарт [15] дає наступне визначення якості ПЗ - весь обсяг ознак та характеристик програмної продукції, який належить до її здатності задовольняти встановлені або передбачувані потреби.

Характеристика якості програмного засобу - набір властивостей програмного засобу, за допомогою яких описується та оцінюється його якість [19]. Згідно [15], характеристика якості ПЗ - набір властивостей (атрибутів) програмної продукції, за якими її якість описується і оцінюється. Характеристики якості можуть вимагатись в тій чи іншій мірі, можуть бути відсутніми або можуть задавати певні вимоги. *Показник якості програмного засобу* - характеристика якості програмного засобу, яка підлягає точному опису та вимірюванню і має кількісне значення [19].

Стандарт [20] об'єднує характеристики якості ПЗ з різних стандартів і пропонує наступну модель якості ПЗ – рис. 3.

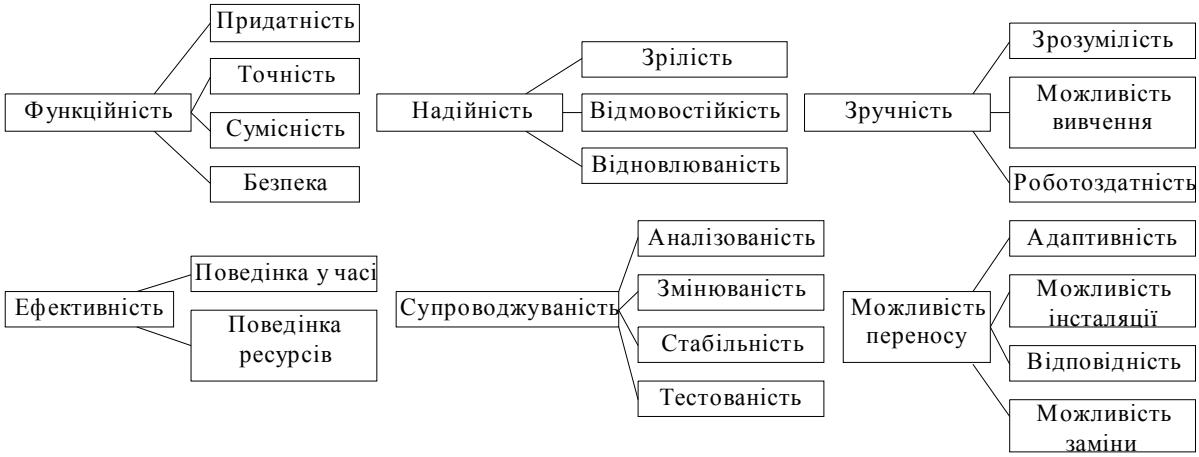

Рис. 3. Модель якості ПЗ за стандартом ISO/IEC 25010:2011 [20]

Отже, якість ПЗ (Q) згідно стандартів [15, 19, 20] є функцією від шести основних характеристик якості: функційності (F), надійності (R), зручності (U), ефективності (E), супроводжуваності (M), можливості переносу (P), які представляють собою значення з певного діапазону:

$$Q = f(F, R, U, E, M, P).$$

Очевидно, що, досягаючи максимального значення кожної з 6 характеристик якості, можна досягти максимального значення якості ПЗ.

На сьогодні оцінювання характеристик якості ПЗ відбувається наступним чином. Спочатку оцінюються показники якості ПЗ - обирається метрика, градуюється шкала оцінки в залежності від можливих ступенів відповідності показника накладеним обмеженням. Набір "вимірюваних" показників представляє собою критерій для оцінки характеристики.

Але вибір метрик для оцінювання показників якості ПЗ виконується суб'єктивно, оскільки створені тисячі метрик, але відсутні єдині стандарти щодо їх вибору та використання. Інтерпретація величин метрик також здійснюється суб'єктивно, оскільки відсутні стандартизовані "еталонні" значення метрик. Градуювання шкали оцінки знову-таки суб'єктивне, тому що залежить від можливих ступенів відповідності показника накладеним обмеженням, а ступені відповідності не стандартизовані і визначаються софтверною організацією.

Отже, оцінювання якості ПЗ як функції основних характеристик є суб'єктивним, оскільки софтверна організація обирає вигідні їй метрики для оцінювання показників, інтерпретує одержані значення обраних метрик як максимальні, градує шкалу оцінки кожної характеристики, виходячи з власної інтерпретації значень метрик та можливих ступенів відповідності показників обмеженням, в результаті чого одержує максимальні значення кожної характеристики, а відповідно й максимальне значення якості ПЗ. Насправді ж відбувається лише формальне задоволення якості ПЗ внаслідок неповного покриття стандартами об'єктів стандартизації, а також внаслідок вибору розробником вигідних для себе стандартів та пристосування цих стандартів до своїх потреб.

Крім того, відсутні комплексні методології, які дозволять оцінити не лише вплив кожної окремої характеристики на якість ПЗ (цьому питанню присвячений ряд робіт), але й дозволять оцінити взаємовплив характеристик.

3. Оцінювання якості програмного забезпечення

Якість ПЗ може підвищуватись шляхом ітеративного процесу постійного покращення. Це вимагає контролю, координації та зворотнього зв'язку в процесі управління багатьма одночасно виконуваними процесами (процесами ЖЦ, процесами виявлення, усунення та попередження збоїв/дефектів та процесами покращення якості).

При дослідженні *моделей якості* програмного забезпечення SquaRE [20] та CMM [14, 21] виявлено наступні проблеми:

1) незрілість технології вимірювання якості;

2) недостатня деталізованість та можливість різних трактувань стандартів та моделей якості в залежності від уявлень аудитора;

3) неточність оцінювання якості процесів за моделлю SQuaRE, задіяних при створенні та впровадженні ПЗ;

4) відсутність у моделі SQuaRE механізмів, які сприяють покращенню існуючих процесів;

5) модель CMM є власністю Software Engineering Institute (SEI) і не є загальнодоступною, тому подальша розробка моделі ведеться без залучення програмістської спільноти, причому оцінювання якості процесів організацій за моделлю CMM може проводитись лише спеціалістами, які пройшли спеціальне навчання та акредитовані SEI;

6) модель CMM орієнтована на застосування у відносно великих софтверних компаніях.

Для *одержання оцінки значень показників якості* за стандартами [15, 22, 23] використовуються такі *методи*: вимірювальний; реєстраційний; органолептичний; розрахунковий; експертний. Зрозуміло, що на етапі проектування ПЗ неможливо виміряти жодної характеристики ще не розробленого ПЗ, неможливо реєструвати моменти процесу виконання ще не існуючого ПЗ і неможливо сприйняти органами чуття інформацію щодо нерозробленого ПЗ. Отже, на етапі проектування є можливість визначати якість ПЗ лише із застосуванням розрахункових та експертних методів.

Найбільшого поширення набули наступні *методи керування якістю програмного забезпечення* [24]:

1) статичні методи (Static techniques) - передбачають детальне дослідження проектної документації, програмного забезпечення та іншої інформації про ПЗ без його виконання;

2) методи колективної оцінки (People-intensive techniques) - ідея полягає в формі прямої ("очної") взаємодії множини фахівців. Форма таких методів, включаючи оцінку і аудит, може варіюватись від формальних зборів до неформальних зустрічей або обговорення продукту навіть без звернень до його коду;

3) аналітичні методи (Analytical techniques) - базуються на специфіці застосовуваних інструментів та можуть передбачати "ручну" роботу. Методи включають різнорідну експертизу (assessment) як складовий елемент загального аналізу якості. Приклади таких методів - аналіз складності, аналіз керуючої логіки або аналіз контролю потоків керування, алгоритмічний аналіз. Кожен тип аналізу має конкретне призначення і не всі типи застосовні до будь-якого проекту.

4) формальні методи (Formal techniques) - застосовуються для перевірки вимог, перевірки коректності (застосовно до критичних фрагментів ПЗ) та верифікації особливо важливих частин критично-важливих систем; є корисними, але мають ряд обмежень – громіздкість та трудомісткість; незастосовність методів для ПЗ, яке складається з багатьох компонент;

5) динамічні методи (Dynamic techniques) - в основному, це методи тестування, а також методи моделювання, перевірки моделей та "символічного" виконання.

Дослідження методів оцінювання якості та керування якістю ПЗ показало, що, при наявності чималої кількості розроблених методів, оцінювання якості ПЗ не забезпечене фундаментальною методологією. Всі розроблені методи розрізнені, використовуються на розсуд розробників, оскільки відсутні чіткі критерії використання того чи іншого методу в кожному конкретному випадку.

Для автоматизації розрахунку показників якості програмного коду застосовуються різні "вимірювальні" програми, одні з яких досліджують характеристики ПЗ комплексно, інші - орієнтовані на цілком конкретні цілі (аналіз вихідного коду, розміру та структури окремих модулів).

До загальних недоліків *засобів оцінювання якості* віднесемо:

1) суб'єктивну залежність вибору метрик, які буде формувати засіб автоматизації побудови метричної інформації;

2) суб'єктивність інтерпретації метрик, адже точні (еталонні) значення метрик, з якими можна було б порівняти одержані метрики, відсутні;

3) спрямованість засобів автоматизації побудови метричної інформації на тестування та метрологію готового вихідного коду, а не на прогнозування і розрахунки метрик ПЗ на етапі проектування, коли готового коду ще немає, а є лише інформаційна, функційна та поведінкова моделі аналізу вимог;

4) всі існуючі автоматизовані засоби спрямовані максимум на оцінювання якості, але жоден з них не спрямований на підвищення або забезпечення належного рівня якості.

Отже, основою вдосконалення якості є попередження та раннє діагностування помилок, постійне вдосконалення ПЗ, увага до вимог замовника. Однак теорія та методологія в галузі оцінювання та забезпечення якості ПЗ, яка б враховувала вплив різних факторів при управлінні якістю, наразі відсутня, а використовувані концепції є суб'єктивно залежними. Існуючі методи і засоби оцінювання якості ПЗ неефективні на етапі проектування та не задовольняють сучасних вимог до ПЗ.

Висновки

Аналіз впливу якості ПЗ на роботу складних апаратно-програмних комплексів виявив важливу тенденцію, яку необхідно враховувати при оцінюванні та забезпеченні якості ПЗ: оцінювання якості проекту та прогнозування якості розроблюваного за проектом програмного забезпечення на етапі проектування надаватиме можливість раннього виявлення помилок, що забезпечить підвищення якості ПЗ та зменшення витрат на його розроблення. Актуальність проблем підвищення якості ПЗ обумовлює необхідність розроблення фундаментальної теорії та методології оцінювання і забезпечення (гарантування) якості програмного забезпечення.

Проведений *аналіз стандартів якості ПЗ* дозволив визначити, що деякі з сучасних стандартів у

галузі оцінювання якості ПЗ не відповідають потребам користувача щодо якості розроблюваного ПЗ через відсутність єдиних процедур стандартизації однакових процесів, неповне покриття об'єктів стандартизації, відсутність єдиних критеріїв вибору стандартів, пристосування стандартів розробниками до своїх потреб та можливість формального задоволення якості ПЗ.

Дослідження моделей, методів та засобів оцінювання якості ПЗ дало можливість виявити наступні характерні *проблеми*:

1) відставання та неефективність методів та засобів оцінювання якості ПЗ через неврахування та незадоволення ними сучасних вимог до ПЗ;

2) використання методів для процесів розроблення та оцінювання якості ПЗ має ряд обмежень через відсутність методології та критеріїв вибору методів;

3) суб'єктивна залежність та штучна пристосовуваність використовуваних концепцій в галузі оцінювання та забезпечення якості ПЗ;

4) відсутність моделей оцінювання та забезпечення якості ПЗ, які б враховували вплив різних факторів при управлінні якістю;

5) відсутність теорії та методології в галузі оцінювання та забезпечення якості ПЗ, яка б при використанні однакових технологій розроблення із застосуванням однакових стандартів гарантувала створення однаково якісного ПЗ з відповідними витратами та в межах заданого часу.

На вирішення цих проблем і будуть спрямовуватись подальші зусилля авторів.

Література

1. Patterson, D. Recovery oriented computing: a new research agenda for a new century [Текст] / D. Patterson // Proceedings of Eighth International Symposium on High-Performance Computer Architecture, 2002

2. Макконнелл С.. Совершенный код. Мастер-класс [Текст] / С. Макконнелл. – М.: Русская редакция, 2013. –- 896 с.

3. Мищенко, В.О., CASE-оценка критических программных систем. В 3-х томах. Т. 1. Качество[Текст] / В.О. Мищенко, О.В. Поморова, Т.А. Говорущенко; под ред. В.С. Харченко – Х.: Нац. аэрокосмический университет "ХАИ", 2012. – 201 с.

4. Цена программной ошибки [Електронний ресурс]. – Режим доступу: http://www.cusoft.ru /error.php. – 18.11.2012 г.

5. Software goes wrong, we all know that, but just how wrong can it go? [Electronic resource]. – Access mode: http://www.datareservoir.co.uk/bugs/. –18.11.2012 г.

6. 20 Famous Software Disasters [Electronic resource]. – Access mode: http://sandipsandilya.wordpress.com/2011/01/17/20-famous-software-disasters/. – 18.11.2012 г.

7. Software glitch blamed for CryoSat loss [Electronic resource]. – Access mode: http:// www. theregister.co.uk/2005/10/27/cryosat_downed_by_software/. –18.11.2012 г.

8. Qantas terror blamed on computer [Electronic resource]. – Access mode: http://www.stuff.co.nz/travel/ australia/6163633/Qantas-terror-blamed-on-computer. – 18.11.2012 г.

9. Падение ракеты-носитель "Зенит-3SL" // [Електронний ресурс]. – Режим доступу: http:// ukrday.com/politika/novosti.php?id=57865. –18.11.2012 г.

10. Explaining Settlement Fails [Electronic resource]. – Access mode: http://www.ny.frb.org/ research/current_issues/ci11-9/ci11-9.html. – 18.11.2012 г.

11. Cost of a bug within a software lifecycle [Electronic resource]. – Access mode: http://www .testically.org/2012/02/09/cost-of-a-bug-within-a-software-lifecycle/. –18.11.2012 г.

12. Fusani, M. Examining software engineering requirements in safety-related standards [Текст] / M.Fusani. // Радіоелектронні і комп'ютерні системи – 2009 – № 7. – С. 268 – 274.

13. ISO/IEC 12207 FDAM Information technology – Software life cycle processes [Текст] / ISO/IEC, 2008.

14. ISO/IEC 15504-2 Information technology -- Process assessment -- Part 2: Performing an assessment [Текст] / ISO/IEC, 2003

15. ISO/IEC 9126-1 Software engineering- Product quality- Part 1: Quality model [Текст] / ISO/IEC, 2001

16. Скляр В.В. Оценка качества и экспертиза программного обеспечения. Лекционный материал [Текст] . - Харьков: НАУ "ХАИ", 2008. - 204 с.

17. ISO 8402:1994 Quality management and quality assurance [Текст] / ISO/IEC, 1994

18. IEEE Std. 1061-1998 IEEE Computer Society: Standard for Software Quality Metrics Methodology [Текст] , 1998 - 20 p.

19. ГОСТ 28806-90. Качество программных средств. Термины и определения. // [Електронний ресурс] - Режим доступу: http://vsegost.com/ Catalog/10/10605.shtml. –18.11.2012 г.

20. ISO/IEC 25010:2011. Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models[Текст] / ISO/IEC, 2011

21. Chrissis M. CMM: Guidelines for Process Integration and Product Improvement [Текст] / M. Chrissis, M. Konrad, S. Shrum. - Addison-Wesley Professional, 2006.

22. ГОСТ 28195-89. Оценка качества программных средств. Общие положения [Текст].

23. ISO/IEC 14598-6 Software engineering - Product evaluation / ISO/IEC, 2001 [Текст].

24. Guide to the Software Engineering Body of Knowledge (SWEBOK) [Текст]. – A project of the IEEE Computer Society Professional Practices Committee, 2004

Надійшла до редакції 28.02.2013, розглянута на редколегії 13.03.2013

Рецензент: д-р техн. наук, проф., зав. каф. комп'ютерних систем і мереж В.С. Харченко, Національний аерокосмічний університет ім. М.Є. Жуковського «ХАІ», Харків, Україна.

СОВРЕМЕННЫЕ ПРОБЛЕМЫ ОЦЕНИВАНИЯ КАЧЕСТВА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

О.В. Поморова, Т.А. Говорущенко

Проведен анализ влияния качества программного обеспечения (ПО) на работу сложных аппаратно-программных комплексов, стандартов качества ПО, а также моделей, методов и инструментов оценивания качества ПО, который позволил обнаружить важную тенденцию - необходимость оценивания и прогнозирования качества ПО на ранних этапах жизненного цикла. Доказана субъективная зависимость и искусственная приспосабливаемость используемых концепций, а также отсутствие необходимых математических моделей, теории и методологии в области оценивания и обеспечения качества ПО.

Ключевые слова: программное обеспечение (ПО), качество ПО, стандарты качества ПО, характеристики качества ПО, показатели качества ПО, оценивание качества ПО.

THE ACTUAL PROBLEMS OF SOFTWARE QUALITY EVALUATION

O.V. Pomorova, T.O. Hovorushchenko

The authors have analyzed the impact of the software quality for work of complex hardware-software systems, the software quality standards, the models, methods and tools of software quality evaluation. The analysis provides to discover an important tendency - the need of software quality evaluation and prediction at the early stages of the life cycle. The authors have proved the subjective dependence and artificial adaptable of used concepts and the lack of mathematical models, theory and methodology in the area of software quality evaluation and assurance.

Key words: software, software quality, software quality standards, software quality characteristics, software quality indicators, software quality evaluation.

Поморова Оксана Вікторівна – доктор технічних наук, професор, завідувач кафедри системного програмування Хмельницького національного університету; e-mail: o.pomorova@gmail.com.

Говорущенко Тетяна Олександрівна – кандидат технічних наук, с.н.с., доцент кафедри системного програмування Хмельницького національного університету; e-mail: tat_yana@ukr.net.