

УДК 004.415.53

С.О. ВОЛКОВА, О.М. ТРУНОВ

Миколаївський державний гуманітарний університет ім. Петра Могили, Україна

ДОСЛІДЖЕННЯ ІСНУЮЧИХ ПІДХОДІВ ПІДВИЩЕННЯ ЯКОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КРИТИЧНОГО ЗАСТОСУВАННЯ

Розглянуто існуючі методи та засоби оцінки якості та моделі керування якістю програмного забезпечення (ПЗ). Описано методологію та обґрунтовано застосування технології розробки ПЗ через тестування (TDD) для систем критичного застосування з метою підвищення якості та надійності на прикладі медичних програмно-апаратних комплексів.

програмне забезпечення, якість, надійність, моделі та атрибути якості, розробка через тестування, ітерація

Враховуючи основні тенденції розвитку інформаційних технологій на сучасному етапі становлення інформаційного суспільства, виникає потреба в підвищенні якості та надійності програмного забезпечення (ПЗ). Не викликає сумніву той факт, що вже відомі розроблені методології та технології контролю якості та надійності ПЗ, які знаходять широке застосування в процесі розробки програмно-апаратних комплексів критичного застосування (див. рис. 1.), дозволяють ефективно проводити оцінку вищевказаних характеристик програмних продуктів (ПП) [1-3].

Саме тому варто відзначити внесок відомих науковців у вирішення завдання забезпечення якості ПЗ: в Україні - Харченко В.С. [4], Коваль Г.І. [5], Тоценко В.Г. [6] та ін.; у СНД - Липаев В.В. [7], Корольков Ю.Д. [8] та ін. [9,10]; за кордоном - Boehm [11], Avizenis A. [12] та ін. [13].

Однак, через прагнення розробників скоротити собівартість ПЗ, формуються нові концепції щодо процесів його розробки та застосування новітніх технологій створення, таких як: Rapid Application Development (RAD) [14], Extreme Programming (XP) [15], Agile Software Development (ASD) [16], з використанням таких методів як: Test Driven Development (TDD) [17-18], Behavior Driven Development (BDD) та Feature Driven Development (FDD) [19-20]. В зв'язку з цим актуальності набуває завдання підвищення якості ПП

шляхом модифікації існуючих методів, технологій та моделей якості.

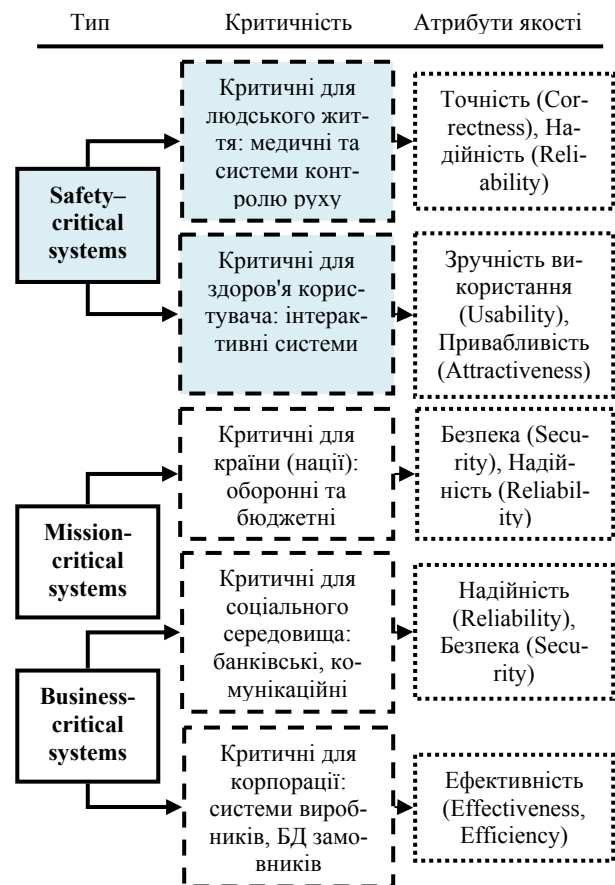


Рис. 1. Типологія систем критичного застосування

У контексті завдання підвищення якості ПЗ варто звернути увагу на медичні програмно-апаратні комплекси (МПАК), які відносяться до типу Safety-Critical Systems, вищевказаної типології систем критичного

застосування (рис. 1). Забезпечення якості та надійності медичних систем є не менш важливою науковою задачею, зважаючи на статистику відмов МПАК [21], яка приведена по основних галузях медицини та зображена на рис.2.

Для формування наукових гіпотез щодо процесу забезпечення якості МПАК розглянемо розробку програмного забезпечення критичного застосування у галузі медицини з використанням технології Test Driven Development (TDD) [22], тобто розробки через тестування та сформуємо загальні рекомендації щодо підтримки, оцінки та підвищення їх якості, та відповідно надійності.

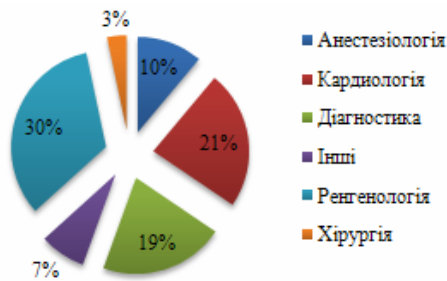


Рис. 2. Статистичні дані щодо відмов МПАК

Аналіз літературних джерел в області якості та надійності ПЗ дозволив сформуванати визначення якості у відповідності до відомих спеціалістів в галузі ІТ таких як: Джозефа А. Джурана (Joseph A. Juran) [23], Волтера А. Шевхарта (Walter A. Shewhart) [24], Філіпа Б. Кросбі (Philip B. Crosby) [25], В. Едвардса Демінга (W. Edwards Deming) [26], Арманда В. Фейгенбаума (Armand V. Feigenbaum) [27], Каору Ішікава (Kaoru Ishikawa) [28]. Отже, якість ПЗ – динамічна характеристика, яка визначає відповідність кінцевого ПП вимогам замовника та характеризує відсутність дефектів в ньому.

Огляд наукових досліджень в області якості ПЗ дозволяє зробити висновок, що основною проблемою в досліджуєміій галузі є розробка конструктивних підходів до побудови базової моделі якості ПЗ, котра б була прийнятною для різних класів та методологій розробки ПЗ, та визнавалась одночасно розробником, замовником і користувачами. Шляхом проведення аналізу відомих моделей якості ПЗ визначено еволюцію під-

ходів до оцінки якості та побудови моделей якості ПЗ, яка представлена на рис. 3.

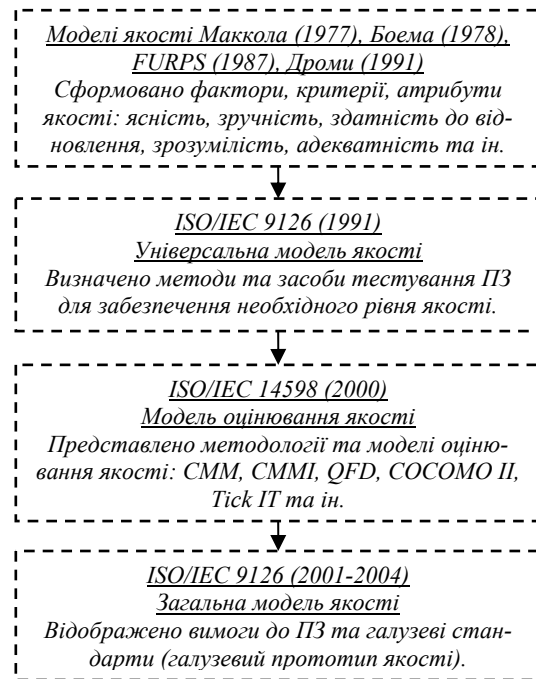


Рис. 3. Еволюцію підходів до оцінки якості ПЗ

На основі проведеного аналізу щодо існуючих моделей оцінки якості ПЗ здійснено порівняльну характеристику складових якості ПЗ за різними моделями (табл.1).

Таблиця 1
Порівняльна характеристика атрибутів якості ПЗ за різними моделями

Атрибути якості	1	2	3	4	5
Тестуємість (Testability)	+	+		+	
Правильність (Correctness)			+		
Результативність (Efficiency)	+	+	+	+	+
Зрозумілість (Understandability)	+			+	
Надійність (Reliability)	+	+	+	+	+
Гнучкість (Flexibility)		+	+		
Функціональність (Functionality)			+	+	+
Інженерна психологія (Human Engineering)	+				
Цілісність (Integrity)		+	+		
Сумісність (Interoperability)		+	+		
Завершеність (Maturity)					+
Ремонтпридатність (Maintainability)	+	+	+	+	+
Модифікуємість (Changeability)	+				
Переносимість (Portability)	+	+		+	+
Повторне використання (Reusability)		+			+
Зручність використання (Usability)		+	+	+	+

1 - Модель Боєма, 2 - Модель Макгола, 3 - Модель FURPS, 4 - Стандарт ISO 9126, 5 - Модель Дроїми

В галузі якості існує безліч стандартів - національних, галузевих, відомчих, корпоративних і т.д. Основні поняття про якість ПЗ, її складові характеристики, відповідну документацію та сертифікацію зафіксовані у відповідних державних стандартах України: ДСТУ 2844-94 [29], ДСТУ 2850-94, ДСТУ 2851-94, ДСТУ 2853-94, ДСТУ 3415-96, ДСТУ 3419-96 та ін.

Згідно ДСТУ 2850-94 і ДСТУ 2844-94 під *якістю* розуміють сукупність властивостей, що визначають ступінь придатності ПЗ для використання за призначенням, а також виділяються наступні характеристики якості програмної продукції: функціональність, надійність функціонування, зручність використання, раціональність, супроводжуємість та мобільність.

Міжнародним стандартом якості ПЗ є ISO 9126:2001 [30]. Він складається з наступних частин під загальним заголовком "Інформаційна технологія – характеристики та метрики якості програмного забезпечення": Частина 1. "Характеристики та субхарактеристики якості"; Частина 2. "Зовнішні метрики якості"; Частина 3. "Внутрішні метрики якості"; Частина 4. "Метрики якості у використанні".

Модель внутрішніх і зовнішніх характеристик якості ПЗ у відповідності до ISO 9126:2001 складається із шести груп базових показників, кожна з яких деталізована набором нормативних субхарактеристик. Додатково кожна характеристика супроводжується субхарактеристикою узгодженість, що повинна відображати відсутність протиріч з іншими стандартами і нормативними документами, а також з іншими показниками в даному стандарті.

З точки зору процесів вимірювання якості, дана характеристика ПЗ поділяється на *внутрішню якість ПЗ* – якість, яка вимірюється за статистичними властивостями коду; *зовнішню якість ПЗ* – якість, яка вимірюється за динамічними властивостями коду в процесі виконання; *якість використання ПЗ* – якість, яка вимірюється за показником, яко-

му ПЗ відповідає за вимогами користувача в робочому середовищі.

Більш детально взаємозв'язок між різними представленнями якості ПЗ приведено на рис. 4.

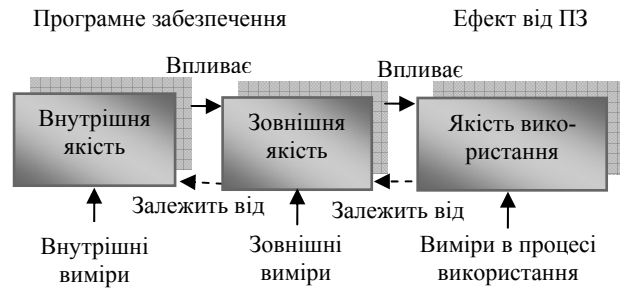


Рис. 4. Взаємозв'язок між видами якості ПЗ

Характеристики, субхарактеристики та атрибути якості ПЗ із позиції можливості та точності їх виміру можна розділити на три типи: *категорійний* - описовий, що відбиває набір властивостей і загальні характеристики об'єкта - його функції, які можуть бути представлені номінальною шкалою категорій-властивостей; *кількісний* - представляється множиною впорядкованих, числових точок, що відбивають неперервні закономірності та описуються інтервальною або відносною шкалою, які можна об'єктивно виміряти та чисельно зіставити з вимогами; *якісний* – складається з кількох впорядкованих або окремих значень – категорій, які характеризуються порядковою або точковою шкалою набору категорій, що встановлюються, вибираються та оцінюються в значній мірі суб'єктивно та експертно.

Переходячи до процесу забезпечення якості, необхідно визначити поняття *інженерії якості*, що являє собою процес забезпечення у програмних продуктах властивостей, які характеризують їх якість. У відповідності до [30] у проблематиці якості розрізняють три аспекти її забезпечення: *запобігання* появи дефектів у програмному продукті, *виявлення* та усунення дефектів безпосередньо на тих стадіях життєвого циклу, на яких вони були внесені, а також підвищення якості під час *тестування* продукту. Враховуючи вищеописане, призначення про-

цесів інженерії якості ПЗ необхідно переглянути у відповідності до новітніх технологій розробки ПЗ [14-20, 22].

Як результат детального огляду відомих методологій загального керування якістю ПЗ, таких як: TQM (Total Quality Management) [31]; розгортання процесів забезпечення якості QFD (Quality Function Deployment) [32]; гнучкі технологічні лінії (Lean Software development, Quality Improvement Paradigm); сходинки до якості (CMM [33], SPICE, Bootstrap, Trillium), виникає необхідність вдосконалення процесу керування якістю ПЗ на всіх етапах ЖЦ ПЗ для вищеписаних новітніх методологій розробки ПЗ [14-20, 22].

У відповідності до стандартів якості ПЗ, одним із найуживаніших методів забезпечення якості є процес тестування ПЗ [34], що визначається як спостереження за функціонуванням ПЗ в специфічних умовах з метою визначення ступеню відповідності програмного продукту вимогам. Концепція забезпечення якості ПЗ за рахунок тестових специфікацій приведена у відповідності до стандарту ISO 9126 та продемонстрована на рис. 5.

Зіставимо тестування з іншими методами оцінки якості ПЗ. Для цього скористаємося звітами по проєкту SCOPE [35], в якому виділено засоби та методи оцінки різних характеристик якості програмного забезпечення. Отже, для оцінки якості ПЗ використовуються такі методи як:

1. *Вивчення документів* з метою пошуку проблемних місць і перевірки відповідності стандартам, стилям, прийнятним правилам та угодам: цільове вивчення коду (code inspection); цільове вивчення документації (documents inspection).

2. *Формальний аналіз*: формальний доказ властивостей ПЗ (formal verification).

3. *Аналіз*: автоматичний аналіз коду (static analysis); аналіз архітектури та проєкту (architecture review, design review); аналіз розробки (process analysis).

4. *Вимірювання*: визначення метрик ПЗ, проєкту, документації; вимірювання продуктивності (benchmarks); профілювання (profiling).

5. *Моделювання шляхом використання моделей для оцінки властивостей ПЗ*: моделі використання (usability model); моделі надійності (reliability model); моделі функціонування (model checking); прототипування (prototyping).

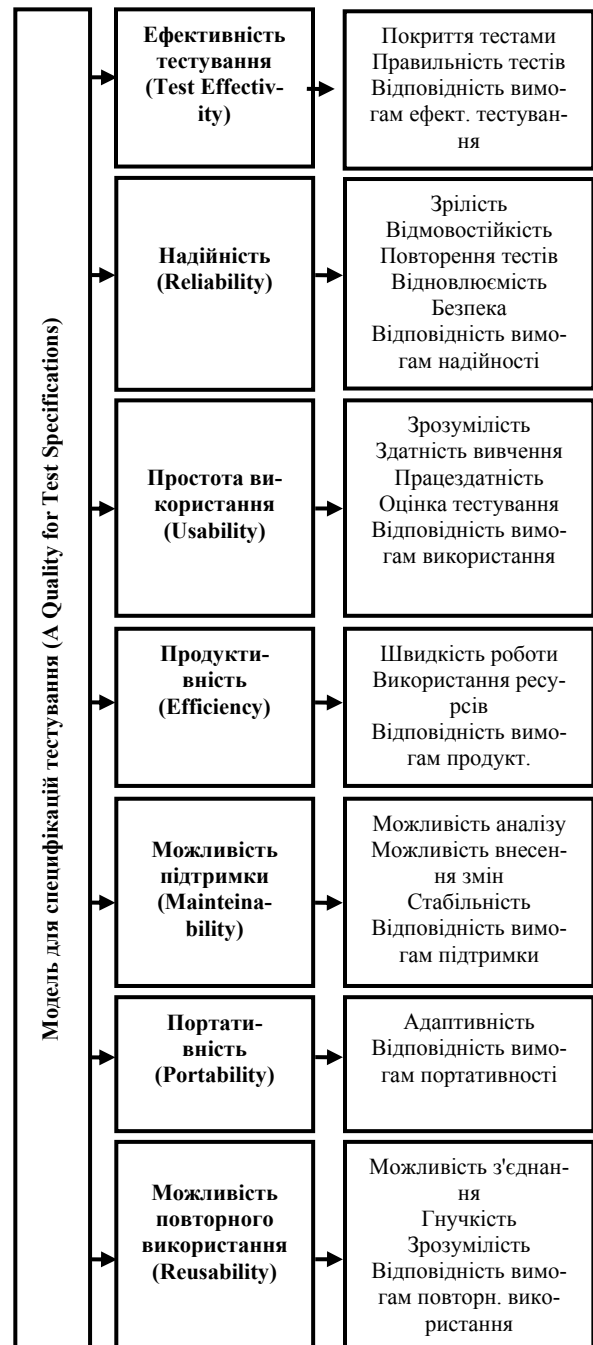


Рис. 5. Модель для специфікацій тестування ПЗ

Крім того існують такі методи оцінки якості ПЗ, що засновані на її суб'єктивному сприйнятті та інтуїції експертів.

В результаті аналізу існуючих методів оцінки якості ПЗ можна зробити висновок, що вищеописані методи забезпечення та керування якістю ПЗ, які засновані на системному підході та передбачають формування комплексної оцінки якості програмного продукту, не можуть бути застосовані в технології TDD повною мірою.

Однак, за аналогією з існуючими методами оцінки та підвищення якості ПЗ технологія TDD дозволяє: проводити модульне тестування та тестування функціональності; виявляти помилки в процесі виконання ПЗ; проводити аналіз коду щодо покриття тестами.

Техніка програмування Test Driven Development є однією з основних практик екстремального програмування, при якій модульні тести для ПЗ або його фрагменти пишуться до процесу кодування самого ПЗ, та як результат управляють процесом його розробки.

Розробка в стилі TDD складається з коротких циклів (ітерацій), що наведено на рис. 6.

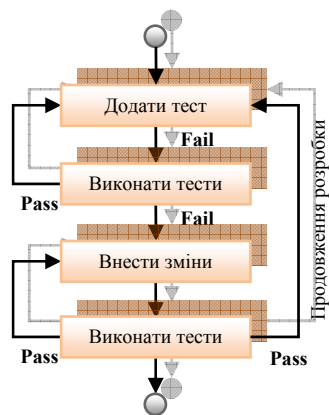


Рис. 6. Візуалізація ітерації за технологією TDD
Базові характеристики Test Driven ітерації:

- починається зі створення набору тестів;
- закінчується успішним виконанням тестів.

Виникнення помилки при компіляції та помилкове виконання набору тестів не є обов'язковим, що в свою чергу підтверджує ймовірність успішного

виконання TDD ітерації. На рис. 7а та 7б продемонстровано можливі варіанти проходження ітерації за технологією TDD.

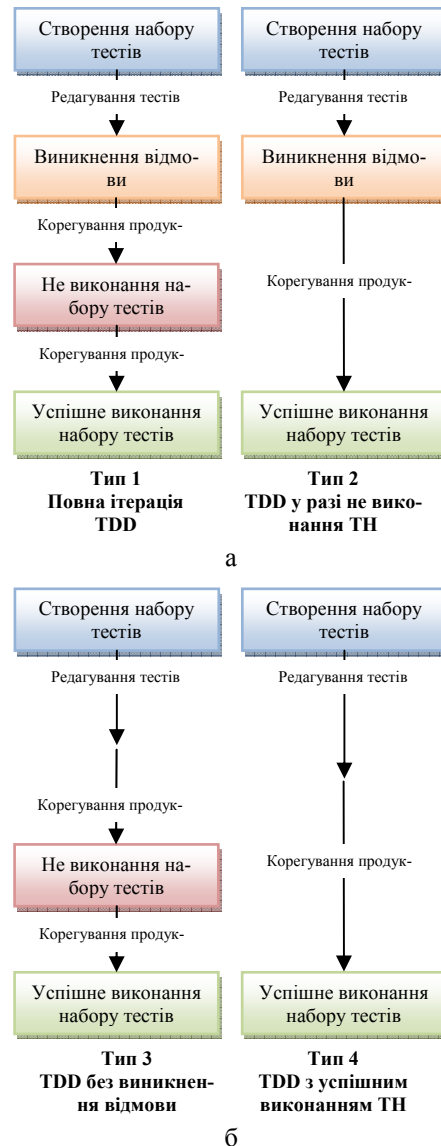


Рис. 7. Можливі варіанти виконання ітерації в TDD

Більш того, варто відмітити переваги застосування методології Test Driven Development контексті підвищення якості програмних засобів критичного застосування:

1. Тести запобігають появі помилок у новому коді, що безпосередньо підвищує якість програмного продукту на всьому циклі його розробки.
2. Тести дозволяють проводити рефакторинг коду, що по-перше дозволяє поліпшити дизайн коду та по-друге цілком замінює такі вищеописані методи підвищення якості ПЗ як: статистичний аналіз коду

та архітектури, формальну верифікацію та ін. Крім того рефакторинг коду замінює процес верифікації ПЗ з використанням таких методик як: наскрізний контроль, колегіальні перевірки, інспекції та, поперше, підвищує якість ПЗ, по-друге, зменшує часові та виробничі затрати на розробку ПЗ.

3. Тести можуть використовуватися як документація, що взагалі виключає необхідність цільового вивчення документації для пошуку проблемних місць та перевірку відповідності стандартам з метою підвищення якості ПП.

4. Тести сприяють підвищенню кваліфікації розробників (PSP, Personal Software process та TSP, Team Software process) та прискорюють процес розробки.

Отже, відповідно до вимог сучасного замовника, та з врахуванням новітніх концепцій розробки ПЗ вирішено завдання підвищення якості та надійності програмно-апаратних комплексів критичного застосування (МПАК) з використанням технології розробки TDD шляхом формування рекомендацій щодо модифікації існуючих методів, технологій та моделей якості ПЗ. Проведений аналіз моделей та підходів до оцінки якості ПЗ визначає:

– необхідність *вдосконалення процесу керування якістю ПЗ на всіх етапах ЖЦ* та необхідність *перегляду призначення процесів інженерії якості ПЗ* для новітніх методологій розробки ПП;

– необхідність *створення конструктивних концепцій побудови базової моделі якості ПЗ*, котра б була прийнятною для різних класів та сучасних технологій розробки ПП.

Література

1. IEEE 1012-1998. IEEE Standard for Software Verification and Validation. – New York: IEEE, 1998. – 81 p.

2. Тарасюк О.М., Харченко В.С. Динамические радиальные метрические диаграммы в задачах управления качеством программного обеспечения //

Зб. наук. праць ін-ту проблем моделювання в енергетиці ім. Г.Є. Пухова. Вип. 22. – К.: НАНУ, ИПМЕ, 2003. – С.202-205.

3. Харченко В.С., Тарасюк О.М., Гордеев А.А. Мамутов С.С. Инструментальные средства оценки качества программного обеспечения // Материалы конф. «Информационные технологии – в науку и образование». – Х: Харьк. нац. ун-т радіоелектроніки, 2005. – С. 98-99.

4. Харченко В.С., Тарасюк О.М. Оценка экспертизы программного обеспечения: показатели, методика и инструментальные средства. Информационные технологии и безопасность // Сб. научн. тр. – К.: НАНУ, Ин-т проблем регистрации информации, 2003. – Вып. 4. – С. 128-139.

5. Андон Ф.И., Коваль Г.И. Основы инженерии качества программных систем. – К.: Академперіодика, 2002. – 503 с.

6. Тоценко В. Корректность, устойчивость, точность ПО. – К.: Наукова думка, 1990. – 197 с.

7. Липаев В.В. Обеспечение качества программных средств. Методы и стандарты. – М.: СИНТЕГ, 2001. – 380 с.

8. Корольков Ю. Д.. Математические модели качества программных средств. – Иркутск: Издательство Иркутского университета, 1995. – 160 с.

9. Интеллектуализация и качество ПО под ред. В.Н. Касьянова. – Новосибирск: РАН, 1994. – 184 с.

10. Стекольников Ю.И. Живучесть систем: Теоретические основы. – Спб: «Политехника», 2002. – 155 с.

11. B. W. Boehm, J. R. Brown, H. Kaspar, M. Lipow, G. MacLeod, and M. J. Merritt. Characteristics of Software Quality. North Holland, 1978.

12. Avizienis A. and Kelly J. Fault tolerance by design diversity: Concepts and experiments. IEEE Computers, August 1984. – P. 67-80,

13. Trauboth H. Software-Qualitätssicherung: konstruktive und analytische. München, 1993. – 357 p.

14. James Martin: Rapid Application Development, Macmillan Coll Div.

15. Beck K., Extreme Programming Explained: Embrace Change, Addison-Wesley, 1999.
16. Manifesto for Agile Software Development, Agile Alliance, 2001 [Electronic resource]. – Data mode: Manifesto-for-Agile-Software-Dev.
17. Agilian - All-in-one Modeling Tool supporting Agile Modeling. Support latest UML, BPMN, ERD, DFD and Textual Analysis [Electronic resource]. – Data mode: <http://www.visual-paradigm.com/product/ag/>.
18. Beck, K. Test-Driven Development by Example, Addison Wesley, 2003.
19. Palmer, S.R., & Felsing, J.M., A Practical Guide to Feature-Driven Development. Prentice Hall, 2002.
20. Introduction to Behavior Driven Development, [Electronic resource]. – Data mode: <http://behaviour-driven.org/>.
21. Dolores R. Wallace, D. Richard Kuhn, Failure modes in medical device software: an analysis of 15 years of recall data [Electronic resource]. – Data mode: http://www.sticky-minds.com/s.asp?F=S6449_ART_2.
22. Craig Murphy, Improving Application Quality Using Test-Driven Development [Electronic resource]. – Data mode: <http://www.methodsandtools.com/archive/archive.php?id=20>.
23. Juran, Joseph M., Frank M. Gryna, Juran's Quality Control Handbook. McGraw-Hill.
24. Walter A. Shewhart, Economic Control of Quality of Manufactured Product // 50th Anniversary Commemorative Issue. American Society for Quality December 1980. – 501 p.
25. Grady R., Caswell D. Software Metrics: Establishing a Company-Wide Program, Prentice Hall, 1987.
26. Gabor A., The Man Who Discovered Quality: How W. Edwards Deming Brought the Quality Revolution to America. Penguin, 1992.
27. Feigenbaum A.V. Total Quality Control, McGraw-Hill Professional, 2004
28. The Legacy Of Ishikawa, Greg Watson, Quality Progress. – April, 2004. – P. 54- 57.
29. ДСТУ 2844-94. Програмні засоби ЕОМ. Забезпечення якості. Терміни та визначення. Введ. 1.08.1995. К.: Держстандарт України, 1995. – 57 с.
30. ISO/IEC 9126-2001 Software engineering – Product quality (Part 1 – 4).
31. Dubois, HFW (2002). "Harmonization of the European vaccination policy and the role TQM and re-engineering could play". Quality Management in Health Care 10 (2): 47-57.
32. Akao, Yoji [1994]. "Development History of Quality Function Deployment", The Customer Driven Approach to Quality Planning and Deployment. Minato-ku, Tokyo 107 Japan: Asian Productivity Organization, 339 p.
33. Capability Maturity Model / M.C. Paulk, B. Curtis, M.B. Chrissis, Ch.V. Weber // IEEE Software. – 1993. – Vol. 10, № 4. – P. 18-27.
34. Канер С., Фолк., Нгуен Е. Тестирование программного обеспечения. – К.: ДиаСофт, 2000. – 554 с.
35. A. K. Rae, H. L. Hausen, and P. Robert (Editors). Software Evaluation for Certification: Principles, Practice and Legal Liability. McGraw Hill, International Software Quality Assurance Series, 1995.

Надійшла до редакції 13.02.2008

Рецензент: д-р техн. наук, проф. М.Т. Фісун, МДГУ ім. Петра Могили комплексу «Києво-Могилянська академія», Київ.