

УДК 004.05

DOI <https://doi.org/10.32782/tnv-tech.2024.5.11>

АНАЛІЗ МОДЕЛЕЙ ЯКОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Тузова І. А. – доцент кафедри технічної кібернетики й інформаційних технологій імені професора Р. В. Мерктя

Одеського національного морського університету

ORCID ID: 0009-0002-4198-378X

Тузов О. В. – старший викладач кафедри технічної кібернетики й інформаційних технологій імені професора Р. В. Мерктя

Одеського національного морського університету

ORCID ID: 0009-0006-5443-4957

Панченко Т. Д. – старший викладач кафедри технічної кібернетики

й інформаційних технологій імені професора Р. В. Мерктя

Одеського національного морського університету

ORCID ID: 0009-0007-4629-9537

Чумак О. А. – старший викладач кафедри технічної кібернетики й інформаційних технологій імені професора Р. В. Мерктя

Одеського національного морського університету

ORCID ID: 0009-0002-5802-9765

Стародуб В. І. – старший викладач кафедри технічної кібернетики

й інформаційних технологій імені професора Р. В. Мерктя

Одеського національного морського університету

ORCID ID: 0009-0009-4601-2326

Забезпечення якості програмного засобу включає в себе дії, котрі проводяться на кожній стадії його розробки. Ціль – надати гарантію того, що продукт відповідає функціональним та нефункціональним вимогам.

На перший погляд, «якість програмного забезпечення» може здаватись абстрактним поняттям. Проте для менеджерів проекту, програмістів, спеціалістів з тестування, QA-інженерів та інших учасників процесу розробки продукту критерії якості прозорі та вимірювані.

Отже, якість програмного забезпечення – це комплекс характеристик програмного продукту, який визначає здатність виконувати покладені на нього функції. Аналіз якості програмного забезпечення включає початкову оцінку, яка відповідає процесам виробництва якісного програмного продукту: моніторингу, плануванню, виконанню, зміні або підтримці. Поняття якості програми багатопланове й може бути виражене адекватно тільки деякою структурованою системою характеристик або атрибутів. Така система характеристик називається моделлю якості.

Базова модель якості включає характеристики і відноситься до будь-якого типу програмних продуктів. Під час розробки вимог замовник формує ті вимоги до якості, які найбільше підходять для програмного продукту, який замовляється.

Визначення і планування якості ПЗ ґрунтується на положеннях стандартів у галузі, складанні планів-графіків робіт, процедури перевірки та ін. План забезпечення якості включає набір дій для перевірки процесів забезпечення якості (верифікація, валідація тощо) і формування документа з управління якістю. Управління якістю застосовується до процесів, продуктів і ресурсів, а також включає вимоги до процесів та їх результатів.

Планування якості включає:

- визначення продукту в термінах заданих характеристик якості;
- планування процесів для отримання необхідної якості;
- вибір методів оцінки планованих характеристик та встановленню відповідності продукту сформульованим вимогам.

Мета дослідження: проаналізувати моделі якості програмного забезпечення. *Предмет дослідження:* моделі якості за Макколом, за Боемом, FURPS, Гецці, SATC, ISO 9126. *Методи дослідження:* аналіз, узагальнення та систематизація науково-методичних джерел з питань моделей якості для визначення найбільш оптимальної для використання. *Основні результати й висновки:* після попереднього аналізу різних моделей якості програмного забезпечення було визначено, що деякі з них є логічним розширенням продовженням попередніх моделей з використанням метрик якості, які дозволяють легше вимірювати й оцінювати якість програмних засобів.

Ключові слова: фактори, метрики, атрибут якості, технічні вимоги, функціональність, продуктивність, зручність використання, стандарт.

Tuzova I. A., Tuzov O. V., Panchenko T. D., Chumak O. A., Starodub V. I. Analysis of software quality models

Software quality assurance includes activities that are carried out at each stage of its development. The goal is to ensure that the product meets functional and non-functional requirements.

At first glance, 'software quality' may seem like an abstract concept. However, for project managers, programmers, testers, QA engineers, and other participants in the product development process, quality criteria are transparent and measurable.

So, software quality is a set of characteristics of a software product that determines its ability to perform its functions. Software quality analysis includes an initial assessment that corresponds to the processes of producing a quality software product: monitoring, planning, execution, modification, or support. The concept of software quality is multifaceted and can only be adequately expressed by some structured system of characteristics or attributes. Such a system of characteristics is called a quality model.

The basic quality model includes characteristics and applies to any type of software product. When developing requirements, the customer formulates those quality requirements that are most suitable for the software product being ordered.

Determining and planning software quality is based on the provisions of industry standards, drawing up work schedules, verification procedures, etc. The quality assurance plan includes a set of actions to verify the quality assurance processes (verification, validation, etc.) and create a quality management document. Quality management applies to processes, products and resources, and includes requirements for processes and their results.

Quality planning includes:

- defining the product in terms of specified quality characteristics;*
- planning processes to achieve the required quality;*
- selection of methods for evaluating the planned characteristics and establishing the product's compliance with the formulated requirements.*

Purpose of the study: to analyse software quality models. Subject of research: quality models by McCall, Boehm, FURPS, Gezzi, SATC, ISO 9126. Research methods: analysis, synthesis and systematisation of scientific and methodological sources on quality models to determine the most optimal one for use. Main results and conclusions: after a preliminary analysis of various software quality models, it was determined that some of them are a logical extension of previous models using quality metrics that make it easier to measure and evaluate the quality of software.

Key words: factors, metrics, quality attribute, technical requirements, functionality, productivity, ease of use, standard.

Вступ. Досягнення якості програмного забезпечення залежить від процесу проектування, який повинен бути планованим і включати систематизований набір дій із забезпечення адекватності й довіри до продукту, створеного відповідно до поставлених технічних вимог. При цьому верифікація та валідація програмного забезпечення належать до управління якістю і є важливими процесами забезпечення якості програмного продукту на етапах його життєвого циклу. Програмне забезпечення повинно виконувати свої функції, відповідати заданим критеріям якості, безпеки, надійності. Оцінка продукту, вимог до нього, проектної документації – задача інженерів із забезпечення якості, або QA-інженерів.

Виклад основного матеріалу. На перший погляд, «якість програмного забезпечення» може здаватись абстрактним поняттям. Проте для менеджерів проекту, програмістів, спеціалістів з тестування, QA-інженерів та інших учасників процесу розробки продукту критерії якості прозорі та вимірювані.

Отже, якість програмного забезпечення – це сукупність характеристик програмного забезпечення, які відносяться до його здатності задовольняти встановлені та запропоновані потреби [1].

Функція забезпечення якості може виконуватись внутрішнім відділом компанії, а може делегуватись незалежному підрядчику, котрий об'єктивно оцінить саме рішення, налаштує процеси забезпечення якості і тим самим дозволить випустити на ринок продукт високої якості, який відповідає бізнесвимогам та очікуванням користувачів.

Однак для підвищення ступеня задоволення всіх користувачів програмного забезпечення, для досягнення ним стабільного становища на ринку й підвищення потенціалу розвитку важливо враховувати всі характеристики. Таким чином, якість програмного забезпечення може бути описана великою кількістю різноманітних характеристик.

Характеристики програмного забезпечення [2]:

1. **Функціональність (Functionality)** – визначається здатністю програмного забезпечення вирішувати задачі, котрі відповідають зафіксованим та запропонованим потребам користувача при заданих умовах використання програмного забезпечення. Тобто це характеристика відповідає за те, що програмне забезпечення працює справно та точно, функціонально сумісне, відповідає стандартам галузі та захищене від несанкціонованого доступу.

2. **Надійність (Reliability)** – здатність програмного забезпечення виконувати необхідні задачі у визначених умовах протягом заданого проміжку часу чи вказану кількість операцій. Атрибути даної характеристики – це завершеність та цілісність усієї системи, здатність самостійно та коректно відновлюватись після збоїв в роботі, відмовостійкість.

3. **Зручність використання (Usability)** – можливість легкого розуміння, вивчення, використання та привабливості програмного забезпечення для користувачів.

4. **Ефективність (Efficiency)** – здатність програмного забезпечення забезпечувати необхідний рівень продуктивності у відповідності із виділеними ресурсами, часом та іншими визначеними умовами.

5. **Зручність супроводу (Maintainability)** – легкість з якою програмне забезпечення може аналізуватись, тестуватись, змінюватись для виправлення дефектів, для реалізації нових вимог, для полегшення подальшого обслуговування та адаптуватись до змін навколишнього середовища.

6. **Портативність (Portability)** – характеризує програмне забезпечення з точки зору легкості його переносу з одного оточення в інше.

Виходить, поняття якості програми багатопланове й може бути виражене адекватно тільки деякою структурованою системою характеристик або атрибутів. Така система характеристик називається моделлю якості.

Модель якості програмного забезпечення за Макколом. Першою широко відомою моделлю якості програмного забезпечення стала запропонована в 1977 році Макколом й іншими модель [3]. У ній характеристики якості розділені на три групи:

- фактори (factors), що описують програмне забезпечення з позицій користувача та заданих вимог;
 - критерії (criteria), що описуються програмне забезпечення з позицій розробника й задаються як цілі;
 - метрики (metrics), що використовуються для кількісного опису й вимірювання якості.
-

Фактори якості, яких було виділено 11, групуються в три групи відповідно до роду роботи людей з програмним забезпеченням. Отримана структура зображується у вигляді трикутника Маккола. Критерії якості – це числові рівні факторів, поставлені як цілі при розробці. Об'єктивно оцінити або виміряти фактори якості досить важко. Тому Маккол використав метрики якості, які дозволяють легше вимірювати й оцінювати. Оцінки за шкалою набувають значень від 0 до 10.

Метрики якості:

- зручність перевірки на відповідність стандартам (audiability);
- точність управління й обчислень (accuracy);
- ступінь стандартності інтерфейсів (communication commonality);
- функціональна повнота (completeness);
- однорідність використовуваних правил проектування й документації (consistency);
- ступінь стандарності форматів даних (data commonality);
- стійкість до помилок (error tolerance);
- ефективність роботи (execution efficiency);
- розширюваність (expandability);
- широта сфери потенційного використання (generality);
- незалежність від апаратної платформи (hardware independence);
- повнота протоколювання помилок й інших подій (instrumentation);
- модульність (modularity);
- зручність роботи (operability);
- захищеність (security);
- самодокументованість (selfdocumentation);
- простота роботи (simplicity);
- незалежність від програмної платформи (software system independence);
- можливість порівняння проекту з вимогами (traceability);
- зручність навчання (training).

Кожна метрика впливає на оцінку декількох факторів якості. Числове вираження фактора являє собою лінійну комбінацію значень метрик, що впливають на нього (рис.1). Коефіцієнти вираження визначаються по різному відповідно до різних організацій, команд розробки, видів програмного забезпечення.

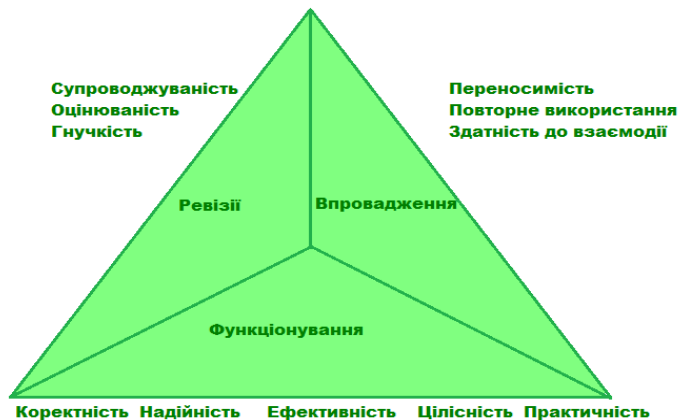


Рис. 1. Трикутник Маккола

Модель якості програмного забезпечення за Боемом. В 1978 році Боем запропонував свою модель, власне кажучи, вона являла собою розширену модель Маккола. Атрибути якості класифікуються за способами використання програмного забезпечення. Визначено 19 проміжних атрибутів, що включають всі 11 факторів якості за Макколом [3]. Проміжні атрибути розділяються на примітивні, які у свою чергу можуть бути оцінені на основі метрик.

На додаток до факторів Маккола атрибути якості за Боемом включають таке:

- ясність (clarity);
- зручність внесення змін (modifiability);
- документованість (documentation);
- здатність до відновлення функцій (resilience);
- зрозумілість (understandability);
- адекватність (validity);
- функціональність (functionality);
- універсальність (generality);
- економічну ефективність (economy).

Модель якості програмного забезпечення FURPS. У 1987 році компанія Hewlett Packard запропонувала нову модель якості, так би мовити індустріальну інтерпретацію моделей Маккола та Боема, FURPS.

Модель містить 5 атрибутів, відповідно до назви, а саме:

- функціональність (functionality);
- зручність використання (usability);
- надійність (reliability);
- продуктивність (performance);
- супроводжуваність (supportability).

Сучасна модель FURPS+ містить додаткові атрибути: вимоги до проекту (Design requirements), вимоги до реалізації (Implementation requirements), вимоги до інтерфейсу (Interface requirements), вимоги до фізичних компонент (Physical requirements) [3].

Модель якості програмного забезпечення FURPS, яка була запропонована Грейді та Hewlett Packard, як вже зазначалось, побудована на основі моделей Маккола та Боема, але на відміну від них складається з двох шарів: перший з яких визначає характеристики, а другий пов'язаний із атрибутами. Основною концепцією, яка лежить в основі моделі, є декомпозиція характеристик програмного забезпечення на дві категорії вимог, а саме, функціональні (F) та не функціональні (URPS) вимоги. Ці виділені категорії можуть бути використані як в якості вимог до програмного продукту, так і в якості оцінки якості програмного продукту. На даний момент модель FURPS+ широко застосовується у розробці програмного забезпечення та при ідентифікації вимог до розроблюваної системи доцільне використання цієї моделі у якості універсального контрольного переліку характеристик програмного забезпечення.

До недоліків даної моделі якості програмного забезпечення можна віднести наступне:

- вказану модель варто було б доповнити атрибутами якості, які мають безпосереднє відношення до веб-додатків: доступність, знаходжуваність та правову відповідність;
- вказана модель не приділяє достатньої уваги переносимості програмного забезпечення.

Модель якості програмного забезпечення Гецці. Карло Гецці та його співавтори розрізняють якість продукту та процесу [3]. Згідно моделі Гецці до якості

програмного забезпечення відносяться наступні характеристики програмного забезпечення:

- цілісність;
- надійність та стійкість;
- продуктивність;
- практичність;
- верифікованість;
- супроводжуваність;
- можливість багаторазового використання;
- мобільність;
- зрозумілість;
- можливість взаємодії;
- ефективність;
- своєчасність реагування;
- видимість процесу розробки.

Модель якості програмного забезпечення SATC. В Центрі забезпечення якості програмного забезпечення NASA (Software Assurance Technology Center, SATC) була розроблена програма метрик, яка забезпечувала оцінку ризиків проекту, якості продукції та ефективність процесів [3]. Програма SATC рекомендує окремо відслідковувати якість вимог, якість програмного забезпечення та інших продуктів (документації), якість тестування та якість виконання процесів. Модель якості SATC визначає набір цілей, пов'язаних із програмним продуктом та атрибутами процесів у відповідності до структури моделі якості програмного забезпечення ISO 9126-1.

Модель якості програмного забезпечення ISO 9126. У 1991 році стандартною була прийнята модель якості програмного забезпечення ISO 9126. Ця модель не є прямим розширенням раніше запропонованих. У ній оцінка якості програмного забезпечення заснована на таких характеристиках [1]:

- мета (goals) – те, що ми хочемо бачити в програмному забезпеченні;
- атрибути (attributes) – властивості програмного забезпечення, що показують наближення до мети;
- метрики (metrics) – кількісні характеристики ступеня наявності атрибутів.

Виділено шість цілей:

- функціональність (functionality);
- надійність (reliability);
- практичність або зручність використання (usability);
- ефективність (efficiency);
- супроводжуваність (maintainability);
- мобільність (portability).

В 2001 році цей стандарт був переглянутий і розширений. У нього було додано шість додаткових атрибутів якості: привабливість як атрибут практичності й ступінь відповідності стандартам. Повний список атрибутів якості ПЗ за стандартом ISO 9126 (рис.2):

- функціональність (functionality);
- надійність (reliability);
- практичність або зручність використання (usability);
- ефективність (efficiency);
- супроводжуваність (maintainability);
- мобільність (portability).



Рис. 2. Характеристики і атрибути якості програмного забезпечення згідно з ISO 9126

Висновки. Перераховані характеристики і атрибути якості програмного забезпечення дозволяють систематично описувати вимоги до нього, визначаючи, які властивості програмного забезпечення по даній характеристиці можуть бачити зацікавлені сторони.

Наведені атрибути якості визначені в стандартах, але це не означає, що вони цілком вичерпують поняття якості програмного забезпечення. Так, в стандарті ISO 9126 відсутні характеристики, пов'язані з мобільністю ПЗ, тобто здатність програми працювати при фізичних переміщеннях машини на якій вона працює.

Замість надійності багато дослідників воліють розглядати більш загальне поняття добротності (dependability), що описує здатність програмного забезпечення підтримувати визначені показники якості за основними характеристиками (функціональності, продуктивності, зручності використання) із заданими ймовірностями виходу за їх рамки та визначенням максимальним збитком від можливих порушень. Крім того, активно досліджуються поняття зручності використання, безпеки і захищеності програмного забезпечення – вони здаються більшості фахівців набагато більш складними, ніж це описується даним стандартом.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ:

1. ISO/IEC 9126-1:2001. Software engineering – Software product quality – Part 1: Quality model.
2. Табунщик Г.В. Інженерія якості програмного забезпечення: навчальний посібник / Г.В. Табунщик, Р.К. Кудерметов, Т.І. Брагіна. – Запоріжжя: ЗНТУ, 2013. 180с.
3. Процеси та системи підтримки якості програмних систем [Електронний ресурс]. Режим доступу: <https://dspace.uzhnu.edu.ua/jspui/bitstream/lib/16456/1/> (дата звернення: 03.10.2024).
4. Проектування інформаційних систем [Електронний ресурс]. Режим доступу: https://elearning.sumdu.edu.ua/free_content/lectured:delc9452f2a161439391120eef364dd8ce4d8e5e/20160217112601/165292/index.html (дата звернення: 10.11.2024).

5. Специфікація вимог до ПЗ [Електронний ресурс]. Режим доступу: <https://studfile.net/preview/5375896/> (дата звернення: 03.10.2024).

6. Принципи встановлення вимог розробки системи [Електронний ресурс]. Режим доступу: http://org2.knuba.edu.ua/pluginfile.php/28593/mod_resource/content/1 (дата звернення: 03.11.2024).

REFERENCES:

1. ISO/IEC 9126-1:2001. Software engineering – Software product quality – Part 1: Quality model.

2. Tabunshik G.V., Kudermetov R.,K., Bragina T.I. (2013) Software quality engineering: a study guide. Zaporizhzhia: ZNTU. (75-90)

3. Processes and systems for supporting the quality of software systems. Retrieved from <https://dspace.uzhnu.edu.ua/jspui/bitstream/lib/16456/1/>

4. Information systems design. Retrieved from https://elearning.sumdu.edu.ua/free_content/lectured:de1c9452f2a161439391120eef364dd8ce4d8e5e/20160217112601/165292/index.html

5. Specification of software requirements. Retrieved from <https://studfile.net/preview/5375896/>

6. Principles of establishing system development requirements. Retrieved from http://org2.knuba.edu.ua/pluginfile.php/28593/mod_resource/content/1