

Резанова Вікторія Георгіївна кандидат технічних наук, доцент кафедри комп'ютерної інженерії та електромеханіки, Київський національний університет технологій та дизайну, м. Київ, <https://orcid.org/0000-0003-4630-2701>

ДОСЛІДЖЕННЯ МОДЕЛЕЙ ЯКОСТІ ТА НАДІЙНОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Анотація. У даній роботі розглянуто теоретичні та практичні аспекти забезпечення якості та надійності програмного забезпечення в контексті програмної інженерії. Проаналізовано сутність поняття якості ПЗ як багатовимірної характеристики, що визначається сукупністю властивостей програмного продукту, процесів його створення та супроводу. Детально досліджено ієрархічну модель якості програмного забезпечення, яка включає характеристики, атрибути, метрики та оціночні елементи, що забезпечують комплексне оцінювання.

Особливу увагу приділено характеристикам якості, таким як функціональність, надійність, зручність використання, ефективність, супроводжуваність та переносимість, а також методам їх кількісного оцінювання. Розглянуто сучасні підходи до побудови метрик якості програмного забезпечення, їх класифікацію та застосування на різних етапах життєвого циклу програмних систем.

У роботі проаналізовано методи стандартної оцінки показників якості, включаючи вимірювальні, реєстраційні, розрахункові та експертні підходи, а також їх комбіноване використання для отримання об'єктивних результатів. Значну увагу приділено процесам управління якістю програмних систем, зокрема плануванню, забезпеченню та контролю якості відповідно до міжнародних стандартів.

Окремий розділ присвячено моделям оцінки надійності програмного забезпечення, їх математичному апарату та практичному застосуванню. Розглянуто основні поняття теорії надійності, такі як імовірність безвідмовної роботи, інтенсивність відмов та середній час між відмовами. Проаналізовано класифікації моделей надійності.

Отримані результати дослідження дозволяють зробити висновок про важливість використання моделей якості та надійності для підвищення ефективності розроблення програмних продуктів, зниження ризиків та забезпечення відповідності сучасним вимогам програмної інженерії.

Ключові слова: якість програмного забезпечення, надійність ПЗ, метрики, програмна інженерія, життєвий цикл, математичні моделі

Rezanova Viktoriia Georhiivna PhD in Engineering, Associate Professor of the Department of Computer Engineering and Electromechanics, Kyiv National University of Technologies and Design, Kyiv, <https://orcid.org/0000-0003-4630-2701>

RESEARCH ON SOFTWARE QUALITY AND RELIABILITY MODELS

Abstract. This paper examines the theoretical and practical aspects of ensuring software quality and reliability in the context of software engineering. The essence of the concept of software quality as a multidimensional characteristic, determined by the set of properties of a software product, the processes of its creation and support, is analyzed. The hierarchical model of software quality is studied in detail, which includes characteristics, attributes, metrics and evaluation elements that provide a comprehensive assessment.

Particular attention is paid to quality characteristics such as functionality, reliability, usability, efficiency, maintainability and portability, as well as methods for their quantitative assessment. Modern approaches to building software quality metrics, their classification and application at different stages of the software systems life cycle are considered.

The paper analyzes methods for standard assessment of quality indicators, including measurement, registration, calculation and expert approaches, as well as their combined use to obtain objective results.

Considerable attention is paid to the processes of quality management of software systems, in particular, planning, quality assurance and control in accordance with international standards.

A separate section is devoted to software reliability assessment models, their mathematical apparatus and practical application. The basic concepts of reliability theory are considered, such as the probability of failure-free operation, failure intensity and mean time between failures. The classifications of reliability models are analyzed.

The obtained research results allow us to conclude about the importance of using quality and reliability models to increase the efficiency of software product development, reduce risks and ensure compliance with modern software engineering requirements.

Keywords: software quality, software reliability, metrics, software engineering, life cycle, mathematical models

Постановка проблеми. Сучасний розвиток інформаційних технологій зумовлює стрімке ускладнення програмних систем, що вимагає застосування інженерних підходів до їх розроблення, аналізу та оцінювання. Особливого значення набувають питання забезпечення якості та надійності програмного забезпечення, оскільки саме вони визначають ефективність функціонування систем, їх безпеку та конкурентоспроможність. Якість розглядається як комплексна характеристика, що формується протягом усього життєвого циклу програмного забезпечення. При цьому важливу роль відіграють моделі якості, які дозволяють формалізувати вимоги, а також моделі надійності, що забезпечують кількісну оцінку стабільності функціонування систем.

Аналіз останніх досліджень і публікацій. Проблематика забезпечення якості та надійності програмного забезпечення є однією з центральних у програмній інженерії та активно досліджується як у класичних, так і в сучасних наукових працях. Формування підходів до оцінювання якості ПЗ бере початок із робіт, присвячених стандартизації характеристик програмних продуктів та їх формалізації.

Сучасні дослідження у сфері якості та надійності програмного забезпечення демонструють перехід від класичних моделей до адаптивних, контекстно-орієнтованих підходів, що враховують специфіку домену, архітектуру систем і сучасні методології розроблення.

У роботах останніх років значна увага приділяється розвитку та адаптації моделей якості програмного забезпечення. Зокрема, систематичний огляд моделей якості показує, що сучасні підходи базуються на міжнародних стандартах, але водночас адаптуються до конкретних умов застосування (наприклад, для open-source або AI-систем) [1]. Дослідження підтверджують, що універсальні моделі поступово доповнюються спеціалізованими, орієнтованими на контекст використання, що підвищує точність оцінювання якості.

Актуальні публікації підкреслюють, що якість програмного забезпечення розглядається як багатовимірна характеристика, що включає функціональність, ефективність, сумісність, безпеку та інші властивості відповідно до сучасних стандартів [2]. При цьому надійність визначається як один із ключових атрибутів якості, що відображає здатність системи працювати без відмов у заданих умовах експлуатації [3].

Окремий напрям досліджень присвячений метрикам якості та методам їх оцінювання. Сучасні підходи включають використання багатокритеріальних методів прийняття рішень (MCDM), таких як АНР, TOPSIS і PROMETHEE, що дозволяють формалізувати процес вибору моделей якості та враховувати суперечливі критерії [4-7]. Крім того, активно застосовуються методи аналітики даних і машинного навчання для прогнозування дефектів та оцінювання якості програмних систем.

У сфері надійності програмного забезпечення сучасні дослідження зосереджені на розвитку моделей зростання надійності (Software Reliability Growth Models - SRGM). У новітніх роботах пропонуються нові моделі, що враховують реалістичні фактори процесу тестування, зокрема недосконалість налагодження, S-подібний характер виявлення помилок та невизначеність, змодельовану за допомогою розподілів.

Такі підходи дозволяють точніше описувати поведінку програмних систем у процесі тестування та експлуатації.

Також сучасні дослідження акцентують увагу на використанні нових математичних підходів для моделювання надійності. Наприклад, застосування стохастичних диференціальних рівнянь і теорії невизначеності дозволяє враховувати невизначеність, пов'язану з людським фактором та складністю програмного коду [8-17].

Крім того, досліджуються методи оптимізації параметрів моделей надійності з використанням метаевристичних алгоритмів, що підвищує точність прогнозування [18].

Важливим напрямом сучасних досліджень є інтеграція процесів забезпечення якості з сучасними підходами розроблення програмного забезпечення. Зокрема, відзначається зростання ролі автоматизованого тестування та безперервної інтеграції, що дозволяє підвищити якість програмних продуктів і зменшити кількість дефектів [19]. Водночас дослідники звертають увагу на нові виклики, такі як складність тестування, управління тестовими даними та пояснюваність результатів.

Окрім цього, сучасні дослідження підкреслюють важливість відтворюваності результатів у програмній інженерії. Аналіз великої кількості наукових робіт показує, що незважаючи на розвиток інструментів і методів, проблема відтворюваності експериментів залишається актуальною, що впливає на достовірність оцінювання якості програмного забезпечення [20].

Таким чином, огляд сучасної літератури свідчить, що розвиток моделей якості та надійності програмного забезпечення характеризується такими тенденціями:

- переходом до адаптивних і доменно-орієнтованих моделей;
- інтеграцією стандартів якості з аналітичними та інтелектуальними методами;
- удосконаленням моделей надійності з урахуванням невизначеності та реальних умов тестування;
- активним впровадженням автоматизації у процесі забезпечення якості.

Це підтверджує актуальність подальших досліджень у напрямі створення комплексних моделей, здатних забезпечити високий рівень якості та надійності програмних систем у сучасних умовах.

Мета статті: Метою даної роботи є аналіз моделей якості та надійності програмного забезпечення, а також дослідження методів їх оцінювання в процесі розроблення програмних систем. Для досягнення поставленої мети необхідно вирішити такі завдання: дослідити структуру моделі якості програмного забезпечення, проаналізувати характеристики та метрики якості, розглянути методи їх кількісного оцінювання, вивчити підходи до управління якістю програмних систем, а також дослідити основні моделі оцінки надійності та їх класифікацію.

Виклад основного матеріалу.

1. Модель якості програмного забезпечення

Якість програмного забезпечення є складною багатовимірною характеристикою, яка визначається здатністю програмного продукту задовольняти встановлені та очікувані потреби користувачів. Вона не є абсолютною величиною, оскільки залежить від умов експлуатації, вимог замовника та особливостей предметної області.

У сучасній програмній інженерії якість розглядається як сукупність трьох взаємопов'язаних компонентів: якості продукту, якості процесів життєвого циклу та якості супроводу. Такий підхід дозволяє оцінювати не лише кінцевий результат, але й ефективність процесів, що призводять до його створення.

Модель якості програмного забезпечення має ієрархічну структуру. На верхньому рівні знаходяться характеристики якості, які відображають загальні властивості програмного продукту. Наступний рівень представлений атрибутами, що деталізують окремі аспекти характеристик. Далі використовуються метрики, які забезпечують кількісне вимірювання властивостей. На нижньому рівні знаходяться оціночні елементи, що дозволяють узагальнити результати оцінювання.

Формально якість програмного забезпечення можна представити у вигляді інтегрального показника:

$$Q = \sum_{i=1}^n w_i \cdot q_i$$

де q_i – значення i -го показника якості, w_i – ваговий коефіцієнт, що визначає його важливість.

1.1 Характеристика показників якості

Функціональність програмного забезпечення визначає його здатність виконувати задані функції відповідно до вимог. Вона характеризується повнотою реалізації функціональних можливостей, точністю отриманих результатів, здатністю до взаємодії з іншими системами та рівнем захищеності даних.

Надійність є однією з ключових характеристик, що визначає здатність програмного забезпечення працювати без відмов протягом певного часу. Вона залежить від кількості помилок, що залишилися в системі після тестування, і визначається такими параметрами, як безвідмовність, стійкість до помилок і здатність до відновлення.

Зручність використання відображає ступінь адаптованості системи до користувача. Вона включає зрозумілість інтерфейсу, легкість навчання та ефективність взаємодії.

Ефективність характеризує використання ресурсів системи та швидкість виконання операцій. Вона безпосередньо впливає на продуктивність системи та її економічну доцільність.

Супроводжуваність визначає складність внесення змін до програмного забезпечення. Вона залежить від структури коду, його зрозумілості та рівня документуваності.

Переносимість характеризує здатність програмного забезпечення функціонувати в різних середовищах, що є важливим у сучасних умовах різноманітності платформ.

1.2 Метрики якості програмного забезпечення

Метрики є основним інструментом кількісного оцінювання якості програмного забезпечення. Вони дозволяють об'єктивно оцінювати властивості продукту, процесу розробки та результатів його використання.

Однією з найпоширеніших метрик є кількість рядків коду (*LOC*), яка використовується для оцінки розміру програмного продукту. Однак ця метрика має обмеження, оскільки не враховує складність алгоритмів.

Більш інформативною є метрика цикломатичної складності Маккейба:

$$V(G) = E - N + 2P$$

де E – кількість ребер графа керування, N – кількість вершин, P – кількість зв'язаних компонентів.

Для оцінки надійності використовується щільність дефектів:

$$D = \frac{N_d}{LOC}$$

де N_d – кількість дефектів, LOC – кількість рядків коду.

Метрики процесу дозволяють оцінити ефективність розробки. Наприклад, середній час виправлення помилки визначається як:

$$MTTR = \frac{\sum t_i}{n}$$

де t_i – час усунення i -ї помилки.

1.3 Стандартна оцінка значень показників якості

Оцінювання якості програмного забезпечення здійснюється на основі вимірювання атрибутів та їх подальшої інтеграції. Важливим етапом є нормалізація показників, яка дозволяє привести різні метрики до єдиної шкали.

Для цього використовується формула:

$$q_i^{norm} = \frac{q_i - q_i^{min}}{q_i^{max} - q_i^{min}}$$

де q_i^{min} та q_i^{max} – мінімальне та максимальне значення показника.

На практиці застосовуються різні методи оцінювання, включаючи вимірювальні, реєстраційні, розрахункові та експертні. Комбінування цих методів дозволяє отримати найбільш об'єктивну оцінку якості.

1.4 Управління якістю програмних систем

Управління якістю є невід'ємною частиною процесу розроблення програмного забезпечення. Воно включає планування, забезпечення та контроль якості на всіх етапах життєвого циклу.

Основною метою управління якістю є забезпечення відповідності програмного продукту встановленим вимогам. Це досягається шляхом впровадження стандартів, проведення аудиту процесів та використання метрик.

Процес забезпечення якості передбачає систематичну перевірку відповідності продукту вимогам, аналіз процесів розробки та усунення виявлених недоліків. Контроль якості включає тестування, верифікацію та валідацію програмного забезпечення.

2. Моделі оцінки надійності

Надійність програмного забезпечення є критично важливою характеристикою, особливо для систем, що працюють у реальному часі або забезпечують безпеку.

Формально надійність визначається як ймовірність безвідмовної роботи системи протягом часу t :

$$R(t) = P(T > t)$$

де T – випадкова величина часу до відмови

2.1 Основні поняття надійності ПЗ

Інтенсивність відмов визначається як:

$$\lambda(t) = \frac{f(t)}{R(t)}$$

де $f(t)$ – щільність розподілу часу до відмови.

Середній час безвідмовної роботи обчислюється як:

$$MTBF = \frac{1}{\lambda}$$

Надійність програмного забезпечення має особливість: вона зростає в процесі експлуатації, оскільки помилки поступово виявляються та усуваються.

2.2 Класифікація моделей надійності

Моделі надійності дозволяють прогнозувати поведінку системи на основі статистичних даних.

У прогнозуючих моделях використовується залежність кількості помилок від складності програми. Наприклад, у моделі Холстеда кількість помилок визначається як:

$$B = \frac{V}{3000}$$

де V – обсяг програми.

У моделях Джелінського-Моранди інтенсивність відмов зменшується після кожного виправлення:

$$\lambda_i = \phi(N - i + 1)$$

де N – початкова кількість помилок, i – номер відмови.

У моделі Гоела–Окумото використовується пуассонівський процес:

$$\mu(t) = a(1 - e^{-bt})$$

де a – загальна кількість помилок, b – коефіцієнт виявлення.

Новизна одержаних результатів полягає у поглибленні теоретичних та методичних засад оцінювання якості й надійності програмного забезпечення та їх інтеграції в єдину системну модель. Зокрема:

- удосконалено підхід до побудови ієрархічної моделі якості програмного забезпечення, який, на відміну від класичних моделей, передбачає узгоджене поєднання характеристик, атрибутів, метрик і оціночних елементів із урахуванням їх вагомості та впливу на інтегральний показник якості;

- дістало подальшого розвитку формалізоване подання якості ПЗ у вигляді інтегральної багатокритеріальної оцінки, що дозволяє поєднати

ISSN 2786-6025 Online

різномірні метрики (продукту, процесу та експлуатації) в єдину систему оцінювання;

- удосконалено методичний підхід до кількісного оцінювання якості, який базується на комбінуванні вимірювальних, розрахункових та експертних методів із використанням нормалізації показників, що підвищує об'єктивність і порівнюваність результатів;

- розвинуто підхід до інтеграції моделей якості та надійності, що дозволяє розглядати надійність не ізольовано, а як ключовий атрибут загальної якості програмного забезпечення;

- систематизовано та узагальнено сучасні моделі надійності програмного забезпечення з позицій їх практичного застосування та адаптації до сучасних умов розроблення;

- обґрунтовано доцільність використання сучасних підходів (DevOps, аналітика даних, MCDM) у процесах забезпечення якості, що дозволяє підвищити точність прогнозування дефектів і ефективність управління якістю.

Практична значущість отриманих результатів полягає у можливості їх використання в реальних процесах розроблення, оцінювання та супроводу програмного забезпечення. Зокрема:

- запропоновані підходи до побудови моделі якості можуть бути використані під час проєктування систем оцінювання якості програмних продуктів у компаніях-розробниках;

- розроблені методи інтегрального оцінювання дозволяють здійснювати об'єктивне порівняння альтернативних програмних рішень та приймати обґрунтовані управлінські рішення;

- використання метрик і моделей надійності дає змогу прогнозувати кількість дефектів, оцінювати готовність програмного продукту до релізу та оптимізувати процес тестування;

- результати дослідження можуть бути застосовані для підвищення ефективності процесів управління якістю в рамках сучасних методологій розроблення;

- запропонований підхід до комбінування методів оцінювання може бути використаний для створення автоматизованих систем моніторингу якості програмного забезпечення;

- матеріали роботи доцільно використовувати у навчальному процесі під час викладання дисциплін з програмної інженерії, зокрема курсів, пов'язаних із якістю та надійністю ПЗ.

У сукупності це забезпечує підвищення якості програмних продуктів, зниження ризиків розроблення та покращення ефективності функціонування програмних систем у практичних умовах.

Висновки. У результаті проведеного дослідження встановлено, що якість та надійність програмного забезпечення є ключовими характеристиками, які визначають ефективність функціонування програмних систем та рівень задоволення потреб користувачів. Вони формуються протягом усього життєвого циклу програмного продукту та потребують системного підходу до управління і оцінювання.

Проаналізована модель якості програмного забезпечення дозволяє зробити висновок, що якість є багаторівневою структурою, яка включає характеристики, атрибути та метрики, що забезпечують можливість як якісного, так і кількісного оцінювання властивостей програмного продукту. Використання інтегральних показників якості з урахуванням вагових коефіцієнтів дозволяє отримати узагальнену оцінку, що враховує різні аспекти функціонування системи.

Дослідження метрик якості показало, що їх застосування є необхідною умовою для об'єктивного оцінювання програмного забезпечення. Метрики дозволяють не лише визначати поточний рівень якості, але й прогнозувати розвиток програмного продукту, оцінювати ефективність процесів розробки та виявляти проблемні області.

Розгляд методів стандартної оцінки показників якості підтвердив доцільність використання комбінованих підходів, які поєднують вимірювальні, розрахункові та експертні методи.

Це забезпечує більш точну та достовірну оцінку характеристик програмного забезпечення.

Аналіз підходів до управління якістю показав, що ефективне забезпечення якості можливе лише за умови інтеграції процесів планування, контролю та забезпечення якості на всіх етапах життєвого циклу. Використання стандартів і методологій дозволяє підвищити передбачуваність результатів розробки та знизити ризики.

Дослідження моделей надійності продемонструвало, що вони є важливим інструментом для прогнозування поведінки програмного забезпечення в процесі експлуатації.

Використання математичних моделей, заснованих на статистичних даних про відмови, дозволяє оцінити залишкову кількість помилок і визначити рівень готовності системи до використання.

Таким чином, застосування моделей якості та надійності в програмній інженерії забезпечує комплексний підхід до створення програмних продуктів, що відповідають сучасним вимогам, підвищує ефективність процесів розроблення та сприяє створенню надійних і конкурентоспроможних програмних систем.

Література

1. Cardenas A., Ruíz-Saavedra F., García-Rivas-Plata C., Valles Coral M.A. Software quality models: Exploratory review // *Endorsed Transactions on Scalable Information Systems*, 2023; DOI:10.4108/eetsis.3982
2. Kekül H., Polatgil M. User feedback-based classification of software quality attributes // *Science of Computer Programming*, Volume 252: 103466, 2026; <https://doi.org/10.1016/j.scico.2026.103466>
3. Vasileva M.T., Penchev G. Modeling and Assessing Software Reliability in Open-Source Projects, *Computation*, 2025, 13(9): 214; <https://doi.org/10.3390/computation13090214>
4. Silva D. A Review of Software Quality Models for the Evaluation of Software Products, 2014, *International Journal of Software Engineering & Applications*, Last updated February 06, 2026; <https://doi.org/10.5121/IJSEA.2014.5603>
5. Verma A. Preferential Selection of Software Quality Models Based on a Multi-Criteria Decision-Making Approach, *International Journal of Software Innovation*, 2024; <https://doi.org/10.4018/IJSI.315739>
6. Pinedo L., Valles-Coral M., Navarro-Cabrera J. R., Injante R., Cárdenas-García Á., Ruiz-Saavedra F., García-Rivas-Plata C., Flores-Tananta C.A. Software quality models: Exploratory review, *Endorsed Transactions on Scalable Information Systems*, Vol. 10 No. 6, 2023; <https://doi.org/10.4108/eetsis.3982>
7. Deissenboeck F. An Integrated Approach to Quality Modelling, 2007, Last updated September 07, 2025; <https://doi.org/10.1109/WOSQ.2007>
8. Sharma N., Kumar V., Khan A.A. *et al.* Software belief reliability growth model incorporating change point and imperfect debugging based on uncertain differential equation approach. *Sci. Rep.*, **16**, 1685, 2026; <https://doi.org/10.1038/s41598-025-31258-w>
9. Singh V., & Kumar V. Revisiting multi-release software reliability growth model under the influence of multiple software patches. *Life Cycle Reliability and Safety Engineering*, **14**(2), 2025, 275–288 <https://doi.org/10.1007/s41872-025-00309-6>
10. Kumar V., & Pham H. Predictive analytics in system reliability. Springer Nature, 2022; <https://doi.org/10.1007/978-3-031-05347-4>
11. Ke, S. Z. & Huang, C. Y. Software reliability prediction and management: A multiple change-point model approach. *Qual. Reliab. Eng. Int.* 36 (5), 1678–1707, 2020; <https://doi.org/10.1002/qre.2653>
12. Kim, Y. S., Song, K. Y., Pham, H. & Chang, I. H. A software reliability model with dependent failure and optimal release time. *Symmetry* 14 (2), 343, 2022; <https://doi.org/10.3390/sym14020343>
13. Saxena, P., & Ram, M. Two phase software reliability growth model in the presence of imperfect debugging and error generation under fuzzy paradigm. *Mathematics in Engineering, Science & Aerospace (MESA)*, **13**(2), 2022.
14. Kumar, V., Pradhan, S. K., Kumar, A., & Kapur, P. K. Revisiting software reliability growth model under general setup. *International Journal of System Assurance Engineering and Management*, 1–14, 2025; <https://doi.org/10.1007/s13198-025-02988-x>
15. Liu, Z., Wang, S., Liu, B. & Kang, R. Change point software belief reliability growth model considering epistemic uncertainties. *Chaos Solitons Fractals*. **176**, 114178, 2023; <https://doi.org/10.1016/j.chaos.2023.114178>
16. Garg, M., Kumar, V., Chaudhary, K. & Kapur, P. K. Uncertain differential equation based software belief reliability growth model (SBRGM) considering software patching. *Int. J. Syst. Assur. Eng. Manage.* 1–14, 2024; <https://doi.org/10.1007/s13198-023-02225-3>

17. Liu, Z., Yang, S., Yang, M. & Kang, R. Software belief reliability growth model based on uncertain differential equation. *IEEE Trans. Reliab.* **71** (2), 775–787, 2022; <https://doi.org/10.1109/TR.2022.3154770>

18. Shatnawi O. Modelling Open-Source Software Reliability Incorporating Swarm Intelligence-Based Techniques, 2023, *arXiv:2401.02664*; <https://doi.org/10.48550/arXiv.2401.02664>

19. Akhtar S. Software Testing Evolution: Comparative Insights into Traditional and Emerging Practices, *ICCK Journal of Software Engineering*, Volume 1, Issue 1, 2025, P. 46-62; DOI:10.62762/JSE.2025.246843

20. Siddiq M. L., Islam-Gomes A., Sekerak N., Santos J. C. S. Large Language Models for Software Engineering: A Reproducibility Crisis, *Empirical Software Engineering (EMSE) journal*, 112 p., 2025, *arXiv:2512.00651*; <https://doi.org/10.48550/arXiv.2512.00651>

References

1. Cardenas A., Ruíz-Saavedra F., García-Rivas-Plata C., Valles Coral M.A. Software quality models: Exploratory review // *Endorsed Transactions on Scalable Information Systems*, 2023; DOI:10.4108/eetsis.3982

2. Kekül H., Polatgil M. User feedback-based classification of software quality attributes // *Science of Computer Programming*, Volume 252: 103466, 2026; <https://doi.org/10.1016/j.scico.2026.103466>

3. Vasileva M.T., Penchev G. Modeling and Assessing Software Reliability in Open-Source Projects, *Computation*, 2025, **13**(9): 214; <https://doi.org/10.3390/computation13090214>

4. Silva D. A Review of Software Quality Models for the Evaluation of Software Products, 2014, *International Journal of Software Engineering & Applications*, Last updated February 06, 2026; <https://doi.org/10.5121/IJSEA.2014.5603>

5. Verma A. Preferential Selection of Software Quality Models Based on a Multi-Criteria Decision-Making Approach, *International Journal of Software Innovation*, 2024; <https://doi.org/10.4018/IJSI.315739>

6. Pinedo L., Valles-Coral M., Navarro-Cabrera J. R., Injante R., Cárdenas-García Á., Ruiz-Saavedra F., García-Rivas-Plata C., Flores-Tananta C.A. Software quality models: Exploratory review, *Endorsed Transactions on Scalable Information Systems*, Vol. 10 No. 6, 2023; <https://doi.org/10.4108/eetsis.3982>

7. Deissenboeck F. An Integrated Approach to Quality Modelling, 2007, Last updated September 07, 2025; <https://doi.org/10.1109/WOSQ.2007>

8. Sharma N., Kumar V., Khan A.A. *et al.* Software belief reliability growth model incorporating change point and imperfect debugging based on uncertain differential equation approach. *Sci. Rep.*, **16**, 1685, 2026; <https://doi.org/10.1038/s41598-025-31258-w>

9. Singh V., & Kumar V. Revisiting multi-release software reliability growth model under the influence of multiple software patches. *Life Cycle Reliability and Safety Engineering*, **14**(2), 2025, 275–288 <https://doi.org/10.1007/s41872-025-00309-6>

10. Kumar V., & Pham H. Predictive analytics in system reliability. Springer Nature, 2022; <https://doi.org/10.1007/978-3-031-05347-4>

11. Ke, S. Z. & Huang, C. Y. Software reliability prediction and management: A multiple change-point model approach. *Qual. Reliab. Eng. Int.* **36** (5), 1678–1707, 2020; <https://doi.org/10.1002/qre.2653>

12. Kim, Y. S., Song, K. Y., Pham, H. & Chang, I. H. A software reliability model with dependent failure and optimal release time. *Symmetry* 14 (2), 343, 2022; <https://doi.org/10.3390/sym14020343>

13. Saxena, P., & Ram, M. Two phase software reliability growth model in the presence of imperfect debugging and error generation under fuzzy paradigm. *Mathematics in Engineering, Science & Aerospace (MESA)*, 13(2), 2022.

14. Kumar, V., Pradhan, S. K., Kumar, A., & Kapur, P. K. Revisiting software reliability growth model under general setup. *International Journal of System Assurance Engineering and Management*, 1–14, 2025; <https://doi.org/10.1007/s13198-025-02988-x>

15. Liu, Z., Wang, S., Liu, B. & Kang, R. Change point software belief reliability growth model considering epistemic uncertainties. *Chaos Solitons Fractals*. 176, 114178, 2023; <https://doi.org/10.1016/j.chaos.2023.114178>

16. Garg, M., Kumar, V., Chaudhary, K. & Kapur, P. K. Uncertain differential equation based software belief reliability growth model (SBRGM) considering software patching. *Int. J. Syst. Assur. Eng. Manage.* 1–14, 2024; <https://doi.org/10.1007/s13198-023-02225-3>

17. Liu, Z., Yang, S., Yang, M. & Kang, R. Software belief reliability growth model based on uncertain differential equation. *IEEE Trans. Reliab.* 71 (2), 775–787, 2022; <https://doi.org/10.1109/TR.2022.3154770>

18. Shatnawi O. Modelling Open-Source Software Reliability Incorporating Swarm Intelligence-Based Techniques, 2023, *arXiv:2401.02664*; <https://doi.org/10.48550/arXiv.2401.02664>

19. Akhtar S. Software Testing Evolution: Comparative Insights into Traditional and Emerging Practices, *ICCK Journal of Software Engineering*, Volume 1, Issue 1, 2025, P. 46-62; DOI:10.62762/JSE.2025.246843

20. Siddiq M. L., Islam-Gomes A., Sekerak N., Santos J. C. S. Large Language Models for Software Engineering: A Reproducibility Crisis, *Empirical Software Engineering (EMSE) journal*, 112 p., 2025, *arXiv:2512.00651*; <https://doi.org/10.48550/arXiv.2512.00651>

Дата першого надходження статті до видання: 12.04.2026

Дата прийняття статті до друку після рецензування: 26.04.2026