https://doi.org/10.31891/2219-9365-2024-80-1
УДК 004.738.1:316.772.5

ПІХ Ірина
Національний університет «Львівська політехніка»
https://orsid.org/0000-0002-9909-84444
e-mail: pikhirena@gmail.com
СЕНЬКІВСЬКИЙ Всеволод
Національний університет «Львівська політехніка»
https://orsid.org/0000−0002−4510−540X
e-mail: senk.vm@gmail.com
СЕНЬКІВСЬКА Наталя
Національний університет «Львівська політехніка»
https://orsid.org/0000-0002-6096-2282
e-mail: senkivskanata@gmail.com
КАЛИНІЙ Ірина
Бережанський агротехнічний інститут
https://orsid.org/0000-0003-1296-6454
e-mail: iryna.kalynii@gmail.com
БІЛИК Олексій
Національний університет «Львівська політехніка»
https://orsid.org/0009-0002-1355-2333
e-mail: lesykbilyk@gmail.com

ПРОГНОЗУВАННЯ ЯКОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ НА ОСНОВІ НЕЧІТКОЇ ЛОГІКИ
(ЧАСТИНА 1. ФУНКЦІЇ НАЛЕЖНОСТІ ЛІНГВІСТИЧНИХ ЗМІННИХ)

У сучасних умовах розроблення програмного забезпечення важливою проблемою є забезпечення його високої якості. Одним з ефективних підходів до прогнозування якості програмного забезпечення вважається застосування засобів нечіткої логіки, яка дозволяє враховувати невизначеності та нечіткості оцінок різних факторів впливу. В роботі реалізовано підхід до прогнозування якості програмного забезпечення, заснований на використанні лінгвістичних змінних для опису факторів, що впливають на якість програмного забезпечення. Структура публікації передбачає короткі теоретичні описи етапів дослідження, підкріплені практичними рекомендаціями. У зв'язку з великим обсягом, матеріал дослідження поділено на дві частини, перша з яких подається до опублікування.

Виконано аналіз літературних джерел, дотичних до тематики статті. Відзначено, що публікації, які стосуються оцінювання якості програмного забезпечення на основі нечіткої логіки, послуговуються неповним переліком чинників впливу на ступінь добротності програм. Вказане додатково підтверджує актуальність виконаного нами дослідження, в якому вихідна множина впливових факторів максимально повно відтворює характеристики якості програмного забезпечення, наведені у стандарті.

Виокремлено фактори впливу на якість програмного забезпечення, трансформовані лінгвістичними змінними, значення яких описано лінгвістичними термами, що утворюють універсальну терм-множину. Для кожної лінгвістичної змінної побудовано функції належності, що відображають ступінь відповідності кожного терму реальному значенню фактора. Обумовлено передумови доцільності застосування моделі нечіткого логічного виведення для прогнозування якості програмного забезпечення. Запроектовано модель логічного виведення, яка відтворює ієрархію взаємозв'язків між лінгвістичними змінними та ступенями якості програмного забезпечення. Виконано розрахунок значень функцій належності лінгвістичної змінної «тестування», виконаний на основі аналізу та опрацювання матриць попарних порівнянь, елементами яких стали ранги пріоритетності лінгвістичних змінних. Здійснено візуалізацію функцій належності у графічному відображенні, що обумовлює ступінь причетності фактора до певного рівня якості програмного забезпечення залежно від означеного терму.

Ключові слова: лінгвістична змінна, якість програмного забезпечення, модель логічного виведення, терм-множина значень, функція належності, матриця попарних порівнянь, візуалізація функцій належності.

PIKH Iryna, SENKIVSKYY Vsevolod, SENKIVSKA Natalia
Lviv Polytechnic National University
KALYNII Iryna
Berezhany Agrotechnical Institute
BILYK Oleksii
Lviv Polytechnic National University

SOFTWARE QUALITY FORECASTING BASED ON FUZZY LOGIC
(PART 1. MEMBERSHIP FUNCTIONS OF LINGUISTIC VARIABLES)

In today's software development environment, ensuring its high quality is an important issue. One of the effective approaches to predicting the quality of software is the use of fuzzy logic, which allows to take into account uncertainties and uncertainties and fuzziness of estimates of various influence factors. The paper implements an approach to software quality

forecasting based on the use of linguistic variables to describe the factors that affect software quality. The structure of the publication includes brief theoretical descriptions of the research stages supported by practical recommendations. Due to the large volume, the research material is divided into two parts, the first of which is submitted for publication.

The author analyzes the literature sources related to the subject of the article. It is noted that publications related to software quality assessment based on fuzzy logic use an incomplete list of factors influencing the degree of software quality. This additionally confirms the relevance of our study, in which the initial set of influencing factors reproduces the characteristics of software quality given in the standard to the fullest extent possible.

The factors of influence on the quality of software, transformed by linguistic variables, whose values are described by linguistic terms that form a universal term set. For each linguistic variable, membership functions are constructed that reflect the degree of correspondence of each term to the real value of the factor. The prerequisites for the feasibility of using a fuzzy inference model to predict software quality are determined. A model of logical inference is designed, which reproduces the hierarchy of relationships between linguistic variables and degrees of software quality. The values of the membership functions of the linguistic variable «testing» are calculated, based on the analysis and processing of matrices of pairwise comparisons, the elements of which are the priority ranks of linguistic variables. The membership functions are visualized in a graphical representation, which determines the degree of involvement of a factor in a certain level of software quality depending on the defined term.

Keywords: linguistic variable, software quality, logical inference model, term set of values, membership function, pairwise comparison matrix, visualization of membership functions.

ПОСТАНОВКА ПРОБЛЕМИ У ЗАГАЛЬНОМУ ВИГЛЯДІ ТА ЇЇ ЗВ'ЯЗОК ІЗ ВАЖЛИВИМИ НАУКОВИМИ ЧИ ПРАКТИЧНИМИ ЗАВДАННЯМИ

Якість програмного забезпечення (ПЗ) є одним із найважливіших критеріїв, який визначає успіх будь-якої інформаційної системи. Вона впливає на широкий спектр показників, від надійності та безпеки до зручності використання й ефективності. Зі стрімким розвитком технологій, складність програмних систем збільшується, що, в свою чергу, ускладнює процес оцінювання якості ПЗ. Традиційні методи оцінювання, засновані на класичних математичних моделях або статистичних підходах, не завжди можуть повною мірою врахувати всю множину факторів, які впливають на якість продукту. Зазначене стосується як технічних аспектів (наприклад, кількість помилок, продуктивність), так і суб'єктивних оцінок (наприклад, задоволення користувачів, досвід розробників).

Оскільки процес розроблення ПЗ часто супроводжується невизначеностями, появляється необхідність у нових підходах до прогнозування та оцінювання його якості. Нечітка логіка дозволяє працювати з нечіткими величинами та нечіткими правилами, наближаючи процес прогнозування до реальних умов розробки програмного забезпечення. У результаті використання нечіткої логіки стає особливо актуальним прогнозування якості ПЗ, оскільки воно дозволяє не тільки враховувати існуючі невизначеності, а й моделювати складні взаємозв'язки між різними показниками якості.

Крім того, прогнозування якості програмного забезпечення на основі нечіткої логіки забезпечує врахування широкого спектру факторів, включаючи не лише технічні, але й соціальні характеристики. У традиційних підходах оцінка часто ґрунтується на точних числових показниках, що може обмежувати точність прогнозів в умовах невизначеності. Нечітка логіка, навпаки, дозволяє використовувати лінгвістичні змінні (наприклад, "висока", "середня", "низька" якість), що робить підхід більш гнучким та придатним для реальних сценаріїв. Ще однією перевагою використання нечіткої логіки є її здатність інтегрувати думки експертів та інших зацікавлених сторін у процес прогнозування якості ПЗ. Засоби нечіткої логіки дозволяють формалізувати знання експертів, перетворивши їх на конкретні правила та моделі для подальшого аналізу і дослідження.

Отже, актуальність теми дослідження полягає у застосуванні методологічних аспектів нечіткої логіки в процесі прогнозування якості програмного забезпечення, що може, на наш погляд, значно підвищити точність оцінювання, зменшити ризики й оптимізувати процес розроблення. Вказане особливо важливе для проєктів із високими вимогами до якості, таких як системи для критичних галузей (наприклад, охорона здоров'я, авіація, фінанси), де помилки можуть мати значні наслідки.

Мета статті полягає у дослідженні та розробленні моделей прогнозування якості програмного забезпечення на основі поєднання засобів факторного аналізу та нечіткої логіки, що уможливить врахування нечіткості оцінок, залежностей та критеріїв, характерних для процесу розроблення та оцінювання програм, тим самим підвищуючи точність та надійність прогнозів.

АНАЛІЗ ДОСЛІДЖЕНЬ ТА ПУБЛІКАЦІЙ

Дослідження процесу використання нечіткої логіки в оцінюванні якості програмного забезпечення обумовлює аналіз публікацій з цієї теми. У науковій літературі нечітка логіка є популярною методологією для обробки неточних або невизначених даних, що активно застосовується в різних галузях, зокрема для оцінювання програм. Перелік літературних джерел містить частину робіт, достатні для розуміння загальних тенденцій наукової періодики стосовно проблематики, озвученої у поданій статті.

Першочергово не можна оминути збірник праць Лотфі А. Заде, засновника нечіткої логіки [1], що містить важливі статті та дослідження, які пояснюють основи теорії нечітких множин і нечіткої логіки, та вважаються фундаментальним джерелом для тих, хто цікавиться застосуванням нечіткої логіки, зокрема у

сфері оцінювання якості програмного забезпечення. Книга [2] пропонує фундаментальний вступ до нечіткої логіки та методів нечіткого управління, включаючи застосування у сфері програмного забезпечення. Вона надає загальні методології для побудови нечітких моделей, що можуть бути корисні для оцінювання якості ПЗ. Стаття [3] описує різні підходи до оцінювання якості програмного забезпечення та обговорює, як нечітка логіка може бути використана для моделювання якісних характеристик, таких як надійність, підтримуваність і зручність користування. Робота [4] досліджує застосування нечіткої логіки для кількісної оцінки якості програмного забезпечення, зокрема за такими параметрами, як надійність і продуктивність. Публікація [5] наводить практичний приклад використання нечіткої логіки для оцінювання якості програмного забезпечення на основі різних факторів, таких як зручність, продуктивність і підтримуваність. Стаття [6] розглядає застосування нечіткої логіки для прийняття рішень у процесі оцінювання якості програмного забезпечення. Основна увага приділяється тому, як нечітка логіка може допомогти враховувати суб'єктивні критерії. Публікація [7] описує модель оцінювання якості програмного забезпечення, що використовує нечітку логіку для аналізу таких аспектів, як надійність, функціональність та підтримуваність. Автори пропонують новий підхід до створення моделей якості ПЗ. Дослідження [8] описує підхід для оцінки якості програмного забезпечення на основі нечіткої логіки, який включає в себе кілька метрик якості, таких як ефективність та зручність користування. Робота [9] представляє фреймворк для оцінювання якості програмного забезпечення, заснований на нечіткій логіці та стандарті ISO 9126, який охоплює такі показники, як функціональність, надійність, ефективність та переносимість. У статті [10] автори розглядають методологію оцінювання надійності та якості програмного забезпечення на основі нечіткої логіки, де якість розглядається як багатовимірний показник, залежний від різних параметрів ПЗ. Публікація [11] описує нечітку багатокритеріальну модель прийняття рішень для оцінки якості програмного забезпечення, що дозволяє враховувати декілька факторів, таких як підтримуваність, зручність використання та ефективність.

Українські публікації, дотичні до тематики даної статті, відображають результати досліджень, що стосуються: методів і способів досягнення та контролю якості програмного забезпечення з огляду суттєвого впливу на процес фактору тестування [12-14]; методологічних засад формування якості програмного забезпечення, в основі якого базова та оптимізована моделі факторів (лінгвістичних змінних) – визначальних чинників впливу на процес [15, 16]; багатофакторного вибору альтернативних варіантів процесу тестування ПЗ [17]; процесу формування якості програмного забезпечення за методом багатокритеріальної оптимізації з використанням факторів множини Парето [18]; створення інформаційних систем і програмних комплексів багатофакторного аналізу і діагностики (прогнозування) виробничих процесів, в основі яких побудова моделей логічного виведення і фазифікація впливових чинників, проектування нечітких баз знань і нечітких логічних рівнянь, дефазифікація результуючого показника [19, 20]; теоретичних і прикладних основ інформаційної концепції формування та оцінювання якості видавничо-поліграфічних процесів, реалізованої на базі факторного аналізу впливових чинників (лінгвістичних змінних) та розроблених на засадах нечіткої логіки автоматизованих програмних комплексів [21].

ФОРМУЛЮВАННЯ ЦІЛЕЙ СТАТТІ

Огляд літературних джерел додатково обґрунтував актуальність теми статті, дослідження якої обумовлено такими аргументами. Застосування нечіткої логіки для прогностичного оцінювання якості програмного забезпечення стало активною науковою та прикладною тематикою. Характеристики якості ПЗ, визначені стандартом ISO/IEC 25010:2011 [22], містять вісім критеріїв, до яких відносяться: функціональна придатність, ефективність роботи, сумісність, зручність використання, надійність, безпека, зручність супроводження, портативність (переносимість). Водночас, джерела, що стосуються оцінювання якості ПЗ на основі нечіткої логіки, послуговуються неповним їх переліком. Найбільше показників, тобто чотири метрики, містить публікація [9]. На відміну від згаданих у поданій нами статті базова множина лінгвістичних змінних сформована на основі стандарту та містить вісім впливових факторів, повний опис яких буде наведено нижче.

ВИКЛАД ОСНОВНОГО МАТЕРІАЛУ

Для кращого сприйняття суті дослідження наведемо у більш повному відтворенні характеристики якості ПЗ, що стали основою множини впливових факторів – лінгвістичних змінних процесу оцінювання програмної продукції. Отже, маємо таку вихідну інформаційну базу даних [15, 16], сформовану у вигляді графічної багаторівневої моделі рис.1.

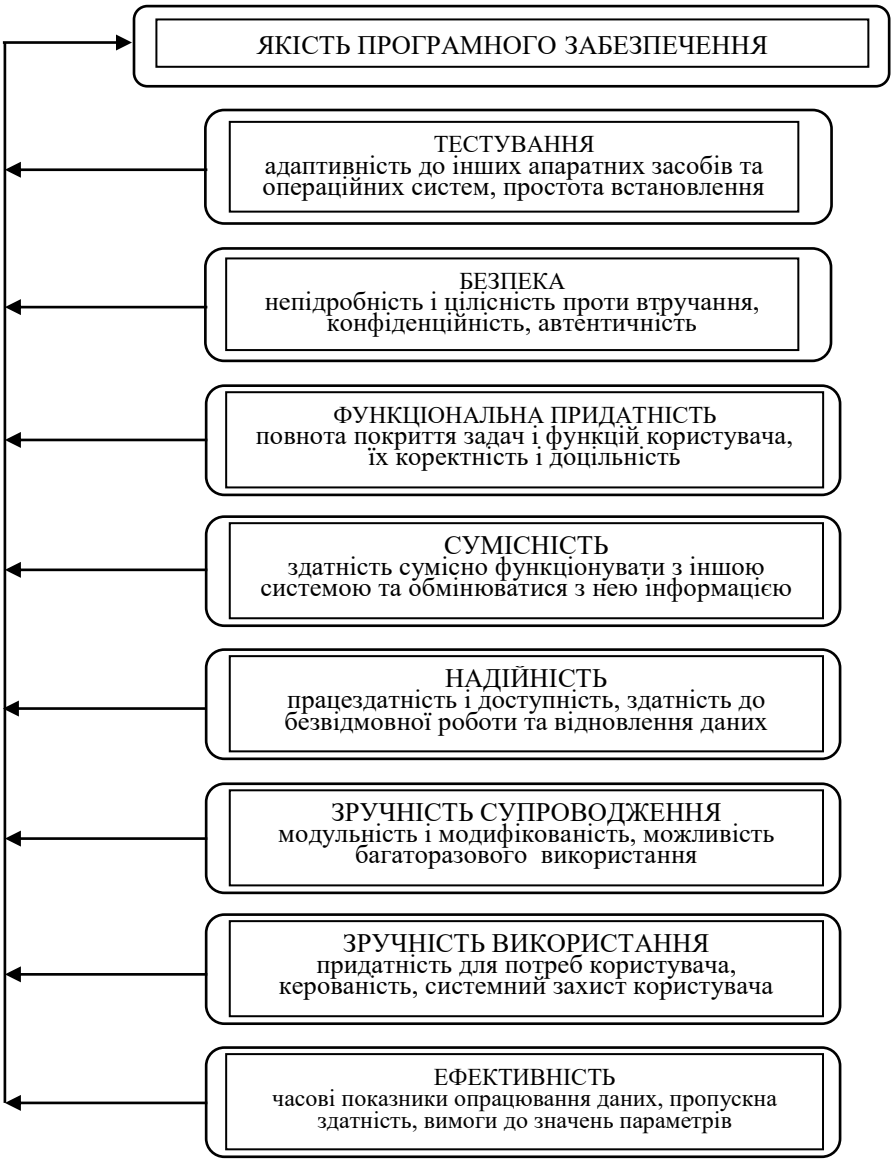

Рис. 1. Графічна модель пріоритетного впливу факторів на якість програмного забезпечення

Модель отримана у результаті опрацювання семантичної мережі факторів впливу на якість ПЗ за допомогою методу аналізу ієрархій та методу попарних порівнянь з використанням шкали відносної важливості об'єктів, що уможливило розрахунок умовних вагових значень та відповідних рівнів пріоритетності досліджуваних чинників. Фактори рис. 1 максимально співпадають з характеристиками якості ПЗ, визначені стандартом ISO/IEC 25010:2011, що додатково підтверджує актуальність та наукову новизну виконаного нами дослідження. Трансформуємо фактори у множину лінгвістичних змінних, виконавши при цьому класифікаційне групування за сутнісними ознаками, що уможливить використання моделі нечіткого логічного виведення – основи формування та прогностичного оцінювання якості програмного забезпечення. Зрозуміло, що кожна з лінгвістичних змінних «вносить» свою частку в груповий показник якості програмного забезпечення, що потребує упорядкування кроків реалізації певних процедур. З цією метою подамо вираз для лінгвістичної змінної, що ідентифікує підсумковий показник у вигляді функції:

$$Q = F_Q\left(T, B, C\right), \tag{1}$$

де: T – лінгвістична змінна, що позначає групу функціональної складової якості ПЗ; B – лінгвістична змінна, орієнтована на технічну складову якості; C – лінгвістична змінна, відповідальна за експлуатаційну складову.

Відтворимо вміст задекларованих груп такими функціями.

$$T = F_T\left(t_1, t_2, t_3\right), \tag{2}$$

де: t_1 – ЛЗ «тестування»; t_2 – ЛЗ «функціональна придатність»; t_3 – ЛЗ «ефективність».

$$B = F_B\left(b_1, b_2, b_3\right), \tag{3}$$

де: b_1 – ЛЗ «безпека»; b_2 – ЛЗ «сумісність»; b_3 – ЛЗ «надійність».

$$C = F_C\left(c_1, c_2\right), \tag{4}$$

де: c_1 – ЛЗ «зручність супроводження»; c_2 – ЛЗ «зручність використання».

Застосування моделі нечіткого логічного виведення для прогнозування якості програмного забезпечення (ПЗ) обумовлюється кількома причинами.

Процес оцінювання якості ПЗ містить характеристики, значення яких часто важко визначити точно, наприклад, надійність, сумісність, безпека, зручність користування і т.п. Нечітка логіка забезпечує перетворення суб'єктивних описових оцінок на кількісні показники через лінгвістичні терми, наприклад, «низька», «середня», «висока». При цьому невизначеності, що виникають через варіативність умов використання, зміни у вимогах користувачів та зовнішніх умовах (оновлення, сумісність), параметрах якості ПЗ інтегруються логічними моделями, що враховують непередбачувані ситуації, забезпечуючи при цьому достовірні прогнози. Нечіткі моделі логічного виведення уможливлюють чітке простеження логіки, на основі якої приймаються рішення, що підвищує довіру до отриманих результатів, та є важливим для аудиту якості ПЗ і підвищення прозорості процесу його оцінювання. Нечіткі моделі можуть працювати з меншими обсягами даних, що вважається суттєвою перевагою при обмеженій інформації, зокрема на ранніх стадіях розроблення ПЗ, яка забезпечує отримання попередніх прогнозів якості навіть при неповних даних.

Остаточно модель логічного виведення зображена на рис. 2.

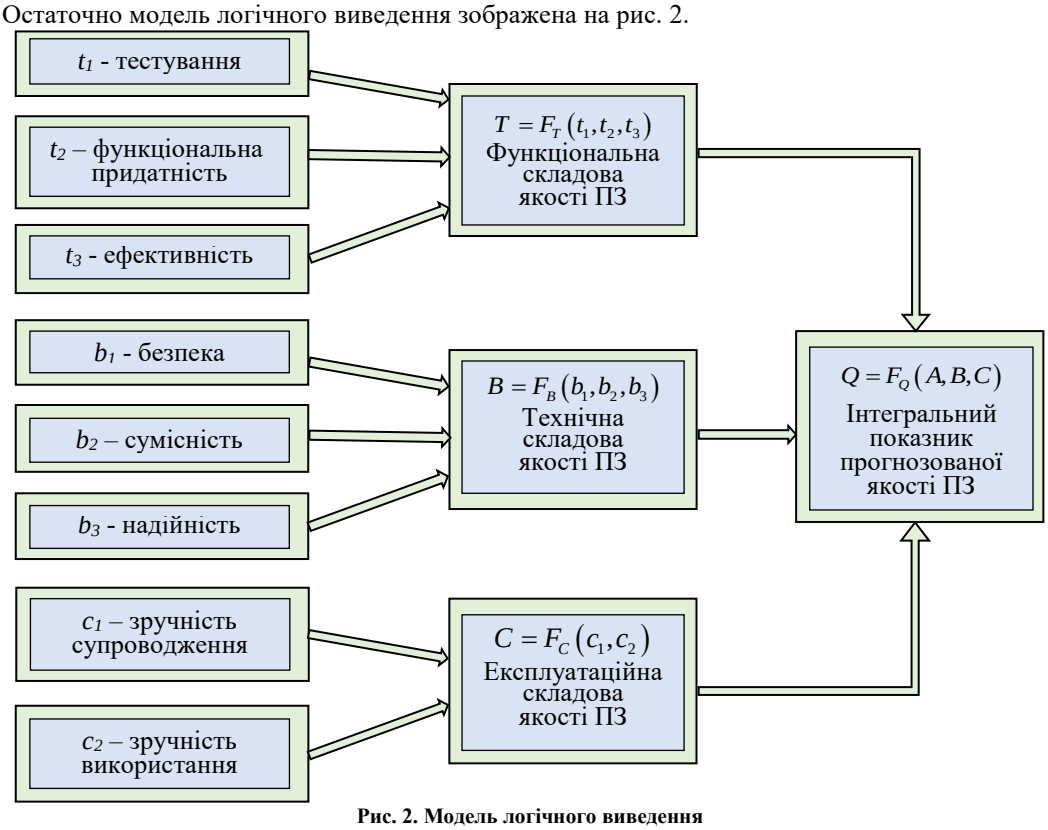

Рис. 2. Модель логічного виведення

Модель логічного виведення стає по суті алгоритмом, що визначає покрокову ієрархію визначення показника якості ПЗ, починаючи з лінгвістичних змінних найнижчого рівня. Для їх ідентифікації згідно вимог нечіткої логіки сформуємо терм-множину значень ЛЗ, що забезпечить можливість описати невизначеність і нечіткість реальних впливових факторів. Вказаний підхід має широкий спектр застосувань – від штучного інтелекту до управління якістю, що підкреслює його універсальність і корисність. Загалом, терм-множина є основною компонентою забезпечення ефективності нечіткої логіки в моделюванні реальних процесів, які не можуть бути точно описані традиційними математичними методами.

Загальну інформацію оформимо таблицею 1, де відображено математичний символ лінгвістичної змінної, її описову сутність, універсальну множину D значень ЛЗ і спеціальну множину лінгвістичних термів у вигляді тривимірних шкал, що ідентифікують якісні стани лінгвістичних змінних в опорних точках множини D. Лінгвістичні терми подамо у текстовій формі, що містить інформацію про значимість лінгвістичної змінної, пов'язану з точками поділу універсальної множини.

Таблиця 1

Терм-множини значень лінгвістичних змінних

ЛЗ	Описова суть змінної	Універсальна множина значень (терм-множина D)	Лінгвістичні терми (множина X)
t_1	Тестування (рівень)	(1-5) у. о.	Низький, середній, високий
t_2	Функціональна придатність	(1-5) у. о.	Низька, середня, висока
t_3	Ефективність	(1-5) у. о.	Низька, середня, висока
b_1	Безпека	(1-5) у. о.	Низька, середня, висока
b_2	Надійність	(1-5) у. о.	Низька, середня, висока
b_3	Сумісність	(1-5) у. о.	Низька, середня, висока
c_1	Зручність супроводження	(1-5) у. о.	Низька, середня, висока
c_2	Зручність використання	(1-5) у. о.	Низька, середня, висока

Оскільки універсальна множина для кожної ЛЗ визначає інтервал (1-5) у. о., приймемо умову, згідно якої значення функцій належності в межах інтервалу будемо прив'язувати до п'яти вузлових точок. Додатково у вказаних точках терм-множини задаємо ранги переваг лінгвістичних термів. Підсумовуючи, маємо у загальних позначеннях такі вихідні компоненти: множину $D=\{d_1,d_2,...,d_n\}$; функції належності $\mu_q(d_i)$, що визначає ступінь приналежності кожного можливого значення лінгвістичної змінної до певного лінгвістичного терму (ЛТ); ранги $r_q(d_i)$, які розкривають пріоритет або вагомість належності певного значення ЛЗ до конкретного терму (наприклад, «низький», «середній», «високий») у точках поділу терм-множини. Для розрахунку значень ФН скористаємося відомими в теорії нечітких множин математичними викладками [20, 21].

З огляду на прийняті умовні позначення для лінгвістичного терму «показник якості програмного забезпечення» сформуємо у загальному вигляді таку нечітку множину [21]:

$$Q_F=\left\{\frac{\mu_q(d_1)}{d_1},\frac{\mu_q(d_2)}{d_2},...,\frac{\mu_q(d_n)}{d_n}\right\}, \tag{5}$$

де: $Q_F\subset D$; $\mu_q(d_i)$ – функція належності d_i до множини Q_F.

За умови, що $\mu_i=\mu_q(d_i)$ та $r_i=r_q(d_i)$ для $i=\overline{1,n}$ розподіл мір належності між змінними має такий вигляд:

$$\frac{\mu_1}{r_1}=\frac{\mu_2}{r_2}=...=\frac{\mu_n}{r_n}, \tag{6}$$

де ФН відповідають умові нормування, тобто $\mu_1+\mu_2+...+\mu_n=1$.

З урахуванням (6) для умовно базової точки $d_1\subset D$ з функцією належності μ_1 можна знайти решту функцій належності з виразів:

$$\mu_2=\frac{r_2}{r_1}\mu_1,\ \mu_3=\frac{r_3}{r_1}\mu_1,...,\mu_n=\frac{r_n}{r_1}\mu_1. \tag{7}$$

Подібним чином знаходимо значення ФН для всіх точок множини D.

З огляду на умову нормування та співвідношення (7) розрахунок функцій належності у вузлових точках універсальної терм-множини виконуємо за такими відношеннями [20, 21]:

ISSN 2219-9365

$$\left.\begin{aligned}
\mu_1 &= \left(1+\frac{r_2}{r_1}+\frac{r_3}{r_1}+\ldots+\frac{r_n}{r_1}\right)^{-1};\\
\mu_2 &= \left(\frac{r_1}{r_2}+1+\frac{r_3}{r_2}+\ldots+\frac{r_n}{r_2}\right)^{-1};\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\
\mu_n &= \left(\frac{r_1}{r_n}+\frac{r_2}{r_n}+\frac{r_3}{r_n}+\ldots+1\right)^{-1}.
\end{aligned}\right\} \quad (8)$$

У підсумку для п’яти точок терм-можини за результатами попарних порівнянь рангів будуємо матрицю, опрацювання якої визначить передумови розрахунку значень функцій належності за виразами (8).

$$A=\begin{bmatrix}
1 & \frac{r_2}{r_1} & \frac{r_3}{r_1} & \frac{r_4}{r_1} & \frac{r_5}{r_1}\\
\frac{r_1}{r_2} & 1 & \frac{r_3}{r_2} & \frac{r_4}{r_2} & \frac{r_5}{r_2}\\
\ldots & \ldots & \ldots & \ldots & \ldots\\
\frac{r_1}{r_5} & \frac{r_2}{r_5} & \frac{r_3}{r_5} & \frac{r_4}{r_5} & 1
\end{bmatrix}. \quad (9)$$

Експериментальну реалізацію наведених вище теоретичних викладок виконаємо для лінгвістичної змінної t_1 – «тестування», рівень якого означений лінгвістичними термами «низький», «середній», «високий», множину значень якої згідно таблиці 1 визначатимемо в опорних точках терм-множини $D(t_1)=[1;2;3;4;5]$. Ранги змінної вибираємо з інтервалу $(1\div 9)$, оскільки відомо, що психологічна межа при порівнянні предметів забезпечується 9-ма градаціями відмінностей між ними [21]. Додамо тільки, що коли експертні оцінки при порівняннях узгоджені, задають тільки останній рядок, а решта елементів матриці отримується на підставі того, що $a_{ji}=a_{5j}/a_{5i}$; $i,\ j=\overline{1,5}$.

Матриця попарних порівнянь для терму «низький» в опорних токах універсальної множини має такий вигляд:

$$A_{низький}(t_1)=\begin{bmatrix}
1 & 7/9 & 5/9 & 4/9 & 1/9\\
9/7 & 1 & 5/7 & 4/7 & 1/7\\
9/5 & 7/5 & 1 & 4/5 & 1/5\\
9/4 & 7/4 & 5/4 & 1 & 1/4\\
9 & 7 & 5 & 4 & 1
\end{bmatrix}. \quad (10)$$

Виконаємо згідно виразів (8) та елементів матриці попарних порівнянь розрахунок функцій належності в усіх точках терм-множини.

$$\mu_{низький}(d_1)=\frac{1}{1+\frac{7}{9}+\frac{5}{9}+\frac{4}{9}+\frac{1}{9}}=\frac{9}{26}=0.35;\ \mu_{низький}(d_2)=\frac{7}{26}=0.27.$$

$$\mu_{низький}(d_3)=\frac{5}{26}=0.19;\ \mu_{низький}(d_4)=\frac{4}{26}=0.15;\ \mu_{низький}(d_5)=\frac{1}{26}=0.04.$$

Для терму «середній» сформуємо таку матрицю:

$$A_{середній}(t_1)=\begin{bmatrix} 1 & 4 & 9 & 5 & 1 \\ 1/4 & 1 & 9/4 & 5/4 & 1/4 \\ 1/9 & 4/9 & 1 & 5/9 & 1/9 \\ 1/5 & 4/5 & 9/5 & 1 & 1/5 \\ 1 & 4 & 9 & 5 & 1 \end{bmatrix}. \quad (11)$$

Опускаючи хід виконаних вище розрахунків, наведемо отримані значення функцій належності:

$$\mu_{середній}(d_1)=0.05;\ \mu_{середній}(d_2)=0.20;\ \mu_{середній}(d_3)=0.45;$$
$$\mu_{середній}(d_4)=0.25;\ \mu_{середній}(d_5)=0.05.$$

Аналогічно будуємо матрицю для терму «високий» стосовно рівня тестування лінгвістичної змінної t_1:

$$A_{високий}(t_1)=\begin{bmatrix} 1 & 5 & 6 & 8 & 9 \\ 1/3 & 1 & 6/3 & 8/3 & 9/3 \\ 1/6 & 5/6 & 1 & 8/6 & 9/6 \\ 1/8 & 5/8 & 6/8 & 1 & 9/8 \\ 1/9 & 5/9 & 6/9 & 8/9 & 1 \end{bmatrix}. \quad (12)$$

Функції належності розраховано з такими значеннями:

$$\mu_{високий}(d_1)=0.034;\ \mu_{високий}(d_2)=0.173;\ \mu_{високий}(d_3)=0.207;$$
$$\mu_{високий}(d_4)=0.276;\ \mu_{високий}(d_5)=0.310.$$

Згідно правил нечіткої логіки отримані значення функцій належності підлягають нормуванню, що обумовлюється необхідністю привести значення функцій належності в інтервал від 0 до 1. Для цього є декілька причин: можливість однозначно інтерпретувати значення належності й порівнювати їх між різними множинами та елементами; нормування забезпечує стабільність та коректність обчислень, особливо у випадках, коли функції належності використовуються в складних операціях; У нечітких системах нормовані функції належності допомагають стандартизувати результати, що полегшує їх використання для побудови комплексних моделей та порівняння результатів у різних контекстах. Нормування виконується з використанням коефіцієнта

$$k_p=1/\max\mu_p(d_i),\ \ (i=1,...,5),$$

де: $p\equiv$ «низький», «середній», «високий»; $\mu_{p_n}(d_i)=k_p\times\mu_p(d_i)$

Унаслідок, отримаємо нормовані значення функцій належності:

$$\mu_{низький_n}(d_1)=1;\ \mu_{низький_n}(d_2)=0.77;\ \mu_{низький_n}(d_3)=0.54;$$
$$\mu_{низький_n}(d_4)=0.43;\ \mu_{низький_n}(d_5)=0.11;$$
$$\mu_{середній_n}(d_1)=0.11;\ \mu_{середній_n}(d_2)=0.44;\ \mu_{середній_n}(d_3)=1;$$
$$\mu_{середній_n}(d_4)=0.56;\ \mu_{середній_n}(d_1)=0.11;$$
$$\mu_{високий_n}(d_1)=0.11;\ \mu_{високий_n}(d_2)=0.56;\ \mu_{високий_n}(d_3)=0.67;$$
$$\mu_{високий_n}(d_4)=0.89;\ \mu_{високий_n}(d_5)=1.$$

Отримані функції належності доцільно подати у графічній формі, що обумовлюється такими особливостями.

Візуалізація функцій належності у графічному вигляді є важливим інструментом у нечіткій логіці, оскільки забезпечує можливість кращого розуміння та аналізу складних нечітких даних. При цьому функції належності показують, наскільки конкретний елемент належить до певної нечіткої множини. Графічне

зображення таких функцій (наприклад, трикутних, трапецієвидних або гаусівських, як у нашому дослідженні) дозволяє швидко оцінити, як елементи впливають на систему та на якій стадії вони найбільше належать до тієї чи іншої множини, що особливо корисно при розробленні нечітких моделей за необхідності чіткого уявлення про «рівень нечіткості» у кожній точці терм-множини. Завдяки графічному поданню можна швидко виявити ймовірні проблеми у функціях належності, як-от перекривання нечітких множин, невідповідність логіці системи або неправильне визначення меж.

Для побудови графіків отримані вище нормовані числові значення функцій належності лінгвістичної змінної «тестування» (рівень тестування) відтворимо згідно виразу (7) нечіткими множинами відносно термів «низький», «середній», «високий».

$$\begin{aligned}
&\textit{тестування}\left(\textit{рівень низький}\right)=\left\{\frac{1}{1};\frac{0,77}{2};\frac{0,54}{3};\frac{0,43}{4};\frac{0,11}{5}\right\}\ \text{у. о.;}\\
&\textit{тестування}\left(\textit{рівень середній}\right)=\left\{\frac{0,11}{1};\frac{0,44}{2};\frac{1}{3};\frac{0,56}{4};\frac{0,11}{5}\right\}\ \text{у. о.;}\\
&\textit{тестування}\left(\textit{рівень високий}\right)=\left\{\frac{0,11}{1};\frac{0,56}{2};\frac{0,67}{3};\frac{0,89}{4};\frac{1}{5}\right\}\ \text{у. о.}
\end{aligned} \tag{13}$$

Остаточно візуальне відображення множин (13) має такий вигляд:

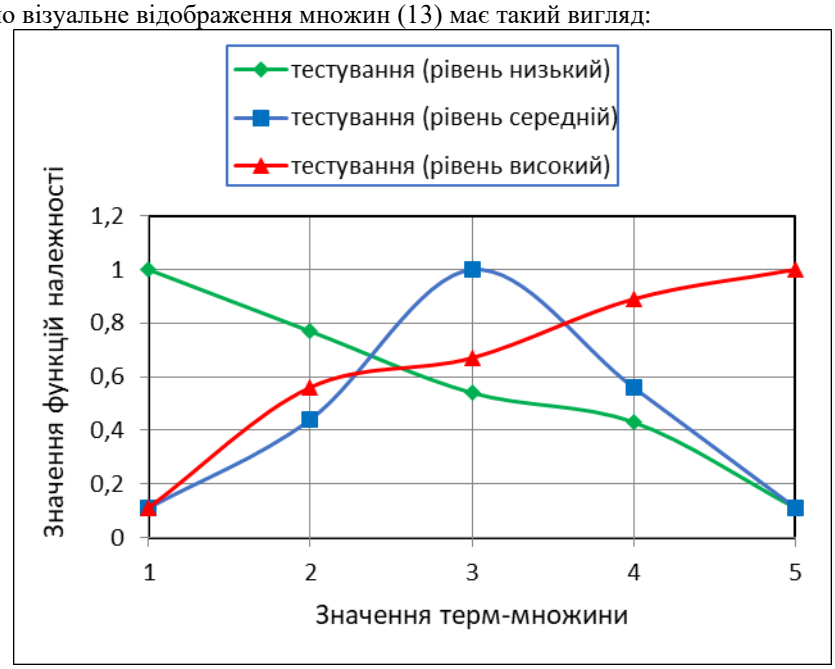

Рис. 3. Візуалізація функцій належності лінгвістичної змінної «тестування»

Як було відзначено вище, у статті застосовано Гаусові функції належності, які на відміну від трикутних або трапецієвидних забезпечують, як видно на графіку, більш плавний перехід у рівнях належності, що корисно для моделювання процесів, де зміни надійності відбуваються поступово.

За реалізованим вище підходом при необхідності виконуються перетворення над рештою лінгвістичних змінних, відображених у моделі логічного виведення, що стануть основою для формування нечітких баз знань і матриць знань та нечітких логічних рівнянь

ВИСНОВКИ З ДАНОГО ДОСЛІДЖЕННЯ І ПЕРСПЕКТИВИ ПОДАЛЬШИХ РОЗВІДОК У ДАНОМУ НАПРЯМІ

У процесі дослідження за темою статті реалізовано початкові етапи процесу формування та прогностичного оцінювання якості програмного забезпечення, моделі і засоби яких побудовані з врахуванням основ нечіткої логіки як однієї з фундаментальних компонент теорії нечітких множин. Подана нами публікація присвячена частині загальної тематики та орієнтована на дослідження базової складової процесу фазифікації, а саме функціям належності лінгвістичних змінних – вирішальних, на наше переконання, чинників (факторів) якості програмного забезпечення. Основою дослідження стала запроектована багаторівнева модель логічного виведення, що послужила алгоритмом подальших кроків, починаючи з лінгвістичних змінних найнижчого рівня. Для їх ідентифікації сформовано терм-множину значень змінних, що забезпечила можливість відображення невизначеностей і нечіткостей реальних

впливових факторів з прив'язкою до опорних точок інтервалу визначення. Побудовано матриці попарних порівнянь рангів лінгвістичної змінної «тестування», рівень якого дотичний до лінгвістичних термів «низький», «середній», «високий». На їх основі розраховано значення функцій належності. Виконано візуалізацію функцій належності, що забезпечило наочність впливу елементів на рівень тестування залежно від означених термів.

Подальший розвиток виконаного дослідження логічно повинен продовжити напрацювання першої частини загальної тематики статті, адже отримані значення функцій належності лінгвістичних змінних утворюють основу для формування нечіткої бази знань, що обумовить проектування нечітких логічних рівнянь для остаточного переходу до розрахунку показника прогнозованої якості програмного забезпечення.

Література

1. Zadeh, L. A. (1996). Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems: Selected Papers by Lotfi A. Zadeh. World Scientific Publishing.

2. Timothy J. Ross. (2004). Fuzzy Logic and Soft Computing. John Wiley &Sons; 2nd edition.

3. Kitchenham, B. (1996). Software quality frameworks. IEEE Software, 13(5), 12-21.

4. Fenton, M., & Ohlsson, N. (2000). Quantitative analysis of software quality using fuzzy logic. Journal of Systems and Software, 53(1), 97-108

5. Gupta, S., & Kumar, S. (2014). Fuzzy logic-based approach for quality assessment of software. International Journal of Computer Applications, 89(6), 27-33.

6. Kaur, C., & Malhotra, R. (2012). Application of fuzzy logic in software engineering. International Journal of Advanced Research in Computer Science and Software Engineering, 2(8), 40-45.

7. Kumar, R., & Singh, M. (2007). A fuzzy logic approach to software quality model. Proceedings of the International Conference on Computing: Theory and Applications, 62-66.

8. Satapathy, S., & Tripathy, R. (2006). Fuzzy logic based software quality estimation. Journal of Software Engineering and Applications, 9(2), 113-120.

9. Mishra, B., & Kumar, R. (2012). Fuzzy-based framework for software quality estimation using ISO 9126. International Journal of Computer Applications, 59(13), 15-19.

10. Jain, N., & Taneja, M. (2011). Fuzzy logic based software reliability and quality evaluation. International Journal of Advanced Computer Science and Applications, 2(5), 23-28.

11. Niazi, A. F. K. (2013). Fuzzy multi-criteria decision-making for software quality assessment. Journal of Software Engineering, 7(4), 35-42.

12. Білас О. Є. (2011). Якість програмного забезпечення та тестування. Навчальний посібник. – Львів: Видавництво Львівської політехніки. 216 с. [Електронний ресурс]. Режим доступу: https://vlp.com.ua/node/7698

13. Авраменко А.С., Авраменко В.С., Косенюк Г.В. (2017). Тестування програмного забезпечення. Навчальний посібник. – Черкаси: ЧНУ імені Богдана Хмельницького. – 284 с. [Електронний ресурс]. Режим доступу: http://lib.istu.edu.ua/index.php?p=22&par=22

14. Крепич С.Я., Співак І.Я. (2020). Якість програмного забезпечення та тестування: базовий курс. Навчальний посібник. Тернопіль: ФОП Паляниця В.А.. – 478с. [Електронний ресурс]. Режим доступу: http://dspace.wunu.edu.ua/handle/316497/39773

15. В.М. Сеньківський, І.В. Піх, І.В. Калиній, Н.Ю. Сеньківський, М.А. Драгоміров. (2022). Методологічні засади формування якості програмного забезпечення (Частина 1: Базова модель факторів якості. Поліграфія і видавнича справа. № 2 (84), Львів: С. 9-21. https://doi.org/10.32403/0554-4866-2022-2-84-9-21

16. В.М. Сеньківський, І.В. Піх, А.В. Кудряшова, О.В., Литовченко, О.З. Білик. (2023). Методологічні засади формування якості програмного забезпечення (Частина 2: Оптимізація моделі факторів якості програмного забезпечення). Поліграфія і видавнича справа. № 1 (85), Львів: С. 11-21. https://doi.org/10.32403/0554-4866-2023-1-85-11-21

17. Піх І.В., Білик О.З., Сеньківський Н. Ю., Андріїв Р.Р. (2024). Браташ С.П. Багатофакторний вибір альтернативних варіантів процесу тестування ПЗ. Вісник Вінницького політехнічного інституту. № 3. С. 78-85. https://doi.org/10.31649/1997-9266-2024-174-3-78-85

18. Піх, І., & Білик, О. (2024). Формування якості програмного забезпечення за методом багатокритеріальної оптимізації. Measuring and computing devices in technological processes, (1), С. 117–124. https://doi.org/10.31891/2219-9365-2024-77-14

19. Ротштейн О. П., Ларюшкін Є. П., Мітюшкін Ю. І. (2008). Soft Computing в біотехнології: багатофакторний аналіз і діагностика: монографія. Вінниця: УНІВЕРСАМ-Вінниця. 144с.

20. Сявавко М.С. (2007). Інформаційна система «Нечіткий експерт». Виавничий центр ЛНУ імені Івана Франка. 320 с.

21. Б. В. Дурняк, І. В. Піх, В. М. Сеньківський. (2022). Теоретичні основи інформаційної концепції формування та оцінювання якості видавничо-поліграфічних процесів. Монографія. – Львів: Українська академія друкарства. 356 с.

22. ISO/IEC 25010:2011. Systems and software engineering. Systems and software Quality Requirements and Evaluation (SQuaRE). System and software quality models. [Introduced 01.03.2011]. Geneva (Switzerland).

References

1. Zadeh, L. A. (1996). Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems: Selected Papers by Lotfi A. Zadeh. World Scientific Publishing.

2. Timothy J. Ross. (2004). Fuzzy Logic and Soft Computing. John Wiley &Sons; 2nd edition.

3. Kitchenham, B. (1996). Software quality frameworks. IEEE Software, 13(5), 12-21.

4. Fenton, M., & Ohlsson, N. (2000). Quantitative analysis of software quality using fuzzy logic. Journal of Systems and Software, 53(1), 97-108

5. Gupta, S., & Kumar, S. (2014). Fuzzy logic-based approach for quality assessment of software. International Journal of Computer Applications, 89(6), 27-33.

6. Kaur, C., & Malhotra, R. (2012). Application of fuzzy logic in software engineering. International Journal of Advanced Research in Computer Science and Software Engineering, 2(8), 40-45.

7. Kumar, R., & Singh, M. (2007). A fuzzy logic approach to software quality model. Proceedings of the International Conference on Computing: Theory and Applications, 62-66.

8. Satapathy, S., & Tripathy, R. (2006). Fuzzy logic based software quality estimation. Journal of Software Engineering and Applications, 9(2), 113-120.

9. Mishra, B., & Kumar, R. (2012). Fuzzy-based framework for software quality estimation using ISO 9126. International Journal of Computer Applications, 59(13), 15-19.

10. Jain, N., & Taneja, M. (2011). Fuzzy logic based software reliability and quality evaluation. International Journal of Advanced Computer Science and Applications, 2(5), 23-28.

11. Niazi, A. F. K. (2013). Fuzzy multi-criteria decision-making for software quality assessment. Journal of Software Engineering, 7(4), 35-42.

12. Bilas O.Ye. (2011). Software Quality and Testing. Textbook. – Lviv: Lviv Polytechnic Publishing House. 216 pages. [Electronic resource]. Access mode: https://vlp.com.ua/node/7698

13. Avramenko A.S., Avramenko V.S., Kosenyuk H.V. (2017). Software Testing. Textbook. – Cherkasy: Bohdan Khmelnytsky National University of Cherkasy. – 284 pages. [Electronic resource]. Access mode: http://lib.istu.edu.ua/index.php?p=22&par=22

14. Krepich S.Ya., Spivak I.Ya. (2020). Software Quality and Testing: Basic Course. Textbook. Ternopil: FOP Palyanytsia V.A. – 478 pages. [Electronic resource]. Access mode: http://dspace.wunu.edu.ua/handle/316497/39773

15. Senkivskyy V. M., Pikh I. V., Kalynii I. V., Senkivskyy N. Y., Drahomirov M.A. (2022). Methodological Foundations of Software Quality Formation (Part 1: Basic Model of Quality Factors). Printing and Publishing. No. 2 (84), Lviv: pp. 9-21. https://doi.org/10.32403/0554-4866-2022-2-84-9-21.

16. Senkivskyy V. M., Pikh I. V., Kudriashova A. V., Lytovchenko O. V., Bilyk O. Z. (2023). Methodological Foundations of Software Quality Formation (Part 2: Optimization of Software Quality Factors Model). Scientific Notes. Issue No. 1 (66). Lviv: UAD, pp. 11-21. https://doi.org/10.32403/0554-4866-2023-1-85-11-21

17. Pikh I.V., Bilyk O.Z., Senkivskyi N.Y., Andriiv R.R., Bratash S.P. (2024). Multifactor Selection of Alternative Options for the Software Testing Process. Bulletin of the Vinnytsia Polytechnic Institute. No. 3. pp. 78-85. https://doi.org/10.31649/1997-9266-2024-174-3-78-85

18. Pikh, I., & Bilyk, O. (2024). Quality Formation of Software Using the Multi-Criteria Optimization Method. Measuring and Computing Devices in Technological Processes, (1), pp. 117–124.https://doi.org/10.31891/2219-9365-2024-77-14

19. Rotshtein O.P., Laryushkin Ye.P., Mityushkin Yu.I. (2008). Soft Computing in Biotechnology: Multifactor Analysis and Diagnostics: Monograph. Vinnytsia: UNIVERSAM-Vinnytsia. 144 pages.

20. Syavavko M.S. (2007). Information System 'Fuzzy Expert'. Publishing Center of Ivan Franko Lviv National University. 320 pages.

21. B. V. Durniak, I. V. Pikh, V. M. Senkivskyy. (2022). Theoreticcal foundations of the information concept for the formation and evaluation of the quality of publishing and printing processes. Monograph. – Lviv: Ukrainian Academy of Printing. 356 pages.

22. ISO/IEC 25010:2011. Systems and software engineering. Systems and software Quality Requirements and Evaluation (SQuaRE). System and software quality models. [Introduced 01.03.2011]. Geneva (Switzerland).