

<https://doi.org/10.31891/2219-9365-2024-77-14>

УДК 004.738.1:316.772.5

ПІХ Ірина

Українська академія друкарства  
Національний університет «Львівська політехніка»  
<https://orcid.org/0000-0002-9909-8444>  
e-mail: [pikhirena@gmail.com](mailto:pikhirena@gmail.com)

БІЛИК Олексій

Українська академія друкарства  
<https://orsid.org/0009-0002-1355-2333>  
e-mail: [lesykbilyk@gmail.com](mailto:lesykbilyk@gmail.com)

## ФОРМУВАННЯ ЯКОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ЗА МЕТОДОМ БАГАТОКРИТЕРІАЛЬНОЇ ОПТИМІЗАЦІЇ

*Якість програмного забезпечення визначається через різні атрибути та характеристики, які впливають на його функціональну придатність, зручність використання, продуктивність або ефективність роботи, зручність супроводження, сумісність, надійність, безпеку, рівень тестування. Забезпечення високої якості програмного забезпечення є неперервним процесом, який вимагає планування, ресурсів і уваги до деталей на кожному етапі розроблення та підтримки програмного забезпечення. Від ефективної його реалізації залежить задоволення потреб користувачів та успіх програмного продукту на ринку.*

*У поданій статті засобами теорії ухвалення рішень досліджено та застосовано інформаційну концепцію до вирішення проблеми формування якості програмного забезпечення, оскільки стандарт перераховує лише умови відповідності програмної продукції вказаним характеристикам, не розкриваючи шляхів отримання результату. Суть запропонованого підходу полягає у виокремленні із загальної множини характеристик якості програмної продукції підмножини Парето, що містить домінуючі фактори пріоритетного впливу на досліджуваний процес. Для оптимізації процесу формування якості програмного забезпечення на етапі його проектування та розроблення використано метод багатокритеріальної оптимізації з теорії дослідження операцій, суть якого полягає у використанні для прийняття рішення нечітких відношень попарних переваг між альтернативами, ступінь яких можна задати числом на одиничному відрізку.*

*З використанням методу багатокритеріальної оптимізації сформовано нечіткі відношення попарних переваг між альтернативами, що ідентифікуються функціями належності. На підставі засобів матричного аналізу теоретично обґрунтовано та розраховано функції корисності та функції належності згорток, що служать інструментом встановлення оптимального варіанту реалізації процесу розроблення програмного забезпечення. З використанням згорток відношень та множини недомінованих альтернатив розраховано функцію належності фінальної згортки, максимальне значення якої визначає оптимальну альтернативу процесу розроблення програмного забезпечення.*

*Ключові слова: множина факторів якості програмного забезпечення, принцип Парето, альтернативний варіант виконання процесу, функція корисності альтернатив, згортка перетину множин, функція належності згорток, оптимальний варіант реалізації процесу.*

PIKH Iryna

Ukrainian Academy of Printing  
Lviv Polytechnic National University

BILYK Oleksii

Ukrainian Academy of Printing

## SOFTWARE QUALITY FORMATION USING MULTICRITERIA OPTIMIZATION METHOD

*The quality of software is determined by various attributes and characteristics that affect its functional suitability, ease of use, performance or efficiency, maintainability, compatibility, reliability, security, and testing level. Ensuring high-quality software is a continuous process that requires planning, resources, and attention to detail at every stage of software development and maintenance. The effective implementation of this process determines user satisfaction and the success of the software product in the market. In this article, the decision-making theory is employed to investigate and apply the information concept to address the issue of forming the quality of software. The standard only enumerates the conditions for compliance of software products with specified characteristics without disclosing the ways to achieve the result. The essence of the proposed approach lies in extracting from the general set of software quality characteristics a Pareto subset containing dominant factors that have a priority impact on the investigated process. To optimize the process of forming software quality during its design and development, a multi-criteria optimization method from operations research theory is utilized. This method uses fuzzy preference relations between alternatives to make decisions, the degree of which can be specified as a number on a unit interval.*

*Using the multi-criteria optimization method, fuzzy preference relations between alternatives are formed, identified by membership functions. Based on matrix analysis tools, utility functions and convolution membership functions are theoretically justified and calculated. These functions serve as a tool for establishing the optimal variant of software development process implementation. Using the convolution of relations and the set of non-dominated alternatives, the membership function of the final convolution is calculated, the maximum value of which determines the optimal alternative for the software development process.*

*Keywords: set of software quality factors, Pareto principle, alternative process execution variant, utility function of alternatives, intersection convolution of sets, convolution membership function, optimal implementation variant of the process.*

### Постановка проблеми у загальному вигляді

#### та її зв'язок із важливими науковими чи практичними завданнями

Основна теза задекларованого напряму дослідження полягає у твердженні та прийнятті того незаперечного факту, що якість продукції доцільного спрямування визначальною мірою залежить від якісного виконання процедур виробничого процесу, що продукує певні вироби. Сказане в повній мірі стосується програмного забезпечення (ПЗ), складний і відповідальний процес розроблення якого потребує системного аналізу, обґрунтованого вибору методів та інструментів реалізації проекту, обдуманого планування. Ефективна реалізація відомих в інженерії програмного забезпечення підходів забезпечить успішність розробки і задоволення потреб замовника.

Додатковим прикладним аспектом у цьому напрямі могла би стати запропонована нами методологія, орієнтована на теоретичному розрахунку за певними критеріями оптимального варіанту процесу розроблення програмного забезпечення, отриманого на підставі аналізу та опрацювання альтернативних варіантів створення програмного продукту. Серед відомих альтернативних підходів до розроблення ПЗ застосовують методології, процеси, інструменти та практики, які відрізняються одне від одного та можуть бути ефективними в різних ситуаціях. Наприклад, гнучка методологія Scrum зазвичай оптимальна для швидкого розвитку проекту, тоді як методологія Waterfall більш корисна для проектів зі статичними вимогами. При цьому важливо, щоб остаточний вибір прийнятного варіанту розроблення ПЗ був зроблений на основі теоретично обґрунтованих критеріїв, застосованих для його розрахунку, враховував конкретні потреби користувачів і вимоги до проекту, зважав на такі суттєві фактори, як розмір та професіоналізм команди, бюджет проекту і терміни його виконання, складність завдань.

В низці публікацій, серед яких праці [1-7], наголос робиться на виборі альтернативних та оптимальних варіантів продукування ПЗ серед відомих. У запропонованому дослідженні розглянуто дещо інший підхід, суть якого полягає у формуванні нечітких відношень попарних переваг між альтернативами, побудові перетину множин невідомованих альтернатив, максимальне значення функції належності якого визначає оптимальний варіант процесу розроблення програмного забезпечення.

Багаторівневі моделі переважаючого впливу виокремлених чинників (визначальних факторів) на формування якості програмного забезпечення [8,9] слугують лише попереднім результатом розв'язання більш загальних завдань, суть яких полягає у використанні методу багатокритеріальної оптимізації з теорії дослідження операцій [15,17], коли для прийняття рішення використовують нечіткі відношення попарних переваг між альтернативами, ступінь яких можна задати числом на відрізку [0;1]. Вказане обумовлює необхідність використання наближених методів, застосування яких забезпечує розв'язки, узгодженість яких за певним показником визначає оптимальний варіант реалізації процесу та належну якість ПЗ.

### Аналіз останніх досліджень та публікацій

У списку літературних джерел наведено частину іноземних публікацій, дотичних до тематики даної статті. Вони стосуються таких питань: [1] – одна з базових робіт з інженерії програмного забезпечення, містить відомості про різні підходи до розробки ПЗ, їх переваги та недоліки; [2] – офіційний маніфест гнучкої розробки, який визначає основні принципи та цінності гнучкого підходу до розробки ПЗ; [3] – розглядає модель спіралі, яка є однією з альтернативних моделей розроблення ПЗ; [4] – описує різні підходи до розроблення ПЗ, вводячи класичні та об'єктно-орієнтовані методології; [5] – уособлює раціональний уніфікований процес як один із варіантів розроблення ПЗ; [6] – зосереджена на практиці використання історій користувачів у гнучких методологіях розроблення ПЗ; [7] – фундаментальна книга з інженерії програмного забезпечення та аспектів його розроблення.

Результати українських наукових розвідок відображають початкові етапи досліджень, спрямованих на створення інформаційної концепції методологічних засад формування належного стану програмного продукту, реалізованого імітаційними моделями, що враховують встановлені відповідними стандартами чинники впливу на якість процесу створення програм [8] та визначають вагові рівні їх переваг [9,14]. Подібні роботи у цьому напрямі орієнтовані на проектування альтернативних варіантів виконання видавничо-поліграфічних процесів та розрахунок оптимального різновиду для досягнення максимальної ефективності виробничої діяльності [10-14].

Водночас, потребує розв'язання завдання пошуку оптимального варіанту процесу розроблення програмного забезпечення, вихідна інформаційна база якого закладена в отриманій раніше моделі факторів якості ПЗ [9]. Для проведення числових розрахунків застосовується принцип, запропонований італійським економістом Парето, що обумовлює виокремлення для дослідження обмеженої кількості чинників, рівень важливості яких є переважаючим [10-15].

### Виклад основного матеріалу дослідження

Запропонована стаття розвиває тематику попередніх публікацій [8,9]. В наведених працях на підставі характеристик стандарту ISO/IEC 25010 [16] виокремлено множину факторів якості програмного забезпечення, формалізовано зв'язки між ними за допомогою семантичної мережі, визначено рівні їх

важливості та числові вагові переваги факторів, синтезовано та оптимізовано багаторівневу модель пріоритетного впливу вказаних чинників на якість процесу розроблення програмного продукту.

Оскільки реалізація теоретичних викладок використаного методу багатокритеріальної оптимізації потребує інформації про вагові значення та відповідні їм рівні переваг факторів, наведемо модифіковану модель пріоритетного їх впливу на якість ПЗ (рис. 1), отриману у публікації [9].



рис. 1. Оптимізована модель пріоритетного впливу факторів на якість програмного забезпечення

На основі рис. 1 формуємо підмножину Парето, до якої віднесемо перші чотири фактори, вагові значення яких переважають інші, а саме:

- $F_1 (X_8)$  – «Рівень тестування», початкова вага якого  $w_{s8} = 335$  у. о.;
- $F_2 (X_6)$  – «Безпека»,  $w_{s6} = 242$  у. о.;
- $F_3 (X_1)$  – «Функціональна придатність»,  $w_{s1} = 171$  у. о.;
- $F_4 (X_5)$  – «Сумісність»,  $w_{s5} = 100$  у. о.

Розкриємо функціональну сутність факторів множини Парето [9], для яких наведено тільки узагальнені назви, що втілюють сукупності процедур, виконання яких забезпечить часткові якісні показники та обумовить певну якість альтернативного варіанту. Отже, враховуючи суть фактора «Рівень тестування», розробник ПЗ повинен забезпечити його строгу відповідність закладеним у технічному завданні можливостям, надійність продукту, його адаптивність до інших апаратних засобів та операційних систем, простоту встановлення. Фактор «Безпека» обумовить непідробність і цілісність проти втручання, конфіденційність та автентичність ПЗ. Фактор «Функціональна придатність» визначить повноту покриття задач і функцій користувача, їх коректність і доцільність. Завершальний фактор множини Парето «Сумісність» зумовить здатність ПЗ сумісно функціонувати з іншою системою та обмінюватися з нею інформацією.

Опишемо коротко суть методу багатокритеріальної оптимізації [14,17], використаного нами для одного з запропонованих варіантів формування якості програмного забезпечення. При цьому для прийняття рішення використаємо нечіткі відношення переваг, коли ступінь наявності попарних переваг між альтернативами можна задати числом на відрізку [0;1].

Нехай маємо пару альтернатив  $(x, y)$ . Відношення нестрогої переваги  $F$  означає, що для двох альтернатив можливі твердження:  $x$  не гірше від  $y$ , що відповідає  $x \geq y$ , аналогічно  $(x, y) \in F$ ;  $y$  не гірше від  $x$ , або скорочено  $(y \geq x)$ , що відповідає  $(y, x) \in F$ ;  $x$  та  $y$  не порівняльні, значить  $(x, y) \notin F$  та  $(y, x) \notin F$ .

У випадку чітких функцій корисності  $f_j$  на множині  $X$  альтернатива  $x$  з вищою оцінкою  $f_j(x)$  є кращою за фактором  $j$  від альтернативи  $y$  з оцінкою  $f_j(y)$  і описується чітким відношенням переваги  $F_j$  на множині  $X$ :

$$F_j = \{(x, y) : f_j(x) \geq f_j(y), x, y \in X\}. \quad (1)$$

Треба вибрати таку альтернативу  $x_0 \in X$ , яка матиме найвищу оцінку за множиною усіх виокремлених факторів (критеріїв), тобто

$$f_j(x_0) \geq f_j(y), \forall j = 1, m; \forall y \in X. \quad (2)$$

Альтернативи такого типу називають Парето-оптимальними або ефективними і вони є розв'язком задачі прийняття рішень при нечіткому відношенні переваги на множині альтернатив.

Оскільки вихідними критеріями задачі багатокритеріального вибору альтернативи є фактори (критерії) множини Парето, необхідно здійснити згортку багатьох критеріїв в один скалярний. Для цього використовуємо спосіб перетину, суть якого полягає в наступному [14].

Позначимо  $Z_1 = \bigcap_{j=1}^m F_j$ . Можна стверджувати, що множина альтернатив з відношенням переваги  $Z_1$

відповідає множині альтернатив з функціями корисності  $f_j(x)$ . Це означає, що набір відношень  $F_j$  ( $j = 1, m$ ) заміняємо їх перетином і знаходимо невідомі альтернативи за нечітким відношенням переваги  $Z_1$ . Якщо  $\mu_j(x, y)$  – функція належності нечіткого відношення переваги  $F_j$ , то

$$\mu_j(x, y) = \begin{cases} 1, \text{ якщо } f_j(x) \geq f_j(y), \text{ тобто } (x, y) \in F_j; \\ 0, \text{ якщо } (x, y) \notin F. \end{cases} \quad (3)$$

З урахуванням (3) функція належності (згортка критеріїв) для нечіткого відношення переваги  $Z_1$  матиме такий вигляд:

$$\mu_{Z_1}(x, y) = \min \{\mu_1(x, y), \mu_2(x, y), \dots, \mu_m(x, y)\}. \quad (4)$$

Згортка критеріїв (4) може бути виражена через вагові значення факторів  $w_j$  та відповідні функції корисності таким чином:

$$R(x) = \min_j w_j f_j(x). \quad (5)$$

Аналогічно попередньому вводиться ще один тип згортки відношень  $\{F_j\}$ , який використовує ваги та функції корисності факторів:

$$Z_2 = \sum_{j=1}^m w_j f_j(x), \text{ де } \sum_{j=1}^m w_j = 1, w_j \geq 0. \quad (6)$$

Згортці  $Z_2$  відповідає функція належності  $\mu_{Z_2}(x, y) = \sum_{j=1}^m w_j \mu_j(x, y)$ .

З урахуванням введених величин та послуговуючись [14], наведемо алгоритм задачі вибору оптимального варіанту процесу розроблення програмного забезпечення на основі нечіткого відношення переваги.

1. Нехай якість процесу розроблення програмного забезпечення на множині альтернатив  $X \{x_1, x_2, x_3\}$  оцінюється за факторами такими нечіткими відношеннями:  $F_1$  – сукупність переваг у варіантах тестування ПЗ;  $F_2$  – безпеки ПЗ;  $F_3$  – функціональної придатності ПЗ;  $F_4$  – сумісності ПЗ. Відношенням  $F_j$  відповідатимуть вагові значення факторів  $w_j, j = 1, 4$  та функції належності  $\mu_j(x, y)$ .

Знаходимо згортку відношень  $Z_1 = \bigcap_{j=1}^4 F_j$ .

2. У множині  $\{X, Z_1\}$  встановлюємо множину невідомі альтернатив  $Z_1^{no}$  з функціями належності

$$\mu_{Z_1}^{no}(x) = 1 - \sup_{y \in X} \left\{ \sum_{j=1}^4 \mu_{Z_1}(y, x) - \mu_{Z_1}(x, y) \right\}. \quad (7)$$

3. На основі виразу (6) для згортки  $Z_2$  знаходимо адитивну згортку відношень з функціями належності

$$\mu_{Z_2}(x, y) = \sum_{j=1}^4 w_j \mu_j(x, y), \quad \sum_{j=1}^4 w_j = 1, w_j \geq 0. \quad (8)$$

4. Визначаємо множину невідомованих альтернатив для  $Z_2$ :

$$\mu_{Z_2}^{no}(x) = 1 - \sup_{y \in X} \left\{ \sum_{j=1}^4 \mu_{Z_2}(y, x) - \mu_{Z_2}(x, y) \right\}. \quad (9)$$

5. Шукаємо спільну множину невідомованих альтернатив як перетин множин  $Z_1^{no}$  та  $Z_2^{no}$ , тобто  $Z_{no} = Z_1^{no} \cap Z_2^{no}$ , функція належності якого

$$\mu_{no}(x) = \min \left\{ \mu_{Z_1}^{no}(x), \mu_{Z_2}^{no}(x) \right\}. \quad (10)$$

Оптимальною вважається альтернатива, функція належності якої  $\mu_{no}(x)$  максимальна, що означає найвищий ступінь невідомованості.

Для початку задамо множину альтернатив  $X \{x_1, x_2, x_3\}$ . Використаємо наведені вище фактори множини Парето, які визначають сукупності відношень переваг, та відповідні їм ваги факторів для згортки  $Z_2$ , а саме:  $w_1 = 0,5$ ;  $w_2 = 0,3$ ;  $w_3 = 0,1$ ;  $w_4 = 0,1$  [18].

Встановимо відношення переваги у варіантах за кожним з факторів.

Рівень тестування ПЗ ( $F_1$ ):  $x_1 > x_2$ ,  $x_1 < x_3$ ,  $x_2 < x_3$ .

Безпека ПЗ ( $F_2$ ):  $x_1 > x_2$ ,  $x_1 > x_3$ ,  $x_2 > x_3$ .

Функціональна придатність ПЗ ( $F_3$ ):  $x_1 < x_2$ ,  $x_1 < x_3$ ,  $x_2 > x_3$ .

Сумісність ПЗ ( $F_4$ ):  $x_1 > x_2$ ,  $x_1 > x_3$ ,  $x_2 = x_3$ .

За наведеними відношеннями та умовою (3) будемо матриці. Для відношень  $F_1$  переваги рівня тестування у варіантах помістимо у табл. 1.

Таблиця 1

	$x_i / x_j$	$x_1$	$x_2$	$x_3$
$\mu_{F_1}(x_i, x_j)$	$x_1$	1	1	0
	$x_2$	0	1	0
	$x_3$	1	1	1

Матриця відношень  $F_2$  для переваг альтернатив у заданих варіантах, що стосуються фактора «Безпека ПЗ», відображена табл. 2.

Таблиця 2

	$x_i / x_j$	$x_1$	$x_2$	$x_3$
$\mu_{F_2}(x_i, x_j)$	$x_1$	1	1	1
	$x_2$	0	1	1
	$x_3$	0	0	1

Для відношень  $F_3$ , що інтерпретують можливі якості функціональної придатності програмного забезпечення, матриця порівняльних характеристик матиме вигляд, відображений у табл. 3.

Таблиця 3

	$x_i / x_j$	$x_1$	$x_2$	$x_3$
$\mu_{F_3}(x_i, x_j)$	$x_1$	1	0	0
	$x_2$	1	1	1
	$x_3$	1	0	1

Відношення  $F_4$ , що стосуються сумісності програмного забезпечення, його здатності функціонувати з іншою програмною системою та обмінюватися з нею інформацією, його порівняльних характеристик матиме таке матричне відображення (табл. 4).

Таблиця 4

$\mu_{F_4}(x_i, x_j)$	$x_i / x_j$	$x_1$	$x_2$	$x_3$
	$x_1$	1	1	1
	$x_2$	0	1	1
	$x_3$	0	1	1

Наступний крок – побудова згортки відношень  $Z_1 = F_1 \cap F_2 \cap F_3 \cap F_4$ , для якої матриця значень функції належності матиме такий вигляд.

Таблиця 5

$\mu_{Z_1}(x_i, x_j)$	$x_i / x_j$	$x_1$	$x_2$	$x_3$
	$x_1$	1	0	0
	$x_2$	0	1	0
	$x_3$	0	0	1

Згортка  $Z_1$  вказує на чітку недомінованість альтернатив. На основі табл. 5 та виразу (7) визначаємо множину недомінованих альтернатив.

$$\mu_{Z_1}^{n0}(x) = 1 - \sup_{y \in X} \{ \mu_{Z_1}(y, x) - \mu_{Z_1}(x, y) \}.$$

Для кожної з альтернатив отримуємо такі значення:

$$\mu_{Z_1}^{n0}(x_1) = 1 - \sup_{y \in X} \{ 0 - 0; 0 - 0 \} = 1;$$

$$\mu_{Z_1}^{n0}(x_2) = 1 - \sup_{y \in X} \{ 0 - 0; 0 - 0 \} = 1;$$

$$\mu_{Z_1}^{n0}(x_3) = 1 - \sup_{y \in X} \{ 0 - 0; 0 - 0 \} = 1.$$

Враховуючи результат обчислень, маємо:  $\mu_{Z_1}^{n0}(x) = [1; 1; 1]$ .

Знаходимо нечітке відношення переваги  $Z_2$ , так звану адитивну згортку відношень [14]  $F_j$ ,  $j = 1, 4$ , за формулою  $Z_2 = \sum_{j=1}^4 w_j f_j(x)$ . Розрахунок функцій належності згортки  $\mu_{Z_2}(x_i, x_j) = \sum_{k=1}^4 w_k \mu_{F_k}(x_i, x_j)$  виконаємо за формулами:

$$\mu_{Z_2}(x_1, x_2) = w_1 \mu_{F_1}(x_1, x_2) + w_2 \mu_{F_2}(x_1, x_2) + w_3 \mu_{F_3}(x_1, x_2) + w_4 \mu_{F_4}(x_1, x_2)$$

$$\mu_{Z_2}(x_1, x_3) = w_1 \mu_{F_1}(x_1, x_3) + w_2 \mu_{F_2}(x_1, x_3) + w_3 \mu_{F_3}(x_1, x_3) + w_4 \mu_{F_4}(x_1, x_3)$$

$$\mu_{Z_2}(x_2, x_1) = w_1 \mu_{F_1}(x_2, x_1) + w_2 \mu_{F_2}(x_2, x_1) + w_3 \mu_{F_3}(x_2, x_1) + w_4 \mu_{F_4}(x_2, x_1)$$

$$\mu_{Z_2}(x_2, x_3) = w_1 \mu_{F_1}(x_2, x_3) + w_2 \mu_{F_2}(x_2, x_3) + w_3 \mu_{F_3}(x_2, x_3) + w_4 \mu_{F_4}(x_2, x_3)$$

$$\mu_{Z_2}(x_3, x_1) = w_1 \mu_{F_1}(x_3, x_1) + w_2 \mu_{F_2}(x_3, x_1) + w_3 \mu_{F_3}(x_3, x_1) + w_4 \mu_{F_4}(x_3, x_1)$$

$$\mu_{Z_2}(x_3, x_2) = w_1 \mu_{F_1}(x_3, x_2) + w_2 \mu_{F_2}(x_3, x_2) + w_3 \mu_{F_3}(x_3, x_2) + w_4 \mu_{F_4}(x_3, x_2)$$

Розраховані за наведеними вище рівняннями значення функцій належності згортки  $\mu_{Z_2}(x_i, x_j)$  для нечіткого відношення  $Z_2$  помістимо у табл. 6.

Таблиця 6

$\mu_{Z_2}(x_i, x_j)$	$x_i / x_j$	$x_1$	$x_2$	$x_3$
	$x_1$	1	0.9	0.4
	$x_2$	0.1	1	0.5
	$x_3$	0.6	0.6	1

При розрахунку згорток використано значення функцій належності, задані матрицями відношень у таблицях 1-4.

Для відношення  $Z_2$  знайдемо множину невідомованих альтернатив за формулою (9):

$$\mu_{Z_2}^{no}(x_1) = 1 - \sup\{(0.1 - 0.9); (0.6 - 0.4)\} = 0.8;$$

$$\mu_{Z_2}^{no}(x_2) = 1 - \sup\{(0.9 - 0.1); (0.6 - 0.5)\} = 0.2;$$

$$\mu_{Z_2}^{no}(x_3) = 1 - \sup\{(0.4 - 0.6); (0.5 - 0.6)\} = 1.$$

У результаті отримаємо  $\mu_{Z_2}^{no}(x_i) = [0.8; 0.2; 1]$ .

Останній крок полягає у знаходженні згортки перетину множин  $Z_1^{no}$  та  $Z_2^{no}$ , тобто  $Z_{no} = Z_1^{no} \cap Z_2^{no}$ , з функцією належності

$$\mu_Z^{no}(x_i) = \min\{\mu_{Z_1}^{no}(x_i), \mu_{Z_2}^{no}(x_i)\}, \quad i = 1, 3. \quad (2.39)$$

З урахуванням того, що функція належності  $\mu_{Z_1}^{no}(x_i) = [1; 1; 1]$ , отримаємо  $\mu_Z^{no}(x_i) = [0.8; 0.2; 1]$ .

Функція належності згортки  $Z$  свідчить, що оптимальною альтернативою процесу розроблення програмного забезпечення із заданими вище відношеннями переваги корисності факторів є варіант  $x_3$ , функція належності якого має максимальне значення.

### Висновки з даного дослідження і перспективи подальших розвідок у даному напрямі

У ході дослідження розглянуто один з підходів покращення якості програмного забезпечення за рахунок формування передумов для визначення теоретично обґрунтованих варіантів виконання процесу розроблення програмної продукції. Для формування якості програмного забезпечення на етапі його проектування та розроблення використано метод багатокритеріальної оптимізації з теорії дослідження операцій, суть якого полягає у використанні для прийняття рішення нечітких відношень попарних переваг між альтернативами. Важливою вихідною умовою при цьому вважається виокремлення факторів множини Парето, упорядкованих за важливістю переважаючого впливу на досліджуваний процес. Нечітким відношенням відповідають вагові значення факторів та розраховані функції належності. На основі згорток відношень та множини невідомованих альтернатив розраховано для фінальної згортки функцію належності, максимальне значення якої визначає оптимальну альтернативу процесу розроблення програмного забезпечення, що уможливує отримання якісного програмного продукту.

Отримані результати можуть стати основою для дослідження прогностичного оцінювання рівня якості процесу розроблення програмного забезпечення з використанням моделей і засобів нечіткої логіки, як однієї з визначальних складових теорії нечітких множин.

### Література

1. Sommerville, I. Software Engineering. Pearson. 2018.
2. Agile Alliance. Manifesto for Agile Software Development. 2001. Retrieved from <https://agilemanifesto.org/>
3. Boehm, B. A Spiral Model of Software Development and Enhancement. ACM SIGSOFT Software Engineering Notes. 1988. 11(4), 14-24.
4. Schach, S. R. Object-Oriented and Classical Software Engineering. McGraw-Hill Education. 2014.
5. Kruchten, P. The Rational Unified Process: An Introduction (2nd ed.). Addison-Wesley. 2003.
6. Cohn, M. User Stories Applied: For Agile Software Development. Addison-Wesley Professional. 2004.
7. Pressman, R. S. Software Engineering: A Practitioner's Approach (8th ed.). McGraw-Hill Education. 2014.
8. Сеньківський В. М., Піх І. В., Калиній І. В., Сеньківський Н. Ю., Драгоміров М.А. Методологічні засади формування якості програмного забезпечення (Частина 1: Базова модель факторів якості). *Поліграфія і видавничі справи*. № 2 (84), Львів: 2022. С. 9-21.
9. Сеньківський В. М., Піх І. В., Кудряшова А. В., Литовченко О. В. Білик О. З. Методологічні засади формування якості програмного забезпечення (Частина 2: Оптимізація моделі факторів якості програмного забезпечення). *Наукові записки*. Вип. № 1 (66). Львів: УАД, 2023. С. 83-93.
10. Гавенко С. Ф., Піх І. В., Сеньківська Н. Є. Розрахунок альтернативних варіантів випуску видання. *Поліграфія і видавничі справи*. № 3 (55), Львів: 2011. – С. 66-71.
11. Піх І. В., Сеньківський В. М., Андріїв Р. Р. Проектування та розрахунок альтернативних варіантів реалізації технологічних процесів. *Технологія і техніка друкарства*. Київ: ВПІ НТУУ «Київський політехнічний інститут», 2015. № 2 (48). С. 55-62.
12. Сеньківський В. М., Піх І. В., Петяк Ю. Ф., Калиній І. В. Теоретичні основи забезпечення якості видавничо-поліграфічних процесів (Частина 3. Проектування альтернативних варіантів). *Наукові записки*. 2016. № 2 (53). С. 47-56.

13. Піх І. В. Кудряшова А. В. Багатофакторний вибір альтернативних варіантів композиційного оформлення видання на основі лінійного згортання критеріїв. *Наукові записки*. 2017. № 2 (55). С. 41-46.
14. Б. В. Дурняк, І. В. Піх, В. М. Сеньківський. Теоретичні основи інформаційної концепції формування та оцінювання якості видавничо-поліграфічних процесів. Монографія. – Львів: Українська академія друкарства, 2022. 356 с.
15. Бартіш М. Я., Дудзяний І. М. Дослідження операцій. Частина 3. Ухвалення рішень і теорія ігор. Львів: Видавн. центр ЛНУ імені Івана Франка, 2009. 278 с.
16. 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).
17. Зайченко Ю. П. Дослідження операцій. Підручник. Сьоме видання, перероблене та доповнене. Київ: Видавничий Дім «Слово», 2006. – 816 с.
18. Свідоцтво про реєстрацію авторського права на твір № 41832. Україна. Імітаційне моделювання в системному аналізі методом бінарних порівнянь (комп'ютерна програма). Авторські майнові права належать І. В. Гілеті, В. М. Сеньківському, О. В. Мельникову. Зареєстровано 17.01.2012.

### References

1. Sommerville, I. Software Engineering. Pearson. 2018.
2. Agile Alliance. Manifesto for Agile Software Development. 2001. Retrieved from <https://agilemanifesto.org/>
3. Boehm, B. A Spiral Model of Software Development and Enhancement. ACM SIGSOFT Software Engineering Notes. 1988. 11(4), 14-24.
4. Schach, S. R. Object-Oriented and Classical Software Engineering. McGraw-Hill Education. 2014.
5. Kruchten, P. The Rational Unified Process: An Introduction (2nd ed.). Addison-Wesley. 2003.
6. Cohn, M. User Stories Applied: For Agile Software Development. Addison-Wesley Professional. 2004.
7. Pressman, R. S. Software Engineering: A Practitioner's Approach (8th ed.). McGraw-Hill Education. 2014.
8. Senkivskyy V. M., Pikh I. V., Kalynii I. V., Senkivskyy N. Y., Drahomirov M.A. Methodological Foundations of Software Quality Formation (Part 1: Basic Model of Quality Factors). Printing and Publishing. No. 2 (84), Lviv: 2022. pp. 9-21. <https://doi.org/10.32403/0554-4866-2022-2-84-9-21>.
9. Senkivskyy V. M., Pikh I. V., Kudriashova A. V., Lytovchenko O. V., Bilyk O. Z. Methodological Foundations of Software Quality Formation (Part 2: Optimization of Software Quality Factors Model). Scientific Notes. Issue No. 1 (66). Lviv: UAD, 2023. pp. 83-93.
10. Havenko S. F., Pikh I. V., Senkivska N. Ye. Calculation of Alternative Release Options. Printing and Publishing Affairs. No. 3 (55), Lviv: 2011. pp. 66-71.
11. Pikh I. V., Senkivskyy V. M., Andriiv R. R. Design and Calculation of Alternative Implementation Options for Technological Processes. Printing Technology and Equipment. Kyiv: VPI NTUU "Kyiv Polytechnic Institute," 2015. No. 2 (48). pp. 55-62.
12. Senkivskyy V. M., Pikh I. V., Petiak Yu. F., Kalyniy I. V. Theoretical Foundations of Quality Assurance in Publishing and Printing Processes (Part 3. Designing Alternative Options). Scientific Notes. 2016. No. 2 (53). pp. 47-56.
13. Pikh I. V., Kudriashova A. V. Multi-factorial Choice of Alternative Design Options for Publications Based on Linear Convolution of Criteria. Scientific Notes. 2017. No. 2 (55). pp. 41-46.
14. Б. В. Дурняк, І. В. Піх, В. М. Сеньківський. Теоретичні основи інформаційної концепції формування та оцінювання якості видавничо-поліграфічних процесів. Монографія. – Львів: Українська академія друкарства, 2022. 356 с.
15. Bartish M. Y., Dudzianyi I. M. Operations research. Part 3. Decision making and game theory. Lviv: Publishing center of Ivan Franko National University., 2009. 278 p.
16. 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).
17. Zaichenko Y. P. Operations research. Seventh edition, Revised and expanded. Kyiv: Publishing house «Slovo», 2006. - 816 p.
18. Certificate of copyright registration for work No. 41832. Ukraine. Simulation modeling in system analysis using binary comparisons method (computer program). The copyrights are owned by I. V. Hileta, V. M. Senkivskyy, O. V. Melnykov. Registered 17.01.2012.