

І. В. Піх^{1,2}
О. З. Білик¹
Н. Ю. Сеньківський¹
Р. Р. Андрійв³
С. П. Браташ²

БАГАТОФАКТОРНИЙ ВИБІР АЛЬТЕРНАТИВНИХ ВАРІАНТІВ ПРОЦЕСУ ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

¹Українська академія друкарства, Львів;

²Національний університет «Львівська політехніка»;

³Львівський державний університет безпеки життєдіяльності

Враховано, що вирішення проблеми забезпечення якості програмного забезпечення суттєво залежить від одного з завершальних етапів, пов'язаного з рівнем тестування програмної продукції, що обумовлює необхідність у розробленні альтернативних підходів до процесу тестування, які б дозволили ефективно враховувати різноманітні фактори впливу на його результативність, складність виробничих та організаційних умов, вимоги до якості остаточної продукції. Запропоновано інформаційне трактуванні концепції формування якості вказаної категорії з використанням засобів теорії ухвалення рішень для моделювання альтернативних та визначення оптимальних варіантів процесу тестування програм. Розглянуто застосування альтернативного підходу до вибору варіантів процесу тестування програмного забезпечення, основаного на методі лінійного згортання критеріїв, концепції Парето, врахуванні вагових переваг факторів впливу на якість тестування програм та їхньої дольовій участі у формуванні альтернатив. Наведено теоретичні засади методу лінійного згортання критеріїв та його застосування до тестування програм. Досліджено практичну реалізацію методу, початковим етапом якого є виокремлення із загального набору характеристик тестування програмної продукції множини Парето, що містить домінуючі фактори пріоритетного впливу на процес тестування. Використання принципу Парето забезпечило визначення оптимального набору комбінацій тестових варіантів, які максимально задовольняють потреби тестування за обмежених ресурсів. Запроектовано альтернативні варіанти виконання процесу тестування програмного забезпечення з використанням функцій корисності та часток ефективності факторів в альтернативних варіантах. Побудовано та опрацьовано матриці попарних порівнянь для розрахунку уточнених вагових пріоритетів факторів множини Парето та значень функцій корисності факторів з огляду на їхню частку у формуванні альтернатив. Розраховано та визначено оптимальний серед альтернативних варіантів процесу тестування програмного забезпечення за критерієм максимального значення об'єднаних цільових функціоналів, який забезпечує належну якість імітаційного моделювання.

Ключові слова: фактори якості тестування програмного забезпечення, метод лінійного згортання критеріїв, принцип Парето, альтернативний варіант, матриця попарних порівнянь, функція корисності, функціонал якості, оптимальний варіант.

Вступ

У сучасному світі, де термін випуску програмного забезпечення (ПЗ) є одним з критичних факторів успіху, важливо забезпечити ефективність та якість процесу тестування. Тестування програмного забезпечення зазвичай вимагає великої кількості ресурсів та часу, що може стати перешкодою для швидкого випуску продукту на ринок. У цьому контексті виникає необхідність в пошуку альтернативних підходів до тестування, які б забезпечили оптимальне використання наявних можливостей та максимальну якість продукції.

Проблема тестування програмного забезпечення набуває нових підходів з появою різноманітних технологій та методів розроблення. Наявні методи тестування можуть виявитися недостатньо ефективними у забезпеченні високої якості ПЗ через їхню обмежену можливість врахування різноманітних критеріїв та сценаріїв тестування. У зв'язку з цим виникає необхідність у розробленні альтернативних підходів до процесу тестування, які б дозволяли ефективно враховувати різноманітні фактори впливу на якість процесу тестування, складність виробничих та організаційних умов, вимоги до якості остаточної продукції.

Одним з можливих варіантів до оптимізації процесу тестування є використання методу лінійного згортання критеріїв та принципу Парето, що обумовлює врахування багаторівневості моделі факторів якості тестування та пошук оптимальних рішень за критерієм максимального значення об'єднаних цільових функціоналів. Концепція Парето, відома також як правило 80/20, в основі якого закладено твердження, що приблизно 80 % результатів залежать від 20 % вихідних даних. Цей принцип уперше сформульовано італійським економістом Вільфредо Парето наприкінці 19 століття. По суті він вказує на те, що не всі зусилля і ресурси рівнозначні у своєму впливі на результати і наголошує на важливості ідентифікації і зосередження зусиль на ключових чинниках, які дадуть найбільший ефект. Незважаючи на емпіричність твердження, ідея пошуку ключових факторів або стратегій, які забезпечують найбільший результат, залишається цінною і застосована авторами у пропонованому дослідженні.

Крім заявленого вище, в публікації використано методи інформаційного спрямування, а саме засоби системного аналізу, теорії ієрархічних багаторівневих систем, теорії моделювання [1]—[3], застосування яких у дослідженні процесів видавничо-поліграфічного напрямку [4]—[6] підтвердило свою ефективність та прикладну доцільність. Важливою умовою при цьому є наявність упорядкованої за ваговими пріоритетами множини факторів впливу на якість тестування програмного забезпечення та підмножини Парето, що містить домінуючі чинники процесу тестування.

Аналіз останніх досліджень та публікацій. Переважна більшість публікацій належить авторам, що узагальнюють свій досвід, пройшовши шлях від розробників програм до керівників підрозділів чи фірм, орієнтованих на продукування складних програмних систем з використанням ефективних методів тестування. Основу літератури з цього напрямку складають праці іноземних авторів [7]—[13] та декілька українських посібників [14]—[17]. Напрями досліджень зосереджені на таких темах: [7] — методи експлораторного тестування, що є одним з альтернативних підходів до тестування ПЗ; [8] — штучний інтелект, який є новою парадигмою в розробленні програмного забезпечення; [9] — архітектура програмних веб-додатків, тестування API, методи кодування та стандарти для покращення розробки та управління автоматизацією тестування API; [10] — різні аспекти тестування програмного забезпечення в Agile командах; [11] — аналіз результатів досліджень стосовно впровадження DevOps (DevOps є культурною та процесною філософією, спрямованою на покращення співпраці між розробниками програмного забезпечення та ІТ-фахівцями з експлуатації, а також на автоматизацію процесів розроблення, тестування та впровадження програмного забезпечення); [12] — книга є класичним джерелом з тестування програмного забезпечення — описує різні методи, включаючи ручне, автоматизоване, модельне тестування; [13] — методи та практики поєднання тестування з Agile та DevOps.

Наявність вищезазначених та інших подібних праць не знімає потреби в удосконаленні методів і засобів тестування програм. Проблема не тільки в унікальності та складності цього завершального етапу життєвого циклу програмного забезпечення, але й у його творчо-пошуковій суті, яка обумовлює, крім загальної кваліфікації, певні індивідуальні риси тестувальника.

Метою роботи є виділення із загального набору чинників (факторів) тестування програмної продукції множини Парето, що містить домінуючі фактори впливу на досліджуваний процес; проектування альтернативних та визначення оптимального варіанта тестування ПЗ за критерієм максимального значення об'єднаних цільових функціоналів.

Результати дослідження

Тести програмного забезпечення можуть виконуватися різними методами, залежно від потреб проекту та можливостей проектувальників. Оскільки тематика статті орієнтована на розширення засобів процесу тестування, наведемо декілька основних методів тестування програм [7]—[13].

Модульне тестування (Unit Testing) — тестування окремих функцій, методів або модулів.

Інтеграційне тестування (Integration Testing) — тестування взаємодії між різними модулями або компонентами програми.

Системне тестування (System Testing) — перевірка програми як цілого, разом зі всіма її функціями та компонентами. Може містити функціональне тестування, тестування з використанням сценаріїв, а також тестування відповідності вимогам.

Тестування прийняття (Acceptance Testing) — тестування з погляду користувачів, щоб переконалися, що програмне забезпечення відповідає їхнім очікуванням та вимогам.

Тестування навантаження (Load Testing) — перевірка роботи програми за великим обсягом даних або під великим навантаженням.

Тестування безпеки (Security Testing) — перевірка програмного забезпечення на вразливість до атак, зловмисних втручань та інших загроз безпеці.

Автоматизоване тестування (Automated Testing) — використання програм для автоматизації процесу тестування, що забезпечує ефективніше проведення тестів.

Пропонований нами підхід орієнтований на розширення можливостей наведених вище методів та оптимізацію процесу тестування програм за рахунок використання методу лінійного згортання критеріїв та принципу Парето. Розгляд альтернативних варіантів тестування сприятиме збільшенню ефективності та якості тестових процедур, що також позитивно вплине на загальну якість розроблюваного програмного продукту. Для кращого розуміння особливостей застосування методу коротко подамо математичне трактування концепції теорії ухвалення рішень, складовою частиною якої вважається вказаний метод [1], [3].

Фундаментальними поняттями вказаної теорії виступають часткові функціонали факторів процесу $f_1(x), \dots, f_m(x)$, що вважаються аргументами загальної функції мети $f(x) = (f_1(x), \dots, f_m(x))$, та множина альтернатив D . З урахуванням цих основних понять задача багатокритеріальної (у нашому випадку багатофакторної) оптимізації на множині D полягатиме у знаходженні максимального значення функцій корисності. Застосування методу лінійного згортання критеріїв для вибору альтернативи передбачає лінійне об'єднання за кожним з факторів часткових функціоналів

$$F(w, x) = \sum_{i=1}^m w_i f_i(x) \rightarrow \max; \quad w \in W, \quad x \in D, \quad (1)$$

$$\text{де } W = \left\{ w = (w_1, \dots, w_m)^T; \quad w_i > 0; \quad \sum_{i=1}^m w_i = 1 \right\}.$$

Вагові пріоритети факторів w_i співвідносяться з числовими значеннями функцій корисності. Вибір оптимальної альтернативи здійснюється на підставі теореми методу багатокритеріальної теорії корисності, яка узалежнює наявність незалежних за корисністю та перевагою критеріїв з існуванням функції корисності, що виражається таким чином:

$$U(x) = \sum_{i=1}^m w_i u_i(y_i). \quad (2)$$

Максимальне значення функції корисності вважається критерієм вибору оптимального варіанта виконання багатофакторного процесу. Згідно з (2) вважається, що $U(x)$ — багатофакторна функція корисності альтернативи x ($0 \leq U(x) \leq 1$); $u_i(y_i)$ — функція корисності i -го фактора ($0 \leq u_i(y_i) \leq 1$); y_i — значення альтернативи x за фактором i ; w_i — умовна нормалізована числова вага (ефективність) i -го фактора в альтернативних варіантах, причому $0 < w_i < 1$.

Етапи реалізації методу опишемо стосовно компонент специфіки тестування ПЗ.

Початковим вихідним базисом є багаторівнева модель факторів тестування ПЗ з інформацією про вагові пріоритети чинників [18] (рис. 1).

У подальшому із загальної множини чин-



Рис. 1. Модель пріоритетного впливу факторів на якість тестування ПЗ

ників тестування вибираємо домінантні фактори:

- X_3 — кваліфікація тестувальників (вага фактора 361 у.о.);
- X_1 — планування тестування (вага фактора 252 у.о.);
- X_4 — інструменти тестування (вага фактора 146 у.о.);
- X_2 — тестова документація (вага фактора 98 у.о.);

Для зручності опрацювання подамо їх множиною $T = \{T_1, T_2, T_3, T_4\}$: $T_1 = 361$ у.о.); $T_2 = 252$ у.о.); $T_3 = 146$ у.о.); $T_4 = 146$ у.о.). Враховуючи формулу (2), матимемо такі параметри для виконання обчислень: $m = 4$; $u_i(y_j) = u_{ij}$ — корисність j -ї альтернативи ($j = 1, 2, 3$) за i -м фактором ($i = 1, \dots, 4$). Остаточо вираз для отримання значень функцій корисності альтернативних варіантів матиме такий вигляд:

$$U_j = \sum_{i=1}^4 w_i u_{ij}, \quad j = 1, 2, 3, \tag{3}$$

де U_j — багатофакторна оцінка корисності j -ї альтернативи.

Оскільки вибрано три альтернативні варіанти, задамо таблицею можливі комбінації трудомісткості факторів у формуванні якості тестування в кожній з альтернатив [1].

Таблиця 1

Комбінації значень часток факторів в альтернативах (у відсотках)

10 – 10 – 80	20 – 10 – 70	30 – 10 – 60	40 – 10 – 50
10 – 20 – 70	20 – 20 – 60	30 – 20 – 50	40 – 20 – 40
10 – 30 – 60	20 – 30 – 50	30 – 30 – 40	40 – 30 – 30
10 – 40 – 50	20 – 40 – 40	30 – 40 – 30	40 – 40 – 20
10 – 50 – 40	20 – 50 – 30	30 – 50 – 20	40 – 50 – 10
10 – 60 – 30	20 – 60 – 20	30 – 60 – 10	50 – 10 – 40
10 – 70 – 20	20 – 70 – 10	60 – 10 – 30	50 – 20 – 30
10 – 80 – 10	70 – 10 – 20	60 – 20 – 20	50 – 30 – 20
80 – 10 – 10	70 – 20 – 10	60 – 30 – 10	50 – 40 – 10

Таблиця 2

Остаточо відобразимо у табл. 2 фактори множини Парето, вибрані для розрахунку та оцінювання функцій корисності альтернативних варіантів за формально вираженими мірами важливості чинників в альтернативах. Умовні позначення альтернатив — АВ1, АВ2, АВ3.

Оцінювання факторів в альтернативних варіантах

Фактори	Ваги факторів	Оцінювання факторів в альтернативних варіантах		
		АВ1	АВ2	АВ3
Фактор T_1	w_{s1}	p_{11}	p_{12}	p_{13}
Фактор T_2	w_{s2}	p_{21}	p_{22}	p_{23}
Фактор T_3	w_{s3}	p_{31}	p_{32}	p_{33}
Фактор T_4	w_{s4}	p_{41}	p_{42}	p_{43}

У табл. 2 використано такі позначення: w_{si} — стартова (початкова) вага i -го фактора; p_{ij} — міра оцінювання у відсотках важливості i -го фактора стосовно j -ї альтернативи. Додаткова умова

$$\sum_{i=1}^4 p_{ij} = 100, \quad j = 1, 2, 3. \tag{4}$$

За даними другого стовпця табл. 2 будемо матрицю попарних порівнянь (табл. 3), після опрацювання якої отримуємо уточнені нормалізовані вагові значення w_i , що використовуватимуться для розрахунку за формулою (3) функцій корисності факторів.

Таблиця 3

Матриця попарних порівнянь початкових ваг факторів

	T_1	T_2	T_3	T_4
T_1	1	w_{s1}/w_{s2}	w_{s1}/w_{s3}	w_{s1}/w_{s4}
T_2	w_{s2}/w_{s1}	1	w_{s2}/w_{s3}	w_{s2}/w_{s4}
T_3	w_{s3}/w_{s1}	w_{s3}/w_{s2}	1	w_{s3}/w_{s4}
T_4	w_{s4}/w_{s1}	w_{s4}/w_{s2}	w_{s4}/w_{s3}	1

Далі обчислюємо значення функцій корисності j -ї альтернативи за i -м фактором на основі матриць попарних порівнянь, що відображають переваги альтернатив відносно кожного з факторів.

Елементи матриці для фактора T_1 формуються за таким алгоритмом:

T_1	AB1	AB2	AB3
AB1	1	p_{11}/p_{12}	p_{11}/p_{13}
AB2	p_{12}/p_{11}	1	p_{12}/p_{13}
AB3	p_{13}/p_{11}	p_{13}/p_{12}	1

У результаті опрацювання матриці отримуємо значення функцій корисності альтернатив за фактором T_1 , тобто: u_{11}, u_{12}, u_{13} . Аналогічно для решти факторів отримуємо: T_2 — u_{21}, u_{22}, u_{23} ; T_3 — u_{31}, u_{32}, u_{33} ; T_4 — u_{41}, u_{42}, u_{43} . На завершальному етапі отримуємо значення функцій корисності об'єднаних часткових цільових функціоналів — підсумкових характеристик альтернативних варіантів процесу тестування програмного забезпечення

$$\begin{aligned} U_1 &= w_1 \cdot u_{11} + w_2 \cdot u_{21} + w_3 \cdot u_{31} + w_4 \cdot u_{41}; \\ U_2 &= w_1 \cdot u_{12} + w_2 \cdot u_{22} + w_3 \cdot u_{32} + w_4 \cdot u_{42}; \\ U_3 &= w_1 \cdot u_{13} + w_2 \cdot u_{23} + w_3 \cdot u_{33} + w_4 \cdot u_{43}. \end{aligned} \quad (5)$$

Максимальне значення функцій корисності (5) визначає серед альтернативних оптимальний варіант виконання процесу тестування.

Експериментальне виконання алгоритму реалізації методу починаємо з формування аналогу табл. 2 щодо оцінювання альтернатив, заповнивши її реальними факторами, стартовими значеннями їхніх пріоритетів та комбінаціями величин оцінювання ролі фактора у проектуванні певного альтернативного варіанта, заданих у відсотках.

Таблиця 4

Оцінювання альтернатив за вибраними факторами

Назви факторів	Ваги факторів (початкові)	Оцінювання альтернатив за факторами		
		AB1	AB2	AB3
Кваліфікація тестувальників (T_1)	361 (w_{s3})	20	70	10
Планування тестування (T_2)	252 (w_{s1})	40	40	20
Інструменти тестування (T_3)	146 (w_{s4})	50	20	30
Тестова документація (T_4)	98 (w_{s2})	10	50	40

Для факторів табл. 4, що утворюють автономну множину, знайдемо нормалізовані вагові значення, необхідні для розрахунку функцій корисності. Матрицю попарних порівнянь будемо стосовно даних другого стовпця.

Таблиця 5

Матриця попарних порівнянь факторів

	T_1	T_2	T_3	T_4
T_1	1	3	5	7
T_2	1/3	1	3	5
T_3	1/5	1/3	1	2
T_4	1/7	1/5	1/2	1

У результаті опрацювання матриці за програмою [19] отримуємо нові значення ваг факторів

$$w_1 = 0,56; w_2 = 0,26; w_3 = 0,11; w_4 = 0,07,$$

коли значення критеріїв такі: $\lambda_{\max} = 4,01$; $IU = 0,02$; $WU = 0,02$. Отримані контрольні величини свідчать про достовірність результатів.

Згідно з виразом (5) подальший крок полягатиме у розрахунку значень функцій корисності факторів u_{ij} . Будуємо матриці попарних порівнянь, які відображають переваги альтернатив щодо кожного з факторів, задані у відсотках у третьому, четвертому і п'ятому стовпцях табл. 4. Для фактора T_1 (кваліфікація тестувальників) формуємо таку матрицю:

T_1	AB1	AB2	AB3
AB1	1	1/4	2
AB2	4	1	6
AB3	1/2	1/6	1

Після опрацювання матриці отримаємо такі результати: $\lambda_{\max} = 3,01$; $IU = 0,00$; $WU = 0,00$; Корисність альтернатив за фактором T_1 : $u_{11} = 0,19$; $u_{12} = 0,70$; $u_{13} = 0,11$.

Для решти факторів наведемо лише результати опрацювання матриць.

Фактор T_2 — *планування тестування*:

$\lambda_{\max} = 3,02$; $IU = 0,01$; $WU = 0,01$; $u_{21} = 0,44$; $u_{22} = 0,39$; $u_{23} = 0,17$.

Фактор T_3 — *інструменти тестування*:

$\lambda_{\max} = 3,01$; $IU = 0,00$; $WU = 0,01$; $u_{31} = 0,54$; $u_{32} = 0,16$; $u_{33} = 0,30$.

Фактор T_4 — *тестова документація*:

$\lambda_{\max} = 3,02$; $IU = 0,01$; $WU = 0,02$; $u_{41} = 0,12$; $u_{42} = 0,56$; $u_{43} = 0,32$.

Результати опрацювання матриць попарних порівнянь відповідають умовам прийнятих критеріїв, тобто максимальним власним значенням векторів пріоритетів λ_{\max} , величинам індексів узгодженості IU та відношень узгодженості WU . Підставивши розраховані нормалізовані ваги факторів та значення функцій корисності у вираз (5), отримаємо остаточні числові характеристики об'єднаних функціоналів, що визначають пріоритетність альтернативних варіантів процесу тестування програмного забезпечення.

$$U_1 = 0,056 \cdot 0,19 + 0,26 \cdot 0,44 + 0,11 \cdot 0,54 + 0,07 \cdot 0,12;$$

$$U_2 = 0,056 \cdot 0,70 + 0,26 \cdot 0,39 + 0,11 \cdot 0,16 + 0,07 \cdot 0,56; \quad (6)$$

$$U_3 = 0,056 \cdot 0,11 + 0,26 \cdot 0,17 + 0,11 \cdot 0,30 + 0,07 \cdot 0,32.$$

Після обчислень маємо: $U_1 = 0,192$; $U_2 = 0,550$; $U_3 = 0,161$.

Максимальне значення отримав функціонал U_2 , що згідно з виразом (1) обумовлює оптимальність варіанта АВ2, який априорі за заданих часток важливості факторів в альтернативах повинен забезпечити належний рівень процесу тестування ПЗ. Ключову роль у цьому рішенні відіграли, як впливає з табл. 4, кваліфікація тестувальників і тестова документація.

Остаточна можна стверджувати, що поставлене в публікації завдання вирішено.

Висновки

Здійснено аналіз досліджень і публікацій, що мають відношення до тематики статті. Реалізовано формування альтернативного підходу до визначення оптимального варіанта процесу тестування ПЗ, оснований на методі лінійного згортання критеріїв, концепції Парето, вагових перевагах факторів тестування програм та їхній дольовій участі в альтернативах. Виконано практичну реалізацію застосування методу, початковим етапом якого є виокремлення домінуючих факторів процесу тестування, що утворюють множини Парето. Запроектовано альтернативні варіанти виконання процесу тестування ПЗ з використанням функцій корисності та часток ефективності факторів в альтернативах. Побудовано та опрацьовано матриці попарних порівнянь, що обумовило отримання уточнених вагових пріоритетів чинників множини Парето та значень функцій корисності факторів з огляду на їхню частку у формуванні альтернатив. Достовірність проведених розрахунків підтверджено відповідними критеріями — максимальним власним значенням головного вектора матриці попарних порівнянь і величинами індексів узгодженості та відношення узгодженості. Визначено за критерієм максимального значення об'єднаних цільових функціоналів серед альтернативних оптимальний варіант процесу тестування програмного забезпечення, що забезпечує належну якість імітаційного моделювання.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

- [1] Б. В. Дурняк, І. В. Піх, і В. М. Сеньківський, *Теоретичні основи інформаційної концепції формування та оцінювання якості видавничо-поліграфічних процесів*, моногр. Львів, Україна: Українська академія друкарства, 2022, 356 с. [Електронний ресурс]. Режим доступу: <https://biblio.uad.lviv.ua>.
- [2] М.З. Згуровський, і Н. Д. Панкратова, *Основи системного аналізу*. Київ, Україна: Видавнича група ВНУ, 2007, 544 с. [Електронний ресурс]. Режим доступу: <https://iszzi.kpi.ua>.
- [3] M. Y. Bartish, and I. M. Dudzianyi, *Operations research. Part 3. Decision making and game theory*. Lviv: Publishing center of Ivan Franko National University, 2009. 278 p. [Electronic resource]. Available: <https://ami.lnu.edu.ua>.
- [4] І. В. Піх, В. М. Сеньківський, і Р. Р. Андріїв, «Проектування та розрахунок альтернативних варіантів реалізації технологічних процесів», *Технологія і техніка друкарства*, № 2 (48), с. 55-62, 2015.

- [5] В. М. Сеньківський, І. В. Піх, Ю.Ф. Петяк, і І. В. Калиній, «Теоретичні основи забезпечення якості видавничо-поліграфічних процесів (Частина 3. Проектування альтернативних варіантів),» *Наукові записки*, № 2 (53), с. 47-56, 2016. [Електронний ресурс]. Режим доступу: <http://nz.uad.lviv.ua/uk/archive/2-53/>.
- [6] І. В. Піх, і А. В. Кудряшова, «Багатофакторний вибір альтернативних варіантів композиційного оформлення видання на основі лінійного згортання критеріїв,» *Наукові записки*, № 2 (55), с. 41-46, 2017. [Електронний ресурс]. Режим доступу: <http://nz.uad.lviv.ua/uk/archive/2-55/>.
- [7] E. Hendrickson, *Explore It!: Reduce Risk and Increase Confidence with Exploratory Testing*. ISBN-10: 1937785025. Pragmatic Bookshelf, 2013, 186 p.
- [8] J. Davenport, J. Olszewska, J. Röbler, A. Leon Smith, and J. Wright. *Artificial Intelligence and Software Testing: Building systems you can trust*, 2022, 253 p.
- [9] Jagdeep Jain, *Learn API Testing: Norms, Practices, and Guidelines for Building Effective Test Automation*. Apress, 2022. 245 p. [Electronic resource]. Available: <https://doi.org/10.1007/978-1-4842-8142-0>.
- [10] Janet Gregory and Lisa Crispin, *More Agile Testing: Learning Journeys for the Whole Team*. Addison-Wesley Professional, 2014, 544 p.
- [11] Nicole Forsgren, Jez Humble, and Gene Kim, *Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations*, 2018, 288 p.
- [12] Glenford J. Myers, and Tom Badgett, *The Art of Software Testing*, 2015, 256 p.
- [13] Eran Kinsbruner, *Continuous Testing for Agile and DevOps: Better Software Through Faster Feedback*. Create Space Independent Publishing Platform, 2018, 366 p.
- [14] Dave Todaro, *The Epic Guide to Agile: More Business Value on a Predictable Schedule with Scrum*, 2019, 520 p.
- [15] О. Є. Білас, *Якість програмного забезпечення та тестування*, навч. посіб. Львів, Україна: Вид-во Львівської політехніки, 2011, 216 с. [Електронний ресурс]. Режим доступу: <https://vlp.com.ua/node/7698>.
- [16] А. С. Авраменко, В. С. Авраменко, і Г. В. Косенюк, *Тестування програмного забезпечення*, навч. посіб. Черкаси, Україна: ЧНУ ім. Б. Хмельницького, 2017, 284 с. [Електронний ресурс]. Режим доступу: <http://lib.istu.edu.ua/index.php?p=22&par=22>.
- [17] С. Я. Крепич, і І. Я. Співак, *Якість програмного забезпечення та тестування: базовий курс*, навч. посіб. Тернопіль, Україна: ФОП Паляниця В. А., 2020, 478 с. [Електронний ресурс]. Режим доступу: <http://dspace.wunu.edu.ua/handle/316497/39773>.
- [18] І. В. Піх, В. М. Сеньківський, Н. Є. Сеньківська, О. З. Білик, і Р. Р. Андріїв, «Модель факторів якості тестування програмного забезпечення. Комп'ютерні технології друкарства,» *Зб. наук. праць*, № 1 (51), с. 62-80, 2024. [Електронний ресурс]. Режим доступу: <https://doi.org/10.32403/2411-9210-2024-1-51-62-80>.
- [19] І. В. Гілета, В. М. Сеньківський, і О. В. Мельников, Свідцтво про реєстрацію авторського права на твір № 41832. Україна. «Імітаційне моделювання в системному аналізі методом бінарних порівнянь.» [Комп'ютерна програма]. Зареєстровано 17.01.2012.

Рекомендована кафедрою програмного забезпечення ВНТУ

Стаття надійшла до редакції 29.05.2024

Піх Ірина Всеволодівна — д-р техн. наук, професор, професор кафедри інформаційних систем та технологій у видавничо-поліграфічних процесах Української академії друкарства, професор кафедри автоматизованих систем управління Національного університету «Львівська політехніка», e-mail: pikhirena@gmail.com ;

Білик Олексій Зіновійович — аспірант кафедри інформаційних систем та технологій у видавничо-поліграфічних процесах;

Сеньківський Назарій Юрійович — аспірант кафедри інформаційних систем та технологій у видавничо-поліграфічних процесах.

Українська академія друкарства, Львів ;

Андріїв Роман Ростиславович — асистент кафедри управління інформаційною безпекою.

Львівський державний університет безпеки життєдіяльності, Львів;

Браташ Соломія Петрівна — аспірантка кафедри автоматизованих систем управління.

Національний університет «Львівська політехніка»

I. V. Pikh^{1,2}
O. Z. Bilyk¹
N. Yu. Senkivskiy¹
R. R. Andriiv³
S. P. Bratash²

Alternative Process Options Software Testing

¹Ukrainian Academy of Printing, Lviv;

²Lviv Polytechnic National University;

³Lviv State University of Life Safety

It is taken into account that the solution to the problem of software quality provision depends significantly on one of the final stages related to the level of testing of software products, which necessitates the development of alternative approaches to the testing process that would effectively take into account various factors influencing its effectiveness, the complexity of production and organizational conditions, and the requirements regarding the quality of the final product. An informational interpretation of the concept of quality formation of this category is proposed using the tools of decision theory to model alternative and determine the optimal variants of the program testing process. The application of an alternative approach to the selection of the software testing process options based on the method of linear convolution of criteria, the Pareto concept, taking into account the weight preferences of factors influencing the quality of software testing and their share in the formation of alternatives is considered. The theoretical fundamentals of the method of linear convolution of criteria and its application to program testing are presented. Practical implementation of the method is investigated, the initial stage of which is the selection of a Pareto set from the general set of characteristics of software testing, which contains the dominant factors of priority impact on the testing process. The use of the Pareto principle ensured the determination of the optimal set of combinations of test cases that best meet the needs of testing with limited resources. Alternative variants of the software testing process are designed using utility functions and efficiency shares of factors in alternative variants. The matrices of pairwise comparisons are constructed and processed to calculate the refined weight priorities of the factors of the Pareto set and the values of the utility functions of the factors, taking into account their share in the formation of alternatives. The optimal variant of the software testing process among the alternatives is calculated and determined according to the criterion of the maximum value of the combined target functions, which ensures the proper quality of simulation modeling.

Keywords: quality factors of software testing, method of linear convolution of criteria, Pareto principle, alternative variant, matrix of pairwise comparisons, utility function, quality functionality, optimal variant.

Pikh Iryna V. — Dr. Sc. (Eng.), Professor, Professor of the Chair of Information Systems and Technologies in Publishing and Printing Processes of Ukrainian Academy of Printing; Professor of the Chair of Automated Control Systems of Lviv Polytechnic National University, e-mail: pikhirena@gmail.com ;

Bilyk Oleksiy Z. — Post-Graduate Student of the Chair of Information Systems and Technologies in Publishing and Printing Processes;

Senkivskiy Nazariy Yu. — Post-Graduate Student of the Chair of Information Systems and Technologies in Publishing and Printing Processes;

Andriiv Roman R. — Assistant of the Chair of Information Security Management;

Bratash Solomiia P. — Post-Graduate Student of the Chair of Automated Control Systems