

УДК 004.738.1:316.772.5

МЕТОДОЛОГІЧНІ ЗАСАДИ ФОРМУВАННЯ ЯКОСТІ  
ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ  
(ЧАСТИНА 1: БАЗОВА МОДЕЛЬ ФАКТОРІВ ЯКОСТІ)

В. М. Сеньківський<sup>1</sup>, І. В. Піх<sup>1,2</sup>, І. В. Калиній<sup>3</sup>,  
Н. Ю. Сеньківський<sup>1</sup>, М. А. Драгоміров<sup>1</sup>

<sup>1</sup>Українська академія друкарства,  
вул. Під Голоском, 19, Львів, 79020, Україна,

<sup>2</sup>Національний університет «Львівська політехніка»,  
вул. Степана Бандери, 12, Львів, 79013, Україна,

<sup>3</sup>Бережанський агротехнічний інститут,  
вул. Академічна, 20, Бережани, 47501, Україна

*Обґрунтовано доцільність використання інформаційної концепції формування якості видавничо-поліграфічних процесів щодо дослідження проблематики якості програмного забезпечення. Виокремлено основні характеристики (фактори) якості програм, зумовлені стандартами в галузі інженерії програмного забезпечення. Відтворено формалізований варіант графічного відображення структурних зв'язків між факторами за допомогою семантичної мережі. Здійснено опис семантичної мережі з використанням мови логіки предикатів, що містить атомарні конструкції та спеціальні логічні висловлювання. З використанням методології моделювання ієрархій визначено рівні переваг факторів на основі побудованої бінарної матриці досяжності, опрацювання якої виконано ітераційними таблицями. За допомогою засобів інфографіки синтезовано базову модель факторів впливу на якість програмного забезпечення.*

**Ключові слова:** інженерія програмного забезпечення, фактори якості програмного забезпечення, семантична мережа, мова логіки предикатів, моделювання ієрархій, матриця досяжності, ітерація, модель факторів якості.

**Постановка проблеми.** Поняття «якість» належить до визначення ступеня відповідності певного продукту (що водночас стосується і процесу, в результаті якого цей продукт виробляється) певним, наперед встановленим правилам, вимогам чи стандартам за їх наявності. Сказане повною мірою стосується програмного забезпечення, стосовно якого якість виступає в ролі певної характеристики, критерію чи показника, ступінь відповідності яких має бути максимально близьким до деякого еталону, що визначає оптимальну якість. Оскільки програмне забезпечення, на відміну від багатьох видів продукції, найчастіше створюється для конкретного замовника, тому вимоги до його якості можуть мати дещо іншу інтерпретацію, адже формуються додатково залежно від потреб майбутнього користувача.

Характеристики, за якими оцінюється якість програмного продукту, регламентуються серією міжнародних стандартів ISO/IEC 25000, що оновлюються з плином часу. Так, стандарт ISO/IEC 9126-1:2001 [1] регламентує зовнішні та внутрішні характеристики якості програмного забезпечення (ПЗ) або так звані метрики, які відображають вимоги до функціонування програмного продукту. Для кількісного встановлення критеріїв якості, що підтверджують відповідність ПЗ заданим вимогам, вводяться зовнішні метрики (наприклад, час виконання окремих компонент), діапазони зміни значень і моделі їх оцінки. Внутрішні атрибути забезпечують досягнення необхідних зовнішніх характеристик якості ПЗ [1]. Вказаний стандарт передбачав шість характеристик якості, суть яких буде розкрита пізніше.

Оновлений стандарт ISO/IEC 25010 [2], що вважається ґрунтовним удосконаленням попереднього, містить термінологію для визначення, вимірювання та оцінювання якості систем і програмних продуктів. Він регламентує на найвищому рівні вісім характеристик якості програмного забезпечення, а саме: функціональна придатність, ефективність роботи, зручність використання, надійність, сумісність, безпека, зручність супроводження, переносимість. На основі вказаних характеристик буде створена модель пріоритетності їх впливу на якість ПЗ.

Квантифікація характеристик якості ПЗ за допомогою метрик визначає рівень програм «апостеріорно» і вимагає ухвалення рішень для усунення недоліків остаточного продукту на етапі тестування програм. Не заперечуючи потребу подібних процедур, вважаємо доцільним доповнити їх методологією апріорного отримання якості результуючої продукції через розроблення засобів (моделей чи інформаційних технологій), що зумовлюють врахування чинників формування та прогностичного оцінювання якості процесів на етапі їх реалізації.

Розкриття початкових етапів пропонованого підходу виконаємо за допомогою засобів системного аналізу, методів теорії ієрархічних систем і теорії моделювання [3–9], на основі яких здійснимо виокремлення факторів досліджуваного процесу та розроблення базової моделі пріоритетного впливу вказаних чинників на якість програмного забезпечення.

**Аналіз останніх досліджень та публікацій.** Пропонована стаття побудована на основі аналізу сучасних публікацій у сфері програмування, серед яких особливо цінною є ретроспективний огляд 50-річної історії становлення та перспектив розвитку інженерії програмного забезпечення, названого визначальним напрямом сучасної інформатики [10]. Від часу введення в обіг терміна «інженерія програмного забезпечення» (software engineering) [11] вказана проблематика постійно фігурує в полі зору теоретиків та професійних програмістів [10, 12–20]. Унаслідок стало можливим становлення індустріального виробництва ПЗ, поява структурного програмування, модуляризація програм (наявність автономних модулів та спеціальних підпрограм) та пов'язане з цим розроблення блок-орієнтованих та об'єктно-орієнтованих мов програмування. Системотворчими в цьому напрямі стали результати, орієнтовані на дослідження життєвого циклу програмного забезпечення, в основі якого супроводження програм, створення умов для їх повторного використання і за потреби утилізації програмного продукту та побудови на його основі нового ПЗ [10].

Характерною особливістю дослідження, пов'язаного з продукуванням програмного забезпечення, могло б стати (на нашу думку) використання моделей і методів, відомих з деяких публікацій, що стосуються підготовки та випуску друкованої продукції [21–26]. Водночас, незважаючи на наявність вищезазначених праць, поза увагою авторів залишилася (в деякій мірі обґрунтовано) практично нерозкрита проблема, орієнтована на дослідження системотехнічних аспектів розроблення програмного забезпечення, яке суттєво впливає на повноту та якість реалізації видавничо-поліграфічних процесів. При позитивному результаті застосування інформаційного підходу щодо дослідження проблематики інженерії програмного забезпечення додатково буде підтверджено універсальність сформованих у праці [26] теоретичних основ інформаційної концепції формування якості видавничо-поліграфічних технологій та її ефективність у вивченні довільних процесів, реалізація яких зумовлена наявністю певної множин взаємопов'язаних чинників.

**Мета статті.** Застосувати інформаційну концепцію для дослідження процесу формування якості програмного забезпечення, початковим етапом якого буде формалізоване відтворення зв'язків між факторами (характеристиками якості ПЗ) за допомогою семантичної мережі, що стане підставою для їх ранжування та синтезування попередньої (первинної) моделі впливу факторів на якість програмного продукту. Виконати опис семантичної мережі з використанням конструкцій мови предикатів, що уможливить (за потреби) уточнене ранжування факторів через врахування додаткового впливу зв'язуючих термів мережі на відношення між ними. Засобами теорії моделювання ієрархій встановити рівні переваг факторів та синтезувати базову (первинну) модель їх впливу на якість програмного забезпечення.

**Виклад основного матеріалу дослідження.** Прийняті міжнародні стандарти, що стосуються програмного забезпечення, свідчать, що з розвитком індустрії програмування виникає необхідність змінювати та удосконалювати підходи і критерії оцінювання якості вказаної продукції. Для підтвердження наведеної тези розглянемо два стандарти в цій галузі. Згідно із стандартом ISO 9126:2001 розрізняють такі характеристики програмного забезпечення [1]:

- **функціональність**, тобто здатність за певних умов вирішувати задекларовані користувачем завдання;
- **надійність**, що зумовлює закладену у вихідних умовах працездатність;
- **зручність використання** та освоєння, практичність, привабливість для користувачів;
- **продуктивність** або **ефективність**, що визначає здатність забезпечувати при заданих умовах та наявних ресурсах необхідний результат;
- **зручність супроводження** за умови передбачених проектом видів діяльності програм;
- **переносимість**, що забезпечується рівнем тестування та зумовлює здатність ПЗ зберігати працездатність під час перенесення з одного оточення в інше.

Розвиток, уточнення і доповнення наведених вище характеристик знаходимо у стандарті ISO/IEC 25010, затвердженому в березні 2011 року [19]. Він містить вісім характеристик якості програмного забезпечення, коротке пояснення яких наведемо нижче.

**Функціональна придатність** (Functional Suitability), суть якої розкривається у повноті покриття задач і функцій користувача, їх коректності та доцільності, значною мірою забезпечується ступенем тестування.

**Ефективність роботи** (Performance efficiency), що визначається часовими показниками опрацювання даних, пропускнуою здатністю та відповідністю вимогам до граничних значень параметрів, тестуванням.

**Сумісність** (Compatibility) як здатність програмної системи сумісно функціонувати з іншою системою та обмінюватися з нею інформацією.

**Зручність використання** (Usability), яку обґрунтовує придатність ПЗ для потреб користувача, керованість з огляду на управління і контроль, системний захист користувача від помилок.

**Надійність** (Reliability) як ступінь працездатності і доступності програмного продукту, здатності до безвідмовної роботи та відновлення даних і належного стану у випадку збоїв.

**Безпека** (Security), основними показниками якої слугують невідомість і цілісність проти несанкціонованого втручання, конфіденційність як обмеження доступу, автентичність, порівняно з початковим проектом.

**Зручність супроводження** (Maintainability) — визначається модульністю, що забезпечує модифікованість програмного продукту, простоту оцінювання впливу між частинами під час виявлення загальних недоліків, ефективність тестування, можливість багаторазового використання.

**Портативність, переносимість** (Portability), що зумовлюється адаптивністю до інших апаратних засобів та операційних систем, простотою встановлення програмного продукту в заданому середовищі, здатністю замінити інший програмний продукт для досягнення аналогічних цілей, рівнем тестування.

Наведені характеристики вважатимемо факторами впливу на якість програмного забезпечення, яка зумовлюється відповідним рівнем процесу його розроблення. Наступне уточнення стосується характеристики «портативність». Досвід створення, впровадження та супроводження програмних комплексів свідчить, що від ефективності та повноти тестування як процесу перевірки відповідності вимог технічного завдання і реалізованої функціональності програмних компонент залежить здатність ПЗ зберігати працездатність при зміні умов функціонування. З огляду на сказане вживатимемо термін «рівень тестування».

Перейдемо до формалізованого опису вихідної бази даних, в основі якої множина факторів, що зумовлює якість програмного забезпечення:

$$X = \left\{ \begin{array}{l} x_1 - \text{функціональна придатність (ФП)}; \\ x_2 - \text{ефективність (ЕФ)}; \\ x_3 - \text{зручність використання (ЗВ)}; \\ x_4 - \text{надійність (НД)}; \\ x_5 - \text{сумісність (СМ)}; \\ x_6 - \text{захищеність (ЗХ)}; \\ x_7 - \text{зручність супроводження (ЗС)}; \\ x_8 - \text{рівень тестування (РТ)} \end{array} \right\} \quad (1)$$

Надалі вважатимемо важливими такі положення [20].

*Твердження 1.* Наявність зв'язків між факторами породжує передумову для побудови їх формалізованого відображення у вигляді семантичної мережі.

*Твердження 2.* Початкові ранги факторів встановлюються через врахування та аналіз типів і кількостей зв'язків між ними у вихідній графічній моделі.

*Твердження 3.* Синтезована багаторівнева модель відображає лише переваги між факторами за умови їх порівняння в межах вихідного графа.

Враховуючи множину (1) та результати експертного опитування, запроєктуємо семантичну мережу зв'язків між факторами (рис. 1).

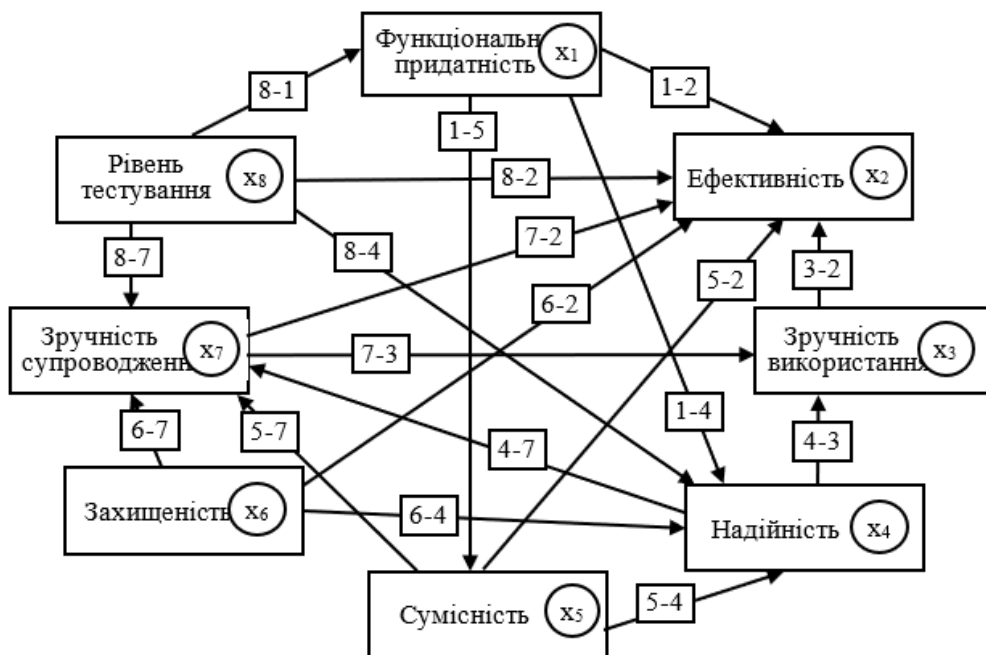


Рис. 1. Семантична мережа зв'язків між факторами якості програмного забезпечення

Вершини мережі відтворюють виокремлені лінгвістичні фактори-аргументи множини  $X = \{x_1, x_2, \dots, x_8\}$  дуги — пари вершин  $(x_i, x_j)$ , для котрих визначено зв'язок  $(i, j = 1 \div 8; i \neq j)$ . У позначеннях на дугах перша цифра вказує номер джерела впливу, друга — залежний від нього фактор.

Для опису відношень між факторами у семантичній мережі використаємо елементи мови предикатів, що містять прості або атомарні предикати та логічні зв'язування:  $\wedge$  — логічне «і»;  $\vee$  — логічне «або»;  $\leftarrow$  — логічне «якщо»;  $\forall$  — квантор спільності (для всіх);  $\exists$  — квантор існування (фігурує принаймні одне) [5]. Внаслідок отримаємо таку формально-лінгвістичну початкову базу даних:

$(\forall x_i) [\exists (x_1, \text{функціональна придатність}) \leftarrow \text{впливає} (x_1, x_2) \wedge \text{обумовлює} (x_1, x_4) \wedge \text{визначає} (x_1, x_5) \wedge \text{формується} (x_1, x_8)]$ ;

$(\forall x_i) [\exists (x_2, \text{ефективність}) \leftarrow \text{отримує } (x_2, x_1) \wedge \text{обумовлюється } (x_2, x_3) \wedge \text{отримує } (x_2, x_5) \wedge \text{визначається } (x_2, x_6) \wedge \text{ґрунтується } (x_2, x_7) \wedge \text{обумовлюється } (x_2, x_8)];$

$(\forall x_i) [\exists (x_3, \text{зручність використання}) \leftarrow \text{обумовлює } (x_3, x_2) \wedge \text{отримує } (x_3, x_4) \wedge \text{передбачається } (x_3, x_7)];$

$(\forall x_i) [\exists (x_4, \text{надійність}) \leftarrow \text{впливає } (x_4, x_3) \wedge \text{обумовлюється } (x_4, x_1) \wedge \text{формується } (x_4, x_5) \wedge \text{обумовлюється } (x_4, x_6) \wedge \text{обумовлює } (x_4, x_7) \wedge \text{ґрунтується } (x_4, x_8)];$

$(\forall x_i) [\exists (x_5, \text{сумісність}) \leftarrow \text{визначається } (x_5, x_1) \wedge \text{впливає } (x_5, x_2) \wedge \text{передбачає } (x_5, x_7)];$

$(\forall x_i) [\exists (x_6, \text{захищеність}) \leftarrow \text{визначає } (x_6, x_2) \wedge \text{обумовлює } (x_6, x_4) \wedge \text{стає основою } (x_6, x_7)];$

$(\forall x_i) [\exists (x_7, \text{зручність супроводження}) \leftarrow \text{стає основою } (x_7, x_2) \wedge \text{впливає } (x_7, x_3) \wedge \text{обумовлюється } (x_7, x_4) \wedge \text{передбачається } (x_7, x_5) \wedge \text{ґрунтується } (x_7, x_6) \wedge \text{визначається } (x_7, x_8)];$

$(\forall x_i) [\exists (x_8, \text{рівень тестування}) \leftarrow \text{формує } (x_8, x_1) \wedge \text{обумовлює } (x_8, x_2) \wedge \text{стає основою } (x_8, x_4) \wedge \text{визначає } (x_8, x_7)].$

Використані в описі семантичної мережі атомарні предикати за наявності спеціальних коефіцієнтів їх вагомості можуть стати основою посилення (для впливів) чи послаблення (для залежностей) попарних зв'язків між факторами при встановленні уточнених рангів пріоритетного впливу на процес.

На основі семантичної мережі визначимо рівні важливості факторів з огляду впливу на якість програмного забезпечення. Використаємо при цьому метод математичного моделювання ієрархій [6, 7], в основі якого матриця досяжності — математичний аналог залежностей між факторами у семантичній мережі. Правило для побудови матриці полягає у такому логічному твердженні:

$$b_{ij} = \begin{cases} 1, \text{ якщо з вершини } i \text{ можна потрапити у вершину } j; \\ 0 \text{ в іншому випадку.} \end{cases} \quad (2)$$

### Матриця досяжності факторів якості ПЗ

	X1 ФП	X2 ЕФ	X3 ЗВ	X4 НД	X5 СМ	X6 ЗХ	X7 ЗС	X8 РТ
X1, ФП	1	1	0	1	1	0	0	0
X2, ЕФ	0	1	0	0	0	0	0	0
X3, ЗВ	0	1	1	0	1	0	0	0
X4, НД	0	0	1	1	0	0	1	0
X5, СМ	0	1	0	1	1	0	1	0
X6, ЗХ	0	1	0	1	0	1	1	0
X7, ЗС	0	1	1	0	0	0	1	0
X8, РТ	1	1	0	1	0	0	1	1

Практично вершина  $x_j$  ( $j = 1, 2, \dots, 8$ ) семантичної мережі (рис. 1) вважається досяжною відносно вершини  $x_i$  ( $i = 1, 2, \dots, 8$ ), якщо з останньої можна потрапити в  $x_j$ . Такі досяжності вважатимемо прямими впливами. Досяжні вершини позначимо підмножиною  $D(w_i)$ . Відповідно вершину  $x_i$  вважатимемо попередницею вершини  $x_j$ , якщо вона досягається з неї. Сукупність вершин попередниць становитиме підмножину  $P(w_i)$ . Перетин вказаних підмножин, тобто

$$Z(w_i) = D(w_i) \cap P(w_i) \tag{3}$$

визначить відповідний рівень фактора. Додатково перевіряється умова

$$P(w_i) = Z(w_i). \tag{4}$$

Для визначення рівнів важливості факторів будемо першу ітераційну таблицю за таким правилом. У другий стовпець таблиці заносимо підмножину  $D(w_i)$  — номери досяжних вершин, або номери одиничних елементів відповідних рядків матриці досяжності; третій стовпець визначає підмножину вершин попередниць  $P(w_i)$  — номери одиничних елементів стовпців цієї матриці. У цьому випадку залежність (4) означатиме виконання умови рівності номерів факторів, заданих у другому і третьому стовпцях таблиці, у результаті чого утворюється певний рівень факторів.

Таблиця 1

$i$	$D(w_i)$	$P(w_i)$	$D(w_i) \cap P(w_i)$
1	1,2,4,5	1,8	1
2	2	1,2,3,5,6,7,8	2
3	2,3	3,4,7	3
4	3,4,7	1,4,5,6,8	4
5	2,4,5,7	1,5	5
6	2,4,6,7	6	6
7	2,3,7	4,5,6,7,8	7
8	1,2,4,7,8	8	8

Як видно з табл. 1, збігання номерів зафіксовано для факторів 6 — захищеність та 8 — рівень тестування ПЗ. Ці фактори згідно з виразом (3) вважатимемо найвищими за пріоритетністю впливу на якість програмного забезпечення. Наступний крок — виключаємо з табл. 1 шостий і восьмий рядки, а в третьому стовпці викреслюємо цифри 6 і 8. Отримана таким чином табл. 2 придатна для обчислення наступної ітерації.

Таблиця 2

$i$	$D(w_i)$	$P(w_i)$	$D(w_i) \cap P(w_i)$
1	1,2,4,5	1	1
2	2	1,2,3,5, 7	2
3	2,3	3,4,7	3
4	3,4,7	1,4,5	4
5	2,4,5,7	1,5	5
7	2,3,7	4,5, 7	7

У табл. 2 визначено фактор чергового рівня впливу, а саме: 1 — функціональну придатність. Вилучаємо з табл. 2 рядок з вказаним номером та викреслюємо у третьому стовпці цифру 1, що зумовить табл. 3.

Таблиця 3

$i$	$D(w_i)$	$P(w_i)$	$D(w_i) \cap P(w_i)$
2	2	2,3,5, 7	2
3	2,3	3,4,7	3
4	3,4,7	4,5	4
5	2,4,5,7	5	5
7	2,3,7	4,5, 7	7

Згідно з табл. 3 черговий рівень за фактором 5 — сумісність ПЗ.

Повторення наведених вище кроків призведе до такого порядку розміщення факторів: 4 — надійність програмного забезпечення; 7 — зручність супроводження; 3 — зручність використання; 2 — ефективність.

За допомогою засобів інфографіки візуалізовано багаторівневу модель пріоритетного впливу факторів на якість програмного забезпечення.



Рис. 2. Багаторівнева модель факторів впливу на якість програмного забезпечення



**Висновки.** Виконано огляд літературних джерел, дотичних до запропонованої тематики. Здійснено виокремлення та формалізований опис множини факторів впливу на якість програмного забезпечення. Відтворено графічне відображення зв'язків між аналізованими чинниками засобом семантичної мережі. Здійснено опис семантичної мережі з використанням конструкцій мови предикатів. Встановлено рівні переваг факторів на основі побудованої квадратної бінарної матриці досяжності, опрацювання якої виконано за допомогою ітераційних таблиць. Синтезовано попередню базову графічну модель пріоритетного впливу факторів на якість процесу проєктування програмного забезпечення.

Результат дослідження стане основою для розрахунку вагових значень факторів та отримання оптимального варіанта вказаної моделі.

### СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. ISO/IEC 9126-1:2001. Software engineering. Product quality. Part 1: Quality model. [Introduced 15.06.2001; revised by ISO/IEC 25010:2011]. Geneva (Switzerland), 2001. 32 p. (International standard).
2. 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).
3. Згуровський М. З., Панкратова Н. Д. Основи системного аналізу. Київ : Видавнича група ВНУ, 2007. 544 с.
4. Баргіш М. Я., Дудзяний І. М. Дослідження операцій. Частина 3. Ухвалення рішень і теорія ігор. Львів : Видавничий центр ЛНУ імені Івана Франка, 2009. 278 с.
5. Матвеев В. Семантичні мережі. URL: [matveev.kiev/exprt/t5.pdf](http://matveev.kiev/exprt/t5.pdf).
6. Матвеев В. Представлення знань з використанням логіки предикатів. URL: [matveev.kiev/exprt/t4.pdf](http://matveev.kiev/exprt/t4.pdf).
7. Лямець В. І., Тевяшев В. І. Системний аналіз. Вступний курс. 2-ге вид., перероб. та допов. Харків : ХНУРЕ, 2004. 448 с.
8. Сявавко М. С. Інформаційна система «Нечіткий експерт». Львів : Видавн. центр ЛНУ імені Івана Франка, 2007. 320 с.
9. Ротштейн О. П., Ларюшкін Є. П., Мітюшкін Ю. І. Soft Computing в біотехнології: багатфакторний аналіз і діагностика : монографія. Вінниця : УНІВЕРСУМ-Вінниця, 2008. 144 с.
10. Сидоров М. О. 50 років інженерії програмного забезпечення. *Проблеми програмування*. 2018. № 4. С. 30–44.
11. Report on a conference sponsored by the NATO science committee, Garmisch, Germany, 7th to 11th October 1968, Editors: Peter Naur and Brian Randell.
12. Lundberg L., Mattson M., Wohlin C. Software quality attributes and trade-offs. Blekinge Institute of Technology, 2005.
13. Білас О. Є. Якість програмного забезпечення та тестування : навч. посіб. Львів : Видавництво Львівської політехніки, 2011. 216 с.
14. Лавріщева К. М. Програмна інженерія. Київ, 2008. 319 с.
15. Табунщик Г. В., Кудерметов Р. К., Брагіна Т. І. Інженерія якості програмного забезпечення : навч. посіб. Запоріжжя : ЗНТУ, 2013. 180 с.

16. Розвиток української IT-індустрії: аналітичний звіт. It Ukraine, 2018.
17. Процеси та системи підтримки якості програмних систем. URL: <https://dspace.uzhnu.edu.ua/jspui/bitstream/lib/16456/1/>.
18. Моделі життєвого циклу, принципи і методології розробки програмного забезпечення. URL: <https://evergreens.com.ua/ua/articles/software-development-metodologies.html>.
19. Стадії циклу розробки ПЗ. URL: <https://mango-test.qalight.com.ua/ua/baza-znaniy/stadiyi-tsiklu-rozrobki-pz/>.
20. Тестування та якість програмного забезпечення. URL: [https://kpi-fict-ip32.github.io/Blog/s07/software\\_quality.html](https://kpi-fict-ip32.github.io/Blog/s07/software_quality.html).
21. Durnyak B., Pikh I., Senkivskyy V. Method of determining the weight of predicates of semantic networks in publishing processes. *Przegląd papierniczy (Polish Paper Review)*. Warszawa, 2018. V (1). Pp. 57–60. (Scopus).
22. Осінчук О. І., Піх І. В., Сеньківський В. М. Модель забезпечення якості тематичного планування книжкових видань. *Science and Education a New Dimension (Natural and Technical Sciences)*. Budapest, 2017. V (16). Issue 148. Pp. 38–40.
23. Senkivskyy V., Pikh I., Babichev S., Kudriashova A., Senkivska N. Modeling of Alternatives and Defining the Best Options for Websites Design. 2-nd International Seminar on Intelligent Information Technologies and Information Security Systems (IntelITSIS-2021). CEUR Workshop Proceedings, 2021, 2853. Khmelnytskyu, April 2021. Pp. 259–270.
24. Піх І. В., Дурняк Б. В., Сеньківський В. М., Голубник Т. С. Інформаційні технології формування якості книжкових видань : монографія. Львів : Українська академія друкарства, 2017. 308 с.
25. Сеньківський В. М., Піх І. В., Кудряшова А. В., Литовченко О. В. Теоретичні основи забезпечення якості видавничо-поліграфічних процесів (Частина 2: Синтез моделей пріоритетності дії факторів). *Поліграфія і видавнича справа*. 2016. № 1 (71). С. 20–29.
26. Дурняк Б. В., Піх І. В., Сеньківський В. М. Теоретичні основи інформаційної концепції формування та оцінювання якості видавничо-поліграфічних процесів : монографія. Львів : Українська академія друкарства, 2022. 356 с.

## REFERENCES

1. ISO/IEC 9126-1:2001. Software engineering. Product quality. Part 1: Quality model. [Introduced 15.06.2001; revised by ISO/IEC 25010:2011]. Geneva (Switzerland), 2001. 32 p. (International standard) (in English).
2. 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) (in English).
3. Zghurovskiy, M. Z., & Pankratova, N. D. (2007). *Osnovy systemnoho analizu*. Kyiv : Vydavnycha hrupa VNU (in Ukrainian).
4. Bartish, M. Ya., & Dudzianyi, I. M. (2009). *Doslidzhennia operatsii. Chastyna 3. Ukhvalennia rishen i teoriia ihor*. Lviv : Vydavnychiy tsentr LNU imeni Ivana Franka (in Ukrainian).
5. Matveiev, V. *Semantychni merezhi*. URL: [matveev.kiev/exprt/t5.pdf](http://matveev.kiev/exprt/t5.pdf) (in Ukrainian).
6. Matveiev, V. *Predstavlennia znan z vykorystanniam lohiky predykativ*. Retrieved from [matveev.kiev/exprt/t4.pdf](http://matveev.kiev/exprt/t4.pdf) (in Ukrainian).

7. Liamets, V. I., & Teviashev, V. I. (2004). Systemnyi analiz. Vstupnyi kurs. 2-he vyd., pererob. ta dopov. Kharkiv : KhNURE (in Ukrainian).
8. Siavavko, M. S. (2007). Informatsiina systema «Nechitkyi ekspert». Lviv : Vydavn.tsentr LNU imeni Ivana Franka (in Ukrainian).
9. Rotshtein, O. P., Lariushkin, Ye. P., & Mitushkin, Yu. I. (2008). Soft Computing v biotekhnolohii: bahatofaktornyi analiz i diahnostyka. Vinnytsia : UNIVERSUM-Vinnytsia (in Ukrainian).
10. Sydorov, M. O. (2018). 50 rokiv inzhenerii prohramnoho zabezpechennia: Problemy prohramuvannia, 4, 30–44 (in Ukrainian).
11. Report on a conference sponsored by the NATO science committee, Garmisch, Germany, 7th to 11th October 1968, Editors: Peter Naur and Brian Randell (in English).
12. Lundberg, L., Mattson, M., & Wohlin, C. (2005). Software quality attributes and trade-offs. Blekinge Institute of Technology (in English).
13. Bilas, O. Ye. (2011). Yakist prohramnoho zabezpechennia ta testuvannia. Lviv : Vydavnytstvo Lvivskoi politekhniki (in Ukrainian).
14. Lavrishcheva, K. M. (2008). Prohramna inzheneriia. Kyiv (in Ukrainian).
15. Tabunshchuk, H. V., Kudermetov, R. K., & Brahina, T. I. (2013). Inzheneriia yakosti prohramnoho zabezpechennia. Zaporizhzhia : ZNTU (in Ukrainian).
16. Rozvytok ukraïnskoi IT-industrii: analitichnyi zvit. It Ukraine. (2018) (in Ukrainian).
17. Protsesy ta systemy pidtrymky yakosti prohramnykh system. Retrieved from <https://dspace.uzhnu.edu.ua/jspui/bitstream/lib/16456/1/> (in Ukrainian).
18. Modeli zhyttievoho tsykladu, pryntsyipy i metodolohii rozrobky prohramnoho zabezpechennia. Retrieved from <https://evergreens.com.ua/ua/articles/software-development-metodologies.html> (in Ukrainian).
19. Stadii tsykladu rozrobky PZ. Retrieved from <https://mango-test.qalight.com.ua/ua/baza-znaniy/stadiyi-tsykladu-rozrobki-pz/> (in Ukrainian).
20. Testuvannia ta yakist prohramnoho zabezpechennia. Retrieved from [https://kpi-fict-ip32.github.io/Blog/s07/software\\_quality.html](https://kpi-fict-ip32.github.io/Blog/s07/software_quality.html) (in Ukrainian).
21. Durnyak, B., Pikh, I., & Senkivskyy, V. (2018). Method of determining the weight of predicates of semantic networks in publishing processes: Przegląd papierniczy (Polish Paper Review). Warszawa, (1), 57–60 (Scopus) (in English).
22. Osinchuk, O. I., Pikh, I. V., & Senkivskiy, V. M. (2017). Model zabezpechennia yakosti tematychnoho planuvannia knyzhkovykh vydan. Science and Education a New Dimension (Natural and Technical Sciences). Budapest, (16), Issue 148, 38–40.
23. Senkivskyy, V., Pikh, I., Babichev, S., Kudriashova, A., & Senkivska, N. (April 2021). Modeling of Alternatives and Defining the Best Options for Websites Design. 2-nd International Seminar on Intelligent Information Technologies and Information Security Systems (IntelliTISIS-2021). CEUR Workshop Proceedings, 2021, 2853. Khmelnytskyi, 259–270 (in English).
24. Pikh, I. V., Durniak, B. V., Senkivskiy, V. M., & Holubnyk, T. S. (2017). Informatsiini tekhnolohii formuvannia yakosti knyzhkovykh vydan. Lviv : Ukrainska akademiia drukarstva (in Ukrainian).
25. Senkivskiy, V. M., Pikh, I. V., Kudriashova, A. V., & Lytovchenko, O. V. (2016). Teoretychni osnovy zabezpechennia yakosti vydavnycho-polihrafichnykh protsesiv (Chastyna 2: Syntez modelei priorytetnosti dii faktoriv): Polihrafiia i vydavnycha sprava, 1 (71), 20–29 (in Ukrainian).

26. Durniak, B. V., Pikh, I. V., & Senkivskyi, V. M. (2022). Teoretychni osnovy informatsiinoi kontseptsii formuvannia ta otsiniuvannia yakosti vydavnycho-polihrafichnykh protsesiv. Lviv : Ukrainska akademiia drukarstva (in Ukrainian).

doi: 10.32403/0554-4866-2022-2-84-9-21

## METHODOLOGICAL PRINCIPLES OF SOFTWARE QUALITY FORMATION (PART 1: BASIC MODEL OF QUALITY FACTORS)

V. M. Senkivskyy<sup>1</sup>, I. V. Pikh<sup>1,2</sup>, I. V. Kalynii<sup>3</sup>, N. Yu. Senkivskyy<sup>1</sup>, M. A. Drahomirov<sup>1</sup>

<sup>1</sup>*Ukrainian Academy of Printing,  
19, Pid Holoskom St., Lviv, 79020, Ukraine*

<sup>2</sup>*Lviv Polytechnic National University,  
12 Stepana Bandery St., Lviv, 79013, Ukraine*

<sup>3</sup>*Berezhanskyi Agrotechnical Institute,  
20, Academichna St., Berezhany, 47501, Ukraine  
senk.vm@gmail.com*

*The concept of “quality” in relation to software acts as a certain characteristic, criterion or metric, the degree of compliance with which should be as close as possible to a certain standard that determines the optimal quality. Since software, unlike many types of products, is often created for (under) a specific customer, the requirements for its quality have a slightly different interpretation, because they are formed additionally depending on the needs of the future user.*

*The adopted international standards related to software indicate that with the development of the software industry there is a need to change approaches and criteria for assessing the quality of these products. Quantification of software quality characteristics with the help of metrics determines the level of programs “a posteriori”, requiring decision-making to eliminate the shortcomings of the final product. This approach is supplemented in the article by the methodology of a priori obtaining the quality of the resulting products through the development of tools (models or information technologies) that determine the factors of formation and prognostic assessment of the quality of processes at the stage of their implementation.*

*The paper contains a review of literature sources related to the proposed topic. The expediency of using the information concept of quality formation of publishing and printing processes in relation to the study of software quality issues is substantiated. The main characteristics (factors) of software quality, determined by standards in the field of software engineering, are highlighted. A formalized graphical representation of the structural relationships between the factors using a semantic network, the description*

*of which is made using the language of predicate logic. Using the methodology of modeling hierarchies, the levels of preferences of factors are determined on the basis of the constructed binary matrix of reachability, the processing of which is performed by iterative tables. The basic model of factors of influence on software quality is synthesized by means of infographics.*

**Keywords:** *software, software quality factors, semantic network, predicate logic language, hierarchy modeling, reachability matrix, iteration tables, quality factors model.*

*Стаття надійшла до редакції 21.09.2022.*

*Received 21.09.2022.*