

# ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ОЦІНКИ ФУНКЦІОНАЛЬНОЇ СКЛАДНОСТІ РОЗРОБКИ ТА ПІДТРИМКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ З ВИКОРИСТАННЯМ МОДЕЛЕЙ НЕЧІТКОЇ ЛОГІКИ

<sup>1</sup> Вінницький національний технічний університет;

## *Анотація*

*Запропоновано математичну модель для інформаційної технології оцінювання складності розробки програмного забезпечення з використанням методів нечіткого виведення при прийнятті рішень та штучних нейронних мереж для визначення впливу факторів поточного середовища при застосуванні методу аналізу функціональних точок.*

**Ключові слова:** метод функціональних точок, нечітке виведення, прийняття рішень, штучні нейронні мережі, оцінка складності розробки програмного забезпечення.

## *Abstract*

*A mathematical model is proposed for an information technology that estimates the size of software using fuzzy inference methods in decision support systems and artificial neural networks to determine the influence of current environmental factors when applying the function point analysis (FPA) method.*

**Keywords:** function point analysis, fuzzy inference, decision support systems, artificial neural networks, software sizing and estimation.

## **Вступ**

Одним із найбільш розповсюджених та стандартних методів оцінки функціональної складності, функціонального розміру, зусиль та витрат на розробку програмного забезпечення є аналіз функціональних точок (FPA). Однак, у методі, що був запропонований Аланом Альбрехтом [1] і який пропонує функціональні точки як незалежний від технології захід вимірювання розміру, містить кілька проблем. До числа типових обмежень використання класичного методу FPA автори [2] відносять суб'єктивність технологічного фактору, потенційний подвійний підрахунок, ускладненість використання на ранніх етапах життєвого циклу, чутливість до якості та зміни вимог, диференціація специфікованих елементів, залежність від технології, тощо. Автором [2-3] було розглянуто та типізовано загальні проблеми, які властиві класичним методам, що базуються на FPA, у сучасних реаліях індустрії розробки та впровадження програмних комплексів: орієнтація виключно на функціональні вимоги і слабка адаптація по поточних реаліях хмарних платформ та комплексів (as-a-service продуктів), детермінованість множини об'єктів і дискретність класифікації функцій, вплив якості (неповноти, нечіткості) даних на ефективність оцінювання. Типовими наслідками розглянутих обмежень є недостатня і чутлива до якості даних точність оцінювання та складність, або навіть неможливість, застосування подібних методів для проєктів професійних сервісів та підтримки. У роботі [4] розглянуто основні напрямки вдосконалення методів оцінки обсягу та трудомісткості процесів розробки програмного забезпечення: використання методів ML для розпізнавання, класифікації, кластеризації і усунення нечіткості або неповноти вимог, використання методів нечіткої логіки, адаптація моделей оцінювання розміру FPA відповідно до потреб сучасної індустрії.

Метою даної роботи є розробка моделі нечіткої функціональної оцінки складності програмного забезпечення із застосування методу аналізу функціональних точок і методу оцінки ступенів втручання [3].

## Результати дослідження

Відповідно до актуального на сьогоднішній день стандарту ISO/IEC 20926:2009 [5] FPA - це стандартизований метод, спрямований на визначення розміру програмного забезпечення на основі його функціональних вимог, який дозволяє визначити обсяг роботи, необхідний для розробки та підтримки. Ключовими термінами, які використовується методом функціональних точок при підрахунках, є функції та складність. Метод функціональних точок розглядає дві категорії типів функцій (дані та транзакції) та п'ять типів функцій. Структура моделі сутностей наведена на рис.1.

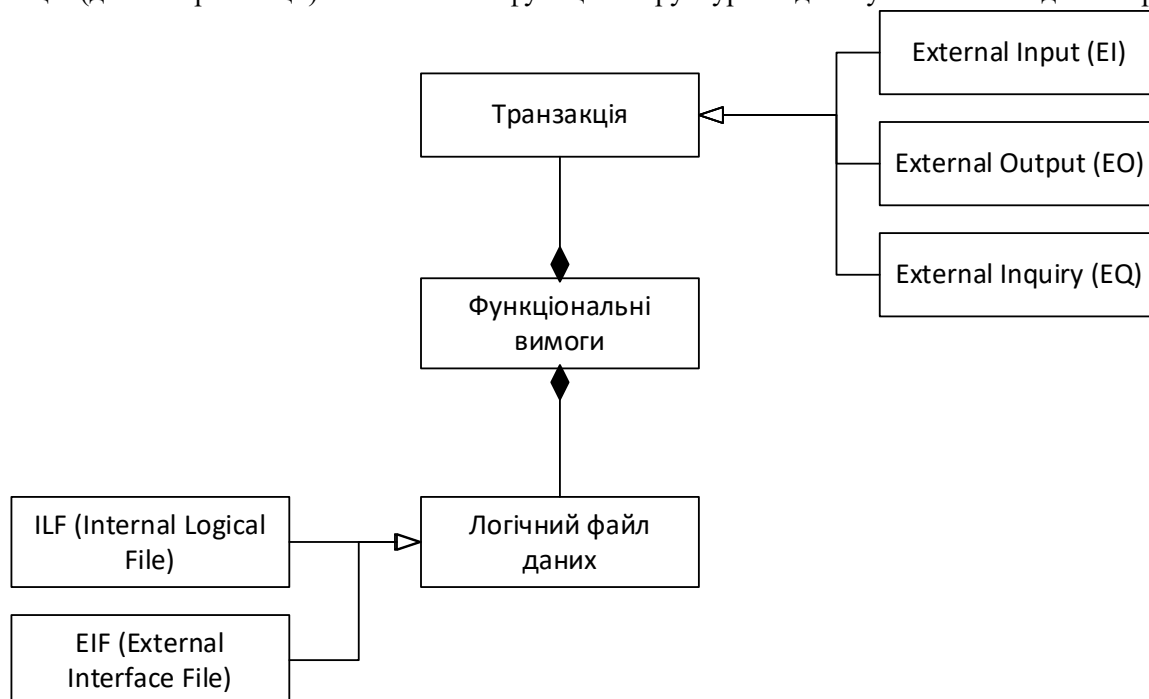


Рис. 1. Модель сутностей FPA за IFPUG

Складність кожної функції визначається відповідно до її характеристик та складових елементів за детермінованою матрицею складності [5]. Загальна кількість функціональних точок обраховується як сума складності усіх функцій з врахуванням загального впливу факторів корегування і визначається за формулою [5]:

$$AFP = (0.65 + 0.01 * \sum_{i=1}^{14} GSC_i) * \sum_{i=1}^5 TE_i * W_i \quad (1)$$

Для адаптації класичного методу FPA до вимог ІТ-проектів підтримки та проектів із динамічною зміною вимог в роботі [3] було введено фактор ступеню втручання (Alteration Adjustment Factor), який приймає значення від 0.1 до 1.0 і визначається експертно відповідно до значень табл.1.

Табл. 1.

Ступінь втручання	Коефіцієнт AAF
Повне	1.0
Критичне	0.5
Середнє	0.25
Низьке	0.1

З врахуванням додаткового фактору формула підрахунку загальної кількості функціональних точок приймає вигляд:

$$AAFP = (0.65 + 0.01 * \sum_{i=1}^{14} GSC_i) * \sum_{i=1}^5 TE_i * W_i * AAF_i, \quad (2)$$

Очевидним недоліком даного методу є можлива суб'єктивність оцінки фактору ступеню втручання від компетенції та обізнаності експерта. Враховуючи, що фактор ступеню втручання має безпосередній вплив на результуючу оцінку, метод висуває суттєві вимоги до володіння предметною областю та

специфікою кожного конкретного рішення.

Авторами [6] було запропоновано метод Fuzzy Function Point Analysis, який розширює традиційну модель FPA із використанням теорії нечіткої логіки та нечітких множин. Основна ідея методу FFPA полягає у тому, що типи функцій даних (ILF і EIF) та транзакційних функцій (EI, EO і EQ) у межах їх відповідних матриць функціональної складності можуть бути відображені на універсум дискурсу  $X$ , який відповідає посиланням на їх складові елементи. Усі ці матриці використовують одні й ті ж лінгвістичні терміни «низький», «середній» і «високий» для вираження складності. Для кожного рядка складності цих матриць були сформовані нечіткі значення для кожного з їх лінгвістичних термінів. Автори методу FFPA експериментальним шляхом [6] доводять, що використання трапецієподібних нечітких чисел є найбільш ефективним для відображення матриць складності функцій. Загалом метод FFPA містить наступні кроки:

1. Фазифікація лінгвістичних терм складності FPA і генерація нечіткого трапецієподібного числа.
2. Розширення матриць складності FPA новою (або новими) лінгвістичними термами.
3. Визначення кількості функціональних точок для нових лінгвістичних термів.
4. Дефазифікація лінгвістичних термів FFPA в значення функціональних точок.

За рахунок застосування методів нечіткого виведення до калібрування виведенням лінгвістичних термів складності функції FPA, автори [7,8] змогли підвищити ефективність методу та зменшити його чутливість при застосуванні для систем, що еволюціонують. Отже, використання подібних підходів, моделей та методів є перспективним з точки зору підвищення загальної ефективності оцінки складності та витрат при розробці та підтримці програмного забезпечення широкого спектру.

Подальше розширення методу FFPA за рахунок введення фактору ступеня втручання потребує прийняття рішень щодо значення самого фактору, а отже є задачею класифікації, для якої можна використати спеціальну інтерпретацію механізму, що базується на композиційному правилі нечіткого виведення Заде [9].

Задача класифікації полягає в віднесенні об'єкта (функції, що оцінюється, та відповідного ступеню втручання, що заданий вектором інформативних ознак  $X = (x_1, x_2, \dots, x_n)$  до одного з наперед означених класів  $\{d_1, d_2, \dots, d_m\}$ , тобто полягає в виконанні відображення:

$$X = (x_1, x_2, \dots, x_n) \rightarrow y \in \{d_1, d_2, \dots, d_m\}. \quad (3)$$

Розглядаючи задачу оцінки складності та супутніх витрат на розробку, підтримку та конфігурацію програмного забезпечення, варто виділити декілька під-задач, які потребують прийняття рішень:

1. Визначення типу функції з детермінованого (класичний FPA) або наперед не детермінованого переліку можливих типів.
2. Визначення складності функції відповідно до детермінованого (класичний FPA) або наперед не детермінованого переліку можливих значень.
3. Визначення значення ступеня втручання відповідно до поточного стану середовища (програмного засобу, що модифікується).
4. Визначення та калібрування значень факторів корегування.

Варто зазначити, що набір факторів, які характеризують поточний стан середовища, є сталим для кожної із під-задач, але із з ступенем впливу, який є динамічним впродовж життєвого циклу середовища. Модель прийняття рішень, що пропонується, описується множиною  $F$ :

$$F = \{X, X', K, R, Y, Z\} \quad (4)$$

де  $X = \{x_i\}$  – множина всіх можливих вхідних параметрів,  $i = 1 \dots I$ ,

$X' = \{x'_j\}$  – корисні вхідні параметри,  $j = 1 \dots J$ ,

$K$  – поле знань, яка характеризує загальну модель бази знань,

$R = \{r_s\}$  – множина критеріїв,  $s = 1 \dots S$ ;

$Y = \{y_t\}$  – множина вихідних рішень (класифікацій)  $t = 1 \dots T$ ,

$Z$  – оператор виведення.

У свою чергу, загальну модель бази знань для класу задач, що розглядаються, можна представити у вигляді множини:

$$K = \{K_n, K_\phi\}, \quad (5)$$

де  $K$  – це поле знань,

$K_n$  – продукційна компонента поля знань,

$K_\phi$  – фреймова компонента поля знань.

Застосування штучних нейронних мереж є ефективним інструментом для підвищення ефективності різних етапів методів оцінки складності розробки програмного забезпечення [6-8]. Наприклад, автори [7] застосовують штучні нейронні мережі для вирішення задачі калібрування взаємозв'язку між 14 факторами корегування і досягають підвищення загальної ефективності методу шляхом застосування двошарової нейронної мережі у поєднанні з В-сплайн функцією входження для визначення впливу кожного із факторів. Із врахуванням підвищеної ефективності методи, яка була досягнута авторами, вбачається доцільним комбінування даного підходу з розглянутою раніше моделлю прийняття рішень на основі моделі знань для застосування при вирішенні усіх чотирьох під-задач прийняття рішень, які були розглянуті раніше, при оцінці складності розробки та підтримки програмного забезпечення. Загальна модель штучної нейронної мережі в цьому випадку зображена на рис. 2.

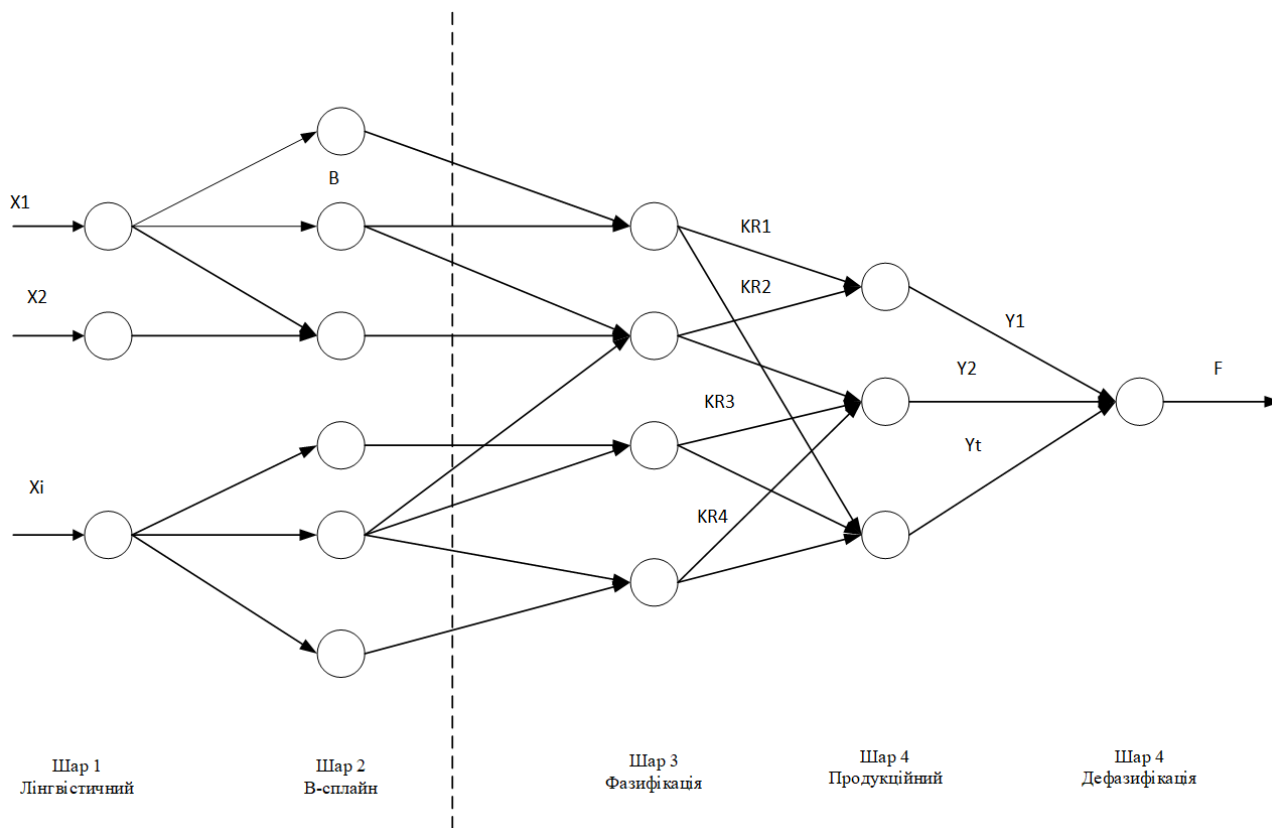


Рис. 2. Модель нечіткої штучної нейронної мережі для прийняття рішень щодо факторів впливу

### Висновки

Розглянуто сучасні підходи до підвищення ефективності та усунення відомих недоліків методів оцінки складності програмного забезпечення та витрат на його розробку та підтримку. Запропоновано загальну структуру моделі прийняття рішень та моделі знань, які за рахунок поєднання нечіткої компоненти знань і продукційної моделі дозволять більш ефективно представляти знання щодо поточного стану середовища, і відкривають можливість для розробки алгоритмів прийняття рішень щодо впливу факторів, включно із фактором оцінки ступеню втручання, які впливають на загальну оцінку складності. Запропоновано структуру штучної нейронної мережі, що базується на використанні В-сплайн функції входження і запропонованої моделі прийняття рішень з врахування накопичених знань про стан середовища, яка дозволить зменшити вплив факторів невизначеності і підвищити загальну ефективність та стійкість методу функціональних точок.

Майбутні дослідження будуть направлені на фіналізацію множини параметрів та факторів, подальшу оптимізацію запропонованих моделей та розробку алгоритмів для оцінки ефективності запропонованого методу.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Albrecht, A.J. Measuring Applications Development Productivity - Proceedings of IBM Application. Dev. Joint SHARE/GUIDE Symposium, 1979.
2. Al-Hagery M.. New complexity weights for function point analysis using artificial neural networks, 2004 – PhD Thesis. Режим доступу URL: [https://www.researchgate.net/publication/328119159\\_NEW\\_COMPLEXITY\\_WEIGHTS\\_FOR\\_FUNCTION\\_POINT\\_ANALYSIS\\_USING\\_ARTIFICIAL\\_NEURAL\\_NETWORKS](https://www.researchgate.net/publication/328119159_NEW_COMPLEXITY_WEIGHTS_FOR_FUNCTION_POINT_ANALYSIS_USING_ARTIFICIAL_NEURAL_NETWORKS).
3. Bilous D., Kozlovskiy A. Using function point analysis for professional service and maintenance IT projects: a tailoring approach for enhanced size and effort estimation.- XI International Scientific and Practical Conference «Modern science: theoretical and practical view», April 16-17, 2024, Madrid. Spain. Pp.82-90, Режим доступу URL: <https://www.sconferences.com/wp-content/uploads/2024/04/Madrid.Spain-11.pdf>
4. Білоус Д.А. Аспекти використання методу аналізу функціональних точок для оптимізації виробничих процесів - Молодь в науці: дослідження, проблеми, перспективи (МН-2023) – Режим доступу URL: <https://conferences.vntu.edu.ua/index.php/mn/mn2023/paper/view/17974>
5. ISO/IEC 20926:2009. Software and systems engineering. Software measurement. Режим доступу URL: <https://www.iso.org/standard/51717.html>
6. Lima Júnior O., Muniz Farias P., Belchior A. A Fuzzy Model for Function Point Analysis to Development and Enhancement Project Assessments - Vol. 5 No. 2 (2002): Special Issue of Best Papers presented at CLEI'2001. Merida, Venezuela. Guest Editor: Jose Aguilar (U de Los Andes, Venezuela). Режим доступу URL: <https://www.clei.org/cleiej/index.php/cleiej/article/view/358>
7. Tsoi H. L. Tuning Function Point Analysis Model by Using Fuzzy Neural Network, 2003. – Режим доступу URL: [https://www.researchgate.net/publication/254456675\\_Tuning\\_Function\\_Point\\_Analysis\\_Model\\_by\\_Using\\_Fuzzy\\_Neural\\_Network](https://www.researchgate.net/publication/254456675_Tuning_Function_Point_Analysis_Model_by_Using_Fuzzy_Neural_Network)
8. Lavazza L., Locoro A., Liu G., Meli R. Estimating Software Functional Size via Machine Learning. ACM Trans. Softw. Eng. Methodol. 32, 5, Article 114, 2023, 27 p.
9. Zadeh, L. : Fuzzy sets. Information and Control 8(3), 338–353 (1965).

**Білоус Дмитро Анатолійович**— аспірант кафедри комп'ютерних наук, Вінницький національний технічний університет, Вінниця, e-mail: dmytro.bilous@gmail.com

**Козловський Андрій Володимирович** — канд. техн. наук, доцент кафедри комп'ютерних наук, Вінницький національний технічний університет

**Bilous Dmytro A.** — PhD. Student of Computer Science Department, Vinnytsia National Technical University, Vinnytsia, email: dmytro.bilous@gmail.com

**Kozlovskiy Andrii V.** — Cand. Sc. (Eng), Assistant Professor of Computer Science Department, Vinnytsia National Technical University, Vinnytsia