

КОМП'ЮТЕРНІ НАУКИ

УДК 004.8+004.94

[https://doi.org/10.32515/2664-262X.2025.11\(42\).1.5-13](https://doi.org/10.32515/2664-262X.2025.11(42).1.5-13)

Д. А. Білоус, А. В. Козловський, доц., канд. техн. наук
Вінницький національний технічний університет, м. Вінниця, Україна
e-mail: dmytro.bilous@gmail.com, akozlovskiy@vntu.edu.ua

Загальна модель бази знань інформаційної технології оцінки складності розробки програмного забезпечення із застосуванням SaaS/PaaS платформ

Критичним фактором успішного управління проектами та ефективного планування є оцінювання складності та зусиль для розробки програмного забезпечення. Використання стандартних для індустрії методів, таких як аналіз функціональних точок, є ефективним засобом підвищення точності оцінювання зусиль при одночасному зменшенні вартості безпосередньо процесу оцінювання. Водночас, стандартні методи, більшість з яких була розроблена декілька десятиліть тому, не є достатньо адаптованими для сучасних реалій природи розробки програмного забезпечення з використанням гнучких підходів та готових або типових рішень. У роботі запропоновані моделі представлення знань з поєднанням продукційної та фреймової моделей для вирішення задачі оцінки складності та трудомісткості розробки програмного забезпечення із застосування SaaS (Software as a Service) та PaaS (Platform as a service) рішень. Розроблено модель бази знань, яка враховує особливості поєднання фреймової та продукційної моделей та забезпечує сумісність із методом аналізу функціональних точок (Function Point Analysis).

Розроблені моделі та їх взаємодії можуть стати основою для розробки моделі прийняття рішень у інформаційній технології оцінки складності розробки програмного забезпечення із врахування факторів середовища та критеріїв оцінювання. Застосування отриманих моделей надасть змогу створити алгоритми автоматизованої оцінки складності розробки програмного забезпечення в умовах неповної визначеності функціональних вимог.

оцінка складності розробки програмного забезпечення, модель знань, фреймова модель, продукційна модель, аналіз функціональних точок

Постановка проблеми. Управління витратами та планування ресурсів у розробці програмного забезпечення значною мірою залежить від точності оцінки зусиль, необхідних для реалізації функціоналу. Одним із найбільш розповсюджених методів оцінки складності є Function Point Analysis (FPA), який дозволяє визначати розмір програмного продукту на основі його функціональних можливостей. Проте класична методологія FPA має ряд обмежень, що ускладнюють її застосування в умовах використання SaaS/PaaS-платформ та сучасних підходів до розробки програмного забезпечення. Основні проблеми, що виникають:

– нечіткість вимог на ранніх етапах. Методології, такі як Agile, передбачають ітеративне уточнення вимог, що створює труднощі для класичних підходів FPA, які орієнтовані на стабільну специфікацію функціоналу. На всіх стадіях розробки, особливо на початкових, типовими є нечіткість, неповнота або недостовірність даних щодо функціональних вимог. В результаті, оцінка FPA може не відображати реальну складність розробки таких функцій.

– значні витрати на експертне оцінювання. Точний аналіз складності ПЗ вимагає залучення досвідчених спеціалістів, що збільшує вартість оцінки.

– недостатня адаптація до використання платформних рішень. Багато сучасних систем будуються на основі конфігурації готових сервісів (SaaS/PaaS), однак традиційний FPA використовує кінцеву множину функцій транзакцій та даних, які та

виключно які використовуються для обчислення складності, тому не диференціює використання частково готових рішень від повноцінного процесу розробки, що може призводити до неточних прогнозів.

– складність оцінки професійних сервісів та підтримки: FPA не враховує зусиль, необхідних для початкової конфігурації, адміністрування, підтримки та MACD-модифікації (Moves, Adds, Changes and Deletes) програмного забезпечення протягом його життєвого циклу. Статистичні дані свідчать, що з кінця вісімдесятих років більшість організацій витрачають щонайменше 50% своїх фінансових ресурсів на супровід програмних продуктів. За останні двадцять років ця частка у деяких сферах зросла до 80-95%. Це підкреслює важливість досліджень у галузі підтримки програмного забезпечення. Оскільки професійний сервіс часто є суттєвою, а подекуди і єдиною, складовою проєктної діяльності, ігнорування цього аспекту є значним недоліком.

Аналіз останніх досліджень і публікацій. Останні дослідження присвячені вдосконаленню методів оцінки зусиль у розробці програмного забезпечення шляхом використання нечіткої логіки, нейронних мереж та їх комбінацій. У роботі [1] представлено систематичний огляд літератури та досліджень, автори роблять фокус на визначення актуальності використання методу аналізу функціональних точок, типових проблем, що виникають при його використанні, раціональності та ефективності запропонованих покращень. Автори [1] виявили, що методи повторного калібрування ваг і складності на основі штучного інтелекту мають багатообіцяючі результати, з підвищенням точності оцінки на 13-30% порівняно зі стандартним методом FPA. Водночас, фундаментальні проблеми, наприклад технологічна незалежність, особливо у розрізі застосування готових чи автоматизованих рішень, залишаються невирішеними.

У роботі [2] розглядається використання методів машинного навчання для оцінки показників функціонального розміру програмного забезпечення як альтернатива більш традиційним підходам, таким як метод високорівневого аналізу функціональних точок (High-level FPA, NESMA method). Дослідники провели емпіричне дослідження, використовуючи великий набір даних предикторів функціонального розміру, навчаючи та оцінюючи три моделі машинного навчання: випадкові ліси, опорну векторну регресію та нейронні мережі. Автори демонструють, що методи машинного навчання можуть досягти рівня точності, порівнянного з усталеними методами оцінювання, такими як FPA високого рівня, вимагаючи меншого рівня достовірності та точності вхідних даних про програмний проєкт.

Дослідження [3] представляє керований даними підхід до покращення оцінки зусиль програмного забезпечення шляхом поєднання логарифмованих формул FPA/PDR, вдосконаленої техніки регресії (Adam Optimizer) і перехресної перевірки. Результати демонструють ефективність запропонованого методу Adj-Effort у забезпеченні більш точних і надійних прогнозів зусиль порівняно з існуючими підходами. Суттєвим здобутком роботи є формалізація набору критеріїв ефективності, який можна використовувати для майбутніх досліджень.

Автори [4] аналізують нечітку модель із використанням субтрактивної кластеризації як підходу до подолання обмежень, пов'язаних із каскадними нечіткими логічними контролерами. Ця методика спрямована на подальше скорочення бази правил і покращення оцінок зусиль розробки програмного забезпечення. Дослідження вивчає техніку кластеризації шляхом поділу даних на кластери, а потім представлення кожного кластера за допомогою одного правила, ефективно мінімізуючи базу правил.

Робота доводить, що застосування нечіткої логіки для оцінки зусиль програмного забезпечення, зокрема за допомогою використання каскадного підходу і субтрактивної кластеризації, показує багатообіцяючі результати в підвищенні точності та ефективності оцінки зусиль розробки програмного забезпечення.

Робота [5] містить детальне дослідження нечітких множин, лінгвістичних змінних і систем нечіткого висновку, відзначає концепцію нечіткої логіки як ефективного інструменту для обробки неповноти даних і невизначеності вхідних параметрів у ранніх моделях оцінки програмного забезпечення. Автори доводять переваги нечітких систем, систем, заснованих на знаннях, і їх нелінійне відображення між вхідними та вихідними векторами.

Роботи [6-9] розглядають застосування методів нечіткої логіки для підвищення ефективності і точності методів оцінки складності розробки програмного забезпечення та керування ризиками в ІТ-проєктах. Автори доходять висновку про перспективність застосування нейронечітких моделей для калібрування значень вагових коефіцієнтів компонентів функціональних точок і наголошують на необхідності постійної переоцінки в залежності від фактичними витратами у завершених проєктах. Крім того, зазначені роботи підкреслюють перспективність застосування підходів застосування розмірних систем оцінювання складності та витрат на розробку, але лише за умови їх адаптації до реалії поточного середовища.

В статті [10] представлено модель, засновану на техніках глибокого та машинного навчання із поєднанням згорткових нейронних мереж і алгоритму рою частинок для оцінку кінцевої вартості розробки програмного забезпечення. Автори використовують критерії оцінювання аналогічні до наведених у роботі [3] і досягають суттєвого підвищення ефективності традиційних методів. У статті підкреслюється актуальність сучасних методів оцінки вартості програмного забезпечення та окреслюються майбутні напрямки досліджень для усунення обмежень і підвищення практичної користі запропонованої моделі в динамічних середовищах розробки програмного забезпечення.

Задача формалізації нечітких функціональних вимог на ранніх стадіях життєвого циклу проєктів з розробки програмного забезпечення за допомогою моделей-трансформерів розглядається авторами роботи [11]. Отримані результати є перспективними і можуть дозволити розширення запропонованого підходу на інші сфери.

Отже, більшість останніх досліджень фокусується на усуненні або мінімізації впливу факторів нечіткості або неповної визначеності даних при застосуванні класичних методів квантифікованої оцінки обсягу, складності та трудомісткості розробки програмного забезпечення із застосуванням сучасних підходів та здобутків у галузі машинного навчання, таких як нейронні мережі, нечітка логіка, традиційні алгоритмічні підходи та їх поєднання. Аналіз публікацій показує, що застосування наведених підходів дозволяє суттєво підвищити ефективність процесу оцінки складності програмного забезпечення. Водночас, слід зазначити, що більшість досліджень зосереджені на підвищенні ефективності класичних методів оцінки зусиль для традиційної розробки повного циклу. Їх застосування у сучасному ІТ-ландшафті, який значною мірою орієнтований на професійні сервіси, SaaS і PaaS, є обмеженим [12]. Більшість моделей не враховують адаптивні методи розгортання, використання готових платформних сервісів та складність інтеграції в існуючі екосистеми. Це підтверджує необхідність розробки підходу, який дозволить оцінювати складність

реалізації функціональних вимог з урахуванням специфіки сучасних програмних архітектур і із застосуванням моделей нечіткої логіки та нейронних мереж [13].

Постановка завдання. Таким чином, метою дослідження є розробка базової моделі знань інформаційної технології для оцінки складності та трудомісткості розробки програмних продуктів з використанням SaaS/PaaS рішень. Основою цієї моделі стане комбінована структура знань, що поєднує фреймову та продукційну моделі, дозволяючи ефективно представити вимоги та механізми їхнього виконання. Основні завдання дослідження:

- Вибір базової моделі представлення знань при прийнятті рішень щодо складності, обсягів та трудомісткості розробки і впровадження програмного забезпечення.

- Побудова загальної моделі знань із використанням фреймової та продукційної моделей, що забезпечить ефективне збереження, оновлення та використання знань у процесі оцінювання.

- Побудова загальної моделі прийняття рішень для задачі оцінки складності програмних продуктів, включаючи конфігурацію, інтеграцію та розгортання на основі платформних сервісів (SaaS/PaaS).

Запропоновані моделі мають дозволити створити інформаційну технологію підтримки прийняття рішень, яка забезпечить точну оцінку трудомісткості розробки та стане основою для ефективного керування програмним продуктом у сучасних умовах розробки.

Виклад основного матеріалу. В якості моделі знань для представлення предметної області, що розглядається, та відображення взаємозв'язків між сутностями, пропонується використання поєднання продукційної і фреймової моделей. Авторами роботи [14] була розглянута фреймова модель для зберігання та обробки інформації про вимоги до створення програмних систем, яка містить такі основні компоненти: фрейми вимог, фрейми реалізації, фрейми контексту та фрейми типових для FPA сутностей. Базова фреймова модель знань для предметної області, що розглядається, наведена на рисунку 1.

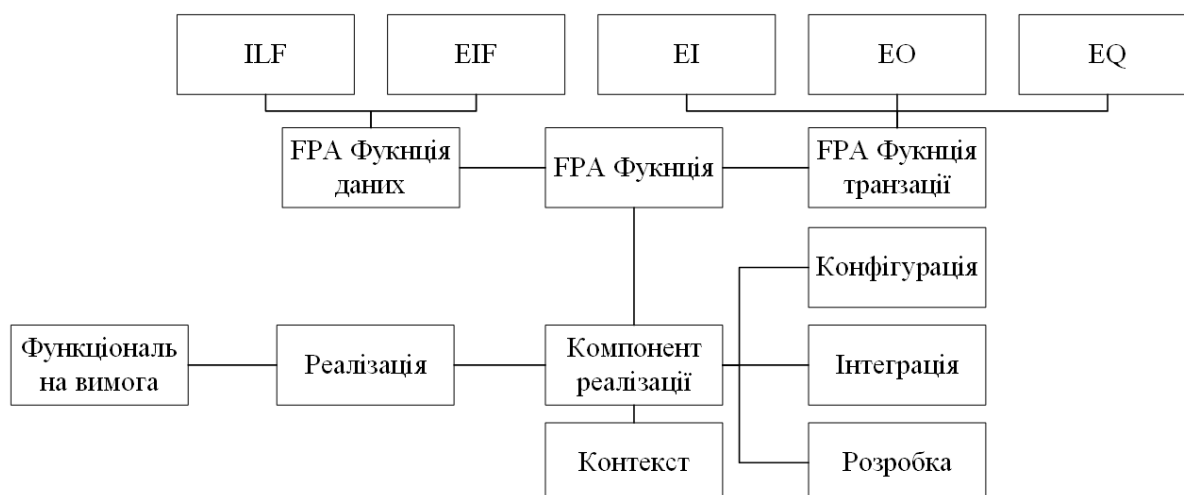


Рисунок 1 – Базова фреймова модель знань

Джерело: розроблено авторами

Типовим завданням при проведенні будь-якої експертної оцінки, а особливо в області оцінки розміру та відповідних трудовитрат на розробку програмного

забезпечення, є використання набутого досвіду з розробки аналогічних або сумісних програмних продуктів. Зазначений досвід складається з певного набору порад, правил та рекомендації щодо аспектів оцінювання, які потрібно враховувати в разі аналізу вимог певного типу або елементарного фрагменту вимог. Такі апіорні дані можна формалізувати у вигляді множини правил-продукції. Відомим недоліком продукційних моделей знань є вірогідність надлишковості за умови використання виключно продукцій звичайного вигляду і нехтуванням появи дублювання умов.

В предметній області оцінки складності розробки програмного забезпечення з врахуванням набутого досвіду та із застосуванням готових рішень або з повторним використанням вже розроблених рішень, доцільно для організації знань застосувати поєднання продукційної та фреймової моделей, що буде мати наступні переваги кожної із складових:

- Простота реалізації та поступової модифікації при додаванні, видаленні, редагуванні та калібруванні знань.
- Можливість застосування наслідування у фреймовій частині знань.
- Відсутність необхідності аналізу складних структур даних у продукційній частині знань, і, як наслідок, зменшення алгоритмічної складності кінцевого методу прийняття рішень.
- Зменшення кількості правил продукції та результуючої надлишковості за рахунок використання фреймової складової.
- Адаптованість та легкість у використанні та сприйнятті людиною-експертом.

Важливим аспектом кінцевої інформаційної технології для оцінювання складності розробки програмного забезпечення є її пристосування до обробки нечітких або неповних вхідних даних, а також введення характеристики впевненості у коректності рекомендації, яка б базувалася на набутому досвіді, мала широкий діапазон для калібрування та могла б бути легко інтерпретована кінцевим користувачем.

Виходячи з вищенаведеного, загальну модель бази знань для класу задач оцінювання складності розробки програмного забезпечення, що розглядається, представимо у вигляді множини:

$$Z = \{Z_{\phi}, Z_{\pi}\} \quad (1)$$

де Z – поле знань,

Z_{ϕ} – фреймова компонента поля знань,

Z_{π} – продукційна компонента поля знань.

У свою чергу, продукційна компонента поля знань складається із множини правил-продукції:

$$Z_{\pi} = \{P_1, P_2, \dots, P_n\} \quad (2)$$

де кожне із правил продукції має наступний вигляд:

$$P_i: (r)S; P; A \rightarrow B; R, (i = 1 \dots n) \quad (3)$$

де r – ім'я продукції,

S – сфера застосування продукції, за допомогою якою створено можливість структурувати множину правил-продукції,

P – тригер або умова застосування ядра продукції,

$A \rightarrow B$ – ядро продукції,

R – постумови продукції.

Як було зазначено вище, оптимізація процесу виведення та зменшення алгоритмічної складності є однією із ключовим умов інформаційної технології, що проектується. Одним із підходів для досягнення цієї умови є виділення серед всієї множини правил продукції у базі знань активної та пасивної частин, яке було запропонована у [15]. За рахунок застосування даного підходу досягається зменшення кількості правил продукції, які будуть розглядатися у кожен конкретний момент часу, за рахунок постійної структуризації правил продукції.

Фреймова компонента поля знань у стандартному підході складається із множини фреймів, кожен з яких має один із чотирьох типів: фрейм-структура, фрейм-ситуація, фрейм-роль, фрейм-сценарій.

Для задачі, що розглядається, найбільш ефективним вбачається використання фреймів-структур для опису множини стійких сутностей із сталим набором властивостей і використання наслідування для представлення знань з певною структурною ієрархією. Водночас, використання фреймів-ролей, фреймів-ситуацій і фреймів-сценаріїв є недоречним через специфіку предметної області та використання продукційної компоненти бази знань для представлення знань подібного типу.

Таким чином, структура бази знань схематично представлена на рисунку 2.

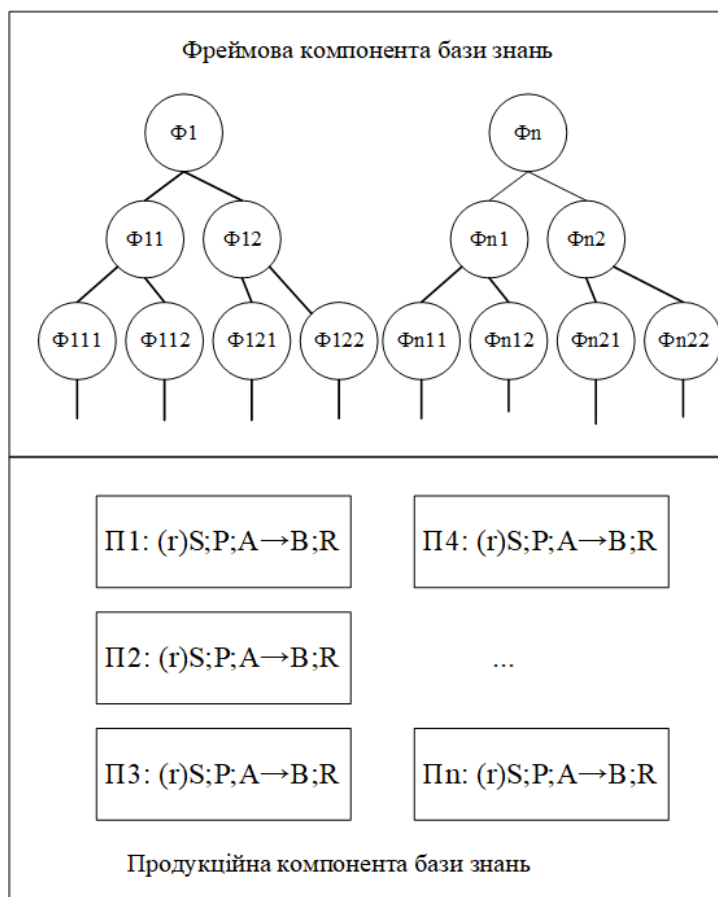


Рисунок 2 – Структура бази знань інформаційної технології оцінки складності розробки програмного забезпечення

Джерело: розроблено авторами

Висновки. Процес оцінювання складності та планування зусиль є невід’ємною складовою управління життєвим циклом розробки програмного забезпечення. Сучасні

тенденції у сфері оцінки складності трудомісткості розробки програмного забезпечення демонструють зростаючу потребу в адаптивних моделях, що враховують специфіку використання SaaS/PaaS-рішень та гнучких підходів до розробки. Традиційні методи, такі як FPA, залишаються ключовими в оцінці складності програмних систем, проте вони не враховують особливості інтеграції, налаштування та використання готових платформних рішень, що може призводити до завищених або некоректних оцінок витрат і ресурсів.

У статті було запропоновано модель знань для інформаційної технології оцінки функціональних вимог, яка поєднує фреймовий та продукційний підходи для більш точного моделювання знань та прийняття рішень. Основні результати дослідження включають:

– Формалізацію базової моделі, що враховує різні аспекти реалізації програмних рішень, зокрема традиційну розробку, конфігурацію та інтеграцію платформних сервісів.

– Розробку загальної моделі знань на основі гібридного підходу з поєднанням фреймової та продукційної компонент, яка дозволяє зберігати, оновлювати та аналізувати інформацію про вимоги, їхню складність та зв'язки з контекстними факторами.

– Забезпечення можливості впровадження механізму зворотного зв'язку, що використовує історичні дані для покращення точності оцінки складності та адаптації моделі до нових умов.

Запропонований підхід дозволяє більш точно оцінювати трудомісткість реалізації функціональних вимог з урахуванням контексту їх виконання, що особливо важливо для сучасних IT-проектів. Крім того, поєднання фреймової та продукційної моделей створює основу для подальшої інтеграції з методами штучного інтелекту та машинного навчання для автоматизації процесу оцінки зусиль.

Перспективи подальших досліджень включають розробку та формалізацію моделі прийняття рішень, експериментальні дослідження ефективності запропонованої моделі відносно традиційних підходів, розширення запропонованої моделі через використання нечіткої логіки та нейронних мереж для динамічного коригування оцінки складності залежно від змін у вимогах. Також доцільним є дослідження можливостей впровадження розробленої інформаційної технології у системи підтримки прийняття рішень для керування програмними продуктами та ресурсами.

Список літератури

1. de Freitas Junior M., Fantinato M., Sun, V. Improvements to the Function Point Analysis Method: A Systematic Literature Review. *IEEE Transactions on Engineering Management*. 2015. Vol. 62, №4, P. 495-506. URL: <https://doi.org/10.1109/TEM.2015.2453354> (дата звернення: 22.02.2025).
2. Lavazza L., Locoro A., Liu G. Estimating software functional size via machine learning. *ACM Transactions on Software Engineering and Methodology*. 2023. Vol. 32, №5. URL: <https://doi.org/10.1145/3582575> (дата звернення: 22.02.2025).
3. Hoc H. T., Hai V. V., Nhung H. L. T. K. An approach to adjust effort estimation of function point analysis. *Software Engineering and Algorithms*. 2021. P. 522-537. URL: https://doi.org/10.1007/978-3-030-77442-4_45 (дата звернення: 22.02.2025).
4. Sree P. R., Ramesh S. N. S. V. S. C. Improving efficiency of fuzzy models for effort estimation by cascading & clustering techniques. *Procedia Computer Science*. 2016. Vol. 85, P. 278-285. URL: <https://doi.org/10.1016/j.procs.2016.05.234> (дата звернення: 22.02.2025).
5. Nwaiwu J. C., Oluwadare S. Analytic study of fuzzy-based model for software cost estimation. *2nd ACM International Conference on Computing Research and Innovations (CoRI)*. 2016. Vol. 1755. URL:

- https://www.researchgate.net/publication/313351062_Analytic_Study_of_Fuzzy-based_model_for_Software_Cost_Estimation (дата звернення: 22.02.2025)
6. Setyohadi D. B. Estimation of software development project success using fuzzy logics. *Advances in Science, Technology and Engineering Systems Journal*. 2019. Vol. 4, №2, P. 280-287. URL: <https://doi.org/10.25046/aj040236> (дата звернення: 22.02.2025).
 7. Xia W., Ho D., Capretz L. F. Updating weight values for function point counting. *International Journal of Hybrid Intelligent Systems*. 2019, Vol. 6, №1, P. 1-14. URL: <https://doi.org/10.48550/arXiv.2005.11218> (дата звернення: 22.02.2025).
 8. Kaushik A., Soni A., Soni R. A type-2 fuzzy logic based framework for function points. *International Journal of Intelligent Systems and Applications*, 2013, Vol. 5, №3, P. 74-82. URL: <https://doi.org/10.5815/ijisa.2013.03.08> (дата звернення: 22.02.2025).
 9. Bou Nassif A., Azzeh M., Idri A. Software development effort estimation using regression fuzzy models. *Computational Intelligence and Neuroscience Journal*. 2019. URL: <https://doi.org/10.48550/arXiv.1902.03608> (дата звернення: 22.02.2025).
 10. Mohamed M., Emam O., Azzam S. M. Software cost estimation prediction using a convolutional neural network and particle swarm optimization algorithm. *Scientific Reports*. 2024, Vol.14, URL: <https://doi.org/10.1038/s41598-024-63025-8> (Дата звернення: 22.02.2025).
 11. Malik G., Cevik M., Bera, S. Software requirement specific entity extraction using transformer models. *The 35th Canadian Conference on Artificial Intelligence*. 2024. URL: <https://doi.org/10.21428/594757db.9e433d7c> (дата звернення: 22.02.2025)
 12. Білоус Д.А., Козловський, А.В. Використання аналізу функціональних точок для ІТ-проектів підтримки та професійного сервісу: модифікований підхід для покращення оцінки розміру та витрат. *XI Міжнародна науково-практична конференція «Сучасна наука: теоретичний та практичний погляд»*, 16-17 квітня 2024, Мадрид, Іспанія, С.82-90. URL: <https://doi.org/10.5281/zenodo.11065612> (дата звернення: 22.02.2025).
 13. Білоус Д.А., Козловський, А.В. Підвищення ефективності оцінки функціональної складності розробки та підтримки програмного забезпечення з використанням моделей нечіткої логіки. *Молодь в науці: дослідження, проблеми, перспективи (МН-2025)*. 2025. URL: <https://conferences.vntu.edu.ua/index.php/mn/mn2025/paper/view/22288> (дата звернення: 22.02.2025).
 14. Білоус Д.А., Козловський А.В. Фреймова модель знань для нечіткого аналізу функціональних точок у сучасній розробці з використанням SaaS технологій. *LIV Всеукраїнська науково-технічна конференція факультету інтелектуальних інформаційних технологій та автоматизації*. 2025. URL: <https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2025/paper/view/23168> (дата звернення: 22.02.2025).
 15. Юхимчук С. В. Математичні моделі ризику для систем підтримки прийняття рішень: монографія. Вінниця: УНІВЕРСУМ-Вінниця. 2003. 188 с.

References

1. de Freitas Junior, M., Fantinato, M., & Sun, V. (2015). Improvements to the Function Point Analysis Method: A Systematic Literature Review. *IEEE Transactions on Engineering Management*, 62(4), 495-506. <https://doi.org/10.1109/TEM.2015.2453354>.
2. Lavazza, L., Locoro, A., Liu, G., & Meli, R. (2023). Estimating software functional size via machine learning. *ACM Transactions on Software Engineering and Methodology*, 32(5), Article 114. <https://doi.org/10.1145/3582575>.
3. Hoc, H. T., Hai, V. V., & Nhung, H. L. T. K. (2021). An approach to adjust effort estimation of function point analysis. *Software Engineering and Algorithms*, 522-537. https://doi.org/10.1007/978-3-030-77442-4_45.
4. Sree, P. R., & Ramesh, S. N. S. V. S. C. (2016). Improving efficiency of fuzzy models for effort estimation by cascading & clustering techniques. *Procedia Computer Science*, 85, 278-285. <https://doi.org/10.1016/j.procs.2016.05.234>.
5. Nwaiwu, J. C., & Oluwadare, S. (2016). Analytic study of fuzzy-based model for software cost estimation. *2nd ACM International Conference on Computing Research and Innovations (CoRI)*. https://www.researchgate.net/publication/313351062_Analytic_Study_of_Fuzzy-based_model_for_Software_Cost_Estimation.
6. Setyohadi, D. B. (2019). Estimation of software development project success using fuzzy logics. *Advances in Science, Technology and Engineering Systems Journal*, 4(2), 280-287. <https://doi.org/10.25046/aj040236>
7. Xia, W., Ho, D., Capretz, L. F., & Ahmed, F. (2019). Updating weight values for function point counting.

- International Journal of Hybrid Intelligent Systems*, 6(1), 1-14. <https://doi.org/10.48550/arXiv.2005.11218>.
8. Kaushik, A., Soni, A., & Soni, R. (2013). A type-2 fuzzy logic based framework for function points. *International Journal of Intelligent Systems and Applications*, 5(3), 74-82. doi.org/10.5815/ijisa.2013.03.08.
 9. Bou Nassif, A., Azzeh, M., Idri, A., & Abran, A. (2019). Software development effort estimation using regression fuzzy models. *Computational Intelligence and Neuroscience Journal*. doi.org/10.48550/arXiv.1902.03608.
 10. Mohamed, M., Emam, O., & Azzam, S. M. (2024). Software cost estimation prediction using a convolutional neural network and particle swarm optimization algorithm. *Scientific Reports*, 14, Article 13129. <https://doi.org/10.1038/s41598-024-63025-8>.
 11. Malik, G., Cevik, M., Bera, S., & Yildirim, S. (2022). Software requirement specific entity extraction using transformer models. *The 35th Canadian Conference on Artificial Intelligence*. doi.org/10.21428/594757db.9e433d7c.
 12. Bilous, D.A., & Kozlovskiy, A.V. (2024). 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, 82-90 [in Ukrainian]. <https://doi.org/10.5281/zenodo.11065612>.
 13. Bilous, D., & Kozlovskiy, A. (2025). Improving the efficiency of assessing the functional complexity of software development and support using fuzzy logic models. *Youth in Science: Research, Problems, Prospects (MH-2025)*. <https://conferences.vntu.edu.ua/index.php/mn/mn2025/paper/view/22288> [in Ukrainian]
 14. Bilous, D., & Kozlovskiy, A. (2025). A frame knowledge model for fuzzy analysis of functional points in modern development using SaaS technologies. *LIV All-Ukrainian Scientific and Technical Conference of the Faculty of Intellectual Information Technologies and Automation*. <https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2025/paper/view/23168> [in Ukrainian].
 15. Yukhymchuk, S.V. (2003). *Mathematical risk models for decision support systems*. Vinnytsia: Universum-Vinnytsia [in Ukrainian].

Dmytro Bilous, Andrii Kozlovskiy, Assoc. Prof., PhD tech. sci
Vinnytsia National Technical University, Vinnytsia, Ukraine

Knowledge Representation Model for the Complexity Analysis of the Software Development using SaaS/PaaS Platforms

A critical factor in successful project management and effective planning is the analysis of software development complexity and effort. The use of industry-standard methods, such as Function Point Analysis, serves as an effective means of improving estimation accuracy while simultaneously reducing the cost of the estimation process itself. However, standard methods, most of which were developed several decades ago, are not sufficiently adapted to modern realities, including agile software development and the use of pre-built or standardized solutions. This study proposes knowledge representation model that combine production and frame-based approaches to address the challenge of assessing the complexity and effort of software development involving SaaS (Software as a Service) and PaaS (Platform as a Service) solutions. A knowledge base model has been developed, integrating both frame-based and production models while ensuring compatibility with Function Point Analysis.

The developed models and their interactions can serve as the foundation for a decision-making model within an information technology framework for software development complexity assessment, considering environmental factors and evaluation criteria. The application of the proposed models will enable the creation of automated algorithms for software development complexity estimation under conditions of incomplete functional requirements.

The proposed approach improves effort estimation by considering real-world implementation contexts, which is particularly relevant for modern IT projects. Additionally, the combination of frame-based and production models lays the groundwork for further integration with artificial intelligence and machine learning to automate effort estimation processes.

Future research should focus on refining the decision-making model, conducting experimental validation against traditional methods, and expanding its capabilities using fuzzy logic and neural networks for dynamic complexity assessment. The integration of this approach into decision support systems (DSS) for software project and resource management also remains a promising direction.

software development complexity analysis, knowledge representation model, frame-based model, production rule model, function point analysis

Одержано (Received) 24.02.2025

Прорецензовано (Reviewed) 28.02.2025

Прийнято до друку (Approved) 14.03.2025