

ГІБРИДНИЙ АЛГОРИТМ ОЦІНЮВАННЯ STORY POINTS НА ОСНОВІ СЕМАНТИЧНОГО ПОШУКУ ІСТОРИЧНИХ АНАЛОГІВ І LLM

Губченко Володимир Денисович

студент групи МПМ-І-25-14д,

Київського столичного університету імені Бориса Грінченка,

науковий керівник – к.т.н., доц. Яскевич В.О.

ВСТУП

Актуальність і постановка проблеми. Сучасні програмні проекти характеризуються великою кількістю текстової та структурованої інформації, що накопичується у системах керування задачами. У гнучких методологіях розробки істотною складовою планування є оцінювання трудомісткості задач у Story Points. Огляд досліджень з Agile-оцінювання показує, що результат залежить від досвіду експертів, організації роботи команди та якості історичних даних, тому автоматизовані методи розглядаються як засіб підтримки, а не повної заміни експертної оцінки [1].

Водночас пряме застосування великих мовних моделей не гарантує стабільної точності. Дослідження Deep-SE продемонструвало можливість прогнозування Story Points за текстом задач, а подальші роботи на основі Transformer-моделей показали потенціал кращого врахування контексту [2, 3]. Реплікаційні дослідження, однак, вказують, що сама семантична близькість задач не є достатньою для надійного розрізнення їхньої трудомісткості [4]. Тому доцільним є поєднання семантичного пошуку історичних аналогів із LLM-модулем, який отримує ці приклади як контекст для оцінювання.

Мета дослідження. Метою дослідження є формалізація та експериментальна перевірка алгоритму оцінювання трудомісткості задач у Story Points на основі семантичного пошуку історичних аналогів і LLM-модуля. Для досягнення мети передбачено формалізувати алгоритм, визначити протокол експериментальної перевірки та порівняти результати гібридного підходу із zero-shot оцінюванням.

Аналіз останніх досліджень і публікацій. Систематичний огляд Usman et al. показує, що оцінювання трудомісткості в Agile є окремим напрямом досліджень, у якому поєднуються експертні, статистичні та машинні методи [1]. Choetkiertikul et al. запропонували Deep-SE та відкритий набір із 23 313 задач із 16 open-source проектів, що став основою для наступних порівнянь [2].

Fu та Tantithamthavorn запропонували GPT2SP - Transformer-підхід для оцінювання Story Points. На 23 313 задачах із 16 проектів автори отримали медіанне MAE 1,16 у межах внутрішньопроєктного сценарію та показали покращення відносно низки базових методів [3]. Водночас реплікаційне дослідження Tawosi et al. засвідчило, що перевага складних моделей над простими базовими оцінювачами не є стабільною в усіх сценаріях [4]. Це підтверджує необхідність експериментальної перевірки конкретного алгоритму, а не лише факту підключення ШІ.

У 2025 році Neo et al. окремо дослідили zero-shot, few-shot та fine-tuning LLM для оцінювання Story Points на Deep-SE. У їхньому експерименті fine-tuned модель мала менше MAE, ніж few-shot підхід, у 14 із 16 проектів та перевершила TF-IDF + Linear Regression у 11 із 16 проектів [5]. Окреме дослідження Ahmad та Kuo поєднало RoBERTa і BiLSTM та повідомило про покращення на 4-32 % відносно стану мистецтва на 21 064 задачах [6]. Отже, сучасні результати свідчать про перспективність семантичного подання тексту, але також про потребу у порівняльній та відтворюваній валідації.

Короткий опис дослідження, його методів і засобів

Запропонована система розглядається як багаторівнева інформаційна система для управління проектами. Методологічною базою є системний аналіз, об'єктно-орієнтоване проектування та експериментальне порівняння методів автоматизованого оцінювання.

Архітектура системи передбачає такі основні компоненти:

- Frontend на React.js для роботи з дошкою задач та аналітикою;
- Backend на Node.js/NestJS для бізнес-логіки, авторизації та обміну даними;
- LLM-модуль, доступний через API, для семантичного аналізу та формування прогнозу Story Points;
- PostgreSQL для зберігання задач, користувачів і результатів оцінювання;
- векторне сховище для пошуку семантично близьких історичних задач;
- BullMQ + Redis для асинхронної обробки звернень до LLM-модуля.

У разі використання зовнішнього LLM-сервісу дані, що містять потенційно чутливу інформацію, повинні проходити попереднє очищення або деперсоналізацію перед передачею зовнішньому сервісу. Таким чином, LLM-модуль є функціональним компонентом системи, а не окремим програмним агентом.

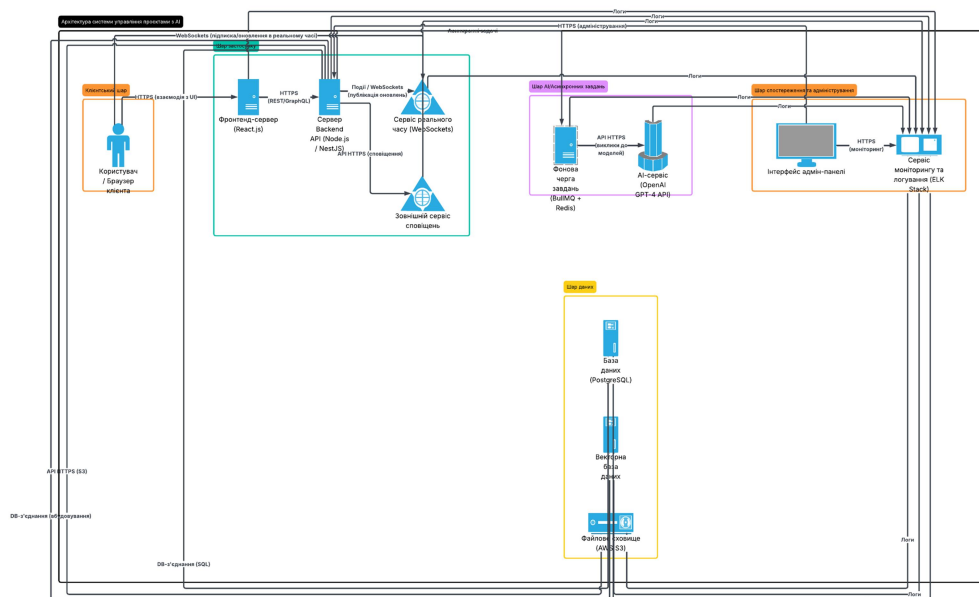


Рис. 1. Архітектура системи управління проектами

Наукову складову дослідження становить алгоритм, який використовує історичні задачі як контекст для оцінювання нової задачі. На відміну від простого запиту до LLM, алгоритм спочатку знаходить релевантні приклади у предметній області поточного проєкту, після чого передає їх до LLM-модуля.

- На етапі попередньої обробки об'єднуються назва та опис задачі, вилучаються HTML-конструкції, URL та технічний шум.

- Для кожної задачі формується семантичне векторне представлення e . Близькість нової задачі x до історичної задачі x_i визначається косинусною схожістю: $(x, x_i) = (e \cdot e_i) / (|e| \cdot |e_i|)$.

- За значенням sim відбираються k найближчих історичних задач. Для них зберігаються фактичні Story Points, тому контекст містить не лише текстові аналоги, а й історичні оцінки.

- У цій експрес-апробації початкова оцінка SP_0 як окремий базовий орієнтир не використовувалася; LLM-модуль безпосередньо отримував текст задачі, три відібрані аналоги та їхні фактичні Story Points.

- LLM-модуль отримує нову задачу, k відібраних аналогів та їхні Story Points. Після цього модель формує остаточну оцінку та коротке пояснення вибору.

- Результат оцінювання округлюється до допустимої шкали Story Points, наприклад 1, 2, 3, 5, 8, 13. Окремий показник упевненості в цій експрес-апробації не використовувався.



Рис. 2. Схема алгоритму семантичного оцінювання задач

Експериментальна методика

Для експрес-апробації використано відкритий набір Deer-SE, який містить 23 313 user stories із 16 open-source проєктів [2]. Публічна версія набору на Hugging Face містить поля project, issuekey, title, description та storypoint, що відповідають структурі, необхідній для оцінювання.

У повному експерименті для кожного проекту дані мають розділятися у часовому порядку: перші 70 % задач використовуються для формування історичного набору, а наступні 30 % - для тестування. У тестовій вибірці модель не повинна мати доступу до фактичних Story Points оцінюваних задач [5]. Для наведеної експрес-апробації використано 15 задач проекту appceleratorstudio та їхні попередні задачі як джерело історичного контексту; результати призначені для первинної перевірки концепції, а не для повної статистичної валідації.

В експрес-апробації порівняно два режими: LLM zero-shot, коли моделі подається лише текст задачі, та гібридний підхід Semantic-KNN + LLM, коли до тексту додаються три семантично близькі історичні задачі з їхніми фактичними Story Points. Для кожної тестової задачі аналогі відбиралися лише серед задач того самого проекту з меншим номером issuekey, причому всі 15 тестових задач були виключені з множини кандидатів. Таким чином, у гібридному режимі не використовувалася інформація з майбутніх задач або фактичні Story Points інших тестових задач. Значення $k = 3$ зафіксовано для однакового протоколу в усіх 15 тестових випадках: воно дає LLM компактний контекст із кількох найближчих аналогів без надмірного збільшення обсягу вхідного тексту. У доступному протоколі цієї експрес-апробації зафіксовано використання GPT-5.6 Luna; відомості про конкретний режим reasoning, значення temperature, дату запуску та кількість повторних запусків у первинному описі експерименту відсутні, тому вони не відновлюються заднім числом.

Основною метрикою є Mean Absolute Error (MAE): $MAE = (\frac{1}{n}) \cdot \sum \nu SP_i - \hat{SP}_i \nu$, де SP_i - фактична оцінка, а $\hat{SP}_i$ - прогноз. Для додаткового аналізу доцільно використовувати Median Absolute Error (MdAE). Нижче наведено контрольні значення з опублікованого порівняння на 16 проєктах Deep-SE; вони використовуються як орієнтир для майбутнього відтворення експерименту у межах цієї роботи [7].

Таблиця 1.

Контрольні результати інших авторів за джерелом [7]

Метод	Середній MAE	Медіанний MAE
GPT2SP	1.44	1.16
Deep-SE	2.09	1.79
FastText + SVM (модель авторів [7])	3.15	2.42

За наведеними опублікованими даними GPT2SP має середній MAE 1,44 та медіанний MAE 1,16, тоді як для Deep-SE відповідні значення становлять 2,09 та 1,79. Для моделі авторів [7] (FastText + SVM) наведено середній MAE 3,15 та медіанний MAE 2,42. Відносна різниця медіанних значень GPT2SP та Deep-SE становить близько 35 %. Ці цифри не є результатами власного запуску системи, а слугують контрольним орієнтиром для майбутньої експериментальної перевірки запропонованого алгоритму [7].

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Проведено експрес-апробацію запропонованого алгоритму на 15 задачах проекту appceleratorstudio з набору Deep-SE. Для кожної задачі отримано прогноз GPT-5.6 Luna у zero-shot режимі та прогноз гібридного підходу з трьома історичними аналогами, відібраними без доступу до майбутніх задач і без використання інших тестових задач. За вибіркою з 15 задач MAE zero-shot становив 1,20, тоді як для гібридного підходу – 1,27. Отже, після усунення витоків даних гібридний підхід не показав переваги: його MAE виявився приблизно на 5,6 % більшим за zero-shot.

Таблиця 2.

Результати експрес-апробації запропонованого підходу на 15 задачах Deep-SE (проект appceleratorstudio) після усунення витоків даних

Задача	Факт. SP	Zero-shot	Hybrid	Середнє SP 3 аналогів
TISTUD-383	8	8	8	1
TISTUD-415	3	3	3	1
TISTUD-434	13	8	13	1
TISTUD-448	13	13	13	1
TISTUD-483	8	8	5	1
TISTUD-486	8	8	5	1
TISTUD-501	13	13	13	1
TISTUD-504	8	8	8	5
TISTUD-509	8	8	8	1
TISTUD-564	3	3	5	4,67
TISTUD-582	5	5	5	7,33
TISTUD-584	21	13	13	1
TISTUD-589	8	8	5	1
TISTUD-600	13	8	13	1
TISTUD-623	8	8	8	1

Як простий базовий метод без LLM використано середнє Story Points трьох відібраних аналогів; його MAE на тих самих 15 задачах становив 7,93, що суттєво гірше за zero-shot (1,20) та Hybrid (1,27), тому на цій вибірці саме додавання LLM, а не лише використання трьох аналогів, дало значно нижчу абсолютну похибку.

Детальна таблиця експрес-апробації показує, що використання історичного контексту не забезпечує стабільного покращення на цій малій вибірці. Для частини задач гібридний підхід відтворює точну оцінку, однак для окремих задач із графічним інтерфейсом, навігацією та імпортом проєктів помилка збільшується. Це підкреслює, що якість результату залежить не лише від наявності історичних аналогів, а й від їхньої релевантності та коректного часового відбору.

Отриманий результат розглядається як первинна емпірична перевірка запропонованої ідеї. Для повноцінного дослідження необхідно провести повторний експеримент на всіх 16 проєктах із часовим розділенням,

зафіксованими параметрами пошуку аналогів, однаковим протоколом виключення тестових задач та кількома запусками LLM.

ВИСНОВКИ

Формалізовано гібридний алгоритм оцінювання Story Points, що включає попередню обробку тексту, семантичне векторне представлення, пошук трьох найбільш схожих історичних задач та контекстне уточнення оцінки LLM-модулем. У гібридному режимі аналоги для кожної тестової задачі відбиралися лише серед задач того самого проєкту з меншим номером `issuekey`, без включення 15 тестових задач до множини кандидатів, що забезпечує відсутність часового та перехресного витоку даних.

На вибірці з 15 задач проєкту `appceleratorstudio` MAE zero-shot становив 1,20, MAE hybrid - 1,27, а простого baseline у вигляді середнього Story Points трьох аналогів - 7,93. Отже, у цій експрес-апробації гібридний підхід не продемонстрував переваги над zero-shot: його MAE був приблизно на 5,6 % більшим, хоча обидва підходи з LLM суттєво перевершили простий baseline. Наукова новизна роботи полягає у поєднанні семантичного пошуку історичних аналогів із LLM для контекстного оцінювання Story Points. Водночас на вибірці з 15 задач статистично значущих висновків про перевагу одного підходу над іншим зробити не можна; для підтвердження результату потрібне відтворення експерименту на більшій вибірці та на всіх проєктах Deep-SE.

ДЖЕРЕЛА

1. USMAN M., MENDES E., WEIDT F., BRITTO R. Effort estimation in agile software development: a systematic literature review. *Proceedings of the 10th International Conference on Predictive Models in Software Engineering*. New York : ACM, 2014. P. 82–91. DOI: 10.1145/2639490.2639503.
2. CHOETKIERTIKUL M., DAM H. K., TRAN T., PHAM T., GHOSE A., MENZIES T. A deep learning model for estimating story points. *IEEE Transactions on Software Engineering*. 2019. Vol. 45, No. 7. P. 637–656. DOI: 10.1109/TSE.2018.2792473.
3. FU M., TANTITHAMTHAVORN C. GPT2SP: A Transformer-Based Agile Story Point Estimation Approach. *IEEE Transactions on Software Engineering*. 2023. Vol. 49, No. 2. P. 611–625. DOI: 10.1109/TSE.2022.3158252.
4. TAWOSI V., MOUSSA R., SARRO F. Agile Effort Estimation: Have We Solved the Problem Yet? Insights From a Replication Study. *IEEE Transactions on Software Engineering*. 2023. Vol. 49, No. 4. P. 2677–2697. DOI: 10.1109/TSE.2022.3228739.
5. NEO G. da S., MOURA J. A. B., NEO A. V. B. da S., FREITAS JÚNIOR O. de G. Effort Estimation in Story Points from User Stories with Large Language Models. *Anais do XXXIX Simpósio Brasileiro de Engenharia de Software (SBES 2025)*. 2025. P. 720–726. DOI: 10.5753/sbes.2025.11121. (Стаття португальською мовою).
6. AHMAD Z., KUO M. M. Y. Improving story points estimation using ensemble machine learning. *Software Quality Journal*. 2025. Vol. 33, No. 4. Art. 35. DOI: 10.1007/s11219-025-09731-6.
7. ATOUM I., OTOOM A. A. Enhancing Software Effort Estimation with Pre-Trained Word Embeddings: A Small-Dataset Solution for Accurate Story Point Prediction. *Electronics*. 2024. Vol. 13, No. 23. Art. 4843. DOI: 10.3390/electronics13234843.