

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»
МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

Кваліфікаційна наукова праця
на правах рукопису

РОГОВИЙ МИКИТА АНТОНОВИЧ

УДК 004.89

ДИСЕРТАЦІЯ

**МОДЕЛІ ТА МЕТОДИ ІНТЕЛЕКТУАЛЬНОГО ПЛАНУВАННЯ РОБОТИ
КОМАНДИ ІТ ПРОЄКТУ НА ОСНОВІ ГНУЧКОЇ МЕТОДОЛОГІЇ**

122 – Комп'ютерні науки
12 – Інформаційні технології

Подається на здобуття наукового ступеня доктор філософії

Дисертація містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

М.Роговий М.А. Роговий

Наукова керівниця:
Гринченко Марина Анатоліївна,
кандидат технічних наук, доцент

Харків – 2025

АНОТАЦІЯ

Роговий М.А. Моделі та методи інтелектуального планування роботи команди ІТ проєкту на основі гнучкої методології. - Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктор філософії (PhD) за спеціальністю 122 «Комп'ютерні науки». - Національний технічний університет «Харківський політехнічний інститут» Міністерства освіти і науки України, м. Харків, 2025.

У дисертаційній роботі вирішена науково-практична задача створення моделей та методів інтелектуального планування роботи команди ІТ проєкту на основі гнучкої методології, яке має істотне значення для розвитку інформаційних технологій.

Об'єкт дослідження – процеси планування роботи команди ІТ проєкту.

Предмет дослідження – моделі та методи інтелектуального планування роботи команди ІТ проєкту на основі гнучкої методології.

Метою дисертаційної роботи є підвищення якості роботи команди ІТ-проєкту шляхом створення і застосування інтелектуального планування роботи команди ІТ проєкту на основі гнучкої методології з інтеграцією штучного інтелекту в планування спринтів, розподіл задач та управління ризиками.

При вирішенні задач дисертаційного дослідження використані: методи обробки природної мови та методи великих мовних моделей (LLM), методи математичної статистики, методи класифікації текстових описів, методи векторної репрезентації текстів, методи оцінювання узгодженості думок експертів, методи семантичного аналізу тексту, класифікатор Байєса та нейронні класифікатори для класифікації текстових виразів, методи PCA та t-SNE для наочного відображення результатів експериментів, алгоритми розподілу задач в ІТ-командах, що працюють за гнучкими методологіями розробки (EADAM та SOSM), методи штучного інтелекту. Для тренування нейронної мережі на основі векторизованої репрезентації тексту використано претреновану модель BERT (bert-base-uncased) з пакету Python для моделей класу «трансформери».

У вступі обґрунтовано актуальність теми дослідження, яка пов'язана з необхідністю підвищення ефективності планування роботи команди ІТ- проєкту на основі гнучкої методології. Визначено основну мету дослідження – розробку моделей та методів інтелектуального планування, що дозволять оптимізувати розподіл задач, управління ризиками та підвищити продуктивність команди. Сформульовано основні завдання роботи, сформульовано наукову новизну та практичну значущість отриманих результатів та особистий внесок здобувача. Наведено відомості про апробацію матеріалів дисертації та публікації, структуру та обсяг дисертації.

У першому розділі проведено аналіз сучасного стану досліджень у сфері планування роботи команд ІТ-проєктів на основі гнучких методологій, таких як Scrum, Kanban та їх модифікацій. Розглянуто існуючі моделі та методи планування роботи команди, що дозволяють ефективно керувати завданнями та контролювати виконання спринтів. Окрему увагу приділено аналізу трекінгових систем, що використовуються в управлінні ІТ-проєктами для ведення обліку задач, витраченого часу, коментарів, виправлення помилок та інших аспектів, що впливають на продуктивність команди. Також досліджено підходи до аналізу текстових даних, які можуть використовуватися для покращення формулювання задач, їх класифікації та групування. Проведено аналіз методів оцінки ризиків у процесі планування та управління ІТ-проєктами. На основі аналізу існуючих рішень виявлено основні недоліки та сформульовано постановку задачі дослідження.

У другому розділі представлено теоретичні основи вирішення поставленої задачі. Запропоновано концептуальну модель (фреймворк) інтелектуального планування роботи команди, що включає аналіз текстових описів задач, оцінку їх складності та взаємозв'язків. Розроблено метод покращення текстових описів задач спринту, який дозволяє стандартизувати їх та підвищити зрозумілість для команди. Запропоновано модель кластеризації текстових описів задач, що дає змогу автоматично групувати схожі завдання, оптимізуючи їх розподіл між членами команди. Розглянуто моделі розподілу задач у спринті, що враховують

компетенції розробників, їхню завантаженість, складність задач та можливі ризики.

У третьому розділі розглянуто методологічні засади розробки інтелектуальної системи підтримки прийняття рішень для управління ІТ-проєктами в умовах невизначеності та ризиків. Запропоновано рекомендаційну систему для планування роботи команди, яка аналізує історичні дані та продуктивність команди для формування оптимальних рекомендацій щодо розподілу задач. Розроблено інформаційну технологію інтелектуального планування роботи команди ІТ-проєкту, яка використовує алгоритми обробки текстових описів задач, кластеризацію та машинне навчання для адаптивного планування. Визначено принципи інтеграції розробленої технології з існуючими трекінговими системами, що дозволяє підвищити автоматизацію процесів управління проєктом.

У четвертому розділі представлено результати експериментальних досліджень, що підтверджують ефективність запропонованих моделей та методів. Проведено тестування алгоритмів обробки текстових описів задач, оцінено їхній вплив на якість формулювання задач та рівень їх розуміння командою. Виконано оцінку запропонованих методів кластеризації та розподілу задач, що дозволило встановити підвищення точності та швидкості прийняття рішень. Проведено експериментальне впровадження розроблених підходів у реальних ІТ-командах, що дозволило оцінити їх вплив на ефективність планування, швидкість виконання спринтів та управління ризиками.

У висновках узагальнено результати дослідження, представлено основні теоретичні та практичні досягнення роботи. Окреслено перспективи подальшого розвитку запропонованих моделей та методів, а також можливості їхнього впровадження в сучасні трекінгові системи для підвищення ефективності управління ІТ-проєктами.

За результатами дослідження отримано такі наукові результати:

1 вперше розроблено модель інтелектуального планування роботи команди ІТ проєкту, яка дозволяє підвищити цінність спринту враховуючі

ризика та дефекти за рахунок визначення та розподілу задач спринту на основі пріоритетів виконавців та оцінки чіткості змісту задач з використанням моделей штучного інтелекту;

2 вперше розроблено метод покращення текстових описів задач спринту, який дозволяє підвищити точність розуміння завдання виконавцем та знизити ризики невиконання задач за рахунок використання моделей машинного навчання та великих мовних моделей (LLM) для оцінки чіткості та вдосконалення тексту;

3 набув подальшого розвитку метод планування роботи команди IT проєкту за рахунок комплексного підходу до визначення рекомендацій щодо розподілу, вибору та оцінки задач на основі інтелектуальної обробки ретроспективних та текстових даних та контексту проєкту, що, на відміну від існуючих підходів, дозволяє врахувати невизначеність та ризики, притаманні проєктам з гнучкою методологією управління та знизити рівень дефектів;

4 Удосконалено модель розподілу задач спринту між виконавцями на основі моделі визначення стабільних зіставлень, що, на відміну від існуючих підходів, враховує специфіку гнучкої методології управління проєктами та дозволяє знизити ризики невиконання спринту за рахунок підвищення узгодженості призначень та задоволеності виконавців.

Практичне значення отриманих результатів полягає у наступному:

- оптимізовано процеси управління задачами в Agile-проєктах: інтеграція розробленої інформаційної системи з JIRA та налаштування автоматичного аналізу беклогу забезпечують безперервні рекомендації щодо пріоритезації й розподілу задач, підвищуючи загальну продуктивність команди;

- скорочено час планування спринту на 30-40 %: автоматизоване формулювання та призначення задач істотно зменшує витрати часу Scrum-майстра й розробників на підготовку та затвердження плану робіт;

- підвищено точність призначення виконавець-задача, що зменшує кількість переназначень і змін задач приблизно на 20 %; це знижує ризики перевантаження окремих фахівців і мінімізує затримки у виконанні спринту;

– покращено командну комунікацію: система генерує ясніші, структуровані формулювання user-story та критеріїв приймання, що скорочує кількість уточнень і непорозумінь під час розробки та сприяє більшій погодженості дій між розробниками, тестувальниками й аналітиками.

Ключові слова: AGILE, проєкт, ризик, модель системи; управління проєктами, підтримка прийняття рішень, обробка природної мови, команда проєкту, інформаційна система, інформаційна технологія, метод, опис задач, призначення задач

Список публікацій здобувача

Публікації здобувача за темою дисертації, в яких опубліковані основні наукові результати:

1 Rohovyi M., Grinchenko M. Towards the improvement of project team performance based on large language models. *Radioelectronic and computer systems*, 2024. №. 4 (112). P. 229–247. (Scopus, Україна, А)

2 Grinchenko M., Rohovyi M. A model for identifying project sprint tasks based on their description. *Innovative technologies and scientific solutions for industries*. Kharkiv, 2023. №4 (26). P. 33-44. (Б)

3 Rohovyi M., Grinchenko M. Project team management model under risk conditions. *Вісник Національного технічного університету «ХПІ». Серія: Стратегічне управління, управління портфелями, програмами та проєктами*. Харків, 2023. № 1 (7). С. 3–11. (Б)

4 Rohovyi M., Grinchenko M. Comparative Analysis of stable matching algorithms for intelligent work planning of IT teams. *Біоніка інтелекту*. Харків: ХНУРЕ, 2024. №2 (101). С. 56–63. (Б)

5 Гринченко М.А., Роговий М.А. Розробка інформаційної технології інтелектуального планування роботи команди ІТ проєкту на основі гнучкої методології. *Вісник Національного технічного університету «ХПІ». Серія: Стратегічне управління, управління портфелями, програмами та проєктами*. Харків, 2024. № 2 (9). С. 9–15. (Б)

Опубліковані праці апробаційного характеру:

6 Гринченко М.А., Роговий М.А. Підхід щодо класифікації текстового опису задач спринту проєкту. *Управління проектами у розвитку суспільства*: тези доповідей XX Міжнародної конференції, (м. Київ, 12 травня 2023 р.). – Київ: КНУБА, 2023. – С.86–90.

7 Гринченко М.А., Роговий М.А. Ідентифікація опису задач спринту проєкту. *Інтегроване стратегічне управління, управління портфелями, програмами, проєктами*: тези доповідей XIV Міжнародної науково-практичної конференції, (м. Харків, 13-14 лютого 2024 р.). – Харків: НТУ «ХПІ», 2024. – С. 33.

8 Grinchenko M., Rohovyi M. Intelligent decision support system in the face of uncertainty and risks in project management. *Інформаційні системи та технології*: тези доповідей 13-ї Міжнародної науково-технічної конференції, (м.Харків, 26 – 28 листопада 2024р). Харків, 2024. – Р. 237–240.

9 Grinchenko M., Rohovyi M. System of recommendations for the it project team in the face of risks. *Scientific trends in the development of education in universities*: proceedings of the 17th International scientific and practical conference, (Athens, Greece, 24 – 27 Desember 2024). Athens, Greece:International Science Group, 2024. P. 237–240

ABSTRACT

Rohovyi M. A. Models and Methods of Intelligent Planning of IT Project Team Work Based on Agile Methodology. Qualifying scientific work on the rights of the manuscript.

Dissertation for the degree of Doctor of Philosophy (PhD) in specialty 122 "Computer Science". - National Technical University "Kharkiv Polytechnic Institute", Kharkiv, 2025.

This dissertation solves the scientific and practical task of developing models and methods for intelligent planning of IT project team work based on Agile methodology, which is essential for the advancement of information technologies.

Object of research: processes of planning the work of an IT project team.

Subject of research: models and methods of intelligent planning of IT project team work based on Agile methodology.

The purpose of the dissertation is to improve the quality of IT project team performance by creating and applying intelligent planning of team work based on Agile methodology with the integration of artificial intelligence into sprint planning, task allocation, and risk management.

The introduction substantiates the relevance of the research topic, which is related to the need to improve the efficiency of IT project team planning based on Agile methodology. The main goal of the research is defined — the development of models and methods of intelligent planning that will optimize task distribution, risk management, and increase team productivity. The main tasks of the work, scientific novelty, practical significance of the obtained results, and the author's personal contribution are formulated. Information on the approbation of dissertation materials, publications, structure, and volume of the dissertation is provided.

Chapter One analyzes the current state of research in the field of IT project team planning based on Agile methodologies such as Scrum, Kanban, and their modifications. Existing models and methods of team work planning are considered, allowing effective task management and sprint execution control. Particular attention is paid to the analysis of tracking systems used in IT project management to track tasks, time spent, comments, bug fixing, and other factors affecting team productivity. Approaches to text data analysis that can be used to improve task formulation, classification, and grouping are also explored. Risk assessment methods in IT project planning and management are analyzed. Based on the analysis of existing solutions, the main shortcomings were identified and the research problem statement was formulated.

Chapter Two presents the theoretical foundations for solving the research problem. A conceptual framework for intelligent team work planning is proposed, including analysis of task text descriptions, assessment of their complexity and interrelations. A method for improving sprint task descriptions is developed, allowing

their standardization and increasing clarity for the team. A model for clustering task text descriptions is proposed, enabling automatic grouping of similar tasks to optimize their distribution among team members. Sprint task distribution models are considered, taking into account developers' competencies, workload, task complexity, and potential risks.

Chapter Three covers the methodological principles of developing an intelligent decision support system for IT project management under uncertainty and risk. A recommendation system for team work planning is proposed, analyzing historical data and team performance to form optimal task distribution recommendations. An information technology for intelligent planning of IT project team work is developed, using algorithms for task text processing, clustering, and machine learning for adaptive planning. Principles of integrating the developed technology with existing tracking systems are defined, which enhances project management automation.

Chapter Four presents the results of experimental research confirming the effectiveness of the proposed models and methods. Testing of task text processing algorithms was conducted, assessing their impact on task formulation quality and team understanding. The proposed clustering and task distribution methods were evaluated, showing increased accuracy and decision-making speed. Experimental implementation of the developed approaches in real IT teams made it possible to assess their impact on planning efficiency, sprint execution speed, and risk management.

The conclusions summarize the research results, presenting the main theoretical and practical achievements of the work. Prospects for further development of the proposed models and methods, as well as their potential implementation in modern tracking systems to improve IT project management efficiency, are outlined.

Scientific results obtained in the research:

1 for the first time, a model of intelligent planning of IT project team work has been developed, which allows increasing sprint value by considering risks and defects through determining and distributing sprint tasks based on developer priorities and clarity assessment using artificial intelligence models.

2 for the first time, a method for improving sprint task text descriptions has been developed, allowing for increased task understanding accuracy and reduced risks of non-fulfillment through the use of machine learning models and large language models (LLM) for clarity assessment and text enhancement.

3 the method of IT project team work planning has been further developed through a comprehensive approach to forming recommendations for task distribution, selection, and evaluation based on intelligent processing of retrospective and textual data and project context, which, unlike existing approaches, allows for consideration of uncertainties and risks inherent in Agile projects and reduces defect levels.

4 the model of sprint task distribution among performers has been improved based on the stable matching model, which, unlike existing approaches, considers the specifics of Agile project management methodology and allows reducing the risk of sprint non-fulfillment by improving assignment consistency and developer satisfaction.

Practical significance of the obtained results includes:

- optimization of task management processes in Agile projects: integration of the developed information system with JIRA and automatic backlog analysis configuration provides continuous recommendations for task prioritization and distribution, increasing overall team productivity;

- reduction of sprint planning time by 30–40%: automated task formulation and assignment significantly reduce Scrum master and developer time spent on plan preparation and approval;

- improved accuracy of developer-task assignment, reducing the number of reassignments and task changes by approximately 20%; this decreases the risk of overloading individual specialists and minimizes delays in sprint execution;

- enhanced team communication: the system generates clearer, structured user stories and acceptance criteria, reducing the number of clarifications and misunderstandings during development and promoting greater coordination between developers, testers, and analysts.

Keywords: AGILE, project, risk, system model, project management, decision support, natural language processing, project team, information system, information technology, method, task description, task assignment.

List of the applicant's publication

Publications of the applicant on the topic of the dissertation, in which the main scientific results are published:

1 Rohovyi M., Grinchenko M. Towards the improvement of project team performance based on large language models. *Radioelectronic and computer systems*, 2024. №. 4 (112). P. 229–247. (Scopus, Україна, А)

2 Grinchenko M., Rohovyi M. A model for identifying project sprint tasks based on their description. *Innovative technologies and scientific solutions for industries*. Kharkiv, 2023. №4 (26). P. 33-44. (B)

3 Rohovyi M., Grinchenko M. Project team management model under risk conditions. "Bulletin of the National Technical University 'KhPI'. Series: Strategic Management, Portfolio, Program, and Project Management. Kharkiv", 2023. № 1 (7). P. 3–11. (Б)

4 Rohovyi M., Grinchenko M. Comparative Analysis of stable matching algorithms for intelligent work planning of IT teams. *Bionics of intelligence*. Kharkiv: KNURE, 2024. №2 (101). P. 56–63. (B)

5 Grinchenko M., Rohovyi M. Development of Information Technology for Intelligent Planning of IT Project Team Work Based on Agile Methodology "Bulletin of the National Technical University 'KhPI'. Series: Strategic Management, Portfolio, Program, and Project Management. Kharkiv", 2024. № 2 (9). P. 9–15. (Б)

Publications of the applicant on the topic of the dissertation, which certify the approbation of the dissertation materials:

6 Grinchenko M., Rohovyi M. An Approach to Classifying Project-Sprint Task Descriptions. *Project Management in the Development of Society: Proceedings of the*

20th International Conference, (Kyiv, 12.05.2023). – Kyiv: KNUCA, 2023. – P.86–90.

7 Grinchenko M., Rohovyi M. Identification of Project-Sprint Task Descriptions.” Integrated Strategic Management, Portfolio, Programme and Project Management: Proceedings of the 14th International Scientific and Practical Conference, (Kharkiv, 13-14 February 2024). – Kharkiv: NTU «KPI», 2024. – P. 33.

8 Grinchenko M., Rohovyi M. Intelligent decision support system in the face of uncertainty and risks in project management. Information Systems and Technologies: Proceedings of the 13th International Scientific and Technical Conference, (Kharkiv, 26 – 28 November 2024). Kharkiv, 2024. – P. 237–240.

9 Grinchenko M., Rohovyi M. System of recommendations for the it project team in the face of risks. Scientific trends in the development of education in universities: proceedings of the 17th International scientific and practical conference, (Athens, Greece, 24 – 27 December 2024). Athens, Greece: International Science Group, 2024. P. 237–240.

ЗМІСТ

ВСТУП	4
РОЗДІЛ 1 АНАЛІЗ ПІДХОДІВ ТА МЕТОДІВ ПЛАНУВАННЯ РОБОТИ КОМАНДИ ІТ ПРОЄКТУ	12
1.1 Особливості планування роботи команди ІТ проєкту на основі гнучкої методології	12
1.2 Підходи до зменшення впливу ризиків на роботи команди ІТ проєкту	13
1.2.1 Методи оцінки поведінки роботи команди.....	13
1.2.2 Методи оцінки впливу опису задач на результати роботи команди проєкту	15
1.2.3 Методи оцінки сценаріїв управління ризиками при виконанні задач проєкту	22
1.2.4 Методи оцінки розподілу задач серед виконавців	24
1.3 Постановка задачі дослідження.....	29
1.4 Висновки до розділу 1	33
РОЗДІЛ 2 МОДЕЛІ ТА МЕТОДИ ІНТЕЛЕКТУАЛЬНОГО ПЛАНУВАННЯ РОБОТИ КОМАНДИ ІТ ПРОЄКТУ	35
2.1 Опорна модель управління проєктною командою в умовах ризиків	35
2.2 Модель задачі інтелектуального планування роботи команди.....	41
2.3 Метод інтелектуального планування роботи команди ІТ проєкту.....	48
2.4 Модель розподілу задач спринту	57
2.5 Висновки до розділу 2.....	65
РОЗДІЛ 3 АНАЛІЗ ТЕКСТОВОГО ОПИСУ ЗАДАЧ СПРИНТУ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ	67
3.1 Особливості аналізу текстового опису задач проєкту	67
3.2 Метод покращення текстових описів задач спринту проєкту	73
3.3 Інформаційна технологія інтелектуального планування роботи команди ІТ проєкту.....	78
3.4 Висновки до розділу 3.....	84

РОЗДІЛ 4 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ МОДЕЛЕЙ ТА МЕТОДІВ ІНТЕЛЕКТУАЛЬНОГО ПЛАНУВАННЯ РОБОТИ КОМАНДИ ІТ-ПРОЄКТУ	86
4.1 Організація експериментального дослідження моделей та методів інтелектуального планування.....	86
4.2 Дослідження чіткості сприйняття текстового опису задач.....	90
4.3 Дослідження результатів розподілу задач на спринт.....	117
4.4 Впровадження результатів.....	124
4.5 Висновки до розділу 4	127
ВИСНОВКИ	130
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	133
ДОДАТОК А	146
Документи впровадження основних результатів дисертаційної роботи	146
ДОДАТОК Б.....	151
Список публікацій здобувача	151

ВСТУП

Обґрунтування вибору теми дослідження. Гнучкі методології розробки програмного забезпечення (Agile, зокрема фреймворки Scrum і Kanban) наразі стали стандартом управління сучасними ІТ-проєктами. Водночас процес планування спринтів і прийняття рішень у Agile-командах досі, значною мірою, здійснюється вручну та ґрунтується на досвіді й суб'єктивних думках учасників команди. Це породжує низку проблем: зокрема, залежність від інтуїтивних оцінок може призводити до неефективного управління ризиками та не оптимального розподілу ресурсів у проєкті. Якість формулювання вимог та задач (user stories) у беклозі також суттєво впливає на успішність проєкту. Описані природною мовою вимоги часто містять неоднозначності чи суперечності, ускладнюючи розуміння задач командою та процес їх планування. Такі дефекти в описах задач можуть спричиняти хибне трактування вимог, що підвищує ризики появи помилок та дефектів у продукті. Додатково, актуальним є завдання оптимального розподілу роботи між членами команди: неналежний або нерівномірний розподіл призводить до перевантаження окремих фахівців і зниження якості колективної роботи. Первинні невдачі у продуктивності команди часто обумовлені саме виснаженням перевантажених розробників та нераціональним розподілом задач, що негативно впливає на якість продукту.

Необхідність підвищення ефективності командної роботи в Agile-проєктах зумовлює потребу в нових підходах до планування – інтелектуального планування, яке базується на аналізі даних та знань, а не лише на інтуїції менеджера. Сучасні досягнення штучного інтелекту, зокрема методи машинного навчання та великі мовні моделі (LLM), відкривають можливості для автоматизації планування. Інтеграція ШІ в процес Agile дозволяє підвищити обґрунтованість рішень. Зокрема, дослідження підтверджують, що AI-орієнтовані системи підтримки прийняття рішень здатні ефективно виявляти ризики та оптимізувати розподіл ресурсів у проєктах. Крім того, використання NLP-інструментів на основі LLM дає змогу автоматично аналізувати текстові

описи задач та рекомендувати їх покращення, що підвищує їх зрозумілість для команди. Перспективним напрямом є також застосування алгоритмів стабільного зіставлення (stable matching) для автоматизованого розподілу задач у команді. Формулювання задачі призначення роботи як задачі пошуку стабільної відповідності дозволяє врахувати пріоритети (вподобання) виконавців і властивості задач та уникнути ситуацій, коли учасники залишаються незадоволеними розподілом. Таким чином, актуальною науковою та практичною проблемою є розробка моделей та методів інтелектуального планування роботи ІТ-проектної команди на основі Agile. Вирішення цієї проблеми сприятиме підвищенню якості планування та продуктивності команд, зменшенню кількості дефектів і ризиків за рахунок кращого формулювання та розподілу задач із залученням засобів штучного інтелекту.

Зв'язок роботи з науковими програмами, планами, темами. Дослідження, результати яких викладені в дисертаційній роботі, виконані на кафедрі «Управління проектами в інформаційних технологіях» НТУ «Харківський політехнічний інститут» відповідно НДР за темою «Моделі та методи управління проектами розвитку організаційних систем» по розділу «Метод вибору методології управління проектом» (ДР № 0124U001735), в якій здобувач брав участь у якості виконавця.

Мета і задачі дослідження. Метою даної дисертаційної роботи є зниження рівня дефектів виконання робіт проекту за рахунок розробки моделей та методів інтелектуального планування роботи команди. Це дозволить підвищити якість планування роботи команди ІТ-проекту шляхом розробки і застосування моделей штучного інтелекту для підтримки планування в умовах гнучкої методології. Інтеграція методів ШІ у процес планування спринтів, розподіл задач і управління ризиками сприятиме оптимізуванню командної роботи та мінімізації впливу людських факторів на виникнення помилок.

Для досягнення поставленої мети в дисертації визначено й вирішено комплекс взаємопов'язаних задач. Зокрема:

- дослідити чинники зростання рівня дефектів виконання робіт проєкту та ризики невиконання робіт спринту;
- розробити модель планування роботи команди ІТ проєкту, яка враховує пріоритети та оцінки змісту задач, та дозволяє формувати рекомендації щодо вибору та розподілу задач спринту;
- розробити методи для оцінки та вдосконалення текстових описів задач спринту на основі використання методів машинного навчання та великих мовних моделей;
- розробити моделі розподілу задач спринту між виконавцями, що враховують специфіку гнучкої методології управління проєктами та дозволяють оптимізувати роботу команди;
- розробити інформаційну технологію підтримки планування роботи команди ІТ проєкту;
- провести апробацію ефективності запропонованих моделей та методів у реальних командах розробників

Об’єкт дослідження – процеси планування роботи команди ІТ-проєкту в умовах використання гнучких методологій розробки.

Предмет дослідження – моделі та методи інтелектуального планування роботи команди ІТ-проєкту на основі гнучкої методології.

Методи дослідження.

- методи машинного навчання і обробки природної мови (Natural Language Processing), включно з великими мовними моделями для аналізу та генерації тексту.
- підходи до класифікації і кластеризації текстових даних (для автоматичної категоризації та групування задач за схожістю описів), методи семантичного аналізу та векторного подання текстів (наприклад, за допомогою претренованих трансформерних моделей BERT).
- методи експертних оцінок та узгодження думок, для врахування експертних знань (для оцінки складності задач, пріоритезації тощо).

- алгоритми стабільного розподілу (stable matching) для розробки моделі розподілу задач. Зокрема, обрано алгоритми EADAM та SOSM, які, згідно з результатами порівняльного аналізу, демонструють збалансовану стабільність, ефективність і високий рівень задоволеності учасників розподілом.

- методи математичної статистики (оцінка кореляцій, критерії значущості) для статистичного аналізу даних та візуалізації результатів, а також методи зниження розмірності (PCA, t-SNE) з метою наочного представлення розподілу задач і результатів кластеризації.

- імітаційне моделювання, комп'ютерні моделі для перевірки розроблених моделей, алгоритмів та експериментального дослідження на тестових і реальних даних, яке відтворює процес планування спринту та дозволяє оцінити вплив запропонованих підходів на ключові показники (якість описів задач, збалансованість розподілу, кількість допущених дефектів тощо).

Наукова новизна отриманих результатів.

Наукова новизна отриманих результатів полягає у розробці нових моделей, методів і технологій інтелектуального планування командної роботи в Agile-проектах, а саме:

- **вперше** розроблено модель інтелектуального планування роботи команди ІТ проєкту, яка дозволяє підвищити цінність спринту враховуючі ризики та дефекти за рахунок визначення та розподілу задач спринту на основі пріоритетів виконавців та оцінки чіткості змісту задач з використанням моделей штучного інтелекту;

- **вперше** розроблено метод покращення текстових описів задач спринту, який дозволяє підвищити точність розуміння задачі виконавцем та знизити ризики невиконання задач за рахунок використання моделей машинного навчання та великих мовних моделей (LLM) для оцінки чіткості та вдосконалення тексту;

- **набув подальшого розвитку** метод планування роботи команди ІТ проєкту за рахунок комплексного підходу до визначення рекомендацій щодо розподілу, вибору та оцінки задач на основі інтелектуальної обробки

ретроспективних та текстових даних та контексту проєкту, що, на відміну від існуючих підходів, дозволяє врахувати невизначеність та ризики, притаманні проєктам з гнучкою методологією управління та знизити рівень дефектів;

- **удосконалено** модель розподілу задач спринту між виконавцями на основі моделі визначення стабільних зіставлень, що, на відміну від існуючих підходів, враховує специфіку гнучкої методології управління проєктами та дозволяє знизити ризики невиконання спринту за рахунок підвищення узгодженості призначень та задоволеності виконавців.

Практична значимість отриманих результатів полягає у можливості практичного застосування результатів дисертаційного дослідження як в ІТ-індустрії, так і в освітній сфері. Зокрема, використання розробленої інформаційної системи дозволить проєктним командам більш ефективно проводити спринт-планування, завдяки чому відбувається підвищення продуктивності команди, а саме:

- оптимізовано процеси управління задачами та підвищення продуктивності команди за рахунок інтеграції системи з JIRA, включаючи налаштування автоматичного аналізу та рекомендацій;
- скорочено час на постановку задач та їх розподіл на 30-40%.
- підвищено точність відповідності задачі виконавцю, що зменшує кількість змін на 20%;
- покращено комунікацію в команді завдяки яснішим формулюванням задач.

Результати досліджень впроваджено (Додаток А):

- в систему планування робіт ТОВ «ДОМІНЕТ» (м.Ірпінь);
- в навчальний процес кафедри «Управління проєктами в інформаційних технологіях» Національного технічного університету «ХПІ».

Особистий внесок здобувача. полягає в розробленні нових та удосконаленні існуючих моделей та методів планування роботи команди ІТ проєкту, що забезпечують підвищення цінності спринту враховуючі ризики та

дефекти, знизити ризики невиконання задач за рахунок використання моделей машинного навчання та великих мовних моделей (Додаток А):

1) фахові видання України:

[1] – здобувачем розроблено технологію дослідження лінгвістичних характеристик задач спринту, виконано збір та підготовка текстових описів задач спринту, проведено валідацію технології, проаналізовано результати. Співатором Гринченко М.А. – обрано концепцію дослідження, сформульовано завдання;

[2] – здобувачем запропоновано комплексний підхід використання алгоритмів машинного навчання для класифікації та залучення експертних оцінок з метою вдосконалення якості сприйняття задач командою проєкту, проведено класифікацію текстових виразів на основі класифікатора Байєса та нейронних класифікаторів. Реалізовано візуальну репрезентацію даних, проведено семантичний аналіз тексту опису та назви задач. Проведено навчання нейронної мережі за допомогою мови програмування Python та фреймворків для глибокого навчання. Співатором Гринченко М.А. – концептуалізація, формулювання завдання, сформульовано ідею підходу;

[3] – здобувачем визначено процеси роботи команди при виконанні спринту проєкту, запропоновано комплексну еталонну модель управління проєктною командою в умовах ризиків у вигляді фреймворку. Співатором Гринченко М.А. – виділено фактори ризику, які впливають на ефективність роботи команди ІТ проєкту, проведено аналіз сучасних підходів управління командою проєкту в умовах ризику;

[4] – здобувачем створено симуляційне середовище для моделювання процесу розподілу задач, генерації синтетичних даних, реалізації та оцінки п'яти ключових алгоритмів: SOSM, EADAM, MESMA, RSD та TTC. Співатором Гринченко М.А. – проведено огляд існуючих підходів до розподілу задач у проєктних командах та обґрунтовано актуальність дослідження алгоритмів стабільного зіставлення;

[5] – здобувачем розроблено інформаційну технологію для інтелектуального планування роботи команди ІТ-проєкту, засновану на принципах гнучкої методології. Співатором Гринченко М.А. – концептуалізація, проєктування архітектури інформаційної технології.

Б) інші публікації:

[6] – здобувачем проведено інтеграцію в систему контролю виконання задач спринту моделі, яка виконує класифікацію якості та однозначності текстового опису задач для прогнозування результатів виконання проєкту;

[7] – здобувачем проведено детальний аналіз лінгвістичних особливостей текстів задач проєктів, включаючи синтаксичну структуру, термінологію, стилістичні відмінності та контекстуальну цілісність;

[8] – здобувачем визначено та проаналізовано ключові фактори, що впливають на ефективність роботи команди ІТ проєкту;

[9] – здобувачем проведено аналіз застосування алгоритмів Gale-Shapley student optimal stable mechanism (SOSM) та efficiency-adjusted deferred acceptance mechanism (EADAM) для покращення процесів розподілу задач спринту.

Апробація результатів. Основні положення дисертаційної роботи були представлені та обговорені на: XX Міжнародна конференція «Управління проєктами у розвитку суспільства» (м. Київ, 2023 р.), XIV Міжнародна науково-практична конференція «Інтегроване стратегічне управління, управління портфелями, програмами, проєктами», (м. Харків, 2024 р.), 13-а Міжнародна науково-технічна конференція «Інформаційні системи та технології» (м.Харків, 2024р), 17th International scientific and practical conference «Scientific trends in the development of education in universities» (Greece, 2024).

Публікації. Основні положення і результати дисертаційних досліджень, які виносяться на захист, опубліковані в 5 наукових працях: 5 статей – у наукових періодичних фахових видання України (категорія А – 1, Б – 4); 4 матеріалів апробаційного характеру.

Структура і обсяг дисертації. Дисертаційна робота складається з анотації двома мовами, вступу, чотирьох розділів, висновків, 128 списку використаних

джерел та 2 додатків. Робота містить 129 сторінок основного тексту, серед них: 31 рисунок, 9 таблиць, 128 найменувань в списках використаних джерел, 8 сторінок додатків. Загальний обсяг роботи викладено на 142 сторінках.

ДОДАТКИ: Документи впровадження основних результатів дисертаційної роботи та список публікацій здобувача

РОЗДІЛ 1 АНАЛІЗ ПІДХОДІВ ТА МЕТОДІВ ПЛАНУВАННЯ РОБОТИ КОМАНДИ ІТ ПРОЄКТУ

1.1 Особливості планування роботи команди ІТ проєкту на основі гнучкої методології

Команда проєкту починає планування роботи, завдяки якій Sprint backlog можна перетворити на працюючий Product Increment. Робота може відрізнитись об'ємом та складністю. Проте, зазвичай, команда проєкту планує такий обсяг роботи, який можна виконати за спринт. Вона сама організовує роботу, плануючи поетапність виконання вимог із Sprint backlog як під час планування спринту, так і, в разі потреби, протягом усього спринту [1,108].

До задач управління проєктом на основі Scrum відносять: контроль за виконанням поставлених задач в рамках спринту чи декількох спринтів (технічного завдання), розподіл ресурсів та оперативну координацію у випадках непередбачених обставин з метою забезпечення ефективної роботи команди. Враховуючи обмежений період управління особливої актуальності набуває дослідження факторів забезпечення ефективної роботи команди. Однак використання гнучких методологій не вирішує, у повній мірі, проблеми впливу ризиків на роботу команди проєкту.

У загальному випадку на ефективність роботи команди проєкту впливає велика кількість факторів. Виділимо, на наш погляд, найвпливовіші:

- кваліфікація розробників, яка визначає професійний рівень команди, здатність реально оцінити задачі спринту та виконати їх вчасно та на необхідному рівні;
- неточність оцінки задач спринту, що може суттєво вплинути на розподіл ресурсів, терміни виконання та кінцевий результат. Помилки щодо оцінки задач спринта можуть мати наслідки, навіть, за межами одного спринту, наприклад, вимагати зміну всіх подальших термінів;

- неточний опис задач, створений технічним керівником, або хибне розуміння їх розробниками разом із недостатньою їх кваліфікацією. Це може стати причиною неточної оцінки задач (попередній фактор) з тими ж наслідками;
- зменшення ресурсів команди (заплановане або раптове), у тому числі тимчасовий простій одного з розробників (хвороба, відпустка тощо), що призводить до різкого падіння обсягів виконаних робіт та ризику зривів термінів виконання спринту;
- внутрішня культура команди, яка визначає правила та процедури розподілу задач спринту, комунікацію всередині команди, стратегію прийняття рішень в умовах кризової ситуації, що в цілому визначає як команда може здолати непередбачувані обставини та виконати поставлені задачі вчасно та на необхідному рівні.

Перелічені фактори впливають на якість виконання задач проєкту, тому актуальним є створення гнучкого механізму за допомогою якого команда зможе знизити вплив цих факторів ризику на ефективність досягнення цілей проєкту.

В роботі досліджується вплив ризиків на виконання проєкту. Головні фактори такого впливу: нераціональна організація робіт членів команди проєкту, не чіткість опису задач спринту проєкту, недостатня кількість кейсів вирішення ситуації при виконанні задач спринту проєкту. Мета дослідження полягає в створенні моделі управління проєктною командою, яка дозволить знизити вплив факторів ризику на результати виконання проєкту.

1.2 Підходи до зменшення впливу ризиків на роботи команди ІТ проєкту

1.2.1 Методи оцінки поведінки роботи команди

Питання щодо підвищення ефективності процесу управління командою проєкту є предметом дослідження багатьох науковців. Проведений аналіз літератури показав, що сучасні дослідження розглядають процеси управління командою проєкту в умовах ризиків в декількох напрямках.

Перший напрямок пов'язано з дослідженням впливу поведінки членів команди на результати виконання задач проєкту. В роботі [119] продемонстровано можливість застосування мультиагентного підходу до моделювання будь-яких процесів управління проєктами. Для підтримки процесу дослідження та розробки цих методологій автори [119] запропонували використання агентної парадигми та використання мультиагентних систем. Їх використання є наслідком адаптованості агентів для імітації роботи проєктних команд. Авторами [118] було запропоновано мультиагентну модель для підтримки управління ІТ-проєкту і проведено експерименти, у яких відтворено виконання задач типового середнього розміру, виконання яких реалізовано за допомогою запропонованої гібридної методології. Але такий підхід дає хороші результати тільки для великих і середніх проєктів. В роботі [86] представлено гібридну, нечітко-онтологічну систему управління командою, яка є основою для позначення її згуртованості. Крім того, вона може бути використана для покращення вибору членів команди з заданого набору кандидатів. Нажаль, обмеженнями цього підходу є недостатній масштаб проєктів і ускладнений доступ до потрібної інформації. У роботі [53] за допомогою агентно-орієнтованого моделювання досліджується вплив функціональної різноманітності команд та взаємозалежності працівників на продуктивність команди в різних економічних умовах. Це дослідження пропонує керівникам використовувати знання про взаємозалежність працівників для захисту працівників з більш високою продуктивністю, зводячи до мінімуму вплив взаємозалежності під час відбору членів команди для покращення продуктивності фірми. В статті [39] автори розглядають моделювання командної роботи, яка охоплює різні дисципліни від управління бізнесом до когнітивної науки і розподіленого штучного інтелекту. Авторами [39] досліджено командну роботу за двома напрямками: соціальна структура команди та соціальна поведінка. Це дослідження дає організаційну основу для аналізу систем моделювання командної роботи.

В роботі [47] створено підхід для моделювання, контролю та управління процесом когнітивного потоку, представлено рішення для застосування цього підходу до розподіленої команди під час розробки програмного забезпечення. Результати експерименту показали, що такий підхід може покращити здатність команди вирішувати проблеми, які виникають в ході виконання проєкту.

1.2.2 Методи оцінки впливу опису задач на результати роботи команди проєкту

Команда проєкту витрачає значний час на ручну класифікацію вхідних задач. На жаль, цей процес є складним, а члени команди не мають чітких рекомендацій щодо оптимальних практик, заснованих на історичних даних. Із зростанням обсягу даних стає критично важливим впровадження автоматизованої класифікації задач. Зазвичай, описи задач спринту є неструктурованими, що ускладнює їх обробку за допомогою природної мови. Ускладнюючими факторами є також друкарські помилки та скорочення у тексті задач. Класифікація задач спринту є критичним процесом, що забезпечує їхню належну адресацію відповідним виконавцям.

Фіксація задач та їх опис в системі тренінгу є важливим елементом для ведення та контролю над процесами розробки проєкту. Така практика допомагає уникнути непорозумінь, встановлювати пріоритети та визначати прогрес кожної задачі під час спринту. Короткі записи задач природньою мовою дозволяють членам команди швидко сприймати суть задач. Вони можуть включати ключові слова та фрази, які ідентифікують основні аспекти задач. Процес ідентифікації та класифікації задач спринту на “зрозумілі” та “не зрозумілі” є критичним на етапі формування беклогу. Це дозволить визначити задачі, які можуть бути джерелом непорозумінь та ризику затримки виконання проєкту через неякісний текстовий опис. Класифікація задач забезпечить команду проєкту можливістю приділяти особливу увагу та ресурси для уточнення та вдосконалення опису цих задач, зменшуючи тим самим ймовірність помилок та непорозумінь.

Розглянемо наукові роботи, які пов'язані з розглядом оцінки впливу опису задач на результати роботи команди проєкту. В роботі [76] виділено та класифіковано фактори ризику відповідно до таксономії створення програмного забезпечення, запропонованої Інститутом розробки програмного забезпечення (SEI). Загалом, було класифіковано 148 різних факторів ризику. Виявлено, що найбільш пов'язаними сегментами управління, на які впливали означені фактори, були нестабільне організаційне середовище та відсутність або неадекватне планування проєкту. Однак у роботі не розглядається зв'язок між визначеними факторами ризику, а також відсутній аналіз впливу середовища розробки програмного забезпечення на фактори ризику. У статті [66] запропоновано метод відносної оцінки задач, який може бути використаний в гнучких методологіях. Він покликаний скоротити час оцінки задач та вимоги до розробників. У статті [76] описано алгоритми, які забезпечують високу точність оцінки задач великих проєктів із значною кількістю задач. Переваги цієї методики полягають у розрахунку похибки оцінки та математичному викладенні еталонного завдання, яке необхідне для оцінювання поточних задач.

У дослідженні [38] запропоновано альтернативний підхід до оцінки часу виконання робіт для Kanban команд проєкту з можливістю врахування додавання задач до резерву та видалення з нього, зміни їх пріоритетності та, загалом, рівня продуктивності роботи команди. Автоматизація цього процесу, шляхом створення експертної системи, дозволить зменшити рівень зусиль, необхідних команді для того, щоб зробити ці оцінки. Використовуючи цей підхід з'являється можливість підвищити ефективність та продуктивність менеджерів проєктів і проєктних груп шляхом автоматизації роботи за рахунок створення експертних систем.

Авторами [21] запропоновано техніку оцінки вартості та часу виконання проєкту на основі програмного забезпечення, яке збирає ці вхідні дані від команди та оцінює їх. Цей підхід до оцінки витрат зменшив проблеми, пов'язані з проєктами розробки програмного забезпечення на основі Scrum. Автори [44] запропонували нову структуру опитувальника для отримання набору даних, в

якому було застосовано класифікатори машинного навчання та виконано прогнозування ризиків для кожної з ідентифікованих моделей програмного забезпечення. Використовуючи цей результат, менеджери проєктів при розробці програмного забезпечення можуть визначити його модель відповідно до вимог разом із прогнозованим відсотком ризику. Автори статті [68] пропонують метод конфігурації та модифікації процесів програмного забезпечення в компаніях на основі зібраних знань і цей підхід дозволяє підтримувати та оптимізувати процеси управління за допомогою формальних методів моделювання та моделювання на основі машинного навчання.

Багато дослідників вивчають проблеми якості опису функціональних та нефункціональних вимог. Робота [10] пропонує підхід, за допомогою якого можна виявляти шаблони, що відповідають загальновідомим стандартам із мірою $F1 = 0,90$. Також, цей підхід дозволяє виявити загальні синтаксичні особливості для нестандартних шаблонів у понад 73,5% випадків. Оцінка показала, що ці результати надійні незалежно від обсягу та тривалості оброблених вимог.

Дослідження [69] фокусується на відмінностях між методами векторизації, які перетворюють вимоги, написані природною мовою, у числові вектори для подальшої класифікації. За допомогою емпіричного експерименту із використанням 5 відкритих наборів даних виявлено, що передові методи векторизації значно покращують ефективність класифікації. Також виявлено, що використання попередньо підготовлених даних для векторизації іноді є найефективнішим методом.

У статті [40] дослідники визначили поняття "Requirement Smells" (запахи вимог) як показника якості опису вимог для швидкого аналізу під час їх формування. Також дослідники представили підхід до виявлення вимог, який емпірично оцінили при дослідженні багатьох випадків. Робота містить висновки, які пов'язують їх із наявними доказами щодо виявлення дефектів якості природної мови в артефактах вимог, а також проаналізовано вплив та обмеження такого підходу та його оцінку. Дослідження базуються на наявних доказах

стосовно дефектів якості природної мови в артефактах вимог. Проведено оцінку цього підходу, виявлення ознак автоматично має середню точність на рівні 59%, при цьому середнє відкликання становить 82%, і значно варіюється. За допомогою запропонованого методу підвищується визначеність про дефекти якості, але не завжди дозволяє чітко розрізняти деякі ознаки. Для деяких ознак потрібне більш точне визначення, виявлення ознак залишається корисним інструментом для підтримки якості в розробці вимог проєкту.

Дослідники в роботі [91] намагалися перевизначити підходи до вимірювання стандартних метрик IEEE (визначає процес супроводу програмного забезпечення) для імпровізації оцінки якості SRS (специфікації програмних вимог). Автори рекомендували шаблон SRS і стратегію оцінки показників якості SRS. На відміну від сучасних методів, цей підхід вивів метрики IEEE стандартів оцінки якості SRS як більш модульні для збору вимог.

У роботі [22] автори запропонували напівконтрольований підхід категоризації тексту для автоматичної ідентифікації та класифікації нефункціональних вимог (NFR). Мета цього підходу - інтеграція в систему рекомендацій для допомоги аналітикам та розробникам програмного забезпечення в процесі архітектурного проєктування. Невелика кількість вимог, визначених під час процесу виявлення, дозволяє вивчати початковий класифікатор для NFR, який може послідовно ідентифікувати подальші вимоги до постановки задачі в ітераційному процесі.

У статті [23] представлено підхід до зменшення ризиків конфіденційності під час розробки гнучкого програмного забезпечення шляхом автоматичного виявлення інформації, пов'язаної з конфіденційністю історії користувача, що є важливою нотацією в гнучкій розробці вимог (RE). Експериментальні результати показали, що алгоритми глибокого навчання дозволяють отримувати кращі прогнози, ніж ті, що досягаються за допомогою звичайних (неглибоких) методів машинного навчання. Застосування Transfer Learning дозволяє значно підвищити точність прогнозів до 10%.

Автори [71] представили підхід до підтримки відстеження вимог на основі онтології. Розроблена онтологія містить теоретичні знання для розробки вимог у гнучкому середовищі, що дозволяє групі розробників ефективно керувати розвитком вимог до програмного продукту. Це сприяє швидкому прийняттю рішень щодо перегляду артефактів проєкту, пов'язаних із змінами, і дозволяє більш точно визначити масштаби цих змін.

В роботі [8] проведено два експерименти для дослідження факторів, які потенційно можуть впливати на отримання статичних концептуальних моделей із специфікацій, написаних у вигляді історій користувачів і випадків використання. Аналіз результатів показав, що позначення вимог має обмежений вплив на якість отриманої концептуальної моделі як щодо достовірності, так і до повноти. Систематичний граматичний аналіз призводить до концептуальних моделей високого рівня повноти та валідності в обох експериментах.

Використовуючи лінгвістичну класифікацію проблем неоднозначності, автори [111] пропонують новий погляд на поточні знання про розуміння, уникнення та вирішення неоднозначності в історіях користувачів. Досліджено можливості підвищення ефективності техніки історій користувачів у підтримці діяльності з розробки вимог (RE) у проєктах розробки систем застосування (ASD).

Запропонований дослідниками підхід [80] складається з набору правил, які отримують розсіяну інформацію в історіях користувачів та організовують її в символах Lexicon. Lexicon забезпечує консолідовану та організовану структуру для полегшення проблеми заплутаної інформації, що породжує відсутність відстеження між різними спринтами. Цей підхід має переваги для відстеження вимог у гнучких методологіях, що підтримується попередніми результатами оцінки.

У дослідженні [92] визначено класифікацію стилів когнітивного представлення розробників і шаблонів когнітивної взаємодії в гнучкій RE. Досліджено когнітивні здібності розробників на основі їх висловлювань при розділенні та визначенні історій користувачів. Висновки показали, що

розробники, як правило, орієнтуються на технологію когнітивного стилю репрезентації, навіть у RE, і мають когнітивні труднощі в активності та деталізації історій користувача.

У статті [55], автори досліджують ефективність використання вбудовування слів для класифікації задач у службу підтримки ІТ. Експериментальні результати вказують на те, що традиційний метод Frequency Inverse Document Frequency (TFIDF), у поєднанні з методом Support Vector Machines (SVM), надає конкурентоспроможні результати та іноді виявляється ефективнішим, ніж статичні методи побудови векторних репрезентацій слів, такі як, наприклад, word2vec. Використання TFIDF-SVM дозволяє досягти хороших результатів при низькій обчислювальній вартості.

Автори статті [31] пропонують інструментальний підхід, який поєднує візуалізацію інформації з двома техніками обробки природної мови: вилучення концептуальної моделі та семантична подібність. Експерименти показали, що виявлення термінологічних неоднозначностей вимагає значного часу, навіть з використанням технічних засобів, і що важко визначити, чи може синонім завадити правильному напрямку розробки програмної системи.

У роботі [71] представлено метод візуального аналізу DeepNLPVis, який використовується для розуміння та аналізу поведінки моделі в задачах класифікації тексту та вивчення причин успішних і неуспішних випадків. Дослідження показали, що описи задач і коментарі можуть бути використані для прогнозування успіху задачі з високим рівнем точності.

У статті [33] автори розглядають корисність текстових описів задач і коментарів для прогнозування того, чи будуть проблеми вирішені успішно. Експерименти показали, що описи проблем і коментарі можуть бути використані для прогнозування успіху розв'язання проблеми з більш ніж 85% точністю, і що прогнози успіху вирішення проблеми змінюються з часом.

В роботі [24] автори пропонують метод класифікації задач шляхом кластеризації та візуалізації характеристик кожної категорії. Запропоновано метод візуалізації згорткових нейронних мереж, які працюють у текстових

програмах. Дослідники використали оптимізатор Адама та стохастичний градієнтний спуск швидкості навчання 0,001 на всіх моделях. Запропонований метод може використовуватися для виявлення дефектів попередньо навченої моделі та навчальні дані.

Мовні моделі RoBERTa та CodeBERT-MLM [17] використовуються для створення вбудованих вихідних кодів, які фіксують семантичні та контекстні функції, які допомагають у вирішенні задач розуміння мови, таких як SDP. Проводились експерименти в міжверсійному сценарії SDP для Apache Calcite, фреймворку з відкритим кодом для керування даними. Результати оцінки класифікаторів дефектів показують статистично значуще покращення, коли представлення коду вивчаються попередньо навченими моделями порівняно із семантичними представленнями, наданими іншими моделями на основі природної мови, doc2vec і LSI.

Проведені дослідження [2] підходів до глибокого навчання для задач SDP (Software defect prediction) з використанням семантичних особливостей вихідних кодів, для того щоб оцінити продуктивність різних методів DL над моделями SDP на основі семантичних функцій. В дослідженні використовували глибоке навчання для створення моделей SDP на основі семантичних ознак. Отримані результати допоможуть розробникам програмного забезпечення обрати більш глибокі методи навчання та методи подання коду за різними сценаріями, вони можуть визначити взаємозв'язки між типом проєктів, методами представлення коду, методами глибокого навчання та необхідними заходами оцінки таких ситуацій.

Проведені дослідження [42] для визначення відповідних факторів якості вимог при цьому враховуючи дані із записів інтерв'ю, звітів про випуск. Інструмент TABASCO [78] запропоновано для виявлення та ідентифікації неоднозначностей у вимогах до програмного забезпечення та інших документах. Автори застосували алгоритм кластеризації К-середніх із косинусною метрикою подібності, що дозволяє створити кілька кластерів для іменника-кандидата, щоб кожен кластер містив усі екземпляри, які використовувалися в подібному

контексті. Проведені експерименти показують, що TABASCO може бути корисним для виявлення внутрішньодомених неоднозначностей у вимогах до програмного забезпечення та інших документах, пов'язаних з проектом, написаних з використанням тексту NL.

Метод автоматизованого аналізу синтаксису вимог та виявлення їх шаблонів [101]. використовує прийоми обробки природної мови для виділення синтаксичних особливостей формулювань вимог, а також використовує алгоритм виявлення спільноти для групування їх у логічні класи на основі синтаксичної схожості. Результати експериментів демонструють, що цей підхід ефективно розпізнає відомі стандартні шаблони з точністю 90%.

У дослідженні [14] представлена методика, що базується на машинному навчанні (ML) для класифікації та визначення пріоритетів запахів, що потребують уваги. Автори акумулювали 3100 вимог та отримали їхнє експертне маркування. Для побудови моделей класифікації та пріоритезації використовувались загальноприйняті алгоритми, такі як LR, NB, SVM, DT та KNN, а їх ефективність порівнювалася. Отримані результати показали, що LR з використанням TF-IDF досяг найвищої точності - 94% у класифікації запахів за вимогами. Щодо визначення пріоритетності запахів, SVM перевершив інші алгоритми, досягнувши точності 99%.

1.2.3 Методи оцінки сценаріїв управління ризиками при виконанні задач проекту

Розглянемо дослідження процесу управління командою проекту в умовах ризику пов'язані з оцінкою сценаріїв управління ризиками при виконанні задач проекту. В роботі [48] на основі дослідження 139 проектів створення програмного забезпечення запропоновано стратегії пріоритезації ризиків (пріоритезація, орієнтована на ризик, і актуалізована продуктивність визначення пріоритетів) і пояснено їх відповідне застосування, переваги та недоліки. Авторами з'ясовано, що залежно від достатності проектних, організаційних ресурсів та профільних вимог до виконання проекту, менеджери проектів можуть

використовувати ці стратегії як практичні вказівки для систематичного, ретельного та ефективного зменшення ризиків програмного проєкту. Автори [103] запропонували анкету, яка дозволяє вимірювати ризики та результати проєктів розробки програмного забезпечення. За її допомогою провели опитування по виконанню 145 програмних проєктів світових ІТ-компаній. В результаті було розроблено модель структурного рівняння, що показує взаємозалежність основних факторів ризику та їх вплив на результат проєкту.

Дослідники у [74] охарактеризували різницю між ризиком і невизначеністю та виявили п'ять методів і 18 практик для зменшення, саме, невизначеностей. В роботі запропоновано підхід до управління невизначеністю та описано стратегії, які дозволяють членам команди чітко формалізувати та керувати невизначеністю в проєктах програмного забезпечення. У дослідженні [112] представлено імітаційну модель процесу стратегічного управління проєктами розробки програмного забезпечення. Основна перевага запропонованої моделі полягає в тому, що вона забезпечує інтегровану структуру управління, в якій пов'язані управління ризиками, оцінка витрат і планування управління проєктами.

У статті [3] авторами запропонована багатоагентна модель управління ризиками. Вона розроблена з використанням багаторазового індивідуального навчання з використанням алгоритму Q-навчання. Це може допомогти керівникам проєктів визначати стан свого проєкту та виконувати певні дії, відповідно до стану означеного ризику. Пропонована модель включає дію або набір дій по кожному джерелу чинника ризику. Але важливим недоліком моделі є те, що в роботі не розглядається модель з множинними можливими ризиками та відповідні пропоновані дії для досягнення цілей проєкту в цьому випадку.

Дослідники у [105] запропонували модель машинного навчання для прогнозування невдач у сферах знань управління проєктами для компаній, що займаються програмним забезпеченням. Модель включає три фактори: контекст керівника проєкту, контекст проєкту та контекст компанії. Результати дослідження показали, що техніка опорних векторів ефективніша за інші

потенційні алгоритми для прогнозування процесу управління проєктами компанії.

У [32] запропонована модель системної динаміки компанії, що враховує процеси фінансування, аутсорсингу діяльності, коригування графіку, управління працею тощо. Керівники проєкту повинні враховувати динамічні петлі поведінки зворотного зв'язку щодо затримок і збурень. Це дозволить їм порівняти альтернативну стратегію управління змінами з точки зору показників ефективності.

У статті [65] запропоновано допоміжний інструмент, що дозволяє використовувати онтологію в якості гнучкого постачальника практичних знань. Авторами, за допомогою опитування експертів, представлено розширену онтологію, яка дозволить керівникам вирішити, яку практику прийняти, як її застосувати тощо.

Авторами [79] запропоновано метод визначення чинників впливу проєкту на розвиток ІТ-компанії з урахуванням її діяльності, цілей розвитку і стратегії. Впровадження цього методу дозволило проєктному офісу раціонально спланувати ресурси компанії, визначити напрямки діяльності на ринку ІТ та скорегувати подальший розвиток.

1.2.4 Методи оцінки розподілу задач серед виконавців

Розподіл задач серед виконавців у спринті проєкту є важливим етапом в управлінні проєктами, особливо в рамках гнучких методологій. Ефективний розподіл задач допомагає оптимізувати роботу команди, підвищити продуктивність і забезпечити виконання задач у встановлений термін. Розглянемо існуючі підходи до розподілу задач серед виконавців проєкту у роботах різних дослідників.

Автори роботи [43] пропонують алгоритм розподілу працівників на проєктні роботи в умовах невизначеності, який враховує рівень професійної компетентності персоналу та кваліфікаційні вимоги до задач проєкту. Вони визначають основні параметри оцінки персоналу, що включають не лише наявні

знання та досвід, а й особистісні якості працівника. Дослідники підкреслюють важливість співвідношення вимог до проєктних задач та кваліфікації трудових ресурсів, що може сприяти ефективнішому розподілу задач у контексті гнучкої методології розробки ІТ-проєктів.

В роботі [41] представлена модель підтримки прийняття рішень з використанням генетичного алгоритму розподілу задач. В якості основних сутностей використовуються задачі, ресурси, цілі, різні параметри. На основі генетичного алгоритму формується хромосома для подвійних фрагментів, спочатку інформація про виділення ресурсів на окремі задачі і далі інформація про виділення часових шкал на окрему комбінацію ресурсів задачі.

Дослідники Barcus A. і Montibeller G. у своїй роботі [13] представили підхід до підтримки розподілу задач в розподілених командах з використанням багатокритеріального аналізу рішень (multicriteria decision analysis, MCDA). Дослідження базується на реальному прикладі, де була створена багатокритеріальна модель для підтримки розподілу роботи в розподілених командах. Ця робота пропонує структурований підхід до вирішення складної проблеми розподілу задач у глобально розподілених проєктах розробки програмного забезпечення.

В статті [123] автори представили метод розподілу задач для краудсорсингу програмного забезпечення в контексті спільної розробки. Автори запропонували підхід до призначення задач у середовищі краудсорсингу, який є перспективною моделлю розробки програмного забезпечення. Ця робота спрямована на вирішення проблеми ефективного розподілу задач у краудсорсингових проєктах розробки програмного забезпечення, що є важливим аспектом для успішної реалізації таких проєктів. Результати показують, що запропонований спосіб дозволяє збільшити корисність приблизно на 25%, а середній рівень успішності приблизно на 30% порівняно з методом послідовного призначення.

Для вирішення проблем розподілу задач у роботі [121] запропоновано три алгоритми: GAN (генеративні змагальні мережі) для генерації тексту, для

генерації даних прийняття рішень та посилення функцій даних; алгоритм Баума–Велча для отримання параметрів моделі; алгоритм Вітербі для отримання оптимальної стратегії розподілу задач. На основі цих алгоритмів створюються ефективні стратегії розподілу задач для максимізації загальної вартості задач, які виконуються працівниками.

Інтегрована система штучного інтелекту [81] забезпечує динамічну оптимізацію розподілу ресурсів і задач на основі даних для підвищення продуктивності програмних проєктів. Ця система включає обробку природної мови (NLP), навчання з підкріпленням та байєсовські мережі. Вона формує вимоги до задач із проєктних документів, прогнозує оптимальний розподіл ресурсів за допомогою навчання з підкріпленням і перевіряє розподіл за допомогою баєсової мережі, навченої на минулих проєктних даних.

У статті [110] авторами пропонується розширений метод і алгоритм за допомогою поєднання оптимізованого гнучкого планування ітерацій і можливості прогнозувати та керувати ризиками в умовах обмежених ресурсів байєсовських мереж. На основі методу розроблено програмне забезпечення як допоміжний інструмент для менеджерів, щоб контролювати свої графіки проєктів. Інструмент також забезпечує надійний набір стратегій секвенування завдання гнучкого планування ітерацій.

У статті [100] автори провели огляд різних підходів до розподілу задач, прийнятих дослідниками гнучкої розробки програмного забезпечення кількісно. У таблиці наведено порівняння прийнятих заявок на розподіл задач разом із їхніми плюсами та мінусами. В результаті дослідження визначено, що більшість підходів до розподілу задач є кількісними, але під час цього процесу якісний аспект не розглядався значною мірою.

Для того, щоб розробити гнучку й ефективну модель для планування програмного проєкту, у роботі [26] розроблено підхід із планування на основі подій (EBS) і алгоритму оптимізації мурашиної колонії (ACO). Запропонований метод дозволяє моделювати конфлікт ресурсів та зберігає гнучкість у розподілі

людських ресурсів. Результати 83 експериментів свідчать про перспективність запропонованого методу.

Багатокритеріальна модель прийняття рішень для планування та точного налаштування планів проєкту [7] була розроблена з використанням когнітивного картування та MACBETH (вимірювання привабливості за технікою оцінювання на основі категорій). Запропонована модель ґрунтувалася на ціннісних судженнях осіб, які приймають рішення, що надає моделі суб'єктивного характеру.

У дослідженні [99] автори об'єднали декілька факторів, використаних у розподілі задач, і визначили їхню важливість, дозволивши встановити порядок пріоритетності серед них. Представлено гібридну методологію, побудовану на методах VDA для класифікації і впорядкування факторів, які керують призначенням задач розподіленням командам у проєктах розробки програмного забезпечення. Задачі були згруповані відповідно до їх типу, вимог, архітектури, реалізації та тестування. Цей метод передбачає класифікацію та впорядкування факторів, які керують розподілом задач у розподіленому сценарії.

Авторами роботи [122] запропоновано алгоритм динамічного розподілу задач (DUTA) та динамічний краудсорсинговий програмний алгоритм на основі утиліт. Для визначення оптимальної відповідності між задачами та працівниками використовується метод Куна–Мункресса та алгоритм зі зваженим дводольним графом. Результати експериментів показали, що DUTA дав задовільні результати продуктивності загальної утиліти розподілу з кращим рівнем успіху розподілу задач, ніж алгоритм, заснований на надійності користувача. DUTA досягла середньої точності розподілу 85,63%, що демонструє ефективне керування задачами.

В статті [6] авторами запропонована система підтримки прийняття рішень щодо планування спринту (SPESS), яка є інструментом допомоги менеджерам у плануванні спринту. SPESS використовує покер для планування та угорський алгоритм як основу, і крім врахування фактору досвіду, враховує рівень

компетенції розробника та залежності задач. Результатом є комплексне та точне планування спринту для швидкої та високоякісної доставки продукції.

В роботі [116] запропоновано підхід до розподілу задач у команді Scrum з використанням мультиагентного моделювання на основі р-значень. Автор розробив алгоритм розподілу задач, який використовує р-значення як ключовий фактор для прийняття рішень щодо призначення задач агентам. Р-значення розглядається як відносний погляд на агента та задачі, над яким він працює. Цей підхід дозволяє ефективно розподіляти задачі в команді Scrum, враховуючи характеристики агентів та особливості задач.

Дослідники в роботі [9] використовують змішане цілочисельне нелінійне програмування (MINLP) для планування проєкту та вирішення проблеми розподілу персоналу із залежним від часу ефектом навчання з урахуванням подібності задач. Ефект навчання в роботі залежить від часу, коли співробітники проєкту починають виконувати ці задачі. Якщо персонал проєкту виконує повторювані та/або подібні задачі, тоді цей персонал може отримати досвід і виконувати задачі швидше, ніж заплановано. Таким чином, послідовність задач є важливою для їх призначення співробітникам проєкту, щоб мінімізувати час завершення проєкту із урахуванням подібності задач з точки зору навчання.

Робота [113] спрямована на побудову дерева цілей, що дозволило відобразити загальну мету і підцілі, які необхідно забезпечити для ефективного розподілу ресурсів. Як математичну модель використано булеву задачу цілочислового програмування, яка з достатнім ступенем адекватності змогла відобразити реалії формування портфеля проєктів сумісно із розподілом ресурсів між відповідними проєктами портфеля. Результатом роботи став прототип програмного забезпечення, що реалізує моделювання розподілу ресурсів за допомогою методу Балаша.

За результатами досліджень, які опубліковані у [90], запропоновано новий підхід до розробки інтегрованої системи оптимізації розподілу ресурсів та задач (RATAOS) з використанням архітектури підприємства для підвищення ефективності управління проєктами в ІТ-компаніях. Основні аспекти їхнього

дослідження включають: інтеграцію інформаційної системи управління проектами (PMIS) з системою оптимізації, розробленою з використанням моделі випадкового лісу та обробки природної мови (NLP). Оптимізація розподілу ресурсів та задач призвело до зниження операційних витрат на 14% та скорочення фаз планування на 88,7%. Ефективність запропонованої системи демонструє зменшення часу завершення проекту на 50,80% [90].

Авторами роботи [82] запропонована комплексна методика формування складу команди IT-проекту, що базується на розв'язанні багатокритеріальної задачі розподілу за допомогою метрики, календарного планування та розрахунку навантаження працівників. Запропоновано два шаблони формування команд: для проекту та для задачі в рамках проекту. Такий комплексний підхід дозволяє скоротити час формування команди проекту та виключає ризик хибного призначення.

Виходячи з проведеного аналізу публікацій, вирішення задачі планування роботи команди IT-проекту є актуальною в контексті підвищення ефективності управління проектами, зокрема за умови застосування гнучких методологій.

1.3 Постановка задачі дослідження

Команда проекту розглядається в рамках відповідного проекту та складається з наступних осіб: менеджера проекту, розробників, технічного керівника, що також виконує роботу з імплементації проекту, та тестувальника. Робота над проектом виконується ітеративно, кожна ітерація (спринт) – завдовжки один тиждень. Менеджер проекту виконує функцію управління та комунікацію із замовником, отримуючи від нього задачу та з'ясовуючи базові технічні умови. Цей процес може проходити у декількох формах, як то особисте спілкування, обмін текстовими повідомленнями за допомогою певної інформаційної системи управління проектами тощо.

В даному дослідженні будемо розглядати функціонування команди проєкту та виконання задач на рівні одного спринту проєкту. Процес виконання задач командою проєкту представлено на рисунку 1.1.



Рисунок 1.1 – Схема виконання задач командою проєкту на рівні спринта

В результаті цього процесу формується документ з описом проєкту, або технічне завдання на розробку (частини проєкту, покращення або нового функціоналу). Кожне завдання розбивається командою на чітко визначені задачі, для яких встановлюється оцінка часу виконання. Цю оцінку проводить та людина, яка буде виконувати цю задачу, але під контролем менеджера проєкту та технічного керівника. Далі відбувається загальна оцінка термінів виконання та вартості проєкту, які повідомляються замовнику.

Наступним кроком, після затвердження задач та оцінки часу, є виконання командою затверджених задач протягом спринту. Під час їх виконання члени команди мають своєчасно відмічати прогрес виконання задач та логувати час кожен день на кожну задачу. У випадках, коли функціонал не відповідає повністю або частково поставленим вимогам, задача повертається до розробника з коментарями. В кінці спринту, команда проєкту розглядає його виконання у вигляді інкременту продукту і, в залежності від результату, з'ясовує список задач для наступного спринту. Всі зміни та виконання фіксуються на Scrum дошці. В

кінці кожного спринту проводиться демонстрація результатів замовнику. В процесі такої демонстрації може змінюватися технічне завдання та коригуватися кінцева задача з урахуванням пріоритетів тієї чи іншої частини завдання. Раз в декілька спринтів (зазвичай раз в місяць) проводиться ретроспектива виконання проєкту з підрахунком швидкостей роботи, адекватності оцінки та інше. Результати ретроспективи фіксуються у текстовому документі, до якого мають доступ всі учасники команди проєкту. Оскільки всі задачі та трекінг часу проводиться в інформаційній системі управління проєктами, завжди є можливість експорту звітів у вигляді таблиць та можливість збирати всю інформацію по задачам у вільному форматі. Ця інформація включає в себе переміщення задачі, всі зміни та коментарі до неї.

Таким чином, вихідні дані для виконання задач протягом спринту зберігаються в інформаційній системі у текстовому форматі природньою мовою. Задачі, які команда виділила на спринт, також сформульовані природньою мовою і зберігаються у текстовому форматі. В процесі роботи кожен член команди змінює статус своєї задачі (зміна фіксується в інформаційній системі) та може залишати коментарі або в карточці задачі, або в чаті. Менеджер може контролювати процес виконання та вчасно передбачити можливі відхилення.

Під час виконання проєкту найчастіше виникають такі проблеми:

- виконавці не вкладаються в оцінений час виконання задачі;
- під час оцінки часу виконання проєктних задач не вистачає даних або недостатньо точно визначені вимоги;
- виконавці не заповнюють або не достовірно заповнюють дані в інформаційній системі і неможливо провести ретроспективу для аналізу помилок тощо.

Аналіз проведених досліджень дозволяє зазначити, що існують підходи до опису задач та історій користувача, але питання якісного опису задач спринту проєкту залишається менш вивченим напрямком. Необхідність якісного текстового опису задач спринта особливо актуальна, оскільки неправильно

сформульований чи незрозумілий опис може призвести до непорозумінь та помилок у виконанні задач.

На відміну від оцінки якості опису вимог та історій користувача, оцінка якості текстового опису задач спринту вимагає врахування особливостей цього контексту. На етапі створення беклогу спринта важливо визначити, наскільки зрозуміло та однозначно сформульована кожна задача. Це дозволить уникнути непорозумінь та невірних виконань задач учасниками команди проєкту.

Отже, сучасні дослідження в галузі розподілу задач у команді проєкту демонструють тенденцію до інтеграції різноманітних підходів, включаючи багатокритеріальні моделі, алгоритмічні рішення та методи штучного інтелекту. Ці підходи спрямовані на підвищення ефективності управління проєктами, оптимізацію використання ресурсів та покращення якості результатів проєктів.

Огляд літератури виявляє різноманітні підходи до розподілу задач у командах проєктів. Однак дослідженню розподілу задач у команді проєкту на основі постановки задачі про стабільні зіставлення приділено не достатньо уваги. На наш погляд, таке дослідження є важливим з точки зору забезпечення стабільного розподілу, що особливо актуально для scrum-команди, збалансованості розподілених задач між виконавцями та оптимізації ресурсів із врахуванням інтересів та компетенцій членів команди.

Метою дисертаційної роботи є зниження рівня дефектів виконання робіт проєкту за рахунок розробки моделей та методів інтелектуального планування роботи команди.

Для досягнення даної мети поставлені наступні завдання:

- 1) дослідити чинники зростання рівня дефектів виконання робіт проєкту та ризику невиконання робіт спринту;
- 2) розробити модель планування роботи команди ІТ проєкту, яка враховує пріоритети та оцінки змісту задач, та дозволяє формувати рекомендації щодо вибору та розподілу задач спринту;

3) розробити методи для оцінки та вдосконалення текстових описів задач спринту на основі використання методів машинного навчання та великих мовних моделей

4) розробити моделі розподілу задач спринту між виконавцями, що враховують специфіку гнучкої методології управління проєктами та дозволяють оптимізувати роботу команди.

5) розробити інформаційну технологію підтримки планування роботи команди ІТ проєкту;

6) провести апробацію ефективності запропонованих моделей та методів у реальних командах розробників.

Сучасні дослідження підтверджують, що людський фактор суттєво впливає на успішну реалізацію проєкту, тому розробка моделей та методів створення ефективної команди є актуальним завданням.

1.4 Висновки до розділу 1

У розділі 1 виконано аналіз джерел інформації з питань планування роботи команди ІТ проєкту. Проаналізовано праці, які присвячено дослідженню зменшення впливу ризиків на роботу команди. Зосереджено увагу на визначенні чинники зростання рівня дефектів виконання робіт проєкту та ризики невиконання робіт спринту. Проаналізовано роботи вітчизняних та закордонних авторів, присвячені саме оцінці поведінки роботи команди, оцінці впливу опису задач на результати роботи команди проєкту, оцінці сценаріїв управління ризиками при виконанні задач проєкту, оцінці розподілу задач серед виконавців

Визначено особливості проєктів в ІТ. Показано, що в умовах частих змін вимог до проєкту та продукту найчастіше використовуються Agile підходи до управління проєктами.

Основну увагу приділено аналізу робіт з дослідження опису задач та розподілу задач серед виконавців з допомогою математичних моделей та

методів. Виявлено, які важливі постановки задач не розглядаються у відомих роботах.

На основі аналізу існуючих робіт сформульовано мету та задачі дослідження.

РОЗДІЛ 2 МОДЕЛІ ТА МЕТОДИ ІНТЕЛЕКТУАЛЬНОГО ПЛАНУВАННЯ РОБОТИ КОМАНДИ ІТ ПРОЄКТУ

2.1 Опорна модель управління проєктною командою в умовах ризиків

Проведений аналіз стану вирішення проблеми управління проєктною командою в умовах ризиків показав, які головні фактори визначають ризик невиконання проєкту. Це неточності формулювання задач, які оцінює команда, низька комунікаційна культура команди та недосконалість процедури розподілу задач в команді. Задачі, які виконує команда протягом спринту, як правило, сформульовано природньою мовою у текстовому форматі, тому для аналізу задачі та оцінки ризиків невірного оцінювання його змісту необхідно використовувати лінгвістичні технології інтелектуального аналізу тексту. Комунікація команди, особливо в умовах віддаленої роботи, повністю залежить від корпоративної культури, необхідності застосування робочих каналів для спілкування, своєчасного трекінгу робочого часу, зміни статусів задач, коментування та контролю, який повністю відбувається в інформаційній системі управління проєктами. Наявність такої інформації надає можливість застосовувати методи інтелектуального аналізу бізнес-процесів для вирішення задач моніторингу та контролю виконання спринту, а також дослідити шляхи вдосконалення організаційної структури команди для зменшення ризиків несвоєчасного виконання задач проєкту. При цьому розподіл задач спринту суттєво залежить від кваліфікації команди проєкту та якісної ретроспективи. Вирішення цієї проблеми можливо на основі аналізу прецедентів.

Поведінка команди проєкту на основі аналізу особливості її роботи відображає активність її членів, стан задач та строки їх виконання або причини невиконання. Вхідна інформація для такої моделі – дані з журналів подій (лог-файлів) в інформаційній системі.

Аналіз бізнес-процесів роботи команди можливо виконати на принципах і методах проведення інтелектуального аналізу даних [1]. Ці технології надають

нові засоби виявлення, моніторингу та вдосконалення процесів у різноманітних сферах застосування. Крім того, журнали можуть зберігати додаткову інформацію про події.

Чіткість опису задачі впливає на результат її виконання. Тому для обробки текстового опису задач спринта доцільно використовувати методи лінгвістичного аналізу обробки тексту для отримання якісного формулювання задач для команди проєкту.

Розподіл задач у команді проєкту може виконуватися на основі навчання з підкріпленням [104]. Навчання з підкріпленням використовує формальну структуру процесів прийняття рішень для визначення взаємодії між агентом навчання та його середовищем у термінах станів, дій та нагород [104].

Ще одним теоретичним підходом, який може бути використаний для розв'язання поставленої задачі – це формування рекомендацій на основі поведінки, оцінки задач та стратегії оптимального розподілу при виконанні задач спринту. Системи рекомендацій застосовують методи інтелектуального аналізу даних і алгоритми прогнозування щоб передбачити інтерес користувачів до інформації [120].

Таким чином, формування рекомендацій щодо розподілу задач та виконання спринту командою проєкту повинно враховувати результати вирішення задач оцінки якості формулювань задач, оцінки ефективності організації робіт, а також ґрунтуватись на попередньому досвіді та симуляції можливих ризиків протягом виконання спринту. Реалізація такої рекомендаційної системи передбачає детальне дослідження означених задач та визначення їхньої взаємодії.

Можемо виділити основні концепти задачі управління проєктною командою в умовах ризиків. По-перше, це команда проєкту, проєкт та спринт, які визначають об'єкт дослідження. З початку команда проєкту отримує з беклогу продукту проєкту завдання на спринт проєкту. Команді треба оцінити якість формування задач, які подаються у текстовому вигляді. На кожному спринту команда проєкту отримує текстовий опис задачі, який зберігається в

фреймворку необхідно визначити еталон представлення задачі, який дозволяє оцінити формулювання щодо чіткості, однозначності, трасування та валідації задачі командою протягом спринту. Проблемою формування такого еталону є недостатня кількість зразків та суб'єктивність щодо оцінки якості формулювання задачі. Тому система управління командою проєкту повинна мати модель оцінки якості формування задач спринту з точки зору ризиків невірної їх тлумачення. Вирішити цю задачу дозволяє об'єднання методу обробки текстової інформації та метод навчання на основі прецедентів, що дозволяє врахувати попередній досвід команди та її поведінку. Опис задачі у текстовому форматі, оцінка якості формулювання та ситуація (прецедент) зберігаються в інформаційній системі.

Поведінка команди під час спринту визначається розподілом та закріпленням задач між членами команди, коментуванням задач, зміною їх статусу із валідацією та ретроспективою включно, зберігається у журналі подій. У журналі подій фіксується історія поведінки команди проєкту в ході роботи над задачами спринту. Ця історія аналізується в моделі аналізу журналу подій за допомогою методів Process Mining. Модель поведінки команди проєкту дозволяє врахувати ризики нераціональної організації робіт протягом спринту, недоліки комунікації всередині команди та особливості команди під час планування наступних спринтів.

Центральним елементом опорної моделі (рисунок 2.1) є модель рекомендацій, яка повинна надати команді проєкту необхідну інформацію для прийняття рішень в умовах невизначеності та ризиків при управлінні проєктом. Така модель враховує результати, які надає модель оцінки якості формування задач спринту. Система рекомендацій створюється на основі критеріїв прийняття рішень, метрик схожості, та налаштування параметрів моделі відповідно до особливостей проєкту, кваліфікації команди, бази прецедентів та даних оцінки якості формулювання задач спринту та оцінки поведінки команди.

Таким чином, опорна модель управління командою проєкту задає основу усіх сутностей та інформаційних потоків, що потрібно врахувати, аби сформулювати рекомендації для спринту.

Розглядаючи частину цієї опорної моделі, наступним кроком є деталізація того, як саме ці потоки перетворюються на рішення. Для цього було побудовано фреймворк інтелектуального планування (рисунок 2.2), в якому реалізовано поетапний підхід до автоматизації та оптимізації процесу планування спринту, що ґрунтується на використанні моделей оцінювання, прогнозування та рекомендаційних систем. Фреймворк конкретизує, як саме модель формування рекомендацій (рисунок 2.1) перетворює зібрані дані на практичні рішення. Основу фреймворку на рисунку 2.2 складає інтеграція моделей штучного інтелекту з практиками Agile-менеджменту.

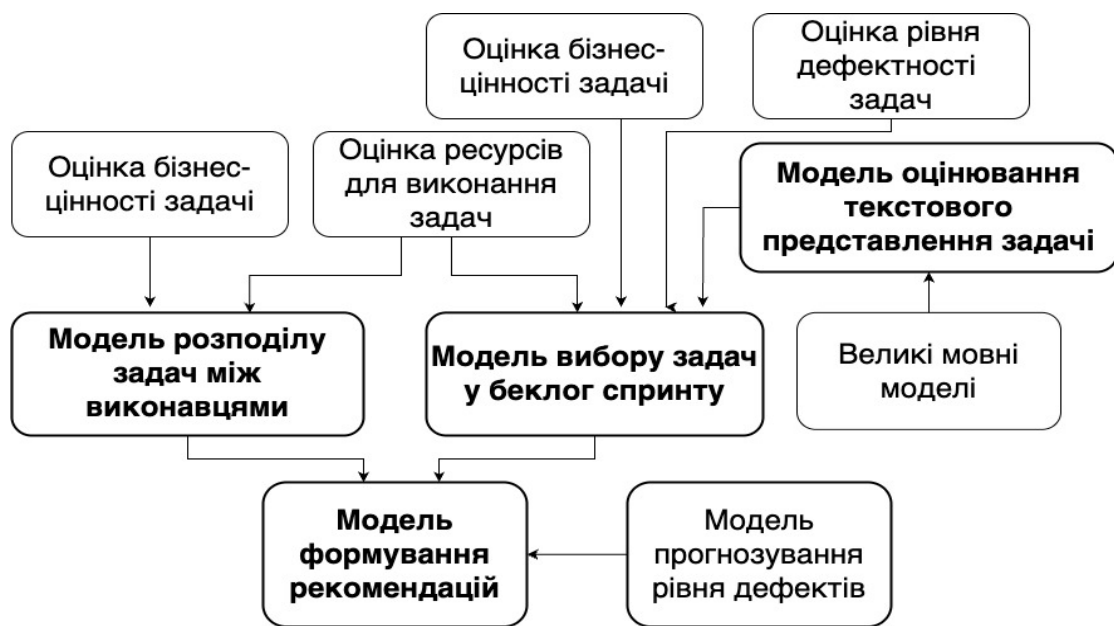


Рисунок 2.2 – Фреймворк інтелектуального планування роботи команди ІТ проєкту

Розглянемо елементи розробленого фреймворку детально.

1 Оцінка бізнес-цінності задачі. Цей компонент дозволяє визначити пріоритет задачі з точки зору її внеску у досягнення бізнес-цілей проєкту.

Результати цієї оцінки використовуються як у моделі вибору задач для беклогу спринту, так і в моделі розподілу задач між виконавцями.

2 Оцінка ресурсів для виконання задач. Даний модуль передбачає прогнозування трудовитрат і необхідних ресурсів для виконання кожної задачі. Ця інформація є критично важливою для планування навантаження команди і використовується у виборі задач для спринту.

3 Оцінка рівня дефектності задач. Передбачає використання історичних даних для оцінки потенційної складності та ризику появи дефектів у результаті реалізації задачі. Даний параметр впливає на прийняття рішень у моделі вибору задач.

4 Модель оцінювання текстового представлення задачі. Ця модель, базується на великих мовних моделях (LLM), здійснює автоматичний аналіз опису задачі з метою виявлення неповноти, неоднозначності чи низької якості формулювання. Якість формулювання задачі, в свою чергу, впливає на ймовірність виникнення дефектів – чим нижче якість, тим вище вірогідність виникнення дефектів.

5 Модель вибору задач у беклог спринту інтегрує результати оцінки бізнес-цінності, ресурсів, ризиків та якості опису задач для формування збалансованого набору задач у спринті.

6 Модель розподілу задач між виконавцями використовує результати попередніх оцінок для ефективного призначення задач на окремих членів команди, враховуючи їхню компетентність, завантаження та спеціалізацію.

7 Модель прогнозування рівня дефектів базується на даних з оцінки задач та якості їх опису для передбачення ймовірності помилок, що можуть виникнути під час реалізації.

8 Модель формування рекомендацій є інтеграційним компонентом, що узагальнює дані з усіх моделей і формує підсумкові рекомендації для менеджера проєкту або для команди щодо плану спринта, включаючи перелік задач, розподіл виконавців та ризику.

Запропонований фреймворк демонструє можливість використання інтелектуальних моделей для покращення процесу планування спринту в ІТ-команді. В ньому шість спеціалізованих аналітичних блоків поєднуються в єдину модель інтелектуального планування спринту. На виході модель прогнозування дефектів перевіряє сформований план, коригуючи його, якщо ризик перевищує припустимий поріг, тоді як модель формування рекомендацій об'єднує результати всіх підсистем у зрозумілий менеджеру набір дій. Таким чином, фреймворк інтегрує пріоритети (бізнес-цінність), оцінки змісту задач (текстова якість, дефектність, трудомісткість) і генерує обґрунтовані рекомендації як щодо наповнення спринту, так і щодо розподілу задач усередині команди. Інтеграція оцінювання задач, ресурсного планування, мовних моделей і прогнозування дефектів дозволяє забезпечити обґрунтовану підтримку прийняття рішень, підвищуючи ефективність командної роботи та якість програмного продукту.

2.2 Модель задачі інтелектуального планування роботи команди

У рамках сучасних ІТ-проектів інтелектуальне планування спринту передбачає використання методів штучного інтелекту для прийняття оптимальних рішень про вибір та розподіл задач, з урахуванням складності їх текстових описів, бізнес-цінності, переваг виконавців і ризиків виникнення дефектів. У межах цього дослідження цінність спринту — це агрегована (бізнес) метрика очікуваного ефекту від виконання запланованих задач спринту. Під ризиками ж будемо розуміти зниження цінності, спричинене певними факторами (нечіткі описи, дефекти, перевантаження команди тощо).

Метою моделі є побудова плану спринту, що мінімізує ймовірність дефектів при максимізації бізнес-цінності за обмежених ресурсах команди.

Нехай:

$T = \{t_1, t_2, \dots, t_N\}$ — множина задач з беклогу проєкту;

$D = \{d_1, d_2, \dots, d_M\}$ — множина розробників (команда проєкту);

$x_i \in \{0,1\}, i = \overline{1, N}$ — ознака вибору задачі t_i для виконання у спринті;

$y_{ij} \in \{0,1\}, j = \overline{1, M}, i = \overline{1, N}$ – ознака призначення задачі t_i виконавцю d_j .

Кожна задача $t_i \in T$ – характеризується оцінкою бізнес цінності $V(t_i)$ для проєкту в цілому, та вимогами $C(t_i)$ щодо її виконання (складність, кваліфікація виконавця, термін виконання).

Для кожного виконавця (розробника) $d_j \in D$ визначена його кваліфікація $Q(d_j)$, яку можна співставити з вимогами задач спринту. Кожен виконавець $d_j \in D$ має власні преференції щодо задач беклогу $P(d_j, t_i)$.

Нехай $E(t_i), \forall i = \overline{1, N}$ визначає необхідні ресурси, у тому числі час виконання, а $A(d_j), \forall j = \overline{1, M}$ – доступний робочий час кожного виконавця $d_j \in D$. Позначимо загальні доступні ресурси команди на спринт E^S , що у загальному випадку може враховувати більш складні залежності, ніж загальний доступний час.

Для кожної пари $(t_i, d_j), t_i \in T, d_j \in D$, вважаємо визначеними наступні оцінки:

$DS(d_j, t_i)$ – задоволеність виконавця d_j призначенням йому задачі t_i , що враховує преференції виконавця;

$CS(t_i, d_j)$ – узгодженість призначення задачі t_i виконавцю d_j , що враховує відповідність виконавця вимогам задачі.

Припускаємо, що узгодженість призначення задачі та задоволеність виконавця, призначеного на виконання цієї задачі посилює її вагу спринті, оскільки позитивно впливає на результати проєкту в цілому.

Високі вимоги до задачі (складність) додають їй та спринту, в цілому, цінності. Тому, враховуючи мету, яка полягає у підвищенні цінності спринту, цінність задачі $t_i \in T$ можна записати наступним чином:

$$V_{total}(t_i) = \alpha * C(t_i) * V(t_i) * \left(\sum_{j=1}^M (CS(t_i, d_j) + DS(d_j, t_i)) * y_{ij} \right) * x_i$$

де: - α – масштабуючий коефіцієнт;

- $C(t_i)$ – складність (вимоги) задачі;
- $V(t_i)$ – бізнес цінність задачі;
- x_i та y_{ij} – ознаки вибору задачі та призначення виконавців;
- $CS(t_i, d_j)$ – узгодженість;
- $DS(d_j, t_i)$ – задоволеність виконавців розподілом.

Виконання обраної на спринт задачі є критично важливим для результатів спринту та проєкту в цілому.

Враховуючи розглянуті раніше чинники ризиків невиконання та дефектності задач, розглянемо наступні оцінки, які можна отримати на етапі планування:

$U(t_i), t_i \in T$ – чіткість текстового опису задачі, що відображає сприйняття суті задачі виконавцем;

$I(t_i, d_j), t_i \in T, d_j \in D$ – індикатор умови, що для задачі t_i існує більш кращий розподіл (існує виконавець, який краще відповідає вимогам задачі, ніж d_j) та для виконавця d_j існує більш приваблива задача, ніж t_i .

Будемо називати таку умову – умовою блокування, а індикатор $I(t_i, d_j)$ – індикатором блокування. Цей індикатор відображає ризик, що задача не буде виконана належним чином або виконавець переключиться на іншу, більш цікаву для нього, задачу.

Слід зауважити, що деякі задачі мають взаємозалежності. Будемо враховувати залежність від послідовності виконання задач. Позначимо $G(t_k, t_i), t_k, t_i \in T$ – індикатор послідовності виконання задач таким чином, що задача t_k повинно бути виконана до початку виконання t_i .

Тоді цінність задачі зростає з урахуванням задач, які залежать від неї, тобто:

$$\begin{aligned} \widetilde{V_{total}}(t_i) = & \alpha * C(t_i) * V(t_i) * \left(\sum_{j=1}^M (CS(t_i, d_j) + DS(d_j, t_i)) * y_{ij} \right) * x_i \\ & + \left(\sum_{k=1}^N (V_{total}(t_k) + D(t_i, t_k)) \right) * x_i, \forall t_i \in T \end{aligned}$$

Отже, оцінка цінності задачі $\widetilde{V_{total}}(t_i)$ враховує цінність всіх задач t_k , які залежать від виконання задачі t_i .

Тоді критерій максимізації сумарної цінності задач спринту можна записати наступним чином:

$$V^S = \sum_{i=1}^N \widetilde{V_{total}}(t_i) \rightarrow \max$$

Цінність спринту визначається як сумарна цінність задач, обраних на спринт, з урахуванням ризиків їх дефектності, тобто:

$$R(t_i) = \beta * (1 - U(t_i)) * V(t_i) * \left(\sum_{j=1}^M I(t_i, d_j) * y_{ij} \right) * x_i, \forall t_i \in T,$$

де: - β – коефіцієнт масштабування;

- $U(t_i)$ – чіткість текстового опису задач;

- $V(t_i)$ – бізнес цінність задачі;

- $I(t_i, d_j)$ – індикатор блокування;

- x_i та y_{ij} – ознаки вибору задачі у спринт та призначення виконавця відповідно.

З урахуванням залежності задач (послідовності виконання), ризик вибору задачі t_i у спринт можна записати наступним чином:

$$\tilde{R}(t_i) = \beta \left[\gamma_1 * (1 - U(t_i)) + \gamma_2 * \sum_{k=1}^N (1 - U(t_k)) * D(t_i, t_k) \right] * C(t_i) \\ * \left(\sum_{j=1}^M I(t_i, d_j) * y_{ij} \right) * x_i, \forall t_i \in T$$

де: γ_1, γ_2 – коефіцієнти масштабування.

Тоді сумарний ризик задач, що обрані на виконання у спринті:

$$R^S = \sum_{i=1}^N \tilde{R}(t_i) \rightarrow \min$$

Планування спринту – це процес відбору, впорядкування та розподілу задач для виконання протягом обмеженого інтервалу часу з урахуванням обмежених ресурсів та за умови оптимізації цінності спринту.

Цінність спринту (критерій планування) – це баланс між загальною цінністю задач та ризиками, що пов'язані з їх виконанням.

Таким чином, критерій вибору задач та їх розподіл між виконавцями в межах спринту можна сформулювати:

$$W = V^S - R^S \rightarrow \max$$

Отже, задача планування роботи команди протягом спринту може бути сформульована наступним чином:

Знайти вектор ознак вибору задач спринту:

$$X^S = (x_1, x_2, \dots, x_N),$$

Та вектор призначення задач виконавцям:

$$Y^S = (y_{11}, \dots, y_{1m}, y_{21}, \dots, y_{2m}, \dots, y_{nm}),$$

які забезпечують максимальне значення функції цінності спринту

$$W = V^S - R^S$$

та задовольняють обмеженням:

1) сукупна трудомісткість задач спринту не перевищує наявний ресурс команди та розподілені (призначені) задачі не перевищують ресурс відповідного виконавця:

$$\sum_{i=1}^N C(t_i) * E(t_i) * x_i \leq E^S,$$

де: $E(t_i)$ – трудомісткість задачі;

$C(t_i)$ – складність задачі.

$$\sum_{i=1}^N C(t_i) * E(t_i) * y_{ij} \leq A(d_j), \forall j = \overline{1, M}$$

2) Кожен виконавець протягом спринту не може виконувати більше певної (фіксованої) кількості задач одночасно:

$$\sum_{i=1}^N y_{ij} \leq L, \forall j = \overline{1, M}$$

де L – константа, яка визначає максимальну кількість задач на спринт для одного виконавця (зазвичай 1-3 задачі);

3) виконання послідовності задач:

$$G(t_i, t_k) * x_i \geq G(t_i, t_k) * x_k, \forall i, k = \overline{1, N}$$

$$4) \quad x_i \in \{0,1\}, \forall i = \overline{1, N}$$

$$y_{ij} \in \{0,1\}, \forall i = \overline{1, N}, j = \overline{1, M}$$

Таким чином, запропонована модель інтелектуального планування роботи команди ІТ-проєкту формалізує процес вибору задач для спринта з урахуванням множини критичних факторів, що впливають на ефективність командної діяльності. Вона ґрунтується на оптимізаційному підході до максимізації цінності спринта, яка визначається як баланс між сумарною корисністю (цінністю) запланованих до виконання задач та сукупними ризиками, що супроводжують їх виконання та формують дефекти. Розроблена модель включає багатокомпонентну функцію цільової вартості задачі, яка враховує її бізнес-цінність, трудомісткість, рівень узгодженості з виконавцем, задоволеність виконавця та наявність блокування (як потенційний ризик невиконання задачі). Водночас ризики оцінюються з урахуванням нечіткості текстового опису задачі, можливих залежностей і конфліктів, а також потенційних дефектів при виконанні.

Інтеграція логічних змінних x_i та y_{ij} , які відповідають за вибір задач, та їхнє призначення виконавцям, дозволяє будувати формальні обмеження, що гарантують відповідність доступності ресурсів, компетентностей команди та вимог до спринта. Таким чином, модель дозволяє забезпечити узгоджене планування, орієнтоване на досягнення як бізнес-цілей, так і підвищення якості розробки.

Загалом, представлена модель є підґрунтям для підтримки прийняття рішень в процесі планування спринтів, що поєднує математичну строгість та адаптивність до командної структури, а також враховує якісні та кількісні характеристики задач. Запропонована модель є основою для побудови інтелектуальних систем підтримки проєктного менеджменту в ІТ-сфері.

2.3 Метод інтелектуального планування роботи команди ІТ проєкту

У сучасному управлінні ІТ-проєктами ключовим завданням є ефективне планування роботи команди в умовах обмежених ресурсів, високої динаміки вимог і необхідності дотримання термінів. Як вже згадувалося раніше, в контексті Agile-методології та спринтового підходу, критичним є процес формування оптимального набору задач для спринта, а також їх збалансованого розподілу між членами команди. Успішне виконання цього завдання потребує врахування численних факторів: бізнес-цінності задач, ресурсної трудомісткості, міжзадачних залежностей, якості опису задач і відповідності задач компетенціям виконавців.

У даній постановці задачі розглядається проблема оптимізації планування спринта як задача вибору підмножини задач з беклогу проєкту та побудови сталого призначення цих задач розробникам. Метою є максимізація цінності спринта за умови обмеженості ресурсів та врахування пріоритетів, ризиків і уподобань команди. Формалізація цієї задачі дозволяє закласти основу для розробки аналітичних і алгоритмічних рішень, що підтримують процес прийняття управлінських рішень у проєктному середовищі.

Для побудови формальної моделі інтелектуального планування роботи команди ІТ-проєкту необхідно чітко визначити множини задач та виконавців, а також ввести основні параметри, що характеризують ці задачі та впливають на процес їх вибору й розподілу. Зокрема, кожна задача повинна бути оцінена за рядом критеріїв, таких як бізнес-цінність, трудомісткість, складність, послідовність виконання, чіткість формулювання та відповідність уподобанням виконавців. Формальне формулювання задачі планування наведено нижче.

Вихідні дані:

$T = \{t_1, t_2 \dots t_N\}$ – множина задач беклогу

Кожна задача t_i ($i=1..N$) характеризується:

$U(t_i), (i = 1..N)$ – чіткість текстового опису (якість);

$E(t_i), (i = 1..N)$ – трудомісткість (нормо годин);

$S(t_i) \subseteq S, (i = 1..N)$ – навички, які необхідні для вирішення задачі (S – множина всіх навичок);

$V(t_i), (i = 1..N)$ – важливість задачі для проєкту;

$H^-(t_i), (i = 1..N)$ – залежності, множина задач, які необхідно виконати до початку t_i .

Відповідно, $H^+(t_i)$ – множина задач, які залежать від $t_i, (i=1..N)$.

$D = \{d_1, d_2 \dots d_M\}$ – множина розробників (команда), де кожен розробник d_j має:

$S(d_j), (j = 1..M)$ – набір навичок, $S(d_j) \subseteq S$;

$A(d_j), (j = 1..M)$ – доступний робочий час;

$P(d_j), (j = 1..M)$ – уподобання (пріоритети) щодо певних задач (або типів задач) у вигляді вектору, компоненти якого відповідають задачам беклогу;

A_{time}^S – часові обмеження спринту для всієї команди.

Тоді формальна постановка задачі може бути сформульована:

1 Знайти підмножину задач $T' \subseteq T$, які будуть обрані на виконання у спринту таку, що сукупна трудомісткість $E(T')$ не перевищує ресурси команди, та задачі T' враховують залежності і ранжовані з урахуванням чіткості, важливості та ризиків.

2 Знайти бієкцію $f: T' \rightarrow D$, що дозволяє розподілити задачі між виконавцями з урахуванням уподобань та забезпечуючи стійкий розподіл. Стійкий розподіл означає, що немає альтернативних перерозподілів, які були б вигіднішими для виконавців або задач.

Таким чином, задача інтелектуального планування роботи команди IT-проєкту полягає в цілісному циклі функціонування запропонованого фреймворку. Тобто, система має обробляти всі попередньо зібрані показники й обмеження, об'єднувати їх у математичних моделях вибору й розподілу, і на виході генерувати оптимізований набір задач разом із рекомендованими виконавцями та оцінками ризику.

Першим кроком є вибір підмножини задач із загального беклогу проєкту, які доцільно включити до спринту. Цей вибір має враховувати обмеження ресурсів команди (насамперед – загальну трудомісткість задач), а також важливість задач з точки зору бізнес-цілей, рівня складності, наявності залежності між задачами (тобто послідовність їх виконання), чіткість формулювання технічного опису та ризику, пов’язані з їх реалізацією.

Другим кроком є призначення вибраних задач конкретним виконавцям із команди таким чином, щоб максимально врахувати їхні професійні компетенції, індивідуальні вподобання щодо типів задач, а також забезпечити рівномірне та ефективне завантаження. При цьому розподіл задач має бути стабільним і не конфліктним, а призначення повинно підвищувати мотивацію та задоволеність членів команди.

Метою задачі інтелектуального планування є досягнення збалансованого плану спринту, який одночасно:

- максимально підвищує сумарну цінність реалізованих задач для проєкту;
- мінімізує пов’язані з цими задачами штрафи;
- забезпечує ефективне використання наявних ресурсів;
- підвищує якість та надійність розробки;
- враховує людський фактор у процесі розподілу задач.

Отже, задача інтелектуального планування є багатоаспектною оптимізаційною проблемою, що поєднує технічні, управлінські та соціальні аспекти командної роботи в ІТ-проєктах.

Для розв’язання сформульованої задачі планування роботи команди ІТ-проєкту доцільно застосувати методи інтелектуального моделювання, які дозволяють ефективно враховувати як формалізовані обмеження (ресурси, залежності між задачами, складність), так і нечіткі, суб’єктивні чинники (уподобання виконавців, чіткість формулювань, потенційні ризики). Враховуючи складність задачі, її багатокритеріальний характер і комбінаторну

природу, використання класичних евристик або ручного планування втрачає ефективність у масштабних проєктах.

У зв'язку з цим у подальшому розглядається підхід, заснований на побудові математичної моделі, що формалізує цільову функцію спринту, як різницю між сумарною очікуваною цінністю виконаних задач і пов'язаними з ними ризиками. Для врахування всіх аспектів задачі використовується система допоміжних оціночних функцій, що відображають якість текстового опису задач, відповідність виконавця вимогам задачі, задоволення від призначення задач, а також узгодженість призначень із залежностями між задачами.

Запропонований метод інтелектуального планування роботи команди ІТ-проєкту передбачає побудову оптимального спринту шляхом поетапного аналізу задач та їхнього розподілу між виконавцями з урахуванням як кількісних, так і якісних характеристик.

На першому етапі здійснюється оцінка кожної задачі з беклогу за низкою критеріїв: бізнес-цінність (тобто її важливість для досягнення цілей проєкту), трудомісткість (обсяг необхідних ресурсів), складність, залежність від інших задач, а також чіткість формулювання технічного опису, що впливає на розуміння задачі виконавцем.

Наступним кроком є відбір підмножини задач для включення у спринт. Відбір відбувається з урахуванням обмеженості ресурсів команди (сумарна трудомісткість не повинна перевищувати доступні зусилля команди), а також з урахуванням послідовностей та залежностей між задачами. У пріоритеті - задачі з високою бізнес-цінністю, чітко сформульовані та такі, що можуть бути виконані без порушення логічного порядку.

Після формування переліку задач для спринту виконується розподіл задач між членами команди. При цьому враховуються наступні фактори:

- відповідність компетенцій виконавця вимогам задачі;
- індивідуальні уподобання та преференції виконавця;
- завантаженість кожного члена команди;
- задоволеність від призначення задачі (мотиваційний компонент);

- уникнення конфліктів і блокувань, спричинених послідовністю задач.

На завершальному етапі для кожної задачі обчислюється інтегральна оцінка: поєднання очікуваної цінності задачі та пов'язаних з нею ризиків. Рішення про включення задачі у фінальний план приймається на основі максимізації різниці між сумарною цінністю та ризиками всього спринта. У результаті формується план задач і їхнє розподілення по виконавцях, яке є збалансованим, обґрунтованим і стійким.

Розглянемо формалізований опис методу інтелектуального планування роботи команди ІТ-проєкту.

Етап 1 Оцінка та ранжування задач.

Мета – визначити пріоритет задач в беклозі.

Сукупний ризик задачі:

$$R(t_i) = \gamma_1(1 - U(t_i))V(t_i) + \gamma_2 \sum_{t_k \in H^+(t_i)} (1 - U(t_k)), i = 1..N$$

враховує чіткість формулювання (якість) та важливість задачі, а також залежності. Пріоритет задачі:

$$P(t_i) = \alpha V(t_i) + \beta C(t_i) - R(t_i), i = 1..N$$

Етап 2 Вибір задач (фільтрація).

Мета – обрати підмножину $T' \subseteq T$ таку, що:

$$\begin{aligned} \max_{T' \subseteq T} \sum_{t_i \in B'} P(t_i), i = 1..N \\ \sum_{t_i \in T'} E(t_i) \leq \sum_{d_j \in D} A(d_j), j = 1..M, i = 1..N \end{aligned}$$

Етап 3 Розподіл задач.

Мета – призначити задачі виконавцям

$$\max \sum_{t_i \in T'} W(t_i, f(t_i)), i = 1..N$$

де: $W(t_i, d_j), i = 1..N, j = 1..M$ – функція задоволеності (відповідності задачі та розробника)

$$\begin{aligned} f_i & T' \rightarrow D, \quad f(t_i) = d_j, \quad S(d_j) \subseteq S(t_i) \\ \sum_{d_j \in f^{-1}(t_i)} E(t_i) & \leq A(t_i) \quad \forall t_i \in T, i = 1..N, j = 1..M \end{aligned}$$

Таким чином, запропонований метод інтелектуального планування роботи команди ІТ-проєкту є комплексним підходом, що забезпечує прийняття обґрунтованих рішень на всіх етапах формування спринта — від вибору задач до їх розподілу між виконавцями. Його ключовою перевагою є здатність інтегрувати різноманітні чинники, як-то бізнес-цінність, технічна складність, міжзадачні залежності, чіткість формулювання та людські уподобання в єдину систему оцінювання та оптимізації. Метод дозволяє не лише підвищити ефективність використання ресурсів команди, а й зменшити ризики, пов'язані з неякісним формулюванням задач або конфліктами в їх розподілі. Завдяки формалізації цінності та ризику задачі, а також завдяки урахуванню преференцій виконавців метод сприяє створенню збалансованих і сталих планів, що орієнтовані як на досягнення проєктних цілей, так і на підтримку мотивації команди.

Незважаючи на те, що оптимізація вибору та розподілу задач є центральним елементом планування, важливо, також, враховувати потенційні ризики, пов'язані з якістю їх виконання. Одним із ключових факторів, що

впливають на загальний успіх спринту, є ймовірність виникнення дефектів під час реалізації задач.

Для забезпечення надійності й стабільності виконання спринту доцільно включити до моделі компонент, що дозволяє прогнозувати рівень дефектності задачі ще на етапі планування. Це дає змогу завчасно виявляти задачі з підвищеним ризиком, уникати їх надмірної концентрації в одному спринті, а також коригувати призначення задач відповідно до рівня впевненості у їх коректному виконанні.

Далі розглядається підхід до оцінювання дефектності задач на основі сукупності характеристик, включаючи чіткість формулювання, складність, рівень відповідності виконавця, а також потенційні залежності між задачами.

Очікуваний рівень дефектності z_i може бути визначений за допомогою предикативної моделі, яка оцінює ймовірність помилок на основі:

- складності текстового опису задачі (метрики обробки природної мови: довжина, кількість термінів, неоднозначність);
- історичних даних про аналогічні задачі (попередня дефектність);
- рівня компетентності потенційних виконавців;
- ступеня залежностей задачі від інших задач.

Метою моделі є побудова плану спринту, що мінімізує ймовірність дефектів при максимізації бізнес-цінності за обмежених ресурсів команди.

Нехай маємо множину задач $T = \{t_1, t_2, \dots, t_N\}$ кожна з яких характеризується вектором ознак: $x_i \in \mathbb{R}^N$:

$t_i \in T$ – тип задачі, наприклад: $T = \{fr, bck, ga\}$;

$c_i \in C$ – складність, наприклад: $C = \{1, 2, 3, 4, 5\}$;

$u_i \in [0, 1], u_i = U(t_i), i = 1..N$ – чіткість текстового представлення (оцінка на основі LLM);

$v_i = V(t_i), i = 1..N$ – важливість (цінність) задачі;

d_j – виконавець (id виконавця);

$\tau_i^{plan}, \tau_i^{fact}$ – час виконання або трудомісткість;

$r_i \in \{0,1\}$ – ознака відкату (1 – був, 0 – ні), тобто, чи поверталася задача після виконання спринту;

$S_i \in [0,1]$ – оцінка задачі після виконання;

$i = 1..N$.

На основі цих ознак можна побудувати оцінку рівня дефектності задачі:

$$z_i = \alpha * \max\left(\frac{\tau_i^{fact} - \tau_i^{plan}}{\tau_i^{plan}}; 0\right) + \beta * r_i + \gamma * (1 - S_i)$$

де $-\left(\frac{\tau_i^{fact} - \tau_i^{plan}}{\tau_i^{plan}}; 0\right)$ – перевищення планового часу;

- $\beta * r_i, i = 1..N$ – наявність відкату;
- $\gamma * (1 - S_i), i = 1..N$ – штраф за низьку оцінку;
- α, β, γ – масштабні коефіцієнти, які необхідні для приведення всіх складових до однієї шкали.

Задача: побудувати функцію

$$f: \mathbb{R}^n \rightarrow \mathbb{R} \geq 0$$

Яка апроксимує залежність

$$y_i = f(x_i) + \varepsilon_i,$$

де $- x_i$ – вектор ознак задачі $i = 1..N$;

- ε_i – випадкова похибка $i = 1..N$.

В якості моделі прогнозування використовується Random Forest Regressor

$$\hat{f}(x) = \frac{1}{T} * \sum_{i=1}^N h_i(x),$$

де $H = \{h_1, h_2, \dots, h_N\}$ – множина дерев, які навчаються на випадковій підмножині ознак та прикладів.

Оцінка якості моделі:

$$MAE = \frac{1}{M} * \sum_{i=1}^M |y_i - \hat{f}(x_i)|$$

Урахування прогнозованого рівня дефектності задачі є важливим доповненням до моделі інтелектуального планування, що дозволяє не лише обирати найбільш цінні та ресурсно обґрунтовані задачі, але й проактивно керувати якістю їх виконання. Запропонований підхід до оцінки дефектності ґрунтується на багатофакторному аналізі: він враховує як об'єктивні характеристики задач (складність, залежності, відповідність виконавцю), так і суб'єктивні аспекти, зокрема чіткість їх формулювання. Такий підхід дозволяє ще до початку реалізації визначати задачі з підвищеним ризиком виникнення помилок і приймати управлінські рішення для зниження ймовірності дефектів.

Таким чином, запропонований метод інтелектуального планування є універсальним інструментом підтримки прийняття рішень у процесі організації командної роботи в рамках ІТ-проєкту. Його концепція базується на поєднанні формального оптимізаційного підходу з інтелектуальними моделями оцінювання, що дозволяє інтегрувати як структуровані дані (трудомісткість, залежності, компетенції), так і слабо формалізовані фактори (уподобання виконавців, якість опису задач, ризики дефектів). Метод охоплює повний цикл планування: від аналізу задач у беклозі та їх відбору до спринту – до персоналізованого розподілу задач між членами команди з урахуванням обмежень та мотиваційних чинників. В результаті формується збалансований план, який орієнтований на досягнення бізнес-цілей, ефективне використання ресурсів та підвищення якості розробки. Побудований метод створює основу для створення адаптивної, обґрунтованої та стійкої стратегії планування в умовах динамічного та невизначеного середовища, що є характерним для сучасних ІТ-проєктів.

2.4 Модель розподілу задач спринту

Ефективне призначення задач – одна з найскладніших управлінських задач у проектах, що розробляються за Agile-підходами. Хоч правильний розподіл робіт є критично важливим для результативності гнучких команд, як вже зазначалося, даний аспект поки що вивчений недостатньо глибоко. Більшість наявних підходів спираються переважно на опитування, що обмежує масштаб і деталізацію зібраних даних. Відсутність зрозумілих принципів призначення задач, зрештою, знижує якість, продуктивність і мотивацію виконавців.

Аналіз різних підходів до розподілу задач показує кілька ключових напрямів, які покликані зменшити невизначеність і підвищити ефективність роботи команди. В дослідженнях [43, 6] розподіл виконується на основі компетенцій. Автори розглядають вирішення цієї проблеми за допомогою узгодження задач проекту із кваліфікацією та компетенціями членів команди. Задачі зіставляють із навичками розробників, щоб кожен отримував роботу, яка відповідає його досвіду. В роботах [41, 26] для розподілу задач серед виконавців проекту застосовують генетичні алгоритми. Вони ефективно вирішують конфлікти ресурсів та оптимізують планування через моделювання різних сценаріїв і пошук оптимальних рішень. Фактично, відбувається розрахунок безлічі можливих розкладів і знаходиться той, де ресурси використано найкраще й конфліктів мінімум. Дослідження [123, 122] використовують потенціал краудсорсингу та розподілених команд. Запропоновані Моделі враховують активність учасників, складність та взаємні залежності задач, завдяки чому підвищується швидкість спільної роботи. Ці підходи демонструють помітні покращення у використанні ресурсів і швидкості виконання задач. В роботах [121, 81, 110, 90] застосовується методи штучного інтелекту та машинного навчання. Нейромережі й інші ML-методи прогнозують, кому й коли краще доручити конкретну задачу, покладаючись на історичні дані команди. Використання багатокритеріальних моделей [13, 7, 82] забезпечує структурований підхід до розподілу задач. Оцінюючи кілька факторів

одночасно, ці моделі забезпечують збалансоване ухвалення рішень та інтегрують якісні та кількісні метрики. Запропоновані інтегровані системи для оптимізації задач [9, 82] підкреслюють важливість поєднання їх розподілу з оптимізацією ресурсів. Такі системи знижують операційні витрати та скорочують терміни планування, прискорюючи виконання результати проєктів.

Підсумовуючи, можна зазначити, що саме комбінація компетентнісного підбору, AI-прогнозування та багато-критеріальної оцінки дає найвідчутніший приріст продуктивності й надійності виконання спринтів

Розглянемо формальну постановку розподілу задач спринта між виконавцями. Нехай

$T^S = \{t_1, t_2, \dots, t_N\}$ – множина задач з беклогу проєкту, які відібрано для виконання на спринт;

$D = \{d_1, d_2, \dots, d_M\}$ – множина розробників (команда проєкту);

Кожна задача $t_i \in T^S$ характеризується вимогами $C(t_i)$ щодо її виконання (складність, кваліфікація виконавця, термін виконання). Для кожного виконавця (розробника) $d_j \in D$ визначена його кваліфікація $Q(d_j)$, яку можна співставити з вимогами задач спринту. Кожен виконавець $d_j \in D$ має власні преференції щодо задач беклогу $P(d_j, t_i)$.

Нехай $E(t_i), \forall i = \overline{1, N}$ визначає необхідні ресурси, у тому числі час виконання, а $A(d_j), \forall j = \overline{1, M}$ – доступний робочий час кожного виконавця $d_j \in D$. Позначимо загальні доступні ресурси команди на спринт E^S , що у загальному випадку може враховувати більш складні залежності, ніж загальний доступний час.

Для кожної пари $(t_i, d_j), t_i \in T^S, d_j \in D$, вважаємо визначеними наступні оцінки:

$DS(d_j, t_i)$ – задоволеність виконавця d_j призначенням йому задачі t_i , що враховує преференції виконавця;

$CS(t_i, d_j)$ – узгодженість призначення задачі t_i виконавцю d_j , що враховує відповідність виконавця вимогам задачі;

$I(t_i, d_j), t_i \in T^S, d_j \in D$ – індикатор умови, що для задачі t_i існує більш кращий розподіл (існує виконавець, який краще відповідає вимогам задачі, ніж d_j) та для виконавця d_j існує більш приваблива задача, ніж t_i ;

$G(t_k, t_i), t_k, t_i \in T^S$ – індикатор послідовності виконання задач таким чином, що задача t_k повинно бути виконана до початку виконання t_i

Тоді постає задача знайти таку бієкцію $f: T^S \rightarrow D$, що дозволяє розподілити задачі між виконавцями з урахуванням уподобань та забезпечуючи стійкий розподіл. Під стійким розподілом будемо розуміти такий розподіл, для якого немає альтернативних перерозподілів, які були б вигіднішими для виконавців або задач.

Отже, розглянемо множину задач спринту $T^S = \{t_1, t_2, \dots, t_N\}$ та множину виконавців $D = \{d_1, d_2, \dots, d_M\}$, які представляють можливі варіанти призначення задач окремим розробникам з урахуванням їх компетенцій, преференцій та доступності ресурсів.

Метою розподілу є побудова бієкції

$$q: D \rightarrow T^S,$$

яка призначає кожному виконавцю $d_j \in D$ задачу $q(d_j) \in T^S$ таким чином, щоб:

- максимізувати загальну відповідність між задачами і виконавцями:

$$\max \sum_{d_j \in D} W(d_j, q(d_j)),$$

де - $W(d_j, t_i)$ - функція відповідності, яка враховує уподобання виконавця, релевантність компетенцій і мотиваційні фактори;

- дотримуватись обмежень на відповідність задачі виконавцю:

$$S(d_j) \subseteq S(t_i), \text{ де } q(d_j) = t_i,$$

тобто виконавець повинен володіти навичками, достатніми для виконання задачі.

Ця модель має очевидну аналогію із задачею про стабільний шлюб, де сторони мають свої пріоритети, і метою є побудова стійкого розподілу, за якого немає «блокуючих пар» - таких, які віддали б перевагу альтернативному партнеру.

Твердження (про існування розв'язку). Нехай кожен виконавець $d_j \in D$ має повний і транзитивний список преференцій щодо задач $t_i \in T^S$, і кожна задача t_i допускає призначення не більше ніж q_i виконавців, з урахуванням обмежень на відповідність і ресурси. Тоді існує стійкий розподіл задач між виконавцями відповідно до функції відповідності $W(d_j, t_i)$, який задовольняє всі обмеження моделі.

Доведення спирається на узагальнення алгоритму Гейла–Шеплі для задачі з ємностями (capacitated stable matching) [34]. Переваги задачі з обмеженнями у тому, що:

- пріоритети (переваги виконавців і задач) є впорядкованими;
- кожен виконавець має обмежене число допустимих задач (через $S(d_j) \subseteq S(t_i)$);
- задачі мають обмежену ємність за ресурсами.

У цьому випадку ітеративне призначення задач з урахуванням найкращих допустимих варіантів призводить до існування стійкого призначення, при якому жодна пара задача-виконавець не утворює блокуючу конфігурацію.

Основою дослідження є створення симуляційного середовища, що оперує штучно згенерованими профілями розробників і задач та дозволяє відтворити весь цикл призначення задач виконавцям з огляду на реалістичні обмеження, пріоритети й інші, характерні для ІТ-проектів, фактори.

Розглядаємо п'ять ключових алгоритмів: SOSM, EADAM, MESMA, RSD та ТТС. Пропонований підхід включає генерацію синтетичних даних, що імітують профілі розробників та задач, обчислення показників сумісності, побудову

списків преференцій, реалізацію алгоритмів та їх оцінку за допомогою ряду метрик. Це дозволяє провести всебічний аналіз ефективності кожного алгоритму в різних сценаріях та умовах. Такий підхід забезпечує ґрунтовну основу для порівняння алгоритмів та формулювання рекомендацій щодо їх застосування в реальних умовах ІТ-проектів.

Алгоритм Student-Optimal Stable Mechanism (SOSM) базується на алгоритмі відкладеного прийняття Гейла-Шеплі [34]. У цьому контексті розробники розглядаються в ролі студентів, які пропонують завданням (школам) утворити пару в порядку своїх списків преференцій. Кожне завдання, також, має список пріоритетних розробників. Алгоритм проходить у декілька раундів.

1 Фаза пропозицій: розробники ранжують задачі за своїми преференціями та кожен розробник "пропонує" себе найбільш бажаній задачі.

2 Фаза прийняття: Задачі мають пріоритетний список розробників. Кожна задача переглядає пропозиції та тимчасово приймає найбільш пріоритетного розробника, інших відхиляє.

3 Ітерація: Відхилені розробники переходять до наступної задачі у своєму списку.

Процес триває, доки всі розробники не будуть призначені або не будуть відхилені всіма задачами. Алгоритм SOSM гарантує стабільне зіставлення, оптимальне для розробників, тобто жоден розробник не може отримати краще призначення без погіршення становища когось іншого. Це сприяє справедливому розподілу та підвищенню мотивації розробників.

Алгоритм Efficiency-Adjusted Deferred Acceptance Mechanism (EADAM) [60, 61] у порівнянні з SOSM прагне досягти Парето-ефективності, але може порушувати стабільність заради підвищення ефективності. EADAM може призвести до кращого загального задоволення розробників їхніми призначеннями порівняно з SOSM. В контексті розподілу задач між розробниками в ІТ-проектах алгоритм EADAM можна описати наступним чином:

1 Спочатку застосовується стандартний алгоритм відкладеного прийняття для отримання початкового розподілу.

2 Ітеративно визначаються та усуваються "переривники" - це пари (розробник, задача), де розробник пропонує себе на задачу, спричиняє відхилення іншого розробника, але пізніше сам відхиляється. В результаті, жоден розробник не може отримати краще призначення без погіршення ситуації для інших.

3 Процес повторюється, доки не будуть усунуті всі переривники.

EADAM прагне знайти баланс між ефективністю та справедливістю розподілу задач, що може бути корисним у динамічному середовищі ІТ-проектів. Алгоритм Maximally Efficient and Stable Matching Algorithm (MESMA) [35] спрямований на знаходження стабільного розподілу задач між розробниками з максимальною загальною вагою (ефективністю) та зосереджується на максимізації загальної ефективності зіставлення, забезпечуючи стабільність. Вхідними даними є система переваг розробників щодо задач (і навпаки) та вагова функція, яка визначає ефективність кожного можливого призначення. MESMA використовує підхід лінійного програмування для вирішення задачі максимізації ваги стабільного зіставлення. Але алгоритм може бути обчислювально складним для великих проектів, оскільки проблема максимально зваженого стабільного зіставлення є NP-складною.

Random Serial Dictatorship (RSD) [16, 12] – це простий, стратегічно безпечний алгоритм, що призначає розробників задачам на основі випадково визначеного порядку. Перший розробник у послідовності обирає найбільш пріоритетну для нього задачу з усього набору доступних задач. Другий розробник обирає свою, найбільш пріоритетну, задачу з тих, що залишились. Процес продовжується, доки всі розробники не оберуть задачу або не закінчатся доступні задачі.

Перевагою RSD є стратегічна безпечність, тобто розробники не мають стимулів неправильно повідомляти свої переваги. RSD забезпечує простий та

справедливий метод розподілу задач, але може потребувати додаткових механізмів для оптимізації ефективності розподілу в контексті ІТ-проектів.

Алгоритм Top Trading Cycles (TTC) [52, 25] також дозволяє оптимізувати розподіл задач з урахуванням побажань розробників, забезпечує Парето-ефективний розподіл та є стратегічно безпечним. Алгоритм працює ітеративно: кожен розробник вказує на найбільш бажану для нього задачу зі списку доступних. Формуються цикли, де розробники вказують один на одного через бажані задачі. Для кожного виявленого циклу відбувається обмін задачами між розробниками. Розробники, які брали участь в обміні, видаляються з подальшого розгляду разом з їхніми новими задачами. Процес повторюється для решти розробників, доки всі не отримають призначення або не залишаться можливостей для обміну.

Таким чином, для порівняльного аналізу обрано п'ять алгоритмів, які дозволяють оптимізувати розподіл задач з урахуванням побажань розробників, їх кваліфікації та преференцій з точки зору проекту.

Для проведення експериментів з п'ятьма алгоритмами зіставлення (SOSM, EADAM, MESMA, RSD, TTC) запропоновано комплексний симуляційний фреймворк [4], який моделює процес розподілу задач між розробниками в ІТ-командах та складається з наступних ключових компонентів:

- генератор синтетичних даних, який створює реалістичні профілі розробників з атрибутами, такими як навички, досвід, навантаження, а також характеристики задач (складність, пріоритет тощо);
- модуль обчислення сумісності, який розраховує показник сумісності між кожною парою розробник-задача;
- генератор списків преференцій для розробників та задач;
- модуль реалізації алгоритмів зіставлення;
- модуль оцінки продуктивності алгоритмів.

Для оцінки алгоритмів були обрані метрики:

1 Загальний показник сумісності (Total Compatibility Score), який обчислюється як сума всіх значень сумісності між призначеними розробниками та задачами:

$$S_{total} = \sum_{(d,t) \in M} C(d, t),$$

де M – множина пар (розробник, задача) у зіставленні, $C(d, t)$ - оцінка сумісності між розробником d і задачею t .

2 Кількість блокуючих пар (Blocking Pairs). Блокуючою парою вважається пара (d, t) якщо розробник d віддає перевагу задачі t над своїм поточним призначенням та задача t віддає перевагу розробнику d над своїм поточним призначенням. Кількість блокуючих пар розраховується наступним чином:

$$B = \sum_{(d,t) \in M} I[(t \succ_d M(d)) \wedge (d \succ_t M(t))],$$

де - $I[\cdot]$ - індикаторна функція, що дорівнює 1, якщо умова виконується, і 0 інакше, $M(d)$ - задача, призначена розробнику d , $M(t)$ - розробник, призначений задачі t ;

- $t \succ_d M(d)$ означає, що задача t є більш бажаною для розробника d , ніж призначена йому;

- $d \succ_t M(t)$ означає, що розробник d є більш бажаним для задачі t , ніж та, що йому призначили.

3 Задоволення розробників (Developer Satisfaction) визначається як середній ранг призначених задач у списках преференцій розробників за формулою:

$$S_{dev} = \frac{1}{|D|} \sum_{d \in D} rank_d(M(d)),$$

де - D – множина всіх розробників, $rank_d(M(d))$ - позиція задачі t у списку преференцій розробника d , $M(d)$ - задача, призначена розробнику d .

Нижче значення S_{dev} означає вище задоволення розробників.

4 Час виконання - обчислювальний час, необхідний кожному алгоритму для досягнення рішення.

Таким чином, розроблена модель узагальнює клас задач із розподілу, для яких можна довести існування стійкого або квазістійкого розв'язку, що відкриває перспективи для подальшого аналізу, розробки ефективних алгоритмів і адаптації до різних контекстів. Вона створює платформу для міждисциплінарного синтезу між формальними методами і реальними прикладними задачами в управлінні IT-проєктами.

2.5 Висновки до розділу 2

Визначено, що критичними джерелами ризиків планування спринту є неякісно сформульовані задачі беклогу, недосконалі канали командної комунікації та непрозорий розподіл задач. Для їх подолання обґрунтовано доцільність використання методів інтелектуального аналізу природної мови (NLP) для оцінки чіткості описів задач, Process Mining – для моніторингу поведінки команди за журналами подій, а також симуляційних і matching-алгоритмів — для стабільного призначення задач виконавцям.

Запропоновано опорну модель управління командою IT проєкту, яка задає мінімально необхідний набір елементів і потоків даних, що дозволяють переходити від вхідних артефактів (лог-файли, текстові описи) до придатних для ухвалення рішень показників якості та ризику.

Запропоновано фреймворк інтелектуального планування роботи команди IT проєкту, який ґрунтується на використанні моделей оцінювання, прогнозування та рекомендаційних систем, і який конкретизує, як саме потоки даних трансформуються в оптимізований план спринту. Основу фреймворку складає інтеграція моделей штучного інтелекту з практиками Agile-менеджменту, що дозволяє забезпечити обґрунтовану підтримку прийняття рішень, підвищуючи ефективність командної роботи та якість програмного продукту.

Запропонована модель інтелектуального планування роботи команди ІТ-проекту формалізує процес вибору задач для спринту з урахуванням множини критичних факторів, що впливають на ефективність командної діяльності.

Запропонована модель розподілу задач у спринті дозволяє формалізувати складний процес призначення задач виконавцям як оптимізаційну задачу на множині бієктивних відображень із численними обмеженнями – ресурсними, кваліфікаційними, мотиваційними. Урахування переваг виконавців, відповідності компетенцій, а також обмежень на доступність задач і ресурсів, робить її придатною для побудови стійких і практично реалізуємих планів. Запропонований фреймворк розподілу задач дозволяє проводити масштабні експерименти з різними конфігураціями команд та проектів, забезпечуючи глибокий аналіз ефективності кожного алгоритму в контексті розподілу задач в ІТ-проектах.

РОЗДІЛ 3 АНАЛІЗ ТЕКСТОВОГО ОПИСУ ЗАДАЧ СПРИНТУ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ

3.1 Особливості аналізу текстового опису задач проєкту

У системі управління проєктами команда проєкту формулює задачу таким чином, щоб описи задач зрозуміло передавали суть необхідних дій. Критично важливо забезпечити, щоб всі учасники команди однаково розуміли задачі, мінімізуючи ризики виникнення двозначностей і непорозумінь. З досвіду практиків визначено, що виконавці часто доповнюють текстові описи задач коментарями, відео, зображеннями тощо, що збагачує контекст. Зберігання всієї цієї інформації в інформаційній системі надає команді зручний інструмент для спільної роботи, спрощує комунікацію та допомагає забезпечити точність та повноту розуміння задач усіма учасниками проєкту.

Для розробки моделі ідентифікації задач з неякісним формулюванням у спринті ІТ-проєкту, пропонується комплексний підхід, який включає на першому етапі збір та обробку текстових описів задач, використання алгоритмів машинного навчання для класифікації та залучення експертних оцінок для вдосконалення якості сприйняття задач командою проєкту.

На другому етапі текстові описи задач спринту перетворюються у векторні представлення для наступного використання в моделі класифікації, що передбачає отримання векторної репрезентації кожної задачі. Цей процес розпочинається з перетворення кожної задачі на набір tokenів за допомогою tokenізатора. Tokenізатор, також відомий як лексичний аналізатор, виконує поділ тексту на невеликі одиниці, які отримують назву токени [15]. Токени можуть представляти собою слова, символи, словосполучення або, навіть, речення, в залежності від типу задачі та мови обробки. Tokenізація представляє текст у форматі, придатному для наступного аналізу та обробки комп'ютерними алгоритмами.

Існує різноманіття підходів до отримання векторних представлень текстових фрагментів. Ці вектори можуть бути згенеровані за допомогою алгоритмів навчання без учителя або використання перенавчених нейронних мереж. Вибір методу векторизації текстів є важливим завданням дослідження. Для отримання векторних репрезентацій можуть використовуватись такі методи як:

- Word2Vec – метод векторного представлення слів [117];
- TF-IDF (Term Frequency-Inverse Document Frequency) – статистичний метод, що широко використовується в галузі інформаційного пошуку та аналізу текстових даних [106];
- GloVe (Global Vectors for Word Representation) – метод є різновидом подання слів з використанням статистичних властивостей слів у великих текстових корпусах [89];
- FASTTEXT – метод є різновидом методів векторного представлення слів, заснованих на нейронних мережах, і має кілька унікальних особливостей, які роблять його популярним і ефективним в обробці природної мови [85].

На третьому етапі виконується розмітка задач за допомогою анотацій експертів, і формується два класи – «зрозумілі» та «не зрозумілі» задачі. Однак виникають ключові питання дослідження: яка кількість експертів є достатньою для забезпечення незалежної та достовірної оцінки якості формулювання задач? Чи може знання експертами контексту проєкту впливати на результати їхнього експертного оцінювання? Важливим аспектом є також визначення методів для оцінки узгодженості думок експертів та встановлення критеріїв відповідності певним класам для подальшого використання у навчанні моделі класифікації.

Четвертий етап включає навчання класифікатора, де для класифікації текстових фрагментів ефективними можуть бути класифікатор Байєса та нейронні класифікатори. Визначення оптимальної моделі класифікації, яка відповідає особливостям вихідних даних конкретного проєкту, є завданням експериментального дослідження. Важливо зауважити, що на результати

класифікації буде впливати не лише обраний метод класифікації, але й підхід до векторизації текстів.

Як було зазначено, для класифікації текстових виразів гарно підходить класифікатор Байєса та нейронні класифікатори.

В якості енкодерно декодерної архітектури, тобто вхідні дані спочатку згортаються для отримання репрезентації, а потім розгортаються, виконуючи специфічне завдання. Основною ідеєю є використання механізму уваги (self-attention mechanism), який дозволяє моделі ефективно взаємодіяти з різними частинами вхідного тексту одночасно. Модель складається з кількох шарів, кожен із яких містить механізми уваги та позначення (нормалізація, активація).

Формула механізму уваги в трансформерах виглядає наступним чином:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V,$$

де - Q, K, V – матриці запитів, ключів та значень відповідно;

- d_k – розмірність вектору ключа.

Також, було виявлено, що використання лише одного рівня уваги менш ефективне, ніж використання механізму уваги на різних лінійних проекціях запиту, ключа та значень, що навчаються, з подальшим збором їх усіх разом та додаванням ще однієї лінійної проекції в оригінальний розмір репрезентації. Таку операцію можна описати наступним чином:

$$MultiHead(Q, K, V) = Concat(head_1, \dots, head_h)W^0,$$

де - $head_i = Attention(QW^Q_i, KW^K_i, V W^V_i)$.

Однією з ключових інновацій моделі глибокого навчання BERT, яка використовує енкодерну модель, що не виконує авторегресійного передбачення, є двостороннє контекстне кодування, що дозволяє моделі використовувати інформацію з усіх напрямків в тексті. Під час тренування використовуються два види вхідних представлень: сегментні та позиційні ембедінги.

Формули для позиційних та сегментних репрезентацій в нейронній мережі BERT виглядають так:

$$Positional = Embedding(w_i) + PositionalEncoding(i),$$

$$Segments = Embedding(s_j),$$

- де
- w_i – токен на позиції i ;
 - s_j – сегментний ідентифікатор речення j ;
 - $PositionalEncoding(i)$ – вектор позначення позиції i у вхідному реченні.

Існуює багато варіантів моделювання позиції, такі як рівномірно зростаючі функції або періодичні функції.

Першим методом класифікації було обрано класифікатор Байєса. Він ґрунтується на теоремі Байєса, яка визначає ймовірність події на основі ймовірностей попередніх подій. У випадку класифікації тексту, класифікатор Байєса використовує ймовірності того, що певне слово або фраза належать до певного класу.

Основна ідея класифікатора Байєса полягає у визначенні ймовірностей для кожного класу, враховуючи входження конкретних слів або фраз у текст. Математично це виражається наступним чином:

$$P(C_x|X) = \frac{P(X|C_k) \cdot P(C_k)}{P(X)},$$

- де
- $P(C_x|X)$ – ймовірність того, що документ X належить класу C_k ;
 - $P(X|C_k)$ – ймовірність документа X за умови класу C_k ;
 - $P(C_k)$ – апіорна ймовірність класу C_k ;
 - $P(X)$ – загальна ймовірність документа X .

Для спрощення розрахунків, часто використовують додаткові припущення, наприклад, незалежність входжень слів, що дозволяє виразити ймовірність документа за умови класу як добуток ймовірностей окремих слів:

$$P(X|C_k) = P(w_1|C_k) \cdot P(w_2|C_k) \cdot \dots \cdot P(w_n|C_k),$$

де w_i – окреме слово з документа.

Для класифікатора Байєса векторні репрезентації отримано з потренованої нейронної мережі BERT. Далі ці репрезентації використовуються як самостійні набори даних, не пов'язані з текстом, на базі яких тренується класифікатор.

Завершальний етап передбачає валідацію роботи моделі на наявних даних та вибір оптимального класифікатора для подальшого застосування його в інших проєктах. Одним із ключових критеріїв оцінки моделі є F-міра. Це метрика, яка комбінує точність (precision) та повноту (recall) у єдиний числовий показник, дозволяючи оцінити збалансованість між цими двома міркуваннями. F-міра надає компроміс між точністю та повнотою, що особливо важливо при роботі з невідсортованими класами чи незбалансованими наборами даних. Загальна мета полягає в досягненні найкращого можливого балансу між цими двома аспектами, щоб модель мала високу точність та, одночасно, уникати пропусків (помилки пропуску) виявлення певного класу.

Запропонований підхід до побудови класифікатора дозволяє систематизувати задачі спринту за категоріями відповідно до якості їх формулювань. Крім того, цей підхід враховує контекст проєкту та експертні думки, що робить його гнучким та адаптивним до специфіки конкретного проєкту та особливостей команди, що є критичним для подальшого вдосконалення процесів управління проєктами.

З метою наочної візуалізації, 2048-вимірні векторні репрезентації були трансформовані в двовимірний простір використовуючи методи PCA [63] та t-SNE [72], що дозволило ефективно їх відобразити.

Розглянемо метод аналізу головних компонент (PCA). Це статистичний метод, який використовується для зниження розмірності даних шляхом перетворення від багатьох змінних до меншої кількості некорельованих змінних, званих головними компонентами. Цей метод допомагає виділити основну структуру даних, мінімізуючи втрату інформації.

Математично, якщо X — вихідний датасет розмірністю $m \times n$ (де m — кількість спостережень, а n — кількість змінних), PCA шукає новий ортогональний базис для X , де перший головний компонент має максимальну дисперсію, другий — максимальну дисперсію серед тих, що є ортогональними до першого, і так далі.

Перший головний компонент $PC1$ можна знайти шляхом максимізації дисперсії:

$$\max\{\text{var}(Xw)\} = \max\left\{\frac{w^T X^T X w}{w^T w}\right\},$$

де w — ваговий вектор.

Метод t-SNE (t-distributed Stochastic Neighbor Embedding) — це метод машинного навчання для візуалізації, заснований на зниженні розмірності, що зокрема ефективний для візуалізації великих об'ємів даних у дво- або тривимірний простір. T-SNE спочатку обчислює ймовірності сусідства між парами об'єктів у високовимірному просторі таким чином, що об'єкти ближчі один до одного мають вищу ймовірність, а далекі об'єкти — низьку. Потім t-SNE шукає низьковимірне представлення даних, що найкраще відображає ці ймовірності.

Математично, ймовірність $p_{j|i}$, що об'єкт j є сусідом об'єкта i у високовимірному просторі, визначається як:

$$p_{j|i} = \frac{\exp(-||x_i - x_j||^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-||x_i - x_k||^2 / 2\sigma_i^2)}$$

де x_i, x_j — вектори високовимірного простору, а σ_i — вибране стандартне відхилення для об'єкта i .

У низьковимірному просторі ймовірності $q_{j|i}$ визначаються аналогічно, але з використанням t-розподілу з одним ступенем свободи замість нормального розподілу, щоб запобігти згущенню об'єктів у центрі мапи. Оптимізація полягає у мінімізації відмінності між $p_{j|i}$ та $q_{j|i}$ за допомогою методу Кульбака-Лейблера:

$$C = KL(P||Q) = \sum_i \sum_j p_{j|i} \ln \frac{p_{j|i}}{q_{j|i}}$$

3.2 Метод покращення текстових описів задач спринту проєкту

В системі трекінгу задачі формулюються таким чином, щоб максимально чітко та зрозуміло передати суть задачі, яку потрібно виконати. Важливо забезпечити, щоб кожен учасник команди розумів задачі однаково, уникнути двозначностей та непорозумінь. Як свідчить досвід, виконавці проєкту зберігають інформацію про задачу у її назві, формують описи та додають коментарі. Фіксація всієї цієї інформації в системі трекінгу надає команді зручний інструмент для спільної роботи, спрощує комунікацію та допомагає забезпечити точність та повноту розуміння задач усіма учасниками проєкту.

Процес ідентифікації та класифікації задач спринту на “зрозумілі” та “не зрозумілі” є критичним на етапі формування беклогу продукту. Це дозволить визначити задачі, які можуть бути джерелом непорозумінь та ризику затримки виконання проєкту через неякісний текстовий опис. Класифікація задач забезпечить команду проєкту можливістю приділяти особливу увагу та ресурси для уточнення та вдосконалення опису цих задач, зменшуючи, тим самим, ймовірність помилок та непорозумінь.

Врахування якості та однозначності текстового опису задач спринту як фактору впливу на виконання задач є ключовим елементом ефективного управління проєктом. Це допомагає забезпечити правильне розуміння задач та їх ефективне виконання членами команди.

Якість текстових описів задач у беклозі проєкту відіграє ключову роль у процесі планування та виконання спринта в ІТ-команді. Нечіткі, неоднозначні або неповні формулювання задач призводять до зростання ризиків, появи дефектів, нерозуміння з боку виконавців і, як наслідок, до зниження ефективності розробки. Тому проблема підвищення якості текстових описів є

надзвичайно актуальною як у контексті командної взаємодії, так і в рамках побудови інтелектуальних систем підтримки прийняття рішень.

Розглянемо метод автоматизованого покращення текстових описів задач, що поєднує сучасні інструменти обробки природної мови (NLP), можливості великих мовних моделей (LLM) та елементи машинного навчання. Метод спрямований на виявлення якісних шаблонів описів, класифікацію існуючих задач за рівнем придатності та формування покращених текстів, що сприяють кращому розумінню й реалізації задач у межах спринту.

Застосування цього підходу дозволяє не лише стандартизувати опис задач, а й підвищити точність інших компонентів моделі планування - зокрема оцінки дефектності, пріоритетності та розподілу задач між виконавцями.

Запропонований метод спрямований на підвищення якості текстових описів задач у беклозі проєкту, що безпосередньо впливає на зрозумілість, реалізованість та точність виконання задач в межах спринту. Метод поєднує сучасні підходи обробки природної мови (NLP) з машинним навчанням і включає три основні етапи:

1 Збір і підготовка даних. На цьому етапі здійснюється систематичний збір та попередня обробка текстових описів задач, які вже використовувалися у попередніх ітераціях спринтів. Включаються як успішно реалізовані, так і ті, що спричинили труднощі або були відхилені. Підготовка даних включає нормалізацію, токенізацію та очищення текстів.

2 Кластеризація описів за якістю. Текстові описи задач автоматично класифікуються (кластеризуються) на три категорії - «хороші», «погані» та «невідомі» - за допомогою попередньо натренованої мовної моделі BERT (bert-base-uncased) із застосуванням бібліотек Python для трансформерів. У якості кластеризатора використовується агломеративний кластерний аналіз. Для підвищення точності кластеризації доцільно залучати експертні оцінки команди під час ретроспектив спринту, що дозволяє адаптувати модель до специфіки команди та проєкту.

3 Генерація покращених описів. На основі отриманих кластерів і аналізу «хороших» описів формується узагальнений варіант найякіснішого опису задач, який може бути використаний як шаблон або рекомендація при формуванні беклогу в наступних спринтах. Це сприяє стандартизації описів, покращенню їх чіткості та зниженню рівня невизначеності при плануванні.

Запропонований метод покращення текстових описів задач спринту демонструє ефективний підхід до вирішення однієї з критичних проблем у процесі планування – недостатньої чіткості та якості формулювання задач. Методологія базується на двох ключових етапах:

- 1) автоматизованій оцінці текстових описів на основі кластеризації з використанням великої мовної моделі;
- 2) генерації покращеного формулювання задачі, орієнтованого на підвищення зрозумілості й відповідності командним стандартам.

На першому етапі метод дозволяє класифікувати існуючі описи задач за рівнем їх якості на «хороші» та «погані» - із залученням попередньо натренованої трансформерної моделі. Така кластеризація забезпечує об'єктивну оцінку чіткості текстів, яка може бути додатково уточнена через зворотний зв'язок від команди.

На другому етапі, базуючись на шаблонах «хороших» описів, система формує рекомендації або варіанти покращених текстів, які можуть бути безпосередньо використані для оновлення задач у беклозі. Таким чином, велика мовна модель виконує роль інструменту як оцінювання, так і трансформації текстового контенту, забезпечуючи сталість і підвищення якості документації.

Метод має значну практичну цінність: він не лише покращує комунікацію всередині команди, але й позитивно впливає на інші компоненти моделі інтелектуального планування – зокрема прогнозування ризиків і дефектів, адекватність оцінки трудомісткості та ефективність розподілу задач. У такий спосіб забезпечується більш точне й надійне планування, що підвищує загальну якість і продуктивність спринтів.

Для оцінювання узгодженості між оцінювачами пропонується застосувати коефіцієнт каппа Коена – це статистика, яка використовується для вимірювання надійності між оцінювачами (а також надійності всередині них) для якісних (категорійних) елементів [75]. Вважається, що це більш надійний показник, ніж простий розрахунок відсотка узгодження, оскільки каппа враховує також можливість випадкової згоди.

Каппа Коена вимірює згоду між двома оцінювачами, кожен з яких класифікує N елементів у C взаємовиключних категорій. Визначення κ таке:

$$\kappa = \frac{p_0 - p_e}{1 - p_e} = 1 - \frac{1 - p_0}{1 - p_e}$$

- де
- p_0 — відносна спостережувана згода серед оцінювачів;
 - p_e — гіпотетична ймовірність випадкової згоди, використовуючи спостережувані дані для обчислення ймовірності того, що кожен спостерігач випадково побачить кожну категорію.

Якщо оцінювачі повністю згодні $\kappa = 1$. Якщо серед оцінювачів немає згоди, крім того, що можна було б очікувати випадково (як зазначено p_e), $\kappa = 0$. Цілком можливо, що статистичні дані будуть негативними, що може виникнути випадково, якщо немає зв'язку між оцінками двох оцінювачів, або це може відображати реальну тенденцію оцінювачів давати різні оцінки.

Для k категорій, N спостережень для класифікації, та n_{ki} - кількість разів, коли оцінювач i передбачив категорію k :

$$p_e = \frac{1}{N^2} \sum n_{k1} n_{k2},$$

В машинному навчанні та статистиці для оцінки бінарних класифікацій, формулу Каппа Коена можна записати так:

$$\kappa = \frac{2 \cdot (TP \cdot TN - FN \cdot FP)}{(TP + FP) \cdot (FP + TN) + (TP + FN) \cdot (FN + TN)},$$

- де
- TP – справжні позитивні результати;
 - FP – помилкові позитивні результати;
 - TN – справжні негативні результати;
 - FN – помилкові негативні результати.

В роботі пропонується система рекомендації, яка заснована на гібридному підході. Цей підхід об'єднує метод рекомендації на основі знань (Knowledge-Based Recommendations) [58] та метод контентної фільтрації (Content-Based Filtering) [58].

Knowledge-Based Recommendations застосовується у контексті розподілу задач серед виконавців проєкту та використовує експертні знання про опис задач та кваліфікацію виконавців для надання релевантних рекомендацій. Цей підхід заснований на явній інформації про кваліфікацію виконавців (освіта, досвід, сертифікації, технічні навички тощо), досвід виконання подібних задач, оцінка продуктивності в попередніх проєктах.

Метод контентної фільтрації (Content-Based Filtering) у контексті розподілу задач серед виконавців проєкту ґрунтується на використанні характеристик самих задач та вподобань виконавців. Цей метод допомагає призначати задачі на основі схожості між описом задач і профілем виконавця, який відображає його компетенції, досвід й уподобання. Профіль виконавця базується на його попередньому досвіді, успішно виконаних задачах і зазначених уподобаннях. Включає компетенції, ключові навички, пріоритетні типи задач та історію виконання. Задачі пропонуються виконавцю, якщо їхні атрибути найбільш схожі на атрибути задач, які цей виконавець успішно виконував у минулому. Content-Based Filtering забезпечує точний розподіл задач, що відповідають компетенціям виконавців, використовуючи детальний опис задач і профілів. Цей підхід є корисним для персоналізації та автоматизації в проєктному менеджменті, особливо в ситуаціях, коли історія виконання є доступною та детальною [58]. Також застосовується підхід (User-Based) згідно якого задача призначається виконавцю, який схожий на того, хто раніше успішно виконував подібні задачі. На основі обчислених показників подібності

створюється список рекомендованих виконавців. Отже, система рекомендацій пропонує перелік членів команди, які із високою ймовірністю успішно виконають задачу, на основі комбінованої інформації про ретроспективу виконання задач, інтереси та кваліфікацію виконавців, пріоритети проєкту та ясність та чіткість опису задачі.

Розроблена рекомендаційна система надає команді проєкту необхідні інструменти для ефективного управління в умовах невизначеності. Вона використовує дані, які зберігаються у трекінговій системі про попередній досвід виконання задач командою, можливості великих мовних моделей щодо удосконалення текстових описів разом із алгоритмами стабільного зіставлення задля зниження ризиків неналежного або невчасного виконання задач. Це дозволяє формувати рекомендації щодо вибору задач спринту проєкту та розподілу між виконавцями з метою підвищення ефективності виконання проєкту.

Включення експертних оцінок від команди проєкту під час ретроспективи спринту виявилось ефективним способом не тільки для покращення якості описів задач, але й для навчання моделі. Експертиза членів команди дозволяє збільшити точність кластеризації, виходячи з глибокого розуміння контексту проєкту.

Менеджеру проєкту при формуванні беклогу продукту проєкту або спринту проєкту надається два варіанта для обрання найбільш якісного опису задач. Для цього йому потрібно переглянути та переформулювати описи задач спринту, для того щоб знизити ризики проєкту та збільшити якість виконання проєкту.

3.3 Інформаційна технологія інтелектуального планування роботи команди ІТ проєкту

Запропонована інформаційна технологія охоплює набір функцій, призначених для обробки текстових описів задач спринту проєкту при плануванні роботи команди ІТ проєкту. Ця технологія базується на моделях та

методах, які охоплюють такі завдання, як збір та очищення даних, токенизація, векторизація, кластеризація описів задач спринту проєкту, розподіл задач серед виконавців.

Інформаційна технологія інтелектуального планування роботи команди ІТ проєкту розроблена на основі моделей та методів запропонованих в роботі, а саме:

1 Метод інтелектуального планування роботи команди ІТ проєкту, який сприяє більш швидкому і точному плануванню спринтів за рахунок аналізу ретроспективних даних та використання моделі штучного інтелекту для надання рекомендацій менеджеру та команді проєкту [95].

2 Модель оцінки текстового опису задач спринту проєкту [93].

3 Метод покращення текстових описів задач, який дозволяє підвищити точність сприйняття задач та знизити ризики невиконання задач, за рахунок використання моделей машинного навчання та LLM для оцінки та вдосконалення тексту [96].

4 Моделі розподілу задач спринту між виконавцями на основі постановки задач визначення стабільних зіставлень що, враховують специфіку гнучкої методології управління проєктами та дозволяють покращити розподіл задач серед членів команди проєкту [46].

5 Метод формування рекомендацій при плануванні роботи команди ІТ проєкту визначає рекомендації щодо опису задач та розподілу задач на основі інтелектуальної обробки ретроспективних даних, текстових даних та контексту проєкту та дозволяє врахувати ризики, що притаманні проєктам з гнучкою методологією [46].

Запропоновані моделі та методи інтегруються в популярні системи трекінгу задач (Jira, Trello), та дозволяють автоматично виявляти проблемні задачі та досягти підвищення ефективності роботи команди.

Концептуальна модель планування робіт команди ІТ-проєкту представлена на рисунку. 3 1.



Рисунок 3.1 – Концептуальна модель планування робіт команди ІТ-проєкту

Вхідними даними в концептуальній моделі є: проєктна документація та експертні оцінки. До проєктної документації відносяться ретроспективні дані описів задач спринту проєкту, які завантажуються із системи управління задачами та ієрархія задач проєкту [93].

Експертні оцінки використовуються для оцінки якості формулювань задач спринту та розподілу задач серед виконавців проєкту, при цьому враховується інформація про компетентності виконавців, кваліфікацію, навички та вподобання.

Вихідними даними інформаційної технології є:

- текстовий опис задач, який задовольняє команду проєкту;
- рекомендації по ранжуванню задач спринту для визначення пріоритезації щодо подальшого розподілу задач серед виконавців;
- рекомендації та варіанти розподілу задач спринту для команди проєкту.

Головні функції інформаційної технології інтелектуального планування роботи команди ІТ проєкту пов'язані:

- зі збором, очищенням та зберіганням даних спринту проєкту;
- поліпшенням опису задач;
- формуванням варіантів опису задач для команди проєкту;
- формуванням рекомендацій щодо розподілу задач серед виконавців.

Ці функції реалізуються за допомогою програмних компонентів, які використовують гнучкий механізм для вирішення проблеми з обробкою текстових описів задач спринту проєкту та розподілом виконавців.

Декомпозиція концептуальної моделі планування робіт команди ІТ-проєкту представлено з використанням методології функціонального моделювання IDEF0 [56, 54] та наведено на рисунку 3.2.

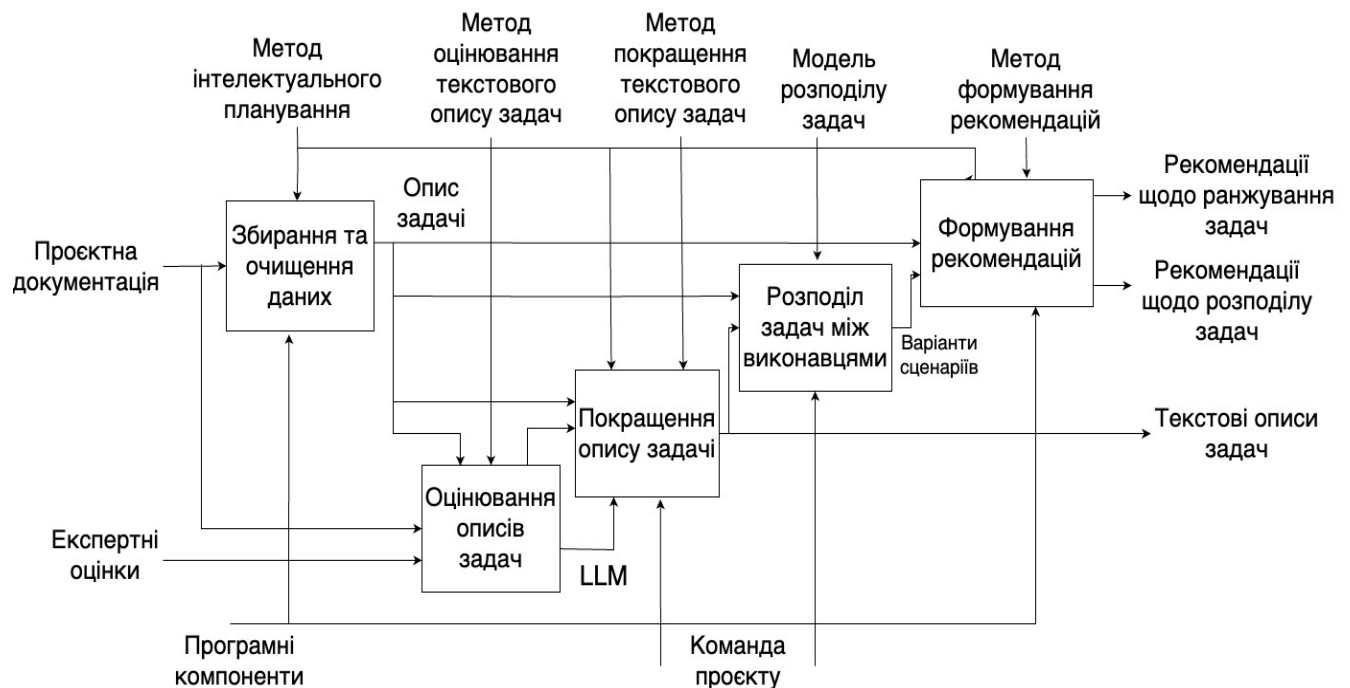


Рисунок 3.2 – Декомпозиція функцій інформаційній технології інтелектуального планування роботи команди ІТ проєкту на основі гнучкої методології (IDEF0)

Розглянемо ключові функції інформаційної технології. Перш за все, приділимо увагу функції, яка стосується збору, очищення та опису задач у системі управління задачами. Цей процес є важливим для організації та контролю процесів розробки проєкту. Фіксація всієї інформації в трекінговій системі створює зручне середовище для командної роботи, сприяє покращенню комунікації та забезпечує повне розуміння задач усіма учасниками проєкту. Для підвищення сприйняття командою проєкту текстового опису задач їх обробка здійснюється із застосуванням алгоритмів машинного навчання та експертних оцінок. Задачі оцінюються експертами, що дозволяє незалежно провести розмітку набору даних для бінарної класифікації. До позитивних задач відносяться задачі з високою якістю, які зрозумілі експерту та оцінка часу для їх виконання, найбільш вірогідно, буде адекватною. Незрозумілі для експерта формулювання задач отримують негативні позначки. Оцінка якості формулювань визначається рівнем обізнаності експерта щодо специфіки та контексту проєкту, при цьому збільшення кількості експертів майже не впливає на результат [93]. Додавання оцінок формулювань під час ретроспективи спринту для тренування моделі класифікатора та залучення членів команди проєкту у якості експертів є доречним.

Експерименти [93] показали, що без участі експертів, які б були обізнані на проєкті, побудувати дієвий класифікатор якості описів задач спринту досить складно. Саме експертні знання потрібні, щоб обрати придатний спосіб векторизації текстів і налаштувати модель під особливості даних із трекінгової системи [17].

На основі отриманих результатів [96] можна зробити висновок що деякі задачі, дійсно, можна кардинально покращити для розуміння за допомогою великих мовних моделей. Їх можна застосовувати в роботі менеджера проєкту для формування задач спринту та покращення ефективності команди через лінгвістичні особливості задач. Менеджер проєкту обирає варіанти формулювання опису задачі, які оцінює LLM, та призначає бали на основі його ясності та зрозумілості [46]. При формуванні беклогу продукту проєкту або

спринту проєкту менеджеру надається два варіанта опису задач для обрання найбільш сприятливого.

Розподіл задач в спринті проєкту між виконавцями може вплинути на виконання задач спринту проєкту. Це може бути пов'язано з нестачею кваліфікованих виконавців або перемиканням виконавців на інші специфічні задачі. Для уникнення вказаних ситуацій застосовуються методи пошуку стійких паросполучень. За результатами досліджень алгоритми стійких паросполучень EADAM [34] та SOSM [60] є найбільш ефективними. Вони забезпечують розподіл задач у спринті, задовольняють вимоги розробників та характеризуються низькою обчислювальною складністю. На базі цих алгоритмів менеджер проєкту отримує стійке зіставлення на основі пріоритезації задач та кваліфікації виконавців, що дозволяє успішно виконати задачу спринту проєкту.

Як було зазначено раніше, для уникнення ризиків, пов'язаних з невдалим розподілом задач, доцільно застосовувати алгоритми стабільного зіставлення EADAM [34] та SOSM [60], які найкраще себе зарекомендували.

Функцій, які пов'язані із формуванням рекомендації команді проєкту щодо пріоритизації задач спринту проєкту та ефективному їх розподілу, виконується з метою підвищення ефективності виконання проєкту.

Розроблена інформаційна технологія враховує контекст проєкту та експертні оцінки, що робить її гнучкою і адаптивною до специфіки конкретного проєкту та особливостей команди, що є важливим для подальшого вдосконалення процесів управління.

Для аналізу впливу ризиків, пов'язаних із невиконанням задач у спринті проєкту, застосовується показник рівня відмов на одиницю часу і використовується для оцінювання надійності та ефективності робочого процесу [97].

Для реалізації інформаційної технології були розроблені програмні компоненти, які враховують вимоги до сучасних інструментів планування роботи команди ІТ проєкту.

У роботі розроблено інформаційну технологію для інтелектуального планування роботи команди ІТ-проєкту, засновану на принципах гнучкої методології. Ця технологія надає команді проєкту механізм для ефективного управління в умовах ризиків. Вона використовує дані із трекінгові системи, застосовуючи методи обробки природної мови на базі LLM, які забезпечують рекомендації для формулювання задач спринту, а також алгоритми стабільного зіставлення, що пропонують рекомендації ефективного розподілу між членами команди проєкту. Використання LLM автоматизує та покращує процеси роботи із задачами, що, у свою чергу, підвищує продуктивність і згуртованість команди. Розроблена інформаційна технологія інтелектуального планування роботи команди ІТ проєкту, яка дає змогу оцінювати та знижувати негативний вплив чинників, що загрожують строкам і якості проєкту, завдяки аналізу історичних даних та моделюванню поведінки команди. Запропоновані моделі легко інтегруються в популярні системи трекінгу задач (Jira, Trello), дозволяючи автоматично виявляти проблемні задачі та досягти підвищення ефективності.

3.4 Висновки до розділу 3

Систематизовано та формалізовано роботу з текстовими описами задач. Запропоновано багатостадійний метод класифікації й покращення опису задач, який поєднує автоматичну обробку природної мови (BERT + кластеризація), експертні оцінки та генерацію покращених формулювань. Такий підхід забезпечує стандартизовані, однозначні описи, зменшує кількість лінгвістичних дефектів і сприяє кращому розумінню команди та виконанню проєкту. Крім того, він підкреслює потенціал LLM для допомоги в постійному вдосконаленні описів задач.

Розроблено модель вибору та розподілу задач спринту. На базі оцінок бізнес-цінності, трудомісткості, текстової якості та прогнозованого рівня дефектів побудовано дві взаємодоповнювальні моделі:

- модель відбору оптимального набору задач до беклогу спринту;

- модель стабільного призначення задач виконавцям, що використовує алгоритми SOSM та EADAM для балансування інтересів команди й бізнес-пріоритетів за мінімальних обчислювальних витрат.

Створено Інформаційну технологію інтелектуального планування роботи команди ІТ проєкту, придатну до впровадження в популярні трекінгові системи (Jira, Trello). Вона охоплює збір та очищення даних, автоматичне підвищення якості описів задач, модель розподілу задач і, за необхідністю, візуалізацію метрик. В результаті отримано підвищення прозорості планування, скорочення часу підготовки спринтів, зниження дефектності й зривів дедлайнів і, як слідство, зростання загальної продуктивності команди.

РОЗДІЛ 4 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ МОДЕЛЕЙ ТА МЕТОДІВ ІНТЕЛЕКТУАЛЬНОГО ПЛАНУВАННЯ РОБОТИ КОМАНДИ ІТ-ПРОЄКТУ

4.1 Організація експериментального дослідження моделей та методів інтелектуального планування

При управлінні ІТ проєктами найбільш поширеним є застосування гнучких або адаптивних методологій управління. Гнучкий підхід ґрунтується на ітеративній моделі, яка спрямована на постійну комунікацію між членами команди, замовниками та іншими зацікавленими сторонами, що дозволяє командам ефективніше справлятися з викликами. Однак існують фактори, які суттєво впливають на успішність проєкту. Зокрема, це нечітке формулювання задач та проблеми з їх розподілом між виконавцями, а також недостатній рівень комунікації в команді. Ці фактори істотно впливають на якість планування та реалізацію проєкту і тому актуальним є створення такого механізму планування роботи команди ІТ проєкту, який би дозволив знівелювати ці негативні впливи. За рахунок впровадження такого механізму команда проєкту зможе знизити вплив зазначених факторів на ефективність виконання проєкту та підвищити продуктивність роботи.

Запропонована інформаційна технологія забезпечує можливість покращити формулювання задач, що виконуються під час спринтів, і оптимізувати їх розподіл між членами команди проєкту. Вона забезпечує автоматизацію процесу аналізу інформації, виявлення можливих проблем й формування рекомендацій команді проєкту щодо опису та розподілу задач. Впровадження цієї інформаційної технології дозволяє знизити ризики та підвищити ефективність управління проєктом. Головні функції інформаційної технології інтелектуального планування роботи команди ІТ проєкту пов'язані з:

- збором, очищенням та зберіганням даних спринту проєкту;
- вдосконаленням текстового опису задач;

- формуванням варіантів опису задач для команди проєкту;
- наданням рекомендацій щодо розподілу задач серед виконавців.

Для досягнення поставленої мети та розробки моделі ідентифікації задач з неякісним формулюванням у спринті ІТ-проєкту пропонується комплексний підхід, що включає в себе збір та обробку текстових описів задач, використання алгоритмів машинного навчання для класифікації та залучення експертних оцінок для вдосконалення якості сприйняття задач командою проєкту. Таким чином, постає питання експериментальної перевірки запропонованих моделей та методів й дослідження параметрів, які впливають на результат.

Для організації експериментів використовується набір даних, який сформовано на базі реального проєкту. Цей набір містить в собі 1000 задач, які експортовано з системи управління проєктом Jira, та містить інформацію щодо різноманітних параметрів задач, таких як назва, опис, унікальний ключ, виконавець, витрачений час, оцінка часу виконання, статус виконання задачі, коментарі та інші. Мова отриманих текстових описів задач для аналізу формулювань – англійська, що є типовим для проєктів з розробки програмного забезпечення. Основними характеристиками для ідентифікації формулювання задачі визначено назва та опис. Слід зауважити, що опис не є обов’язковою характеристикою задачі, тобто її формулювання може складатися лише з назви.

Враховуючи обмеження приватного набору даних, в роботі, також, розглядаються проєкти з відкритих джерел. Дані зібрані з 16 публічних сховищ системи трекінгу Jira, які містять 1822 проєкти та 2,7 мільйона випусків. Дані містять історичні записи про 32 мільйони змін, 9 мільйонів коментарів та 1 мільйон посилань на проблеми, які складно групуються. Для проведення дослідження обрано проєкти, що включають понад 5 тисяч задач. Це дозволить забезпечити репрезентативність даних для аналізу ключових елементів, що впливають на “зрозумілість” опису задач. Кожний проєкт з цього набору даних досліджувався окремо для збереження його власного технологічного контексту.

На першому етапі експериментального дослідження виконується ряд процедур очищення та попередньої підготовки даних. Для стандартного

пайплайну обробки текстових даних спочатку з набору даних вилучаються посилання, фрагменти коду та неінформативні символи, такі як знаки валют та вподобайки. Також проводиться заміна аббревіатур на їх повне значення.

На другому етапі вирішується задача класифікації текстових описів задач із залученням експертних оцінок щодо якості формулювання задач. Їх розмітка виконується за допомогою експертів в результаті аналізу оцінок яких формуються два класи. Для ефективного проведення експериментального дослідження необхідно визначити наступне: яка кількість експертів є достатньою для забезпечення незалежної та достовірної оцінки якості формулювання задач і чи може знання експертами контексту проєкту впливати на результати оцінювання?

Третій етап дослідження пов'язаний з побудовою моделі кластеризації текстових описів. Необхідно з'ясувати, чи достатньо двох класів, чи групування на більшу кількість є більш природнім. Також важливим кроком дослідження є можливість використання сучасних великих мовних моделей для оцінки чіткості текстових формулювань та вплив промтінгу на результат. Технологія дослідження лінгвістичних особливостей текстів задач складається з декілька етапів, представлених на рисунку 4.1. Розглянемо їх більш детально.

1 Використання публічного набору даних Jira включає проведення селекції релевантних частин текстів задач.

2 Проведення трьох серій експериментів і застосуванням методів обробки природної мови та LLM для кластеризації та забезпечення точності результатів:

- перший експеримент пов'язано з тренуванням нейронної мережі на основі векторизованої репрезентації тексту для якого обрано претреновану модель BERT (bert-base-uncased) ;

- другий експеримент спрямований на дослідження впливу запитів на отримання векторних репрезентацій текстових задач з використанням великих мовних моделей;

- в третьому експерименті досліджено використання великих мовних моделей для оцінки якості опису задач на різних наборах даних.

3 Проведення порівняльного аналізу якості модельних оцінок з оцінками експертів.

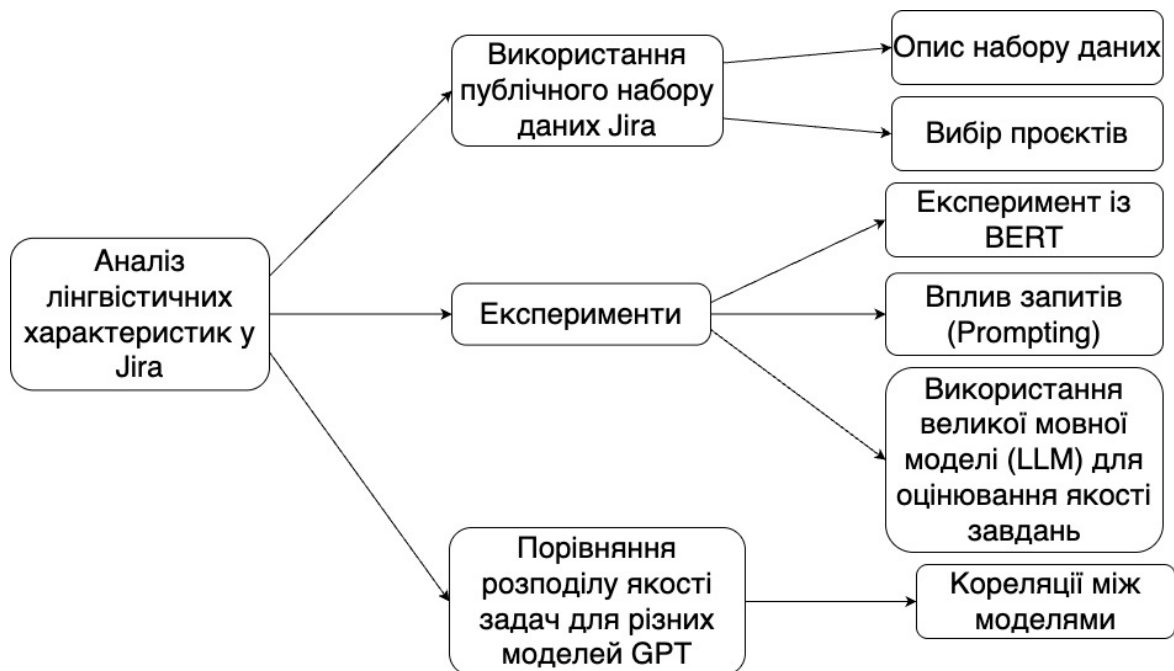


Рисунок 4.1 – Технологія дослідження лінгвістичних характеристик

Таким чином, в результаті обробки текстових описів задач формуються рекомендації щодо покращення чіткості сприйняття задач. Це впливає на оцінку ризику невиконання задачі та, відповідно, на важливість задачі для виконання у наступному спринті. В даному дослідженні передбачається, що відбір задач відбувається за результатами ранжування та з урахуванням обмежень на трудомісткість задач. Однак алгоритми, які реалізують запропоновану в розділі 3 модель розподілу обраних задач серед виконавців, потребують експериментального дослідження. Саме цьому присвячено наступний етап експериментів.

4.2 Дослідження чіткості сприйняття текстового опису задач

Вихідні дані для дослідження наступні. Загальна кількість задач у наборі – 1000, з яких з описом - 461. Аналіз отриманого набору визначив наступні характеристики: середня кількість слів у назві – 6.4, у описі – 37.67.

Дослідження проведено на реальних даних, сформованих із системи управління проектом з метою автоматизації визначення неякісних формулювань задач спринту, які впливають на їх розуміння розробниками та, як наслідок, призводять до ризиків проекту. Особливостями даних, які було зібрано для дослідження, є напівструктурованість, неповнота, текстова форма представлення та відсутність контексту проекту. Отже, набір даних потребує спеціальної обробки та очищення.

Таблиця 4.1 – Статистики набору даних на різних етапах очистки

Етап	Видалення гіперпосилань	Видалення спеціальних символів	Видалення стоп слів	Розширення аббревіатур
Кількість унікальних слів (назва)	1470	1270	1260	1251
Кількість унікальних слів (опис)	3293	2610	2601	2596
Кількість унікальних слів (опис + назва)	3943	3069	3061	3051
Максимальна кількість слів (назва)	18	18	15	16
Середня кількість слів (назва)	6.47	6.42	5.62	5.68



Рисунок 4.2 – Статистика набору даних на різних етапах очистки по загальній кількості слів

Кількість унікальних слів у назві та описі - 5080. Сполучники та артиклі виявилися найпопулярнішими словами, а серед тих, що несуть самостійне семантичне навантаження, найбільш популярними є: "add", "image", "inspection", "defect", "object". Ці слова є релевантними для даного проєкту.



Рисунок 4.3 – Статистики набору даних на різних етапах очистки по кількості унікальних слів

Як було зазначено вище, для задач класифікації пропонується використання текстових описів задач, які містяться в системі трекінгу. Таким чином, кожна задача представлена назвою (title) та її більш детальним описом (description). В проведених експериментах використовувалися або тільки назва, або назва конкатенована з описом.

В процесі обробки тексту задач усі назви (title) проходять декілька етапів:

- заміна усіх відомих абревіатур на їхнє повне значення;
- фільтрація усіх нерелевантних символів (за наявності);
- приведення до нормальної форми усіх елементів тексту задач.

Кожна задача перетворюється на набір токенів за допомогою токенизатора, який використовується моделлю і зрозумілий для неї. Для всіх експериментів використовується мова програмування Python та фреймворк для глибокого навчання, зокрема Transformers. Далі отримано векторну репрезентацію назв задач. На початковому етапі застосовано метод TF-IDF, корпусом якого були усі задачі, наявні у наборі даних реального проєкту.

Набір даних з 1000 задач, спочатку, проанатовано, відповідно, на два класи (позитивні та негативні) відносно думки експерта щодо якості формулювань. Далі, за допомогою засобів зменшення розмірності (PCA та t-SNE), отримано візуальні репрезентації даних.

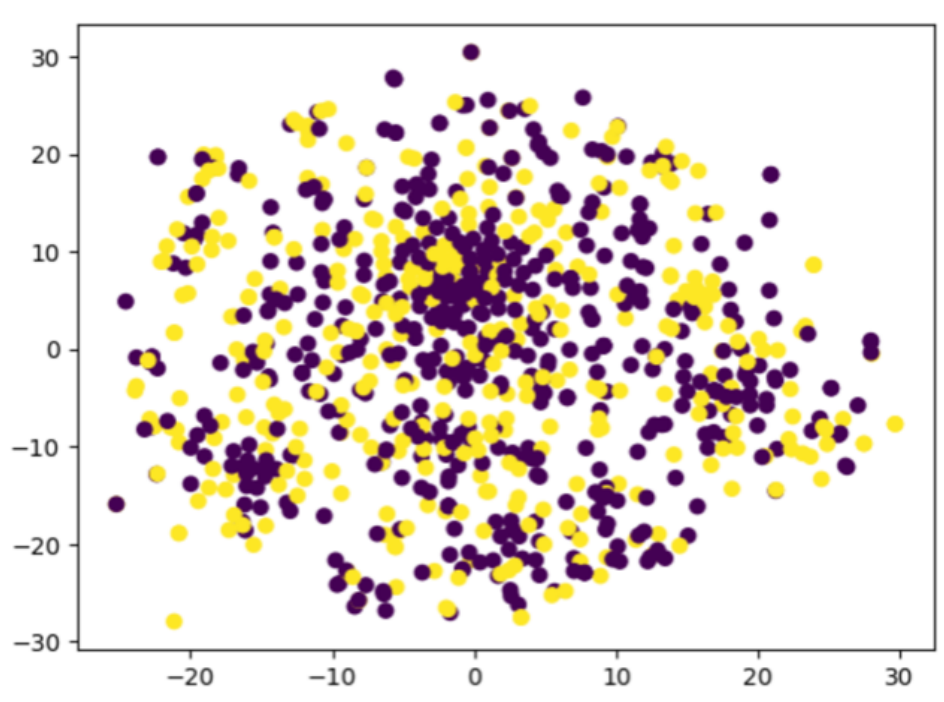


Рисунок 4.4 – Візуальна репрезентація даних за допомогою t-SNE у двомірному просторі (жовті - позитивно проанатовані задачі, фіолетові - негативно проанотовані задачі)

При проведенні семантичного аналізу тексту опису та назви задач було виявлено, що відомі методи не можна використовувати через нестачу контексту в описі. Це є слідством того, що під час формування назви та опису задачі, команда має свій внутрішній контекст, який не передається у текст задачі. Це, в свою чергу, є однією із головних причин виникнення проблем розуміння, що впливає на строки та якість виконання проєкту.

Отже, встановлено, що контекст проєкту знаходиться за межами текстових описів задач, що впливає на їх розуміння новими членами команди, а також обмежує використання методів обробки природньої мови щодо аналізу якості формулювань та визначення факторів впливу на розуміння змісту задач.

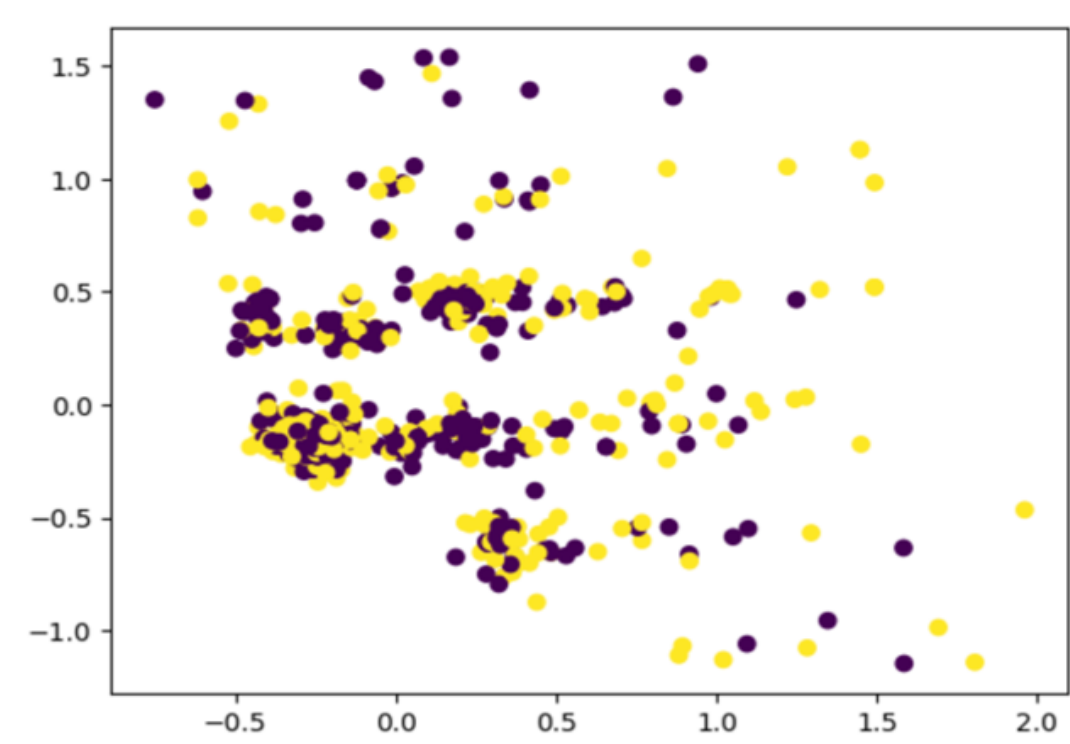


Рисунок 4.5 – Візуальна репрезентація даних за допомогою PCA у двомірному просторі (жовті - позитивно проанатовані задачі, фіолетові - негативно проанотовані задачі)

На наступному етапі експериментів до анотування набору даних залучено додаткових експертів. Командою з чотирьох експертів, незалежно, розмічено набір даних для бінарної класифікації. До позитивних задач відносяться задачі з високою якістю опису, які зрозумілі розмітнику та оцінка часу для них, найбільш вірогідно, буде адекватною. Незрозумілі для експерта формулювання отримують негативні позначки на задачах. На рисунку 4.6 можна побачити розподіл позитивних та негативних позначок для кожного з розмітників (експертів).

Слід зауважити, що під час розмітки даних зі збільшенням часу роботи експерта та кількості оброблених задач, розмітники стають більш впевненими щодо контексту проєкта та більший відсоток задач отримує позитивні оцінки. Зроблено висновок, що, чим більше експерт працює в рамках проєкту, тим точніші стають оцінки, тобто зрозумілість задач, описи яких фіксуються в системі трекінгу, суттєво залежить не тільки від формулювання їх текстового

опису, а також від розуміння змісту самого проєкту. Зроблено припущення, що тільки експерти з середини проєкту можуть надати адекватну розмітку даних для класифікації якості формулювань задач.

Dataset distribution

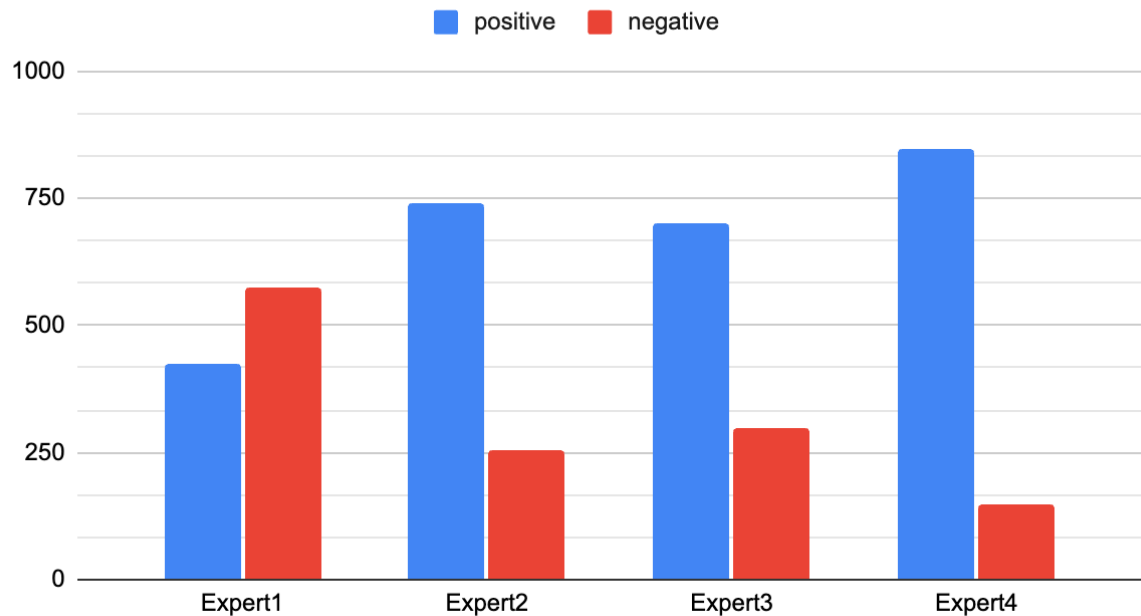


Рисунок 4.6 – Розподіл позначок розмітки експертів

Таким чином, отримано чотири розмічені набори даних та проведено аналіз кореляції оцінок, наданих різними експертами. Як видно з рисунку 4.7, узгодженість думок експертів є низькою – у найкращому випадку розмітники перетинаються лише у 76% задач, а іноді тільки в 43%.

Отримані результати свідчать про низьку узгодженість між експертами, що вказує на суб'єктивність оцінювання та потребу в чіткіших інструкціях або узгодженні критеріїв розмітки.

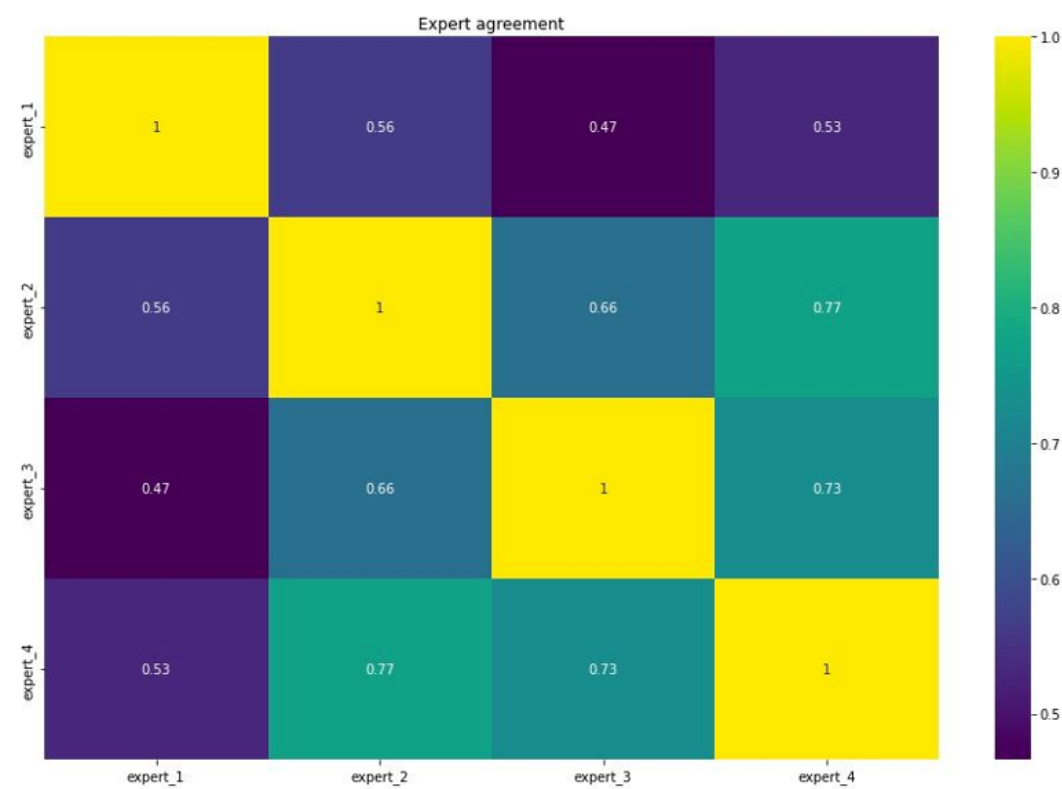


Рисунок 4.7 – Рівень узгодженості думок чотирьох експертів

Для отриманих оцінок експертів розраховано коефіцієнт каппа Коена для кожної пари розмітників. Результати представлено на рисунку 4.8.

На основі матриці коефіцієнтів каппа Коена можна зробити висновок, що узгодженість між експертами є низькою: для більшості пар значення каппа не перевищує 0.29, а в деяких випадках наближається до нуля або навіть негативне (наприклад, між expert_1 та expert_3 – каппа ≈ -0.0015), що свідчить про майже випадкову або несистемну схожість оцінювання. Таким чином, підтверджено припущення про залежність розуміння задач від наявності інформації про проєкт та залучення до команди розробників та інших зацікавлених осіб, які б розумілися на специфіці проєкту і могли більш ефективно оцінювати якість формулювання задач.

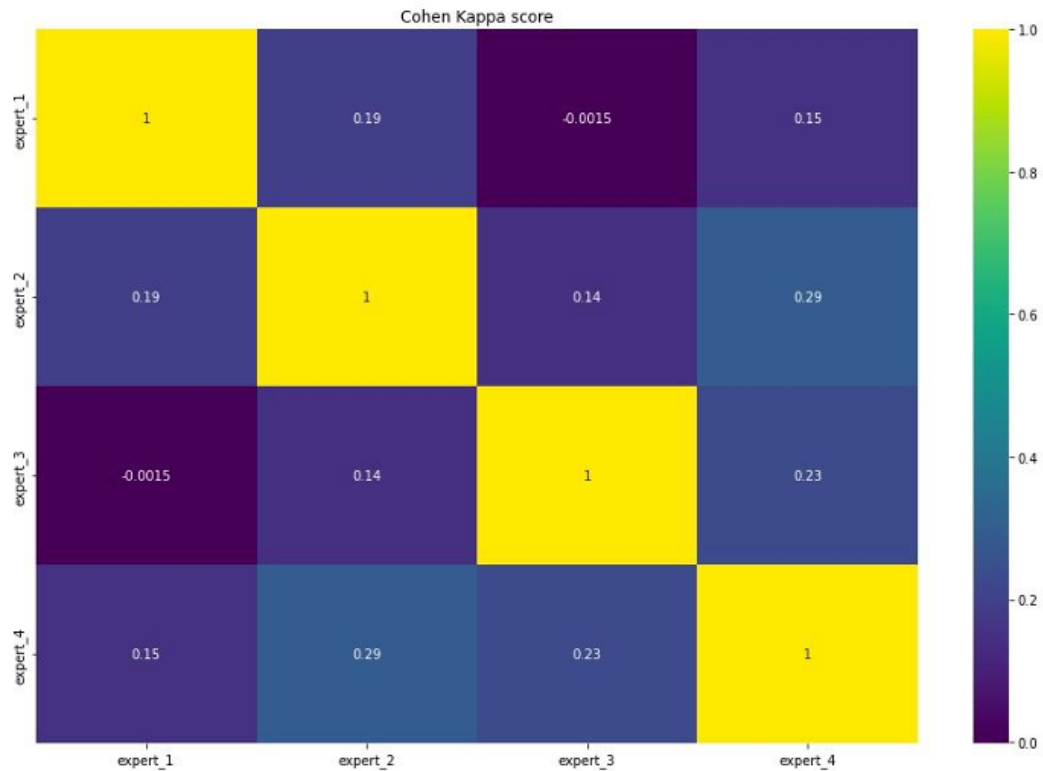


Рисунок 4.8 – Каппа Коена

Для того, щоб отримати достовірні дані для подальшої обробки і враховуючи, що для задовільної узгодженості коефіцієнт каппа Коена має бути понад 0.4 (в нашому випадку максимальне значення склало лише 0,29), для формування набору даних для класифікації було виділено тільки ті задачі, для яких, щонайменше, 3 експерта дали однакову відповідь щодо якості опису задачі. Таким чином, отримано 623 позитивних і 187 негативних оцінок задач.

Сучасна практика машинного навчання показує, що серед моделей, які демонструють високу ефективність у класифікації тексту, особливо добре працюють два підходи: Байєсівські класифікатори (Naive Bayes) завдяки простоті й швидкості та нейронні класифікатори є кращими в задачах, де важливий контекст, складна структура мови або точність. Тому вирішено скомбінувати ці підходи і провести з їх допомогою серію експериментів.

Для класифікатора Байєса векторні репрезентації отримано з потренованої нейронної мережі BERT. Далі, ці репрезентації використовуються як самостійні набори даних, не пов'язані з текстом, на базі яких тренується класифікатор. За

результатами експериментів для класифікатора Байєса отримано точність 70%. Для класифікатора на основі глибокого навчання обрано нейронну мережу для бінарної класифікації на основі архітектури трансформер BERT з вище зазначеного пакету. В якості претренованої моделі було обрано ту ж саму модель. Для оцінок використовується критерій F1 – метрика для вимірювання точності моделей в задачах класифікації, яка є гармонічним середнім між точністю (precision) і повнотою (recall). В результаті отримано класифікатор, який на тестовому наборі даних дає оцінку точності, приблизно, 83%, що є гарним результатом для малого набору даних і з суперечливими мітками. Графік тренування представлено на рисунку 4.9.

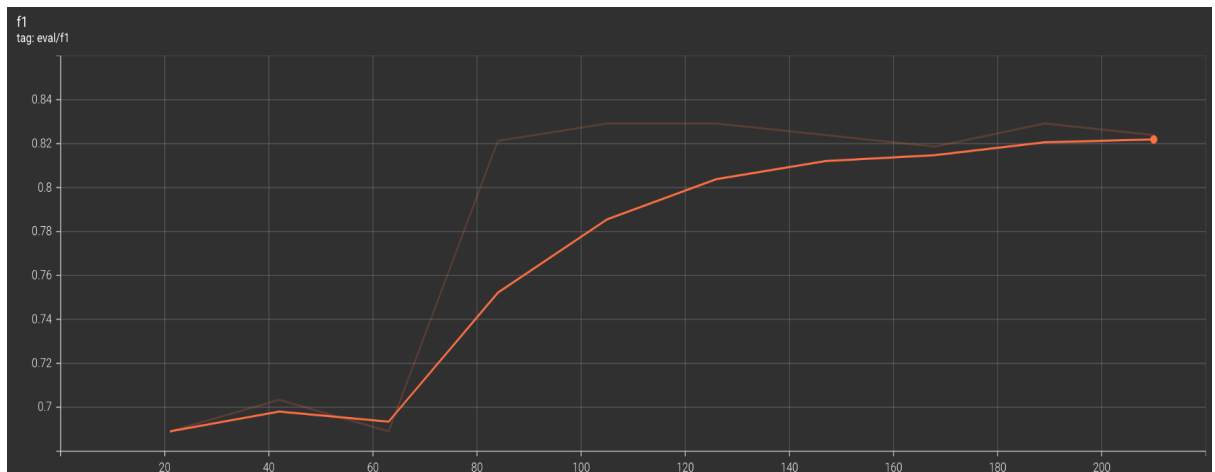


Рисунок 4.9 – Графік залежності метрики F1 від шагу тренування.

На основі проведених експериментів можна зробити висновок, що використання нейронного класифікатора на основі архітектури BERT дозволило досягти значно вищої якості класифікації (F1=83%) порівняно з наївним байєсівським класифікатором (точність 70%), навіть попри невеликий обсяг даних та наявність суперечливих міток. Це свідчить про перевагу контекстуального навчання BERT для задач, якість опису яких розглядається і де важлива глибока семантична інтерпретація. Водночас отримані результати також підкреслюють, що для покращення якості розмітки та ідентифікації текстових описів задач спринту критично важливим є залучення експертів, які

володіють контекстом проєкту, оскільки саме такі знання дозволяють усунути неоднозначності в даних та забезпечити надійніші мітки для навчання моделей.

Для поглибленого лінгвістичного аналізу, зокрема вивченні мовних моделей, що лежать в основі формування задач, їх класифікації та групування проведено ряд експериментів на основі відкритого набору даних The Public Jira Dataset. Він містить задачі з реальних, загальнодоступних проєктів, отриманих через Jira API V2, і містить широкий спектр інформації — від описів задач до історії змін, коментарів та внутрішніх зв'язків між задачами. Це дозволило провести глибокий лінгвістичний аналіз, включаючи вивчення морфологічних особливостей та семантичних характеристик [107].

Селекція реальних проєктів проводилась на базі критеріїв Dataset description та Project Selection. При проведенні аналізу даних 30 проєктів з відкритих джерел, визначено кількісні характеристики проєктів, а саме:

- середня кількість слів в назві задачі;
- середня кількість слів в описі задачі;
- середня кількість слів з пробілами в назві задачі;
- середня кількість символів з пробілами в описі задачі.

Це дало змогу виявити лінгвістичні чинники, що впливають на розуміння задач розробниками, та покращити алгоритми кластеризації та автоматичної категоризації задач, що є особливо актуальним для оптимізації процесів беклог менеджменту в ІТ проєктах.

Результати проведення селекції даних представлено на рисунках 4.10-4.14. Отримані результати свідчать про те, що представлені у розподілу середньої кількості слів. На рисунку 4.10 та 4.11 можна побачити розподіл середньої кількості слів проєктів і символів з пробілами відповідно. Середня кількість слів в назві задачі – 8.27, в описі задачі – 8.27. Також, отримано розподіл середніх проєктних значень кількості символів, який представлено на рисунку 4.12 та рисунку 4.13. Середня кількість символів у описі задач складає 59.56, а в описі – 1051.27.

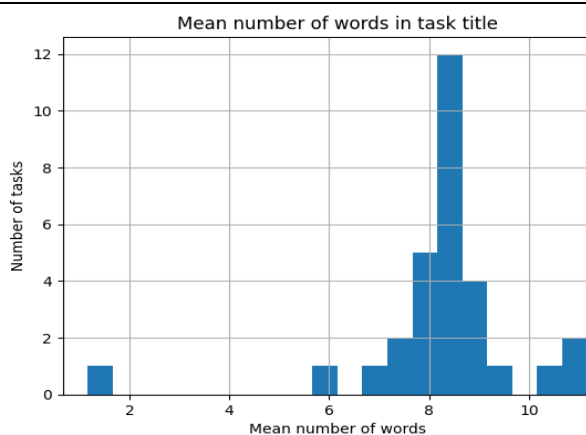


Рисунок 4.10 – Розподіл кількості слів у назві задачі

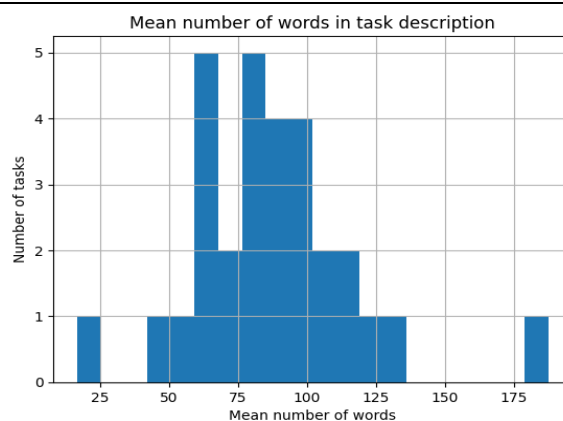


Рисунок 4.11 – Розподіл середньої слів в описі задачі

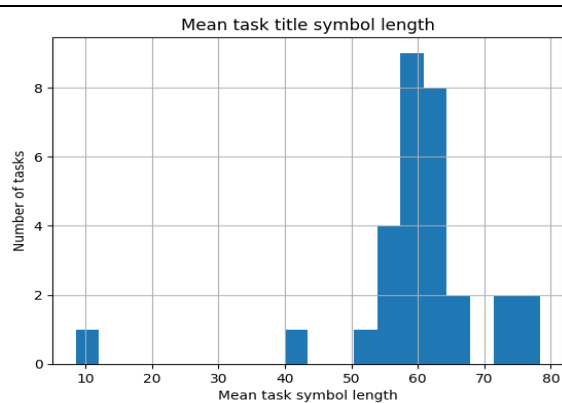


Рисунок 4.12 – Розподіл кількості символів з прогалинами у назві задачі

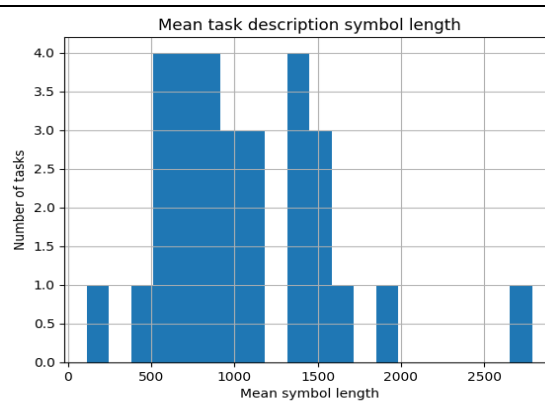


Рисунок 4.13 – Розподіл кількості символів з прогалинами в описі задачі

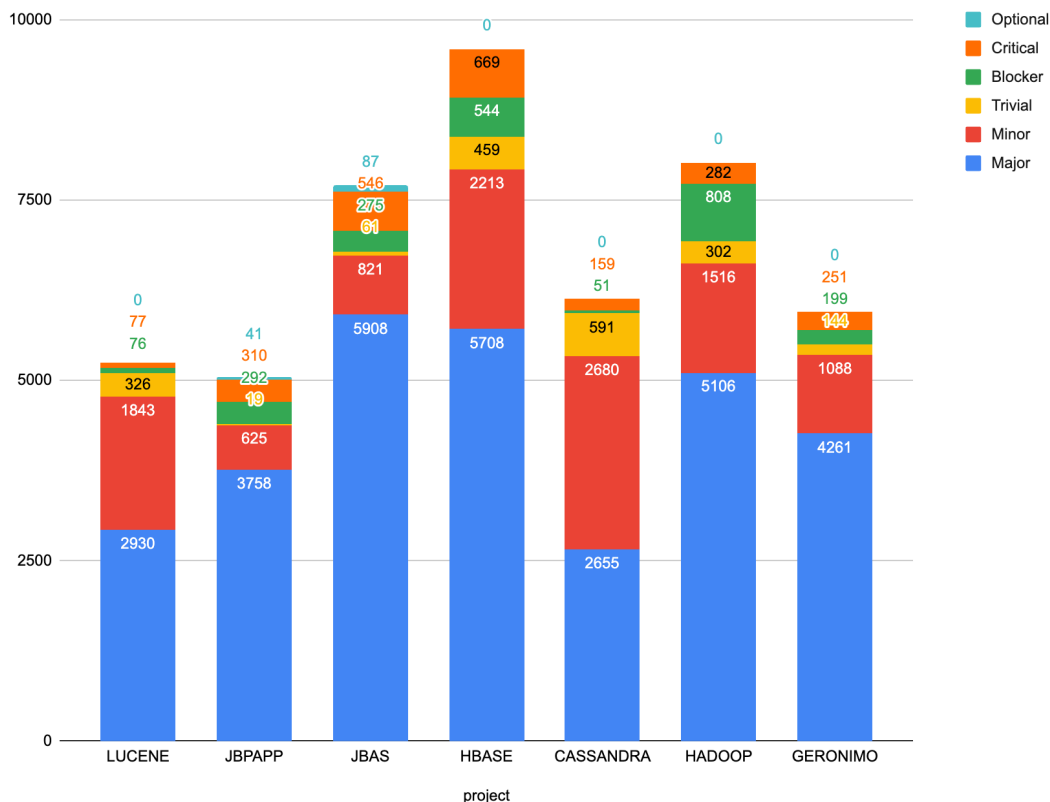


Рисунок 4.14 – Розподіл задач проєктів за пріоритетністю

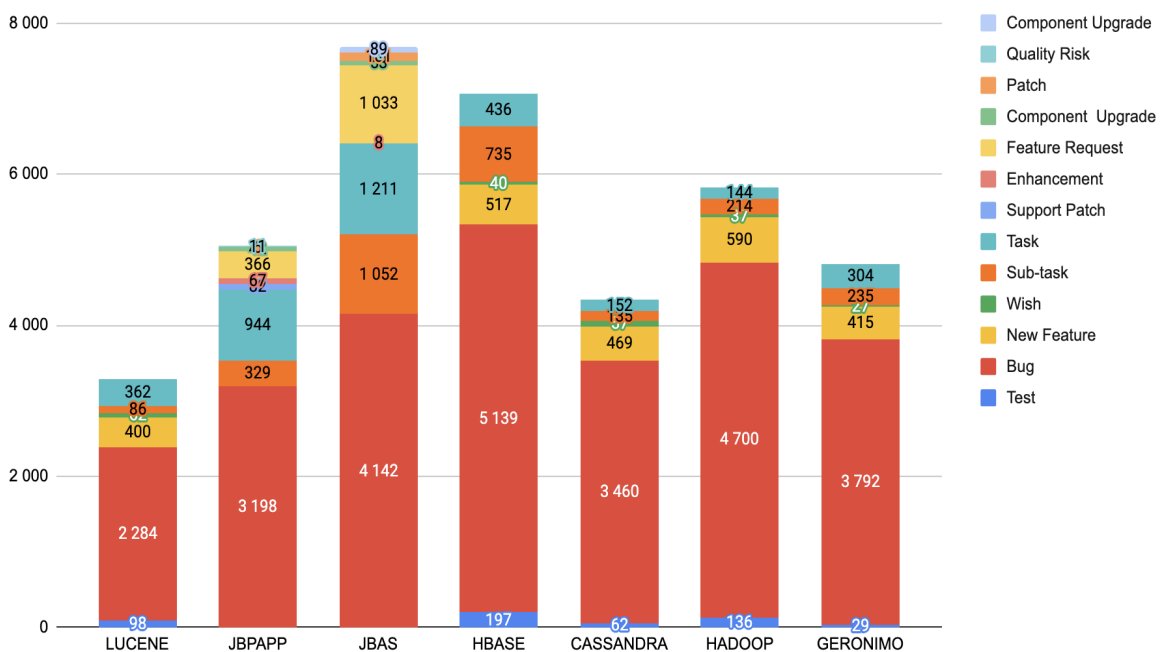


Рисунок 4.15 – Розподіл задач за типом завдання

Зареєстровано ще дві характеристики задач, які дуже важливі для кожного з проєктів – це тип та пріоритетність. Розподіл задач за типом та пріоритетом

представлено на рисунках 4.14 та 4.15. Можна помітити, що найбільша кількість задач мають пріоритет Major та Minor. Далі йдуть пріоритети Critical та Blocker, які є дуже важливими для правильної оцінки часу виконання задач. В розподілі задач за типом абсолютну більшість має тип Bug, що свідчить про те, що проєкти, що розглядаються, вже пройшли етап тестування.

Після проведення етапу аналізу та підготовки вхідних даних текстових описів задач спринту переходимо до другого етапу дослідження, а саме – проведення серій експериментів, метою яких є визначення найкращого підходу для автоматизації процесу класифікації якості описів задач спринту.

Перший експеримент пов'язано з тренування нейронної мережі на основі векторизованої репрезентації тексту. При проведенні цього експерименту обрано претреновану модель BERT (bert-base-uncased) з пакету Python для моделей класу трансформери [33].

Результати першої серії експериментів показали, що розробка моделі класифікації для ідентифікації якості текстових описів задач спринту вимагає залучення експертів, які обізнані змістом проєкту. Внесення формулювання оцінок під час ретроспективи спринту з метою тренування моделі класифікатора та залучення членів команди проєкту, як експертів, є цілком обґрунтованим. Це допоможе збирати дані для регулярного навчання моделі та покращить узгодженість оцінок, які надають різні експерти [93]. Відповідно до запропонованої методології, описаної у [93], задачі перетворені у векторні репрезентації, що дозволило визначити кластери та аналізувати їхні лінгвістичні особливості.

Для наочності, репрезентації задач були трансформовані у двовимірний простір, що дозволило візуалізувати розподіл та зв'язки між категоріями задач. Ці репрезентації та кластери, візуалізовані за допомогою різних кольорів, демонструють розподіл та взаємозв'язки між задачами, які можна переглянути на рисунках 4.16 та 4.17.

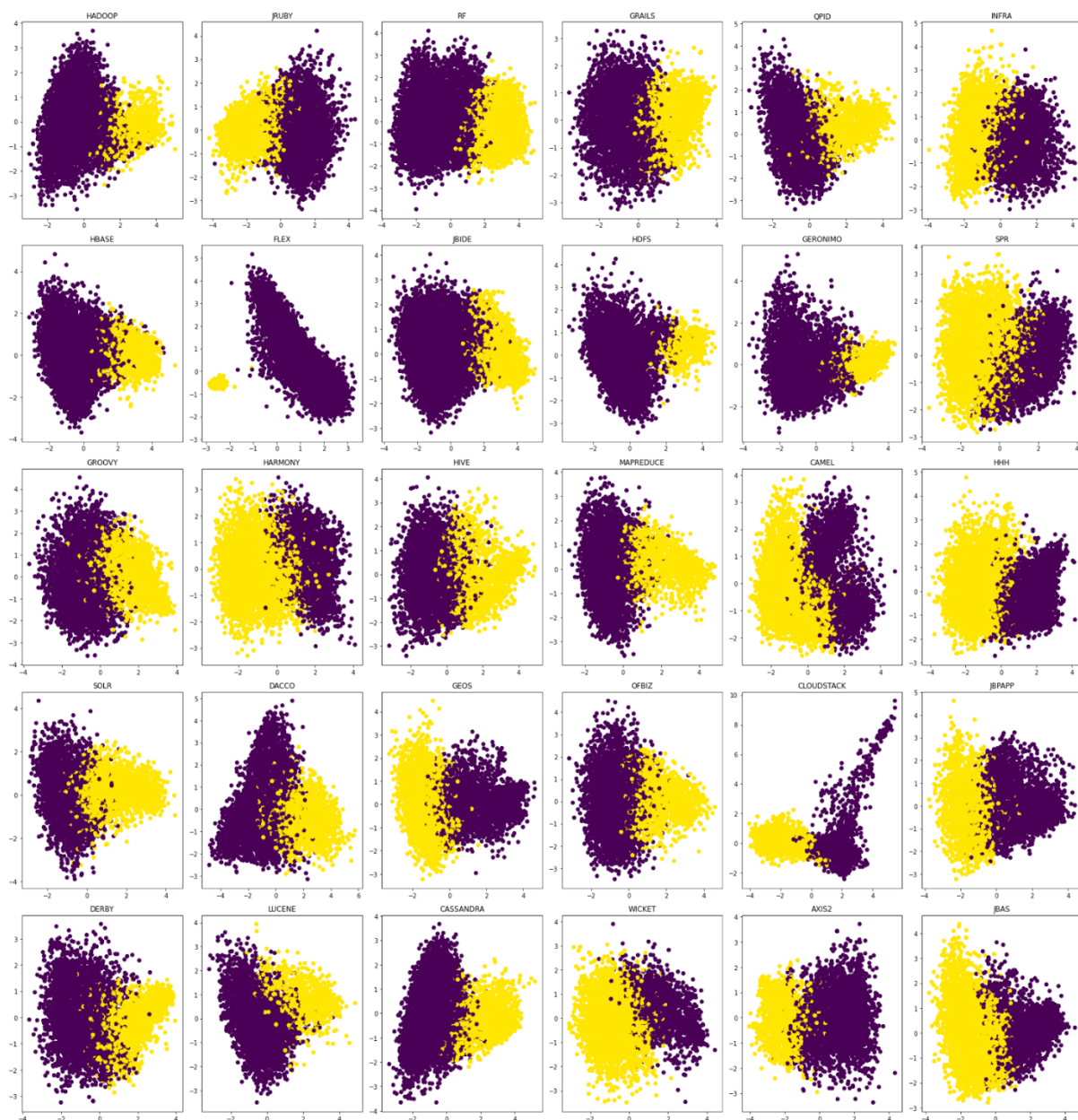


Рисунок 4.16 – Кластеризації задач проєктів методом BERT (двохмірна)

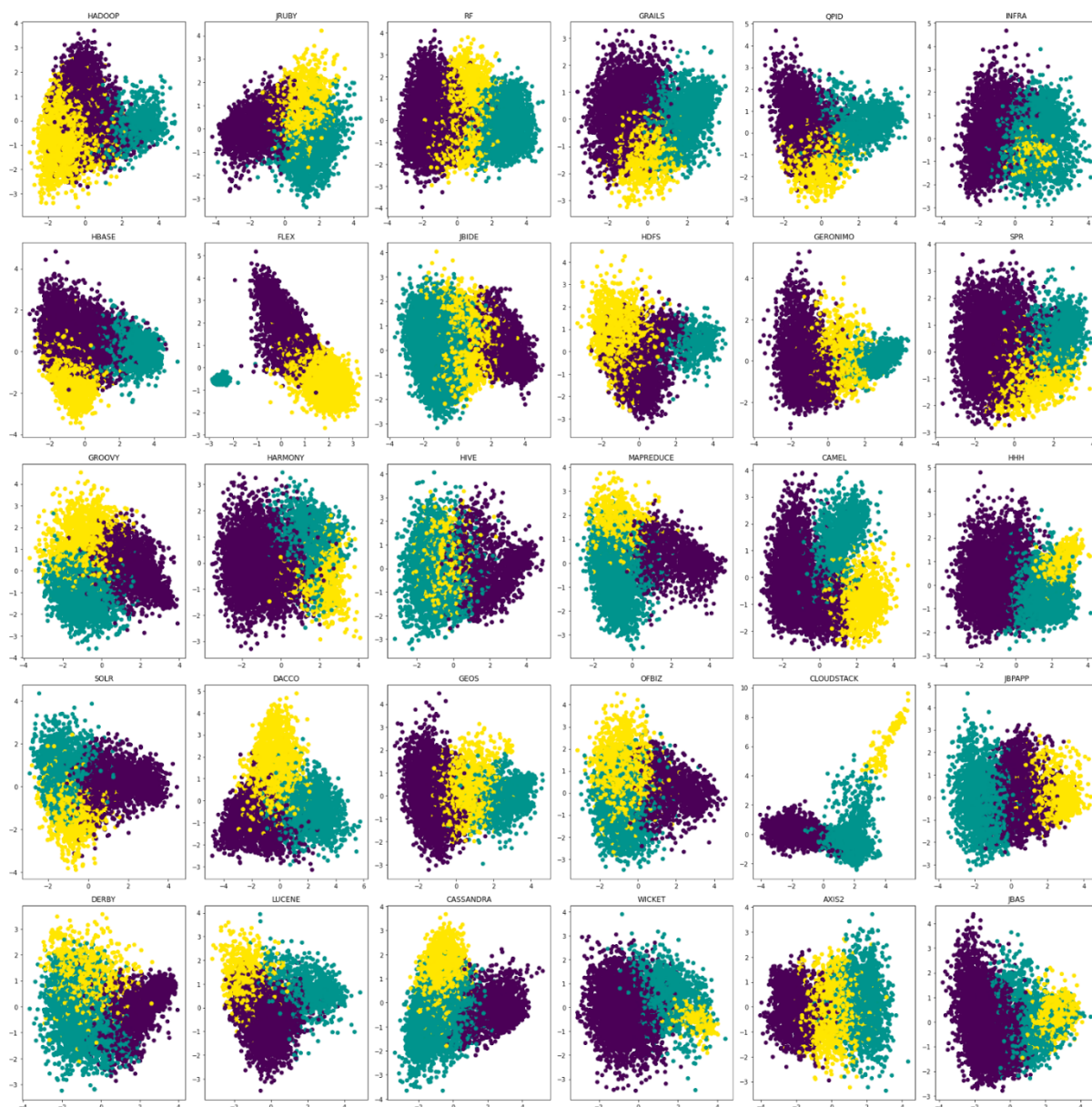


Рисунок 4.17 – Кластеризації задач проєктів методом BERT (трьохвимірна)

Було прийнято рішення кластеризувати задачі на два (рисунок 4.16) або три (рисунок 4.17) класи. Варіант із двома кластерами передбачає умовне розділення задач на «гарні» та «негарні», і це є ідеалізованою моделлю. Однак на практиці часто трапляються задачі, які не підпадають чітко під жодну з цих категорій через свою неоднозначність або атипові характеристики. Тому, для таких випадків, введено третій кластер, що дозволяє врахувати наявність викидів і підвищити гнучкість класифікації.

Аналіз отриманих кластерів показав, що не у всіх випадках можливо чітко та компактно розподілити всі задачі на логічні й однозначні групи. Водночас

було виявлено, що деякі проєкти демонструють тенденцію до групування за типами задач відповідно до їх лінгвістичних характеристик — зокрема, спостерігається окреме кластерування задач типу *Bug* (пов'язаних із помилками в коді) та *Feature/Improvement* (що стосуються розширення функціоналу або вдосконалення продукту). Цей поділ був підтверджений шляхом ручної перевірки експертами [33].

Оскільки кластеризація текстових описів задач спринту не дала бажаної чіткості, було прийнято рішення провести другу серію експериментів, зосереджену на впливі використання техніки промтінгу у взаємодії з великими мовними моделями на якість отриманих векторних репрезентацій задач.

У результаті попереднього експерименту було встановлено, що лише лінгвістичні характеристики текстового опису задачі не дозволяють чітко виявити її якість та зрозумілість. Це напряду узгоджується з практичним досвідом управління ІТ-проєктами, коли ключову роль у сприйнятті та інтерпретації задачі виконавцем відіграє не лише текст, а й додаткові елементи — файли, зображення, посилання, фрагменти коду, конфігураційні формати (наприклад, XML або JSON), а також згадки контактної інформації. Всі ці елементи є важливими джерелами контексту, що впливають на повноту та зрозумілість задачі для виконавців.

У другій серії експериментів було використано той самий набір даних, попередньо підготовлений та очищений. Для кожної задачі були збережені лише назва (title) та опис (description), а також проведено автоматичне визначення наявності специфічних компонентів (зображень, кодових вставок, URL, email, IP-адрес тощо), кожен з яких кодувався бінарним індикатором.

Для побудови векторних представлень задач було використано велику мовну модель phi-1.5 [70]. Важливою особливістю другого етапу було дослідження впливу промтінгу на якість отриманих векторів. Архітектура та токенізатор моделі залишались незмінними, а змінювався лише вхідний текст. У першому варіанті модель отримувала лише фактичний опис задачі (title + description). У другому — задачі подавалися з використанням інструктивного

промпту, який мав на меті акцентувати увагу моделі на ключових аспектах формулювання задачі (рисунок 4.18).

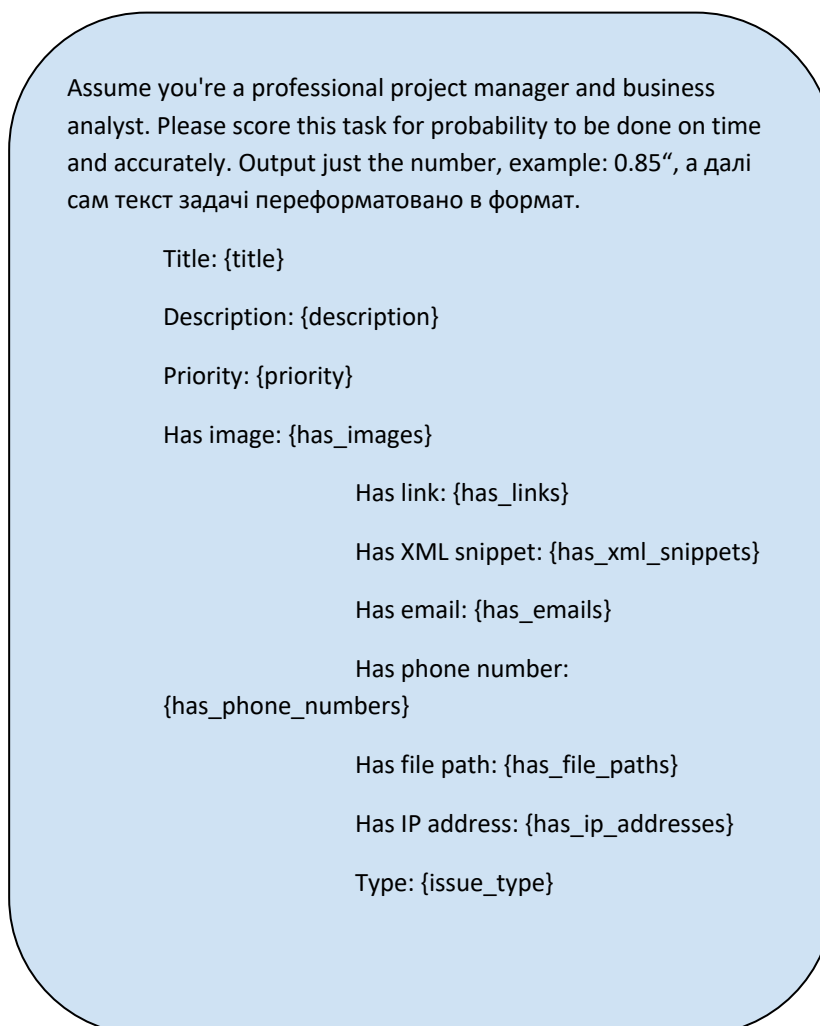


Рисунок 4.18 – Промпт для великої мовної моделі phi-1.5

де title-назва, description - опис, priority - пріоритетність задачі (вказана в даних), has_... - наявність кожного специфічного атрибуту в описі, issue_type - тип задачі такі як Bug, Feature, Improvement і тд.

Висновки цього етапу дозволяють припустити, що не лише зміст задачі, а й її структурна та контекстуальна повнота (зокрема наявність допоміжних технічних атрибутів) значно впливають на якість репрезентації, яку генерує велика мовна модель. Таким чином, промптінг виявився ефективним інструментом, що дозволяє покращити інтерпретацію вхідного тексту і зробити векторну репрезентацію більш релевантною до реального змісту задачі. Це

відкриває перспективи для точнішої автоматичної класифікації та аналізу задач у процесі управління беклогом.

Для побудови векторних репрезентацій проведено аналіз задач у двох текстових форматах, кожен з яких окремо перетворювався на токенизований вхід за допомогою спеціалізованого токенизатора. Після токенизації дані оброблялися великою мовною моделлю phi-1.5, яка генерувала представлення задач у вигляді тренувальних тензорів (hidden states) з останнього шару моделі, розмірністю 2048 елементів. Ці тензори слугували векторними репрезентаціями кожної задачі — компактним числовим узагальненням її семантики.

На основі отриманих векторів було проведено кластеризацію задач методом агломеративної кластеризації, що дозволяє об'єднувати найближчі елементи у групи без попередньо фіксованих форм. Кількість кластерів, як і раніше, обрана за змістовою логікою: два кластери відповідають умовним категоріям «хороші» та «погані» задачі, а третій кластер вводиться для задач, які не мають чіткої відповідності через неоднозначність, нестандартну структуру або відсутність достатнього контексту — тобто тип «невідомо».

Такий підхід дозволив перейти від поверхневого текстового аналізу до глибинної семантичної інтерпретації, де задачі класифікуються не лише за ключовими словами, а за структурою їх значення у векторному просторі. Застосування агломеративної кластеризації у цьому контексті виправдане, оскільки вона враховує ієрархічну близькість задач, що особливо важливо для аналізу об'єктів із частковим перетином змісту [83]. Аналіз показав, що векторні представлення, отримані з phi-1.5, дозволяють виокремити змістові подібності між задачами навіть при лінгвістичних відмінностях, що підтверджує ефективність даного підходу для кластеризації слабоструктурованих текстових описів.

Кількість кожного з атрибутів в задачах 30-ти проєктів представлено в таблиці 4.2. Можна побачити, що найчастіше в задачах зустрічаються шляхи до файлів та картинки.

Таблиця 4.2 – Головні атрибути задач проєктів, що розглядаються

Назва project	Атрибути задач проєктів						
	has images	has links	has xml snippets	has emails	has phone numbers	has file paths	has ip addresses
LUCENE	6	679	86	89	16	1992	15
JBPAPP	11	1432	304	12	20	2580	272
JBAS	5	1110	664	75	35	3458	307
HBASE	7	823	100	69	264	3061	399
CASSANDRA	8	581	29	90	140	2071	307
HADOOP	7	609	93	29	47	2720	159
GERONIMO	7	887	352	74	13	2533	225
HIVE	0	517	40	45	50	1672	40
AXIS2	11	1519	1073	113	26	2719	154
HHH	10	1319	843	61	10	2887	89
HDFS	2	364	46	53	77	1436	185
DACCO	6	76	99	0	13	1483	0
OFBIZ	35	869	313	31	13	2002	130
HARMONY	24	838	70	74	50	2308	63
GEOS	80	1508	589	39	74	2449	101
WICKET	47	1026	410	56	17	2031	31
SPR	17	2032	1891	45	14	5080	163
QPID	1	337	41	38	20	1629	148
GROOVY	11	769	184	38	21	1843	27
SOLR	12	1183	571	80	19	2433	80
DERBY	9	1126	69	91	26	2831	550
JRUBY	19	1184	76	75	73	3717	204
CAMEL	3	1960	477	49	6	3038	89
JBIDE	584	2874	893	50	37	5697	95
FLEX	394	16296	4711	10763	66	21202	237
MAPREDUCE	2	319	50	18	30	1600	91
RF	284	1904	1972	18	12	4130	107
INFRA	74	3380	46	1861	21	4122	250
GRAILS	19	1478	661	58	11	3292	89
CLOUDSTAC K	13	944	121	78	533	2602	1387

Таким чином отримано для кожного проєкту індекси класів. З метою наочної візуалізації, 2048-вимірні векторні репрезентації були трансформовані в двовимірний простір використовуючи методи PCA [63] та t-SNE [72], що дозволило ефективно та наочно їх відобразити.

Результати проведення векторної репрезентації з використанням методів PCA та t-SNE представлено на рисунках 4.19 та 4.20.

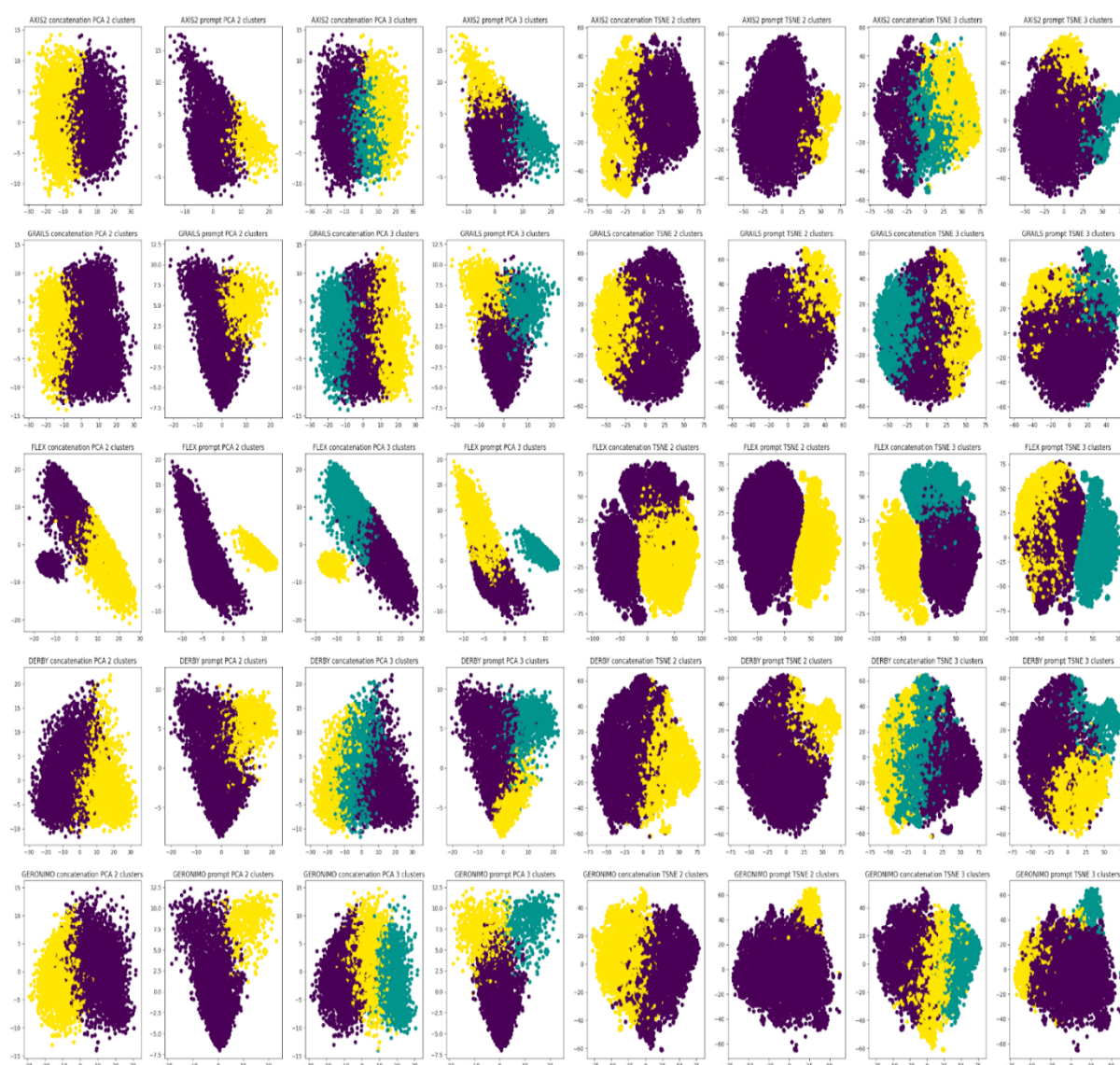


Рисунок 4.19 – Кластеризація задач проєктів методом PCA

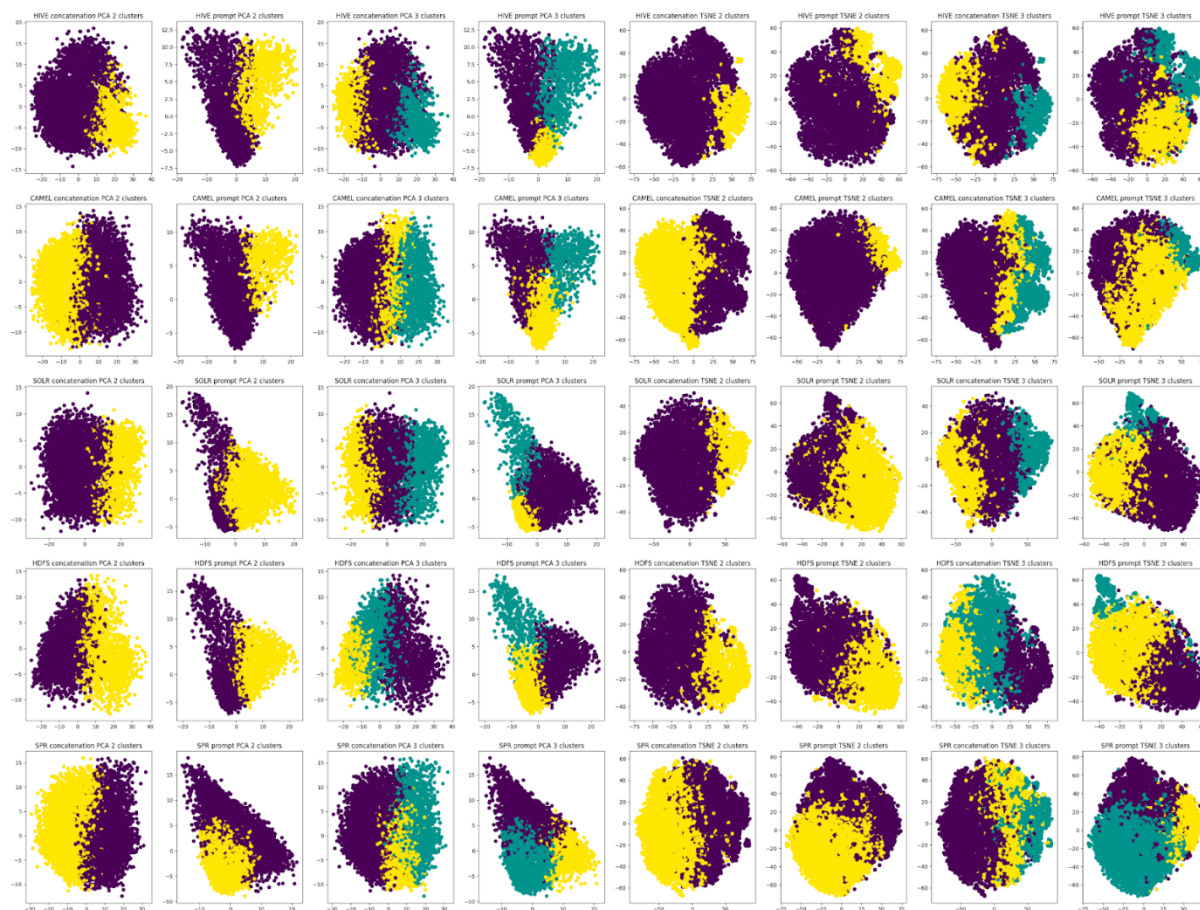


Рисунок 4.20 – Кластеризація задач проєктів методом t-SNE

Жовтим кольором позначені задачі, які потрапили до кластера «хороші», фіолетовим – «погані» і, відповідно, зеленим – «невідомі».

Проаналізувавши результати кластеризації ніякої лінгвістичної закономірності, які б однозначно визначали якість або тип задач, як і в попередніх експериментах з використанням BERT, знайдено не було. Але, слід зазначити, що векторні репрезентації, отримані з використанням промптінгу, виявились більш структурованими та стабільними, що може свідчити про те, що використання промптінгу при взаємодії з великими мовними моделями (LLM) дає більш точні результати аналізу якості задач.

Це відбувається тому, що правильно сформульований промпт допомагає моделі сфокусувати увагу на певних аспектах задачі.

У наступній серії експериментів досліджено використане великих мовних моделей для оцінки якості задач на різних наборах даних. Експеримент

спрямований на ідентифікацію та кластеризацію факторів, що впливають на якість текстових описів задач, та перевірку ефективності використання LLM для їх оцінювання. Під час експерименту порівняно різні LLM. Для цього, окрім набору публічних даних, використано дані з приватного проєкту [95]. Зроблено такий самий пошук релевантних атрибутів в описах задач і той самий промпт. Запит відправлено до LLM OpenAI GPT, використовуючи різні версії цієї моделі (3.5-turbo та 4).

При проведенні аналізу 1000 задач із приватного датасету за допомогою GPT3.5-turbo отримано розподіл вірогідностей своєчасного і точного виконання задачі, який представлено на рисунку 4.21. Можна побачити, що використовуючи підібране порогове значення, можна зробити класифікатор задач по їх опису. Але в результаті відповідність вірогідності та реальної задачі під час аналізу відсутня.

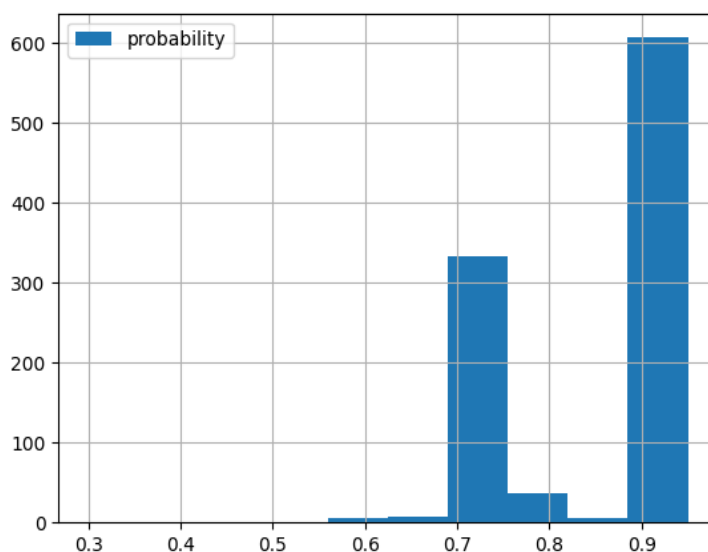


Рисунок 4.21 – Розподіл вірогідностей виконання задач

У наступному експерименті з цього набору даних було обрано 100 прикладів, по 50 кожного класу з розмітки експертами [95]. Ці 100 прикладів було оброблено так само як у всіх попередніх експериментах та проаналізовано за допомогою GPT3.5-turbo та GPT4. На рисунку 4.22 можна побачити, що

розподіли цих двох моделей не збігаються, що показує нам не консистентність моделей при виконанні задачі.

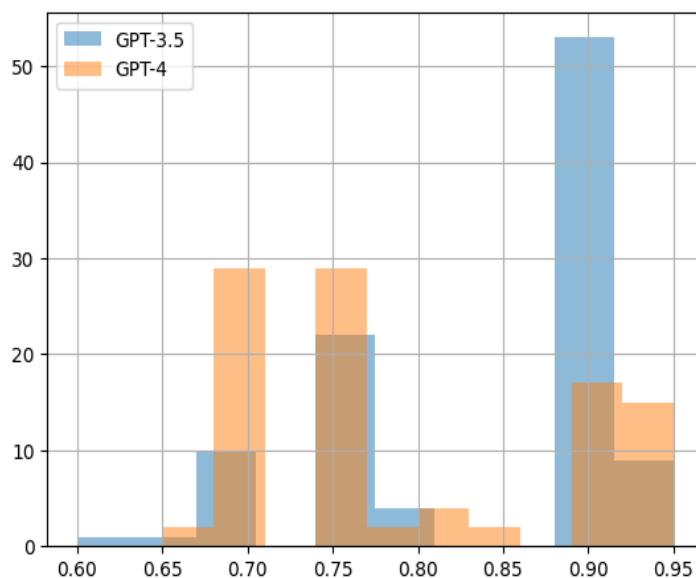


Рисунок 4.22 – Розподіл вірогідностей виконання задач за допомогою GPT3.5-turbo та GPT4

Аналогічно, з одного з проєктів публічного датасету обрано 100 випадкових елементів і проведено той самий експеримент. Отримано розподіл вірогідностей, який представлено на рисунку 4.23. В цьому випадку спостерігається схожість розподілів даних, але проаналізувавши кореляцію між ними, було отримано значення 16%, що підтверджує гіпотезу про можливість використання LLM для класифікації.

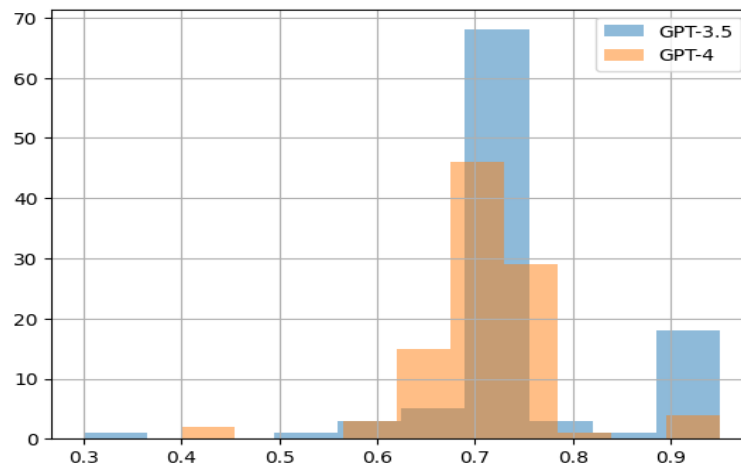


Рисунок 4.23 – Порівняння розподілів вимогностей вчасного виконання задач для різних GPT

Результати експериментів чітко засвідчують, що для розробки ефективної моделі класифікації, метою якої є оцінка якості текстових описів задач спринту, критично необхідно залучення експертів з глибоким розумінням контексту проекту. Саме такі експерти здатні надати чіткі орієнтири для правильної інтерпретації даних, зокрема при формуванні навчальної вибірки. Крім того, експерименти підкреслюють, що ефективність моделі класифікації значною мірою залежить від якісного вибору методу векторизації тексту та адаптації архітектури моделі до специфіки даних, що отримані із систем управління проектами.

На третьому етапі технології дослідження лінгвістичних характеристик виконано порівняльний аналіз якості модельних оцінок з оцінками експертів. Здійснено спробу оцінити та скоригувати задачі для допомоги менеджеру проекту. Для цього відібрано 274 задачі приватного проекту виходячи з рівномірного розподілу позитивно та негативно розмічених експертами задач. Для виконання цього завдання обрано модель GPT-4 через OpenAI API. Кожна задача оброблена для знаходження релевантних показників, як це було зроблено у попередніх експериментах. Для реалізації цієї задачі створено промпт, що представлено на рисунку 4.24.

Assume you're a professional project manager and business analyst. Please score this task for probability to be done on time and accurately. Output the number and how the task can be changed. Example: Probability: 0.85
title: 'improved version of title' description: 'improved version of description'

Title: {title}

Description: {description}

Priority: {priority}

Has image: {has_images}

Has link: {has_links}

Has XML snippet: {has_xml_snippets}

Has email: {has_emails}

Has phone number: {has_phone_numbers}

Has file path: {has_file_paths}

Has IP address: {has_ip_addresses}

Type: {issue_type}

Рисунок 4.24 – Промпт для моделі GPT-4 через OpenAI API

Задачі приватного проєкту послідовно відправлено в модель для отримання вірогідності ясності задачі та покращення можливих назви та опису задачі. Результати експерименту наведено в таблиці 4.3.

Таблиця 4.3 – Результати експерименту з використанням модель GPT-4 через OpenAI API

Input	Output	Probability
Title: Magic wand error on thermal images	Title: ‘Detailed Error report on magic wand tool for thermal images’ Description: ‘Detailed bug report including error message codes, screen-shots and steps to reproduce Magic Wand error on thermal images’	0.6
Title: Research depth map to 3D	Title: ‘In depth Research on 3D Mapping’ Description: ‘Conduct an extensive research on 3D mapping techniques and technologies, focusing on latest models, accuracy, efficiency and operational requirements. The research should include both open source and commercial solutions available in the market. The research output should involve a well compiled, comprehensive document, presenting comparisons, pros, cons and practical applications of each technology.’	0.7
Title: FE error on upload image without geo data Description: Error in console in Asset page in image details when image without geo data	Title: Correct Front-End Error During Upload of Images Lacking Geo-Data Description: An error is being experienced on the Asset page on the console. This error specifically occurs during the upload of image details that do not contain geo-data. A high level of priority has been assigned to this task. To expedite the resolution of this issue and for accurate diagnosis, please include any unique error messages, steps leading up to the error, screenshots, and any relevant links or snippets of code.	0.65
Title: Reidentification doesn’t work	Title: Resolve Reidentification Functionality Issue Description: Carefully trace the issue with	0.6

	the reidentification functionality, identify the root cause, and develop an effective solution to fix it. This task must be rigorously tested before being considered complete to ensure that the issue has been fully resolved and no new issues have been created in the process.	
Title: Add waiting spinner on sign up request process	Title: “Add waiting spinner on sign up request process”	0.9
Title: Create report about regenerated models on Prod	Title: Produce a Report on Regenerated Models in the Production Environment Description: Create a comprehensive report detailing the regenerated models in the production environment. This report should include information about the model regeneration process and the impact of these changes. This task is considered medium priority.	0.85
Title: Icon colors on the map Description: In the map view of the inspection page, the asset and image icon colors should be the following: Asset: We should be able to differentiate on the map between assets that are 1. Inspected with no defects (green), 2. Assets with defects (Red) 3. Assets with images associated but not inspected. (Dark Gray) 4. Assets without images	Title: “Enhanced Asset and Image Icon Color Coordination in Map Inspection” Description: “For the mapping interface of the inspection page, we are to improve the color-coding scheme. This involves utilizing distinct colors to distinguish between various asset and image statuses. The color indications for the following situations shall be: For Assets, we should use: 1. A ‘green’ to denote assets inspected and found without defects. 2. A ‘red’ color for assets with detected defects. 3. Assets with associated images, but yet to be inspected should be distinguished with ‘dark gray’. 4. ‘Light gray’ to denote assets without any images attached.	0.8

(Light Gray) Images” For images 1. Inspected with defects (Red) 2. Inspected without defects (Green) 3. Not inspected (Dark Gray)	For images, the color identification scheme should be: 1. ‘Red’ for inspected images containing defects. 2. ‘Green’ for inspected images that are found to be defect-free. 3. Images yet to be inspected can be distinguished using ‘dark gray’.	
Title: Delete images from raptormaps on Demo account	Title: Delete images from RaptorMaps on Demo account Description: This task involves deleting images which are no longer needed or relevant from the Demo account. As this task will involve some computing skills and knowledge of the RaptorMaps platform, it’s assumed that the team is familiar with it and has access to this account.	0.9

На основі отриманих результатів можна зробити висновок, що формулювання окремих задач справді може бути суттєво покращене за допомогою великих мовних моделей. Такі моделі здатні не лише уточнювати та переформулювати опис задачі, роблячи її більш зрозумілою для виконавців, але й виявляти недоліки формулювання ще на етапі створення. Це дає можливість інтегрувати LLM до інформаційної технології, як інтелектуального помічника для менеджера проєкту, що дозволить підвищити якість формування задач у спринті та загального покращення ефективності команди за рахунок кращого розуміння і трактування вимог.

4.3 Дослідження результатів розподілу задач на спринт

Ще одним ключовим чинником, що безпосередньо впливає на успішність реалізації проєкту, є правильний розподіл задач між членами команди. Це важливо тому, що неузгоджений або нерівномірний розподіл може призвести до

затримок у виконанні спринтів, втрати фокусу або низької якості реалізації задач. В той самий час, точне призначення задач відповідно до компетенцій, завантаженості та пріоритетів дозволяє команді працювати більш злагоджено, уникати простоїв та знизити ризики невиконання.

Спроектовано експерименти для тестування розроблених алгоритмів за різних умов, зосереджуючись на таких змінних:

- розмір команди: кількість розробників (мала, середня, велика);
- кількість задач: варіюється відносно розміру команди для моделювання різного навантаження (легке, помірне, важке);
- розподіл навичок: розробники мають подібний рівень навичок (гомогенний) або розробники мають різноманітні навички (гетерогенний);
- структура преференцій: корельована (преференції узгоджені з навичками) чи випадкова (преференції призначаються випадково).

Це дало можливість сформулювати кілька варіантів експериментальних умов. Було виділено вісім базових сценаріїв, які різняться за такими параметрами, як розмір команди, обсяг задач, розподіл навичок виконавців та характер їхніх преференцій.

Зокрема, щоб оцінити, як алгоритми працюють за різного рівня професійної підготовки команди, було змодельовано кілька типів розподілу навичок. Також, для перевірки впливу узгодженості або випадковості преференцій виконавців на якість розподілу задач, було розглянуто сценарії з різними типами переваг: від чітко визначених до випадкових.

Такий підхід дозволяє глибше проаналізувати стійкість та адаптивність алгоритмів у різних організаційних умовах.

Сформовані варіанти сценаріїв представлено в табл. 4.4. Слід зауважити, що сценарій 5 зосереджується на впливі терміновості задач.

Таблиця 4.4 – Опис умов експериментів

Experiment Number	Name	Team Size	Number of Tasks	Skill Distribution	Preference Structure
1	Exp1_Small_Homogeneous_Light	5	5	homogeneous	correlated
2	Exp2_Small_Heterogeneous_Light	5	5	random	correlated
3	Exp3_Medium_Heterogeneous_Moderate	15	30	random	correlated
4	Exp4_Medium_Homogeneous_Random	15	15	homogeneous	random
5	Exp5_Sensitivity_Urgency	15	30	random	correlated
6	Exp6_Large_Heterogeneous_Heavy	30	90	random	correlated
7	Exp7_Large_Homogeneous_Random	30	30	homogeneous	random
8	Exp8_Large_Heterogeneous_Moderate	30	30	random	correlated

Таким чином, сформовано сценарії експериментів для тестування обраних алгоритмів за допомогою розробленого фреймворку.

Результати експериментів, які проведено за розробленими сценаріями, наведено у таблиці 4.5. Розглянемо більш детально отримані оцінки для різних алгоритмів та сценаріїв.

Аналіз результатів показує, що алгоритми EADAM та SOSM продемонстрували стабільно високі значення індексу сумісності в усіх експериментальних сценаріях, що свідчить про їхню здатність формувати якісні та узгоджені зіставлення між задачами й виконавцями.

Водночас, алгоритм MESMA дещо перевершив EADAM та SOSM за цим показником, що пояснюється його оптимізаційною природою, спрямованою на покращення загальної відповідності.

Отже, можна зробити висновок, що EADAM та SOSM є ефективними алгоритмами для створення зіставлень, особливо у випадках, коли потрібна балансованість і передбачуваність результатів. MESMA, у свою чергу, може забезпечити незначне покращення якості, але за рахунок вищої обчислювальної складності, що варто враховувати під час вибору підходу для практичного застосування.

Таблиця 4.5 – Результати експериментів

Experiment Number	Algorithm	Total Compatibility Score	Blocking Pairs	Developer Satisfaction	Runtime (s)
1	EADAM	4,30	0	3,00	0,0000
	MESMA	4,30	0	3,00	0,0090
	RSD	4,30	0	3,00	0,0000
	SOSM	4,30	0	3,00	0,0000
	TTC	4,30	4	3,00	0,0000
2	EADAM	3,22	0	1,74	0,0000
	MESMA	3,42	0	1,85	0,0055
	RSD	3,21	0	1,82	0,0000
	SOSM	3,22	0	1,74	0,0000
	TTC	2,58	2	2,17	0,0000
3	EADAM	14,43	0	4,34	0,0001
	MESMA	14,53	0	4,57	0,4510
	RSD	14,45	0	4,35	0,0000
	SOSM	14,43	0	4,34	0,0001
	TTC	8,13	17	10,29	0,0003
4	EADAM	13,00	0	7,42	0,0002
	MESMA	13,00	0	7,42	0,3780
	RSD	13,00	0	7,47	0,0000
	SOSM	13,00	0	7,42	0,0001
	TTC	13,00	52	7,99	0,0003

5	EADAM	14,43	0	4,34	0,0001
	MESMA	14,53	0	4,57	0,4435
	RSD	14,45	0	4,35	0,0000
	SOSM	14,43	0	4,34	0,0001
	TTC	8,13	17	10,29	0,0003
6	EADAM	30,76	0	8,34	0,0004
	MESMA	30,84	0	9,00	19,9934
	RSD	30,77	0	8,27	0,0000
	SOSM	30,76	0	8,34	0,0002
	TTC	18,53	87	27,37	0,0022
7	EADAM	26,04	0	14,16	0,0008
	MESMA	26,04	0	14,16	6,7477
	RSD	26,04	0	14,13	0,0001
	SOSM	26,04	0	14,16	0,0004
	TTC	26,04	222	15,57	0,0014
8	EADAM	23,59	0	7,76	0,0003
	MESMA	24,49	0	8,38	1,9629
	RSD	23,64	0	7,76	0,0000
	SOSM	23,58	0	7,76	0,0002
	TTC	18,81	92	10,25	0,0007

За критерієм стабільності розподілу (Blocking Pairs), алгоритми EADAM та SOSM продемонстрували найкращі результати. В усіх експериментальних сценаріях вони забезпечували повністю стабільні зіставлення, тобто не створювали жодної блокуючої пари.

Натомість алгоритм RSD не гарантує стабільності за своєю природою, а TTC часто породжував значну кількість блокуючих пар, особливо у складніших або більших конфігураціях, що свідчить про його вразливість у контексті стабільності.

Варто окремо зазначити, що MESMA не повідомляє блокуючі пари явно, тому його складно об'єктивно порівняти з іншими алгоритмами саме за цим показником.

З огляду на результати, можна зробити висновок, що TTC є менш придатним у ситуаціях, де стабільність розподілу критично важлива, тоді як EADAM та SOSM залишаються більш надійним вибором у таких умовах.

З точки зору критерію задоволення розробників (Developer Satisfaction), алгоритми, що забезпечують стабільність, також сприяють і кращому показнику цього критерію.

Час виконання алгоритмів представлено на рисунку 4.25, де наочно можна побачити, що MESMA має значно довший час виконання, особливо у експериментах з великими командами (майже 20 секунд у експерименті 6), що пояснюється його обчислювальною складністю.

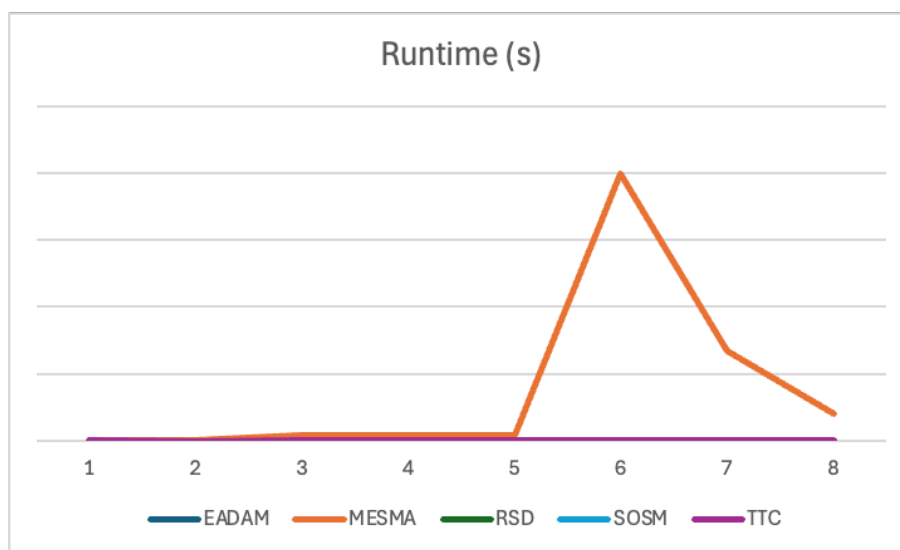


Рисунок 4.25 – Час виконання

Зведені результати використання різних алгоритмів наведено на рисунку 4.26.

У контексті продуктивності різних алгоритмів було встановлено, що EADAM і SOSM демонстрували високу ефективність, забезпечуючи час виконання в межах від мікросекунд до мілісекунд (рисунок 4.26). Алгоритм RSD

виявився найшвидшим через свою просту реалізацію, тоді як TTC показав середній час виконання, який помітно зростав із розміром задачі.

Хоча MESMA продемонстрував незначні покращення за деякими показниками якості, його висока обчислювальна вартість і значне збільшення часу виконання ставлять під сумнів доцільність його використання, особливо в умовах, коли ресурси обмежені або потрібна швидка реакція.

Загалом, EADAM та SOSM забезпечили не лише стабільні та узгоджені розподіли для малих команд, а й високу продуктивність у більших командах з різномірними наборами навичок. Натомість нестабільність TTC стала більш помітною при зростанні розміру команди — зросла кількість блокуючих пар і знизився рівень сумісності, що обмежує його масштабованість. У свою чергу, MESMA, попри окремі переваги, втрачає ефективність на великих даних через значне зростання обчислювальних витрат.

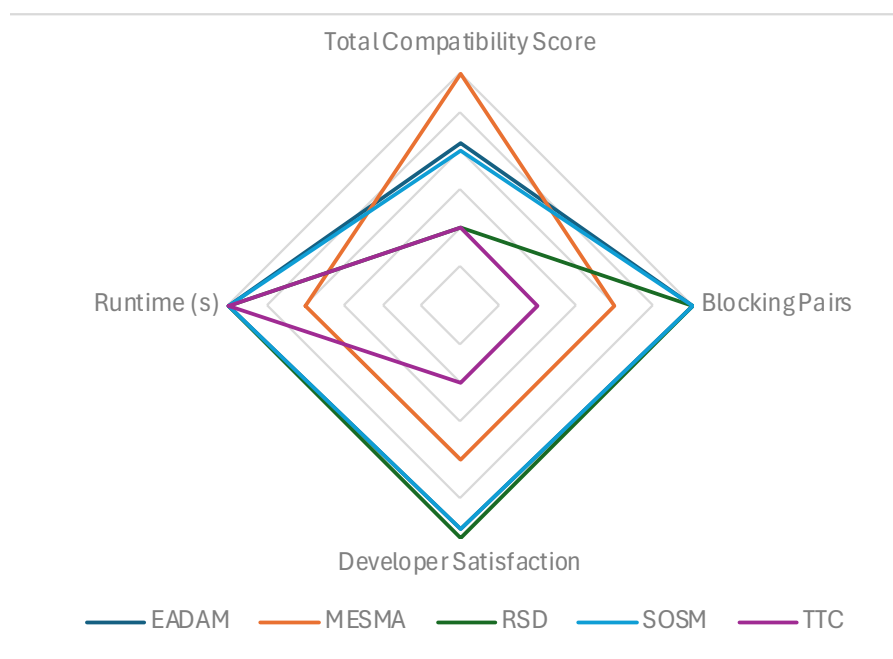


Рисунок 4.26 – Порівняльний аналіз

У четвертому та сьомому сценаріях (гомогенні навички, випадкові преференції) усі алгоритми досягли схожих показників сумісності. TTC показав

обмеження в складних сценаріях – навіть з гомогенними навичками структура преференцій впливає на його стабільність.

У підсумку можна зробити висновок, що алгоритми EADAM та SOSM є найбільш придатними для практичного застосування завдяки оптимальному балансу між стабільністю, ефективністю та рівнем задоволення учасників.

Алгоритм MESMA, хоча й демонструє деякі покращення у якості зіставлень, вимагає значно більше обчислювальних ресурсів, що може стати обмеженням у реальних умовах, особливо при масштабуванні.

Алгоритм TTC показав низьку стабільність і менший рівень задоволення, що робить його менш ефективним для великих або різномірних команд.

Незважаючи на простоту і швидкість, алгоритм RSD має серйозний недолік — відсутність стабільності, що критично впливає на якість формування розподілу задач в проєкті.

Таким чином, можна підсумувати, що алгоритми EADAM та SOSM можна вважати найбільш збалансованими та надійними рішеннями для реалізації в системах автоматизованого розподілу задач між виконавцями.

4.4 Впровадження результатів

Розроблена інформаційна технологія спрямована на покращення ефективності управління проєктами та підвищення продуктивності роботи команд проєкту за рахунок підвищення якості формування задач та кращому їх розподілу між виконавцями.. Ця технологія взаємодіє із існуючими інструментами для відстеження задач, зокрема JIRA, і забезпечує дві основні функції:

- перевірка, оцінка та покращення задач, запланованих для спринтів;
- рекомендації щодо ефективного розподілу задач між учасниками команди на основі їх навичок, з використанням великих мовних моделей для аналізу описів задач спринту проєкту.

Попередні дослідження [95, 93, 96, 46] показали, що ризики, пов'язані з неоднозначністю задач та невідповідністю навичок, можуть перешкоджати успіху проєкту. Завдяки впровадженню розробленої інформаційної технології, яка покращує зрозумілість задач та їх розподіл, зменшиться вказані ризики, підвищується задоволення виконавців та збільшується продуктивність роботи команди проєкту.

Для оцінки ефективності впровадження інформаційної технології розглядався проєкт компанії "Dominet», який передбачав розробку автоматизованої системи. управління, яка забезпечувала узгодженість та керованість умов експерименту. Проведені експерименти оцінки ефективності впровадження інформаційної технології для малої команди (до 5 розробників) та середньої команди (6–15 розробників) протягом чотирьох спринтів.

Отримані дані за ключовими показниками до та після впровадження розробленої інформаційної технології для малої команди (до 5 розробників) наведено у таблицях 4.6. та 4.7.

Таблиця 4.6 - Результати метрик для малої команди

Назва метрики	Значення до впровадження	Значення після впровадження
Рівень дефектів (дефекти на задачу)	0,08	0,05
Індекс виконання графіка (SPI)	1,00	1,03
Продуктивність (задача на розробника)	5	6
Рівень прийняття рекомендацій (%)	Н/Д	75
Рівень прийняття розподілу задач (%)	Н/Д	85

Таблиця 4.7 – Зміни метрик протягом чотирьох спринтів впровадження інформаційної технології

Номер спринту	Рівень дефектів	SPI	Продуктивність	Рівень прийняття рекомендацій (%)	Рівень прийняття розподілу задач (%)
1	0.07	1.01	5.5	65	80
2	0.06	1.02	5.8	70	83
3	0.05	1.03	6.0	77	86
4	0.04	1.04	6.2	80	90

Метрики для середньої команди (6–15 розробників) наведено в таблицях 4.8 та 4.9.

Таблиця 4.8 – Метрики для середньої команди до та після впровадження інформаційної технології

Назва метрики	Значення метрики до впровадження	Значення метрики після впровадження
Рівень дефектів (дефекти на задачу)	0.12	0.09
Індекс виконання графіка (SPI)	0.97	0.99
Продуктивність (задача на розробника)	4	4.5
Рівень прийняття рекомендацій (%)	Н/Д	65
Рівень прийняття розподілу задач (%)	Н/Д	78

Таблиця 4.9 – Зміни метрик протягом чотирьох спринтів після впровадження інформаційної технології

Номер спринту	Рівень дефектів	SPI	Продуктивність	Рівень прийняття рекомендацій (%)	Рівень прийняття розподілу задач (%)
1	0,11	0.97	4,2	55	72
2	0,10	0,98	4,4	60	75
3	0,09	1,00	4,6	68	80
4	0,08	1,01	4,8	75	85

Таким чином, було проаналізовано зміни в рівні дефектів і продуктивності команд до та після впровадження розробленої інформаційної технології. Спостерігається помітне зниження кількості дефектів у обох командах, що свідчить про покращення якості виконання задач. Показник SPI (Schedule Performance Index) також зріс, що вказує на краще дотримання графіків завдяки ефективнішому плануванню після впровадження запропонованої технології. Крім того, в обох командах зросла кількість виконаних задач на одного розробника, що говорить про підвищення індивідуальної продуктивності. Важливо зазначити, що рівень прийняття нових підходів і рішень з часом зростав, демонструючи поступову адаптацію команд до запропонованих змін.

4.5 Висновки до розділу 4

У даному розділі проведено експериментальне дослідження ефективності запропонованих в попередніх розділах моделей та методів інтелектуального планування роботи команди ІТ-проекту. Основна увага приділена перевірці можливостей автоматизованого виявлення неякісних задач у спринті, побудові моделей класифікації якості формулювань, кластеризації задач на основі векторних репрезентацій та дослідженню ефективності алгоритмів розподілу задач з урахуванням структури команди. В якості вихідних даних було використано як приватний набір задач реального проекту, так і публічні дані з відкритих джерел.

Розв'язано задачу автоматизованого виявлення неякісних формулювань задач спринту, що суттєво впливають на ефективність роботи команди. Запропонована інформаційна технологія на основі інтеграції методів машинного навчання та великих мовних моделей дозволила покращити зрозумілість задач і знизити ризики, пов'язані з неправильним або неповним сприйняттям задач виконавцями.

Розглянуто приклад впливу контекстної експертизи на розмітку даних, де продемонстровано, що залучення експертів, які мають глибоке розуміння

проєкту, є необхідною умовою для якісного формування навчальної вибірки. Аналіз розміток показав низьку узгодженість між незалежними експертами (коефіцієнт каппа $\leq 0,29$), що підкреслює важливість контексту у процесі оцінювання задач.

Розв'язано задачу порівняльного аналізу класифікаторів якості описів задач, у якому встановлено, що нейронна мережа на базі BERT досягає значно кращих результатів ($F1 = 83\%$) у порівнянні з наївним байєсівським класифікатором (точність 70%). Це свідчить про високу ефективність використання трансформерних моделей у задачах, пов'язаних з інтерпретацією коротких текстів з обмеженим контекстом.

Вирішено задачу побудови кластерів задач на основі векторних репрезентацій, отриманих за допомогою великої мовної моделі phi-1.5 з використанням промтінгу. Встановлено, що промтінг підвищує структурованість та стабільність репрезентацій, що в свою чергу дозволяє краще виявляти логічні групи задач, зокрема ті, що належать до типів Bug, Feature або Improvement. Це підтверджує потенціал LLM як інструменту підтримки аналітики в беклозі.

Розглянуто вплив різних алгоритмів розподілу задач у спринті на ефективність команди, зокрема алгоритмів EADAM, SOSM, MESMA, RSD та TTC. Встановлено, що EADAM і SOSM забезпечують найкращий баланс між стабільністю, сумісністю та швидкодією, що робить їх оптимальним вибором для практичного застосування в командах різного розміру. Натомість TTC виявився чутливим до структури команди, а MESMA — ресурсозатратним при масштабуванні, хоча і демонстрував трохи вищі результати за деякими метриками.

Результати підтвердили ефективність запропонованого підходу до інтелектуального планування роботи команди ІТ-проєкту на основі гнучкої методології та довели практичну цінність застосування штучного інтелекту для автоматизації аналізу якості задач і їх розподілу. Крім того, підтверджено доцільність використання великих мовних моделей, технік промтінгу та

алгоритмів машинного навчання для покращення якості планування в рамках гнучких методологій управління проєктами.

ВИСНОВКИ

В результаті виконаних теоретичних та експериментальних досліджень було вирішено актуальну науково-практичну задачу, яка полягає в розробці моделей, методів та інформаційної технології інтелектуального планування роботи команди ІТ проєкту на основі гнучкої методології.

1 Визначено ризики зростання рівня дефектів виконання робіт проєкту. Системно проаналізовано вплив основних чинників (кваліфікація розробників, неточність оцінки задач спринту, неточний опис задач, зменшення ресурсів команди, внутрішня культура команди) на підвищення імовірності дефектів та зривів спринту. Це дало змогу закласти підґрунтя для їх цілеспрямованого контролю та усунення.

2 Запропоновано модель інтелектуального планування роботи команди ІТ проєкту, яка сприяє більш швидкому і точному плануванню спринтів за рахунок аналізу ретроспективних даних та використання моделі штучного інтелекту для надання рекомендацій.

Створена модель інтегрує історичні метрики, бізнес-цінність задач і поточні ризики, а далі, за допомогою алгоритмів ML/LLM, формує рекомендації щодо оптимального наповнення спринту. Таким чином менеджер отримує аналітично обґрунтований план, що мінімізує перевантаження та підвищує передбачуваність термінів.

3 Запропоновано метод покращення текстових описів задач проєкту, який дозволяє підвищити точність сприйняття задач та знизити ризики їх невиконання за рахунок використання моделей машинного навчання та великих мовних моделей для оцінки та вдосконалення тексту.

Метод використовує класифікатор на базі BERT і правила LLM-переформулювання, щоб автоматично оцінити ясність, повноту й однозначність кожної задачі. За потреби система пропонує відредагований варіант, підвищуючи зрозумілість вимог для розробників і тим самим скорочуючи кількість дефектів, що виникають через незрозумілі описи.

4 Удосконалено модель розподілу задач спринту між виконавцями на основі постановки задачі визначення стабільних зіставлень що, на відміну від існуючих підходів, враховує специфіку гнучкої методології управління проєктами та дозволяє оптимізувати роботу команди.

Запропонована постановка як задача стабільних зіставлень (SOSM/EADAM) враховує не лише кваліфікацію й завантаженість, а й особисті уподобання розробників і пріоритетність задач. Це забезпечує стійкий розподіл, де не виникають блокуючі пари й мінімізується потреба у переназначеннях під час спринту.

5 Набув подальшого розвитку метод планування роботи команди ІТ проєкту при використанні гнучкої методології управління проєктами за рахунок комплексного підходу до визначення рекомендацій щодо розподілу, вибору та оцінок задач на основі інтелектуальної обробки ретроспективних та текстових даних та контексту проєкту, що, на відміну від існуючих підходів, дозволяє врахувати невизначеність та ризики, притаманні проєктам з гнучкою методологією та знизити рівень дефектів.

Метод поєднує ретроспективні дані, оцінки якості описів, ризикові метрики та контекст проєкту в єдину систему прийняття рішень. Такий комплексний підхід дозволяє оперативно перебудовувати план у відповідь на зміни вимог чи ресурсів, зберігаючи баланс між швидкістю, якістю та ризиками.

6 Розроблено інформаційну технологію підтримки планування роботи команди ІТ проєкту, яка дає змогу оцінювати та знижувати негативний вплив чинників, що загрожують строкам та якості проєкту шляхом застосування методів обробки природної мови на базі LLM, а також алгоритми стабільного зіставлення, що дозволяє надати рекомендації команді ІТ-проєкту.

Розроблена ІТ-платформа інтегрується з JIRA/YouTrack, отримує дані беклогу, пропускає їх через модулі NLP та алгоритми зіставлення й повертає рекомендації щодо уточнення описів та розподілу задач. Таким чином вона як автоматично підказує, як знизити ризики та підвищити продуктивність.

7 Досягнута мета дослідження – зменшення рівня дефектів виконання робіт проєкту шляхом використання моделей та методів інтелектуального планування роботи команди ІТ проєкту на основі гнучкої методології.

Комплексне впровадження запропонованих моделей і методів у пілотних командах показало статистично значуще зменшення дефектів та переробок, скорочення часу планування спринту на 30-40 % і підвищення задоволеності розробників призначеними задачами. Отже, поставлена мета – мінімізувати дефекти через інтелектуальне планування в Agile-середовищі – реалізована на практиці

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1 Aalst, van der, W. M. P., Adriansyah, A., Alves De Medeiros (2012). Process mining manifesto. In F. Daniel, K. Barkaoui, & S. Dustdar (Eds.), Business Process Management Workshops (BPM 2011 International Workshops, Clermont-Ferrand, France, August 29, 2011, Revised Selected Papers, Part I) (pp. 169-194). Vol. 99. Berlin: Springer.
- 2 Abdu, A., Zhai, Z., Algabri, R., Abdo, H., Hamad, K., Al-antari, M. Deep learning-based software defect prediction via semantic key features of source code - systematic survey. *Mathematics*. 2022. Vol. 10. No. 17. P. 3120.
- 3 Adel R, Harb H., Elshenawy A. A Multi-agent Reinforcement Learning Risk Management Model for Distributed Agile Software Projects. *Tenth International Conference on Intelligent Computing and Information Systems (ICICIS)*. 2021. P. 512-520.
- 4 Agathokleous, E., Rillig, M., Peñuelas, J., Yu, Z. One hundred important questions facing plant science derived using a large language model. *Trends in Plant Science*. 2023. Vol. 10.3120.
- 5 Agile software development manifesto [Electronic resource] <https://agilemanifesto.org/iso/uk/manifesto.html> Дата звернення (10.11.2024)
- 6 Alhazmi A., Huang S. A decision support system for sprint planning in scrum practice. *Proc. of SoutheastCon*. 2018. P. 1-9.
- 7 Almeida L. H., Albuquerque A. B., Pinheiro P. R. A multi-criteria model for planning and fine-tuning distributed Scrum projects. *Proc. of 2011 IEEE Sixth International Conference on Global Software Engineering*. 2011. P. 75-83.
- 8 Amna A.R., Poels, G. Ambiguity in user stories: A systematic literature review. *Information and Software Technology*. 2022. Vol. 145. P. 1–12.
- 9 Arık O. A. Project scheduling and staff allocation problem with time-dependent learning effect: a mixed integer non-linear programming approach. *Eskişehir Technical University Journal of Science and Technology A - Applied Sciences and Engineering*. 2019. Vol. 20(3). P. 204-215.

- 10 Asana Available.URL: <https://asana.com> (Дата звернення: 10.11.2024).
- 11 Atlassian Trello. URL: <https://trello.com> (Дата звернення: 15.11.2024).
- 12 Aziz H., Brandt F., Brill M. The computational complexity of random serial dictatorship. *Economics Letters*. 2013. № 121(3). - P. 341-345.
- 13 Barcus A., Montibeller G. Supporting the allocation of software development work in distributed teams with multicriteria decision analysis. *Omega*. 2008. V. 36. № 3. P. 464-475.
- 14 Berhanu, F., Alemneh, E. Classification and Prioritization of Requirements Smells Using Machine Learning Techniques. *International Conference on Information and Communication Technology for Development for Africa (ICT4DA)*. IEEE, 2023. P. 49-54.
- 15 Bird, S., Klein, E., Loper, E. Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit. *O'Reilly Media*. Beijing, 2009. 504 p.
- 16 Brandt F., Greger M., Romen R. Towards a Characterization of Random Serial Dictatorship. *arXi:2303.11976*. 2023. 27 p.
- 17 Briciu, A., Czibula, G., Lupea. M. A study on the relevance of semantic features extracted using BERT-based language models for enhancing the performance of software defect classifiers. *Procedia Computer Science*. 2023. Vol. 225. P. 1601-1610.
- 18 Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., & at al. Language Models are Few-Shot Learners. Arxiv. *Computation and Language*. 2020. P. 1-75.
- 19 Bushuyev, S., Bushuyeva, N., Bushuieva, V., Bushuiev, D. Integrated Intelligence Model for Managing IT Projects. *CEUR Workshop Proceedings*. 2022. V.3382. P.1-11.
- 20 Bushuyev, S., Bushuyeva, N., Bushuieva, V., Bushuiev, D., Ivko, A. Competencies Model for the Socialization of Artificial Intelligence Systems. *CEUR Workshop Proceedings*. 2023. Vol.3641. P. 83–95.

- 21 Butt S.A., Khalid A., Ercan T., etc. A software-based cost estimation technique in scrum using a developer's expertise. *Advances in Engineering Software*. 2022. Vol. 171, 103159. P. 1-10.
- 22 Casamayor, A., Godoy, D., Campo, M. Identification of non-functional requirements in textual specifications: A semi-supervised learning approach. *Information and Software Technology*. 2010. Vol. 52. Issue 4. P. 436–445.
- 23 Casillo, F., Deufemia, V., Gravino, C. Detecting privacy requirements from User Stories with NLP transfer learning models. *Information and Software Technology*. 2022. Vol. 146. P. 106853.
- 24 Chawla, P., Hazarika, S., Shen, H.-W. Token-wise sentiment decomposition for convnet: Visualizing a sentiment classifier. *Visual Informatics*. 2020. Vol. 4 Issue 2. P. 132–141.
- 25 Chen S., Chen Y., Hsu C. L. New axioms for top trading cycles. *Bulletin of Economic Research*. 2023. P. 1-15.
- 26 Chen W. N., Zhang J., Member S. Ant colony optimization for software project scheduling and staffing with an event-based scheduler. *IEEE Transactions on Software Engineering*. 2013. Vol.39. № 1. P. 1-17.
- 27 Cherednichenko, O., Ivashchenko, O., Lincényi, M., Kováč, M. Information technology for intellectual analysis of item descriptions in e-commerce. *Entrepreneurship and Sustainability Issues*. 2023. Vol. 11. N. 1. P. 178-190.
- 28 Cleland, D. L., Gareis, R. Learn and apply successful international project management techniques. *McGraw Hill Professional*. 2010. 575 p.
- 29 Covey, S. R. The 7 Habits of Highly Effective People: Powerful Lessons in Personal Change. *Simon and Schuster*. 2004. 372 p.
- 30 Dalpiaz, F., et al. Detecting terminological ambiguity in user stories: Tool and experimentation, *Information and Software Technology*. 2019. Vol. 110. P. 3–16.
- 31 Dalpiaz, F., Gieske, P., Sturm, A. On deriving conceptual models from user requirements: An empirical study. *Information and Software Technology*. 2021. Vol. 131, 106484. P. 1–13.

- 32 Daming Li, Lianbing Deng, Xi Zeng, Zhiming Cai. Dynamic simulation modelling of software requirements change management system. *Microprocessors and Microsystems*. 2021. Vol. 83. P. 1-6.
- 33 Devlin, J., Chang, M., Lee, K., Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2019. Vol. 1. P. 4171-4186.
- 34 Diebold F., Aziz H., Bichler M., et al. Course Allocation via Stable Matching. *Bus Inf Syst Eng*. 2014. V. 6. P. 97-110.
- 35 Diebold F., Bichler M. Matching with indifferences: A comparison of algorithms in the context of course allocation. *European Journal of Operational Research*. 2017. V. 260. № 1. P. 268-282.
- 36 Edyta Kulej-Dudek. Project management in a partner network in the context of teamwork and project team selection. *Procedia Computer Science*. 2023. Volume 225. P. 4850-4859.
- 37 El Migid M.A., Cail D., Niven T., Voll J, Madampe K., Grundy J., Hoda R. Emotimonitor. A Trello power-up to capture and monitor emotions of Agile teams. *Journal of Systems and Software*. *arXiv:2105.01952*. 2021. Vol. 186. 111206.
- 38 Eric Weflen, Cameron A. MacKenzie, Iris V.Rivero. An influence diagram approach to automating lead time estimation in Agile Kanban project management. *Expert Systems with Applications*. 2022. Vol. 187. 115866. P. 1-12.
- 39 Fan X., Yen J. Modeling and simulating human teamwork behaviors using intelligent agents. *Physics of Life Reviews*. 2004. Vol. 1, Iss. 3. P. 173-201.
- 40 Femmer, H., Fernández, D., Wagner, S., Eder, S. Rapid quality assurance with Requirements Smells. *Journal of Systems and Software*. 2017. Vol. 123. P. 190–213.
- 41 Fernandez J., Basavaraju M. Task allocation model in globally distributed software projects using genetic algorithms. *Proc. of the 2012 IEEE Seventh International Conference on Global Software Engineering*. 2012. 181 p.

42 Frattini, J. Identifying Relevant Factors of Requirements Quality: An Industrial Case Study. *Requirements Engineering: Foundation for Software Quality*. 2024. Vol. 14588. P. 20-36.

43 Gladka M, Kravchenko O., Hladkyi Y., Borashova S. Qualification and appointment of staff for project work in implementing IT systems under conditions of uncertainty: *Proc. of 2021 IEEE International Conference on Smart Information Systems and Technologies*. Astana IT University. Nur-Sultan. Kazakhstan. 2021. P. 28-30.

44 Gouthaman P, Suresh Sankaranarayanan. Prediction of Risk Percentage in Software Projects by Training Machine Learning Classifiers. *Computers & Electrical Engineering*. 2021. Vol. 94. 107362. P. 1-9.

45 Grinchenko M., Rohovyi M. Intelligent decision support system in the face of uncertainty and risks in project management. *Інформаційні системи та технології: матеріали 13-ї Міжнародної науково-технічної конференції, м. Харків, 26-28 лист. 2024 р.* С.86-90.

46 Grinchenko M., Rohovyi M. System of recommendations for the it project team in the face of risks. *XVII International scientific and practical conference "Scientific trends in the development of education in universities"*. Athens, Greece. December 24-27. 2024. P.237-240.

47 Hai Zhuge. Workflow- and agent-based cognitive flow management for distributed team Cooperation. *Information & Management*. 2003. Vol. 40. Issue 5. P. 419-429.

48 Han, WM. Validating differential relationships between risk categories and project performance as perceived by managers. *Empir Software Eng*. 2014. 19. P. 1956–1966.

49 Henning Wachsmuth. Text Analysis Pipelines Towards Ad-hoc Large-Scale Text Mining. *Springer International Publishing Switzerland*. 2015. P. 302.

50 Heubeck T., Storz A.S., Meckl R. Success factors of global virtual teamwork. *A social capital perspective. Digital Business*. 2024. Vol. 4, Iss. 2, 100081. P.1-15.

- 51 Hoda R., Murugesan L. Multi-level agile project management challenges: A self-organizing team perspective. *Journal of Systems and Software*. 2016. Vol. 117. P. 245-257.
- 52 Hong M., Park J. Core and top trading cycles in a market with indivisible goods and externalities. *Journal of Mathematical Economics*. 2022. V. 100. P. 102627.
- 53 Hsu S., Weng K., Cui Q., Rand W. Understanding the complexity of project team member selection through agent-based modeling. *International Journal of Project Management*. 2016. Vol.e 34. Iss. 1. P.82-93
- 54 International Organization for Standardization. Information technology — Modeling Languages — Part 1: Syntax and Semantics for IDEF0 (ISO Standard No. 31320-1).
- 55 Ishizuka, R., Washizaki, H., Tsuda, N., Fukazawa, Y., Oujii, S., Saito, S., Iimura, Y. Categorization and Visualization of Issue Tickets to Support Understanding of Implemented Features in Software Development Projects. *Applied Sciences*. 2022. №. 12. Iss. 7. P. 3222.
- 56 ISO/IEC/IEEE 31320-1:2012 Інформаційні технології — Мови моделювання — Частина 1: Синтаксис і семантика для IDEF0 [ISO/IEC/IEEE 31320-1:2012 | IEC](https://www.iso.org/standard/68411.html)
- 57 James J. Jiang, Gary Klein, Jamie Y.T. Chang. Teamwork behaviors in implementing enterprise systems with multiple projects: Results from Chinese firms. *Journal of Systems and Software*. 2019. Vol. 157. 110392.
- 58 Javed U., Hameed I.A., A Review of Content-Based and Context-Based Recommendation Systems. *International Journal of Emerging Technologies in Learning (iJET)* 2021. Vol. 16. N 03. P. 274-306.
- 59 Jiao Z, et al. Understanding software developers' cognition in agile requirements engineering. *Science of Computer Programming*. 2019. Vol. 178. P. 1–19.
- 60 Jiao Z., Shen Z. School choice with priority-based affirmative action: A responsive solution. *Journal of Mathematical Economics*. 2021. Vol. 92. P. 1-9.

- 61 Jiao Z., Tian G. Two further impossibility results on responsive affirmative action in school choice. *Economics Letters*. 2018. Vol. 166. P. 60-62.
- 62 Jira Software. [The IT industry]. URL: <https://www.atlassian.com/software/jira>. (Дата звернення 10.05.2024).
- 63 Jolliffe, I. T., Cadima, J. Steinerberger, Stefan: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*. 2016. Vol. 374. Iss. 2065. P. 1-16.
- 64 Kalyan, K. S. A survey of GPT-3 family large language models including ChatGPT and GPT-4. *Natural Language Processing Journal*. 2024. Vol. 6. P. 100048.
- 65 Kiva S., Heng S, Wautelet Y., Poelmans S., Kolp M. Using an ontology for systematic practice adoption in agile methods. *Expert system and practitioners-based validation*. 2022. Vol. 195, 116520. P. 1-9.
- 66 Kolychev V., Bezmenskii N.. Estimation of the tasks complexity for large-scale high-tech projects using Agile methodologies. *Procedia Computer Science*. 2018. Vol. 145. P. 266-274.
- 67 Kononenko, I., Sushko, H., Babayev, I., & Abdullayev, R. Solving the problem of ranking applicants for a project team with fuzzy assessment of competencies and requirements. *Radioelectronic and Computer Systems*. 2024. Vol. 2. P. 230-243.
- 68 Košinár M., Štrba R.. Simulations of Agile Software Processes for Healthcare Information Systems Development Based on Machine Learning Methods. *IFAC Proceedings Volumes*. 2013. Vol. 46. Iss. 28. P. 175-180.
- 69 Leelaprute, P., Amasaki, S. A comparative study on vectorization methods for non-functional requirements classification, *Information and Software Technology*. 2022. Vol. 150. P. 106991.
- 70 Li Y., Bubeck S., Eldan R., Del Giorno A., Gunasekar, S., Lee, Y. Textbooks Are All You Need II: phi-1.5 technical report. *Microsoft Research*. 2023, P. 1-16.

- 71 Li, Z., Wang, X., Jing, W., Wu, J., Zhang, Z., Liu, Z., Sun, M., Hui, Z., Liu, S. A Unified Understanding of Deep NLP Models for Text Classification URL: <https://arxiv.org/abs/2206.09355> (Дата звернення 08.05.2024).
- 72 Linderman, G., Steinerberger, S. Clustering with t-SNE, Provably, *SIAM Journal on Mathematics of Data Science*. 2019. Vol. 1. Iss. 2. P. 313-332.
- 73 Lindsjörn Y., Sjöberg D., Dingsøy T., Bergersen G.R. Teamwork quality and project success in software development: *A survey of agile development teams Journal of Systems and Software*. 2016. Vol. 122. P. 274-286.
- 74 Marinho, M., Sampaio, S., Moura, H. Managing uncertainty in software projects. *Innovations Syst Softw Eng*. 2018. Vol.14. P. 157–181.
- 75 McHugh Mary L. Interrater reliability: the kappa statistic. *Biochemia Medica*. 2013. Vol. 22. Iss. 3. P.276-282
- 76 Menezes J., Gusmão C., Moura H. Risk factors in software development projects: a systematic literature review. *Software Qual Journal*. 2019. Vol. 27. P. 1149–1174.
- 77 Microsoft Teams. [Electronic resource] – 2024. URL: <https://www.microsoft.com/> (Дата звернення 15.11.2024).
- 78 Moharil, A., Sharma, A. TABASCO: A transformer based contextualization toolkit. *Science of Computer Programming*. 2023. Vol. 230. P. 102994.
- 79 Moskalenko V., Fonta N., Grinchenko M. The Method of Forming a Dynamic Projects Portfolio of IT Companies. *The 1st International Workshop IT Project Management (ITPM 2020)*: Vol. 1. Lviv, Ukraine, February 18-20, 2020. CEUR Workshop Proceedings (CEUR-WS.org), Vol. 2565. online. P. 152-161.
- 80 Murtazina, M., Avdeenko, T. An Ontology-based Approach to Support for Requirements Traceability in Agile Development. *Procedia Computer Science*. 2019. Vol. 150. P. 628–635.
- 81 Nakra V. Enhancing Software Project Management and Task Allocation with AI and Machine Learning. *International Journal on Recent and Innovative Trends in Computing and Communication*. 2023. № 11. P. 1171-1178.

- 82 Nechvoloda L. V., Shevchenko N. Yu. Increasing the efficiency of IT project management with the application of complex methodology distribution of performers for work. *Taurida Scientific Herald. Series: Technical Sciences*. 2023. No. 2. P. 98-105.
- 83 Nielsen, F. Hierarchical Clustering. *Introduction to HPC with MPI for Data Science, Springer*. 2016. ISSN 1863-7310. P. 221-239.
- 84 Onur Kesten, School Choice with Consent. *The Quarterly Journal of Economics*. 2010. Vol. 125, Iss. 3. P. 1297–1348,
- 85 Open-source FastText URL: <https://fasttext.cc/> (Дата звернення: 08.11.2024)
- 86 Orłowski C., Bach-Dąbrowska I., Kapłański P., Wysocki W. Hybrid Fuzzy-ontological Project Framework of a Team Work Simulation System. *Procedia Computer Science*. 2014. Vol. 35. 1175-118.
- 87 Pattara Leelaprute, Sousuke Amasaki. A comparative study on vectorization methods for non-functional requirements classification. *Information and Software Technology*. 2022. Vol. 150. P. 10699.
- 88 Peggy, G., Strode, D., Sharp, H, Barroca, L. An onboarding model for integrating newcomers into agile project teams. *Information and Software Technology*. 2022. Vol. 143. P. 106792.
- 89 Pennington, J., Socher, R., Manning, C. GloVe: Global Vectors for Word Representation. *In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 2014. P.1532–1543.
- 90 Pratama I. N., Dachyar M., Pratama N. R. Optimization of Resource Allocation and Task Allocation with Project Management Information Systems in Information Technology Companies. *TEM Journal*. 2023. 12. № 3. P. 1814-1824.
- 91 Ramesh M.R.R., Reddy C.S. Metrics for software requirements specification quality quantification. *Computers & Electrical Engineering*. 2021. Vol. 96. Part A. P. 3–11.

- 92 Ramírez-Mora, S., Oktaba, H., Gómez-Adorno, H. Descriptions of issues and comments for predicting issue success in software projects. *Journal of Systems and Software*. 2020. Vol. 168. P. 1–19.
- 93 Rohovyi M., Grinchenko M. A model for identifying project sprint tasks based on their description. *Innovative technologies and scientific solutions for industries*. 2023. №4 (26). P. 33-44.
- 94 Rohovyi M., Grinchenko M. Comparative Analysis of stable matching algorithms for intelligent work planning of IT teams. *Біоніка інтелекту*. 2024. №2(101). С. 56–63. (Б)
- 95 Rohovyi M., Grinchenko M. Project team management model under risk conditions. *Вісник Національного технічного університету «ХПІ». Серія: Стратегічне управління, управління портфелями, програмами та проектами*. 2023. № 1 (7). С. 3-11.
- 96 Rohovyi M., Grinchenko M. Towards the improvement of project team performance based on large language models. *Radioelectronic and computer systems*. 2024. №. 4(112) P. 229-247.
- 97 Schmidt J. Mitigating risk of failure in information technology projects: Causes and mechanisms. *Project Leadership and Society*. 2023. Vol. 4. 100097. P. 1-13.
- 98 Shastri Y., Hoda R., Amor R. The role of the project manager in agile software development projects. *Journal of Systems and Software*. 2021. Vol. 173. P. 110871.
- 99 Simão Filho M., Pinheiro P. R., Albuquerque A. B. Analysis of task allocation in distributed software development through a hybrid methodology of verbal decision analysis. *Journal of Software: Evolution and Process*. 2017. V. 29. № 7. P. 1-18.
- 100 Singh M., Chauhan N., Popli R. A Review on Quantitative Task Allocation in Agile Software Development. *Proc. of International Conference on Sustainable Computing in Science, Technology and Management (SUSCOM)*. Amity University Rajasthan, Jaipur. 2019. P. 268-273.

101 Sonbol, R., Rebdawi, G., Ghneim, N. Learning software requirements syntax: An unsupervised approach to recognize templates. *Knowledge-Based Systems*. 2022. Vol. 248. P. 108933.

102 Strielkina, A., Tetskyi, A., Krasilshchikova, V. Risk and uncertainty assessment in software project management: integrating decision trees and monte carlo modeling. *Radioelectronic and Computer Systems*. 2023. №. 3(107). P. 217-225.

103 Sundararajan. S., Marath, B., K. Vijayaraghavan, P. Variation of risk profile across software life cycle in IS outsourcing. *Software Qual Journal*. 2019. № 27 P. 1563–1582.

104 Sutton R., Barto A. Reinforcement learning: an introduction. Second edition. *Cambridge, MA: The MIT Press*. 2018. P. 525.

105 Taye, G.D., Feleke, Y.A. Prediction of failures in the project management knowledge areas using a machine learning approach for software companies. *SN Appl.* 2022. Sci. 4, 165. P. 2523-3971.

106 TF-IDF (Term Frequency-Inverse Document Frequency). URL: <https://www.learndatasci.com/glossary/tf-idf-term-frequency-inverse-document-frequency/i> (Дата зірнення 08.11.2023).

107 The Public Jira Dataset. URL: <https://zenodo.org/records/5901804>. (Дата звернення 15.05.2024).

108 The Scrum Guide. *The Definitive Guide to Scrum: The Rules of the Game*. URL: <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf#zoom=100> (Дата звернення: 12.12.2024).

109 Todorović M. L., Petrović D. Č., Bushuyev S. D. etc. Project success analysis framework. *A knowledge-based approach in project management. International Journal of Project Management*. 2015. Vol. 33, Issue 4. P. 772-783.

110 Tuan N. N., Hang H. Q. Iteration Scheduling Using Bayesian Networks in Agile Software Development. *Proc. of Vietnamese Academic Workshop*. 2019. P. 300-308.

111 Urbieto, M., etc., The impact of using a domain language for an agile requirements management, *Information and Software Technology*. 2020. Vol. 145, P. 1–16.

112 Uzzafer M. A simulation model for strategic management process of software projects. *Journal of Systems and Software*. 2013. Vol. 86. Iss. 1. P. 21-37.

113 Vasyluk A., Basyuk T. Algebras of algorithms for modeling the distribution of resources in IT projects. *SISN*. 2023. Vol. 13. P. 156-166.

114 Vishnubhotla S.D., Mendes E. Exploring the relation between personality traits and agile team climate: Aggregating results from a twice replicated study in a telecom company. *Journal of Systems and Software*. 2024. Vol. 210, 111937. P.1-27

115 Wahba Y., Madhavji N., Steinbacher, J. A Hybrid Machine Learning Model for Efficient Classification of IT Support Tickets in The Presence of Class Overlap", *Proceedings of the 32nd Annual International Conference on Computer Science and Software Engineering*. 2020. P. 151-156.

116 Wang Z. P-value based task allocation in a scrum team: a multi-agent simulation. *Proc. of the IEEE 10th International Conference on Software Engineering and Service Science (ICSESS)*. 2019. P. 1-4.

117 Word2vec. URL: <https://www.tensorflow.org/text/tutorials/word2vec> (Дата звернення 08.11.2023).

118 Wysocki W. A hybrid software processes management support model. *Procedia Computer Science*. 2020. Vol. 176. P. 2312-2321.

119 Wysocki W, Orłowski C. A multi-agent model for planning hybrid software processes. *Procedia Computer Science*. 2019. Vol. 159. P. 1688-1697.

120 Yagnesh G., Vishal P. A Survey on Various Techniques of Recommendation System in Web Mining. *International Journal of Engineering Development and Research*. 2015. Vol. 3. P. 696-700.

121 Yin X., Huang J., He W., Guo W., Yu H., Cui L. Group task allocation approach for heterogeneous software crowdsourcing tasks. *Peer-to-Peer Networking and Applications*. 2020. № 3. P. 1736-1747.

122 Yu D., Wang Y., Zhou Z. Software crowdsourcing task allocation algorithm based on dynamic utility. *IEEE Access*. 2019. Vol. 7. P. 33094-33106.

123 Yu D., Zhou Z., Wang Y. Crowdsourcing software task assignment method for collaborative development. *IEEE Access*. 2019. Vol. 7. P. 35743-35754.

124 Гринченко М., Роговий М. Гринченко Е. Підхід щодо класифікації текстового опису задач спринту проєкту. *Управління проєктами у розвитку суспільства: тези доповідей XX міжнародної конференції «Управління проєктами післявоєнної розбудови України»* (12 травня 2023 р.). Київ. 2023. – С.86-90.

125 Гринченко М.А., Роговий М.А., Грінченко Є.М. Розробка інформаційної технології інтелектуального планування роботи команди ІТ проєкту на основі гнучкої методології. *Вісник Національного технічного університету «ХПІ»*. Серія: Стратегічне управління, управління портфелями, програмами та проєктами. Харків. 2024. № 2 (8). С.

126 Гринченко М.А., Роговий М.А. Ідентифікація опису задач спринту проєкту: *Інтегроване стратегічне управління, управління портфелями, програмами, проєктами: тези доповідей XIV міжнародної науково-практичної конференції «Інтегроване стратегічне управління, управління портфелями, програмами, проєктами»*, (Харків: НТУ «ХПІ») 13-14 лютого 2024 р. / за ред. проф. Кононенка І. В. С. 33.

127 Свінцицька О. М., Сугоняк І. І., & Пулеко І. В. (2021). Оптимізація бізнес-процесу на основі інформаційної технології в комунікаціях ІТ-проєктів. *Технічна інженерія*. 2021. №1(87). С. 59–65.

128 Сушко Г. В., Баландін Д. С. Розробка інформаційної технології для формування команди ІТ-проєкту. *Вісник Національного технічного університету «ХПІ»*. Серія: Стратегічне управління, управління портфелями, програмами та проєктами. 2023. №1(7). С. 76-80.

ДОДАТОК А

Документи впровадження основних результатів дисертаційної роботи



ЗАТВЕРДЖУЮ
Проректор з наукової роботи
Національного технічного
університету «Харківський
політехнічний інститут»
Андрій МАРЧЕНКО
«15» «11» 2024 р.

АКТ

про використання результатів дисертаційної роботи аспіранта кафедри управління проектами в інформаційних технологіях РОГОВОГО Микити Антоновича в науково-дослідній роботі, виконаній відповідно до тематичного плану Національного технічного університету «ХПІ»

Ми, що нижче підписалися, завідувачка кафедри управління проектами в інформаційних технологіях ГРИНЧЕНКО Марина Анатоліївна, доцент кафедри управління проектами в інформаційних технологіях ЛОБАЧ Олена Володимирівна склали акт про те, що результати дисертаційної роботи РОГОВОГО Микити Антоновича впроваджені в науково-дослідну роботу «Моделі та методи управління проектами розвитку організаційних систем» (ДР№ 0124U001735).

Здобувач брав участь у виконанні робіт за вказаною темою в якості виконавця. В межах виконання робіт здобувачем розроблені: 1) модель оцінки текстових описів задач спринту; 2) модель визначення розподілу задач у команді проекту; 3) інтелектуальна система підтримки прийняття рішень в умовах невизначеності та ризиків при управлінні проектом.

Завідувачка кафедри управління
проектами в інформаційних
технологіях НТУ «ХПІ»

Марина ГРИНЧЕНКО

Доцент кафедри управління
проектами в інформаційних
технологіях НТУ «ХПІ»

Олена ЛОБАЧ



ЗАТВЕРДЖУЮ
Проректор
Національного технічного
університету «Харківський
політехнічний інститут»
Руслан МИГУЩЕНКО
“15” 11 2024 р.

АКТ

про використання результатів дисертаційної роботи РОГОВОГО Микити Антоновича в навчальному процесі кафедри управління проєктами в інформаційних технологіях Національного технічного університету «ХПІ»

Ми, що нижче підписалися, завідувачка кафедри управління проєктами в інформаційних технологіях ГРИНЧЕНКО Марина Анатоліївна, доцент кафедри управління проєктами в інформаційних технологіях ЛОБАЧ Олена Володимирівна, доцент кафедри управління проєктами в інформаційних технологіях ШУБА Ірина Володимирівна склали акт про те, що результати дисертаційної роботи РОГОВОГО Микити Антоновича впроваджені в навчальний процес на кафедрі управління проєктами в інформаційних технологіях.

Модель оцінки текстових описів задач спринту впроваджено в дисципліну «Методології управління проєктами у сфері інформаційних технологій».

Модель визначення розподілу задач у команді проєкту впроваджено в дисципліну «Управління ІТ-проєктами».

Інтелектуальну систему рекомендацій впроваджено в дисципліну «Управління проєктами розробки та реінжинірингу інформаційних технологій».

Завідувачка кафедри управління
проєктами в інформаційних
технологіях НТУ «ХПІ»

Марина ГРИНЧЕНКО

Доцент кафедри управління
проєктами в інформаційних
технологіях НТУ «ХПІ»

Олена ЛОБАЧ

Доцент кафедри управління
проєктами в інформаційних
технологіях НТУ «ХПІ»

Ірина ШУБА



Товариство з обмеженою відповідальністю
«ДОМІНЕТ»
Код ЄДРПОУ 42543669
08200 Київська область, м. Ірпінь, вул. Білокур 9, кв. 54
Телефон : (098) 640-40-26 E-mail: info@dominet.com.ua

ЗАТВЕРДЖУЮ
Директор ТОВ «Домінет»



МОЗОЛЮК Володимир

11 2024 р.

АКТ

про впровадження результатів дисертаційної роботи аспіранта кафедри
Управління проектами в інформаційних технологіях Національного
технічного університету «Харківський політехнічний інститут» Микити
РОГОВОГО

Цим Актом ТОВ «Домінет» підтверджує, що в результаті проведеного з 11 листопада 2024 року по 22 листопада 2024 року розгляду результатів дисертаційного дослідження РОГОВОГО Микити було прийнято рішення про їх впровадження в роботу підприємства.

В результаті аналізу дисертаційної роботи було з'ясовано, що запропонована в ній інтелектуальна система забезпечує інтеграцію з сучасними інструментами відстеження завдань (наприклад JIRA) і дозволяє автоматично аналізувати завдання, призначені для спринтів, оцінювати поточну якість опису завдань, а також забезпечує більшу прозорість у створенні завдань та розподілі ресурсів.

Впровадження зазначених результатів забезпечило:

- оптимізацію процесів управління завданнями та підвищення продуктивності команди за рахунок інтеграції системи з JIRA, включаючи налаштування автоматичного аналізу та рекомендацій;
- скорочення часу на постановку задач та їх розподіл на 30-40%.
- підвищення точності відповідності задачі виконавцю, що зменшує кількість змін на 20%;
- покращення комунікації в команді завдяки яснішим формулюванням завдань.

Висновок.

Результати дисертаційної роботи вважаються успішно впровадженими завдяки:

- збільшення конкурентоспроможності компанії через ефективніше виконання проєктів;
- підвищення задоволеності клієнтів завдяки дотриманню термінів і якості;
- скорочення витрат на управління проєктами.

Система введена в експлуатацію та схвалена для використання в подальшій щоденній діяльності ТОВ «ДОМІНЕТ».

Члени комісії:

Директор

(посада)

Головний інженер

(посада)



МОЗОЛЮК В.А.

(ПІБ)

БАБЕНКО А.А.

(ПІБ)

ДОДАТОК Б

Список публікацій здобувача

Публікації здобувача за темою дисертації, в яких опубліковані основні наукові результати:

1 Rohovyi M., Grinchenko M. Towards the improvement of project team performance based on large language models. *Radioelectronic and computer systems*, 2024. №. 4 (112). P. 229–247. (Scopus, Україна, А). URL: <https://doi.org/10.32620/reks.2024.4.19>

2 Grinchenko M., Rohovyi M. A model for identifying project sprint tasks based on their description. *Innovative technologies and scientific solutions for industries*. Kharkiv, 2023. №4 (26). P. 33-44. (Б). URL: <https://doi.org/10.30837/ITSSI.2023.26.033>

3 Rohovyi M., Grinchenko M. Project team management model under risk conditions. *Вісник Національного технічного університету «ХПІ». Серія: Стратегічне управління, управління портфелями, програмами та проектами*. Харків, 2023. № 1 (7). С. 3–11. (Б). URL: <https://doi.org/10.20998/2413-3000.2023.7.1>

4 Rohovyi M., Grinchenko M. Comparative Analysis of stable matching algorithms for intelligent work planning of IT teams. *Біоніка інтелекту*. Харків: ХНУРЕ, 2024. №2 (101). С. 56–63. (Б). [https://doi.org/10.30837/bi.2024.2\(101\).09](https://doi.org/10.30837/bi.2024.2(101).09)

5 Гринченко М.А., Роговий М.А. Розробка інформаційної технології інтелектуального планування роботи команди ІТ проекту на основі гнучкої методології. *Вісник Національного технічного університету «ХПІ». Серія: Стратегічне управління, управління портфелями, програмами та проектами*. Харків, 2024. № 2 (9). С. 9–15. (Б). DOI: <https://doi.org/10.20998/2413-3000.2024.9.2>

Публікації здобувача за темою дисертації, які засвідчують апробацію матеріалів дисертації:

6 Гринченко М.А., Роговий М.А. Підхід щодо класифікації текстового опису задач спринту проєкту. *Управління проектами у розвитку суспільства*: тези доповідей XX Міжнародної конференції, (м. Київ, 12 травня 2023 р.). – Київ: КНУБА, 2023. – С.86–90. URL:

https://www.researchgate.net/publication/371492759_Analysis_of_Aspects_of_Increasing_the_Efficiency_of_IT_Project_Management#fullTextFileContent

7 Гринченко М.А., Роговий М.А. Ідентифікація опису задач спринту проєкту. *Інтегроване стратегічне управління, управління портфелями, програмами, проєктами*: тези доповідей XIV Міжнародної науково-практичної конференції, (м. Харків, 13-14 лютого 2024 р.). – Харків: НТУ «ХПІ», 2024. – С. 33. <https://repository.kpi.kharkov.ua/server/api/core/bitstreams/91cf93ce-7f86-4fb3-a23d-671396e97461/content>

8 Grinchenko M., Rohovyi M. Intelligent decision support system in the face of uncertainty and risks in project management. *Інформаційні системи та технології*: тези доповідей 13-ї Міжнародної науково-технічної конференції, (м.Харків, 26 – 28 листопада 2024р). Харків, 2024. – Р. 237–240. https://ist-conf-nure.com.ua/IST-2024_part-1.pdf

9 Grinchenko M., Rohovyi M. System of recommendations for the it project team in the face of risks. *Scientific trends in the development of education in universities*: proceedings of the 17th International scientific and practical conference, (Athens, Greece, 24 – 27 Desember 2024). Athens, Greece:International Science Group, 2024. P. 237–240. DOI – [10.46299/ISG.2024.2.17](https://doi.org/10.46299/ISG.2024.2.17)