https://doi.org/10.31891/2307-5732-2026-363-20
УДК 004.4:004.051

МИХАЙЛЮК ІРИНА
Івано-Франківський національний технічний університету нафти і газу
https://orcid.org/0000-0002-6489-3982
e-mail: iryna.mykhailiuk@nung.edu.ua
БАНДУРА ВІКТОРІЯ
Івано-Франківський національний технічний університету нафти і газу
https://orcid.org/0000-0003-3143-0946
e-mail: viktoriia.bandura@nung.edu.ua
ЗІКРАТИЙ СЕРГІЙ
Івано-Франківський національний технічний університету нафти і газу
https://orcid.org/0000-0002-8004-3339
e-mail: s.zikratyi@nung.edu.ua

ОЦІНЮВАННЯ ВПЛИВУ КОМАНДНОЇ ВЗАЄМОДІЇ НА ЯКІСТЬ ПРОГРАМНИХ ПРОДУКТІВ У СТУДЕНТСЬКИХ ІТ-ПРОЄКТАХ

В роботі наведено результати оцінювання впливу характеристик командної взаємодії на якість програмних продуктів, розроблених у межах студентських ІТ-проєктів. Виокремлено основні показники якості програмного забезпечення та проаналізовано їхній зв'язок з організаційними й інженерними аспектами командної роботи. Запропоновано узагальнений підхід до оцінювання впливу командної взаємодії, який може бути використаний у навчальному процесі для підвищення об'єктивності оцінювання результатів проєктного навчання.

Ключові слова: *командна взаємодія, студентські ІТ-проєкти, якість програмного забезпечення, програмний продукт, проєктне навчання.*

MYKHAILIYK IRYNA, BANDURA VIKTORIIA, ZIKRATYI SERHII
Ivano-Frankivsk National Technical University of Oil and Gas

ASSESSING THE IMPACT OF TEAM INTERACTION ON THE QUALITY OF SOFTWARE PRODUCTS IN STUDENT IT PROJECTS

An effective development of software products in modern conditions is possible only under the condition of well-organized team interaction during the implementation of project tasks. In educational practice, particularly within student IT projects, the quality of software products often differs significantly even when participants have similar levels of theoretical training and technical knowledge. This difference is primarily caused by the specific features of team interaction, including communication practices, task coordination, role distribution, and the use of collaborative engineering tools during the software development process.

This paper presents the results of assessing the impact of team interaction characteristics on the quality of software products developed within student IT projects. Software quality is considered as a combination of functional completeness, reliability, code quality, and maintainability, which are the most relevant indicators for evaluating development results in an educational environment. Particular attention is paid to the influence of organizational and engineering aspects of teamwork, such as joint planning of tasks, coordination of activities, use of version control systems, code review procedures, and team-based testing practices.

The assessment is based on a logical sequence that connects a specific team interaction factor with its manifestation in the development process and the resulting impact on software quality indicators. Such an approach makes it possible to analyze the influence of teamwork on development outcomes without the use of complex experimental measurements. The evaluation can be carried out through the analysis of project artifacts, including source code repositories, change histories, documentation, and testing results, which ensures the practical applicability of the proposed approach in the educational process.

The analysis shows that collaborative engineering practices have the most direct impact on the reliability and quality of software products, while organizational factors create the necessary conditions for their systematic application. The obtained results can be used to improve the objectivity of evaluating student IT projects, to identify the causes of typical development errors, and to justify recommendations for improving the organization of team work in project-based learning.

Keywords: *team interaction, student IT projects, software quality, software development, project-based learning.*

Стаття надійшла до редакції / Received 16.01.2026
Прийнята до друку / Accepted 11.02.2026
Опубліковано / Published 26.03.2026

Постановка проблеми у загальному вигляді та її зв'язок із важливими науковими чи практичними завданнями

Сучасні програмні продукти, як правило, створюються командами розробників, у межах яких результат залежить не лише від рівня підготовки окремих фахівців, а й від того, наскільки узгодженою є їхня спільна робота. Саме тому в процесі підготовки фахівців напряму F2 дедалі більшого значення набуває формування навичок командної взаємодії під час виконання практикоорієнтованих завдань, зокрема студентських ІТ-проєктів [1, 2].

Досвід організації проєктного навчання [3, 4] свідчить, що навіть за подібного рівня теоретичних знань результати командної роботи студентів можуть суттєво відрізнятися. Одні команди демонструють стабільну роботу програмного продукту, чітку структуру коду та зрозумілу логіку реалізації, тоді як в інших

спостерігаються проблеми з узгодженістю рішень, появою помилок та ускладненням подальшого супроводу програмного забезпечення. Значною мірою такі відмінності зумовлені особливостями організації командної взаємодії, зокрема підходами до розподілу ролей, комунікації між учасниками та використанням інженерних практик спільної роботи з кодом.

У навчальному процесі оцінювання результатів студентських ІТ-проєктів зазвичай зосереджується на кінцевому програмному продукті або індивідуальному внеску окремих учасників команди. При цьому вплив характеру командної взаємодії на якісні характеристики програмного забезпечення часто залишається поза увагою. Такий підхід ускладнює об'єктивний аналіз причин успішності або недоліків проєктних рішень і обмежує можливості цілеспрямованого вдосконалення організації командної роботи [5].

Зазначені обставини вказують на потребу в оцінюванні впливу командної взаємодії на якість програмних продуктів у студентських ІТ-проєктах, що має практичне значення для підвищення ефективності проєктного навчання та наближення освітнього процесу до реальних умов професійної діяльності в галузі інженерії програмного забезпечення.

Аналіз досліджень та публікацій

У наукових дослідженнях [6] з інженерії програмного забезпечення командна взаємодія розглядається як важливий чинник, що впливає на продуктивність команд та якість програмних продуктів. Зокрема, аналізуються такі аспекти взаємодії, як розподіл ролей, інтенсивність комунікації та узгодженість технічних рішень, що безпосередньо впливають на стабільність процесу розроблення.

Значна кількість робіт присвячена впливу гнучких методологій розроблення програмного забезпечення на результати командної діяльності. Авторами [7, 8] показано, що застосування ітеративного планування, регулярних командних обговорень та ретроспектив сприяє ранньому виявленню проблем і зменшенню кількості дефектів у програмному коді.

Окремий напрям досліджень [9, 10] стосується інженерних практик спільної роботи з кодом, зокрема код-рев'ю, парного програмування та використання систем контролю версій. Результати таких досліджень підтверджують, що регулярне код-рев'ю позитивно впливає на якість коду, його структурованість і супроводжуваність, а також сприяє формуванню єдиних підходів до програмування в команді.

У контексті проєктного навчання наголошується на важливості створення чітких критеріїв оцінювання результатів студентських ІТ-проєктів. Автори [11, 12] зазначають, що відсутність системного підходу до оцінювання командної взаємодії ускладнює аналіз її впливу на якість програмних продуктів і обмежує можливості вдосконалення освітнього процесу.

Таким чином, аналіз сучасних публікацій засвідчує, що, попри значну кількість досліджень окремих аспектів командної роботи та якості програмного забезпечення, питання комплексного оцінювання впливу командної взаємодії на якість програмних продуктів у студентських ІТ-проєктах залишається недостатньо систематизованим, що обґрунтовує доцільність подальших досліджень у цьому напрямі.

Формулювання цілей статті

Метою статті є: оцінювання впливу характеристик командної взаємодії на якість програмних продуктів, розроблених у межах студентських ІТ-проєктів, та визначення найбільш значущих чинників командної роботи, що сприяють підвищенню якості результатів розроблення.

Для досягнення поставленої мети в роботі передбачається розв'язання таких **завдань:**

– виокремити основні показники якості програмних продуктів, релевантні для оцінювання результатів студентських ІТ-проєктів;

– проаналізувати характеристики командної взаємодії, зокрема організацію ролей, особливості комунікації, координацію спільної діяльності та застосування інженерних практик командної розробки;

– визначити взаємозв'язок між параметрами командної взаємодії та показниками якості програмних продуктів;

– обґрунтувати підходи до оцінювання впливу командної взаємодії на результати розроблення програмного забезпечення в освітньому середовищі;

– сформулювати практичні рекомендації щодо організації командної роботи у студентських ІТ-проєктах з метою підвищення якості програмних продуктів.

Виклад основного матеріалу

Критерії та показники якості програмних продуктів у студентських ІТ-проєктах

Оцінювання якості програмних продуктів, розроблених у межах студентських ІТ-проєктів, потребує визначення чітких і зрозумілих критеріїв, які дозволяють здійснювати порівняльний аналіз результатів командної діяльності. У сучасних підходах якість програмного забезпечення розглядається як сукупність взаємопов'язаних характеристик, що охоплюють як функціональні, так і нефункціональні аспекти програмного продукту [8]. Для умов освітнього процесу доцільно використовувати ті показники, які можуть бути об'єктивно оцінені та безпосередньо пов'язані з організацією командної взаємодії.

Одним із базових критеріїв є *функціональна повнота програмного продукту*, що відображає ступінь реалізації поставлених вимог і коректність виконання основних функцій системи. У студентських ІТ-проєктах цей показник оцінюється за відповідністю реалізованого функціоналу початковим вимогам та результатами функціонального тестування [7]. Недостатня узгодженість дій членів команди на етапах аналізу та проєктування часто призводить до пропуску окремих вимог або їх некоректної реалізації.

Важливим показником якості є *надійність програмного продукту,* яка характеризується стабільністю роботи системи та стійкістю до помилок. У межах студентських проєктів надійність може оцінюватися за кількістю виявлених дефектів, частотою збоїв під час тестування та результатами перевірки типових сценаріїв використання [8]. Дослідження підтверджують, що команди з налагодженими механізмами комунікації та взаємного контролю демонструють нижчий рівень критичних помилок у фінальних версіях програмних продуктів.

Окрему увагу слід приділити *якості програмного коду*, яка охоплює дотримання стандартів програмування, структурованість, зрозумілість і відповідність архітектурним принципам. Для студентських команд цей критерій тісно пов'язаний із використанням інженерних практик спільної роботи з кодом, зокрема систем контролю версій і код-рев'ю [9]. Наявність узгоджених правил роботи з кодом сприяє зменшенню кількості помилок і підвищенню загальної якості програмного продукту.

Ще одним суттєвим критерієм є *супроводжуваність програмного забезпечення*, яка визначає можливість подальшого розвитку та модифікації програмного продукту. У студентських ІТ-проєктах цей показник оцінюється за рівнем модульності, зрозумілістю структури проєкту та наявністю технічної документації [6]. Практика показує, що ефективна командна взаємодія на етапах проєктування і реалізації безпосередньо впливає на ці характеристики.

Таким чином, для оцінювання якості програмних продуктів у студентських ІТ-проєктах доцільно використовувати сукупність критеріїв, серед яких ключовими є функціональна повнота, надійність, якість програмного коду та супроводжуваність. Визначені критерії створюють основу для подальшого оцінювання впливу характеристик командної взаємодії на результати розроблення програмного забезпечення.

Оцінювання впливу командної взаємодії на якість програмних продуктів

Оцінювання впливу командної взаємодії на якість програмних продуктів у студентських ІТ-проєктах доцільно здійснювати шляхом аналізу зв'язку між характеристиками командної взаємодії та показниками якості програмного продукту, визначеними в попередньому підрозділі. Такий підхід дозволяє пояснити, яким чином організація спільної діяльності відображається на функціональній повноті, надійності, якості коду та супроводжуваності програмного забезпечення. Основні характеристики командної взаємодії та відповідні показники якості програмних продуктів узагальнено в таблиці 1.

У межах даної роботи оцінювання ґрунтується на логіці *«фактор командної взаємодії → прояв у процесі розроблення → вплив на показники якості»,* що відповідає сучасним підходам до аналізу результативності командної діяльності в програмній інженерії. Застосування такого підходу в освітньому середовищі не потребує складних вимірювань і може бути реалізоване шляхом аналізу артефактів командної роботи, зокрема репозиторію програмного коду, історії змін, результатів код-рев'ю, тестової документації та матеріалів супроводу проєкту.

Таблиця 1

Вплив характеристик командної взаємодії на показники якості програмних продуктів

Характеристика командної взаємодії	Прояв у процесі командної розробки	Очікуваний вплив на показники якості
Чіткий розподіл ролей і відповідальності	Визначені ролі (координатор/ інтегратор/ тестувальник), прозорий розподіл задач, контроль виконання	збільшиться функціональна повнота (менше «втрачених» вимог); підвищиться надійність (менше критичних дефектів через дублювання або пропуски);покращиться супроводжуваність (сталі підходи до структури)
Регулярна комунікація та синхронізація	Короткі регулярні обговорення, узгодження вимог і рішень, фіксація домовленостей	збільшиться функціональна повнота (коректніше тлумачення вимог); підвищиться надійність (менше логічних помилок); покращиться якість коду (менше суперечливих реалізацій)
Спільне планування та декомпозиція задач	Оцінювання задач, визначення пріоритетів, узгодження залежностей	збільшиться функціональна повнота (вища завершеність інкрементів); покращиться супроводжуваність (модульність через правильну декомпозицію); знизиться рівень технічного боргу
Єдині правила роботи з кодом (стандарти, стиль, архітектурні домовленості)	Узгоджені правила неймінгу, структури модулів, форматування, "Definition of Done"	покращиться якість коду (читабельність, структурованість); покращиться супроводжуваність (легше вносити зміни);знизиться кількість помилок інтеграції
Використання системи контролю версій і керування змінами	Гілкування, контроль інтеграції, історія змін, "прозорі" внески	підвищиться надійність (менше дефектів через конфлікти); покращиться якість коду (відстежуваність змін); покращиться супроводжуваність (легше локалізувати й виправляти помилки)

Характеристика командної взаємодії	Прояв у процесі командної розробки	Очікуваний вплив на показники якості
Код-рев'ю (peer review) перед інтеграцією змін	Перевірка логіки, стилю, архітектури; обговорення рішень; виправлення до злиття	покращиться якість коду (менше дефектів, краще дотримання принципів); покращиться супроводжуваність (єдині підходи); підвищиться надійність (менше критичних помилок)
Командна організація тестування (unit/ integration, регресія)	Визначені відповідальні, чек-листи, автоматизовані/ручні тести, фіксація дефектів	підвищиться надійність (менше збоїв і регресій); збільшиться функціональна повнота (перевірка вимог); покращиться якість коду (виявлення проблемних місць)
Наявність спільної документації та ведення артефактів	Опис вимог, рішення з проєктування, інструкції запуску, README, API-опис	покращиться супроводжуваність (легше передавати та підтримувати); збільшиться функціональна повнота (узгодженість вимог);знизиться кількість помилок використання
Механізми зворотного зв'язку та ретроспективи	Аналіз помилок, корекція процесу, узгодження покращень	підвищиться надійність (менше повторюваних дефектів); покращиться якість коду (поступове зменшення технічного боргу); підвищиться керованість процесу

Обговорення результатів оцінювання

Наведена модель показує, що найбільш відчутний вплив на якість програмного продукту в умовах студентських ІТ-проєктів мають *практики спільної роботи з кодом* (контроль версій, код-рев'ю) та *організоване тестування*, оскільки вони безпосередньо знижують ймовірність потрапляння дефектів до фінальної версії продукту та підвищують надійність програмного забезпечення. Водночас *розподіл ролей, комунікація і планування* створюють організаційні умови, за яких технічні практики застосовуються системно, а не епізодично, що позитивно відображається на функціональній повноті та супроводжуваності програмного продукту.

Практична цінність запропонованого підходу полягає в тому, що він може бути використаний у навчальному процесі для: формування прозорих критеріїв оцінювання командних результатів, аналізу причин помилок у програмних продуктах (через прив'язку до командних факторів) та підготовки рекомендацій щодо покращення організації командної роботи в наступних ітераціях проєктів.

Висновки з даного дослідження
і перспективи подальших розвідок у даному напрямі

У ході проведеного дослідження встановлено, що якість програмних продуктів, розроблених у межах студентських ІТ-проєктів, значною мірою визначається не лише рівнем технічної підготовки учасників, а й особливостями організації їхньої командної взаємодії. Аналіз сучасних наукових публікацій та узагальнення практики виконання студентських проєктів підтвердили доцільність розгляду командної складової як одного з ключових чинників впливу на результати розроблення програмного забезпечення.

У роботі виокремлено основні критерії якості програмних продуктів, релевантні для умов освітнього процесу, зокрема функціональну повноту, надійність, якість програмного коду та супроводжуваність. Показано, що ці характеристики можуть бути використані як основа для оцінювання результатів командної розробки програмного забезпечення без застосування складних експериментальних методів, шляхом аналізу артефактів командної діяльності та результатів тестування.

Запропонований підхід до оцінювання впливу командної взаємодії ґрунтується на встановленні зв'язку між організаційними та інженерними характеристиками командної роботи і показниками якості програмного продукту. Результати оцінювання свідчать, що найбільш відчутний позитивний вплив на якість програмних продуктів мають практики спільної роботи з кодом, зокрема використання систем контролю версій, код-рев'ю та командна організація тестування. Водночас такі чинники, як чіткий розподіл ролей, регулярна комунікація та узгоджене планування, створюють умови для системного застосування інженерних практик і зменшення кількості помилок у процесі розроблення.

Практичне значення отриманих результатів полягає у можливості використання запропонованого підходу в освітньому процесі для підвищення об'єктивності оцінювання результатів студентських ІТ-проєктів, а також для формування рекомендацій щодо вдосконалення організації командної роботи. Запропонована модель може бути використана викладачами під час планування проєктних завдань і аналізу причин успішності або недоліків програмних продуктів, створених студентськими командами.

Перспективи подальших досліджень у даному напрямі пов'язані з поглибленням методів оцінювання впливу командної взаємодії на якість програмного забезпечення, зокрема шляхом кількісного аналізу окремих показників, розширення набору критеріїв якості та дослідження впливу складу команд і рівня підготовки учасників на результати командної розробки. Окремий інтерес становить адаптація запропонованого підходу до різних форматів проєктного навчання та типів програмних продуктів.

Література

1. Кузьмінська О. Г., Мазорчук М. С. Групова динаміка і комунікації: добір засобів підтримки проєктного навчання студентів // Відкрите освітнє е-середовище сучасного університету. – 2023. – № 14. – С. 26–39. – DOI: 10.28925/2414-0325.2023.143.

2. Рантюк І. І., Вакалюк Т. А. Development of the model for using ICT in IT companies' specialists non-formal education // Information Technologies and Learning Tools. – 2023. – Vol. 98, № 6. – P. 164–176. – DOI: 10.33407/itlt.v98i6.5288.

3. Moskvin D. V., Vakaliuk T. A. Agile and Scrum: Increasing the Productivity of Software Development Teams // Тези XIV Міжнародної науково-технічної конференції «Інформаційно-комп'ютерні технології – 2024» (Житомир, 28–29 березня 2024 р.). – Житомир : Житомирська політехніка, 2024. – С. 7–8.

4. Stray V., Stabell I., Sæter G. E., Barbala A. M., Lindsjørn Y. Teamwork in agile software development: A mixed-method study of gender diversity and collaboration dynamics // Information and Software Technology. – 2025. – Vol. 187. – Article 107840. – DOI: 10.1016/j.infsof.2025.107840.

5. Nasir N., Usman M., Börstler J., Dzamashvili Fogelström N. Software engineering team project courses with industrial customers: Students' insights on challenges and lessons learned // Journal of Systems and Software. – 2025. – Vol. 226. – Article 112441. – DOI: 10.1016/j.jss.2025.112441.

6. Dorić L. et al. Evaluating Teamwork Components in Large Undergraduate Software Engineering Teams // ACM Transactions on Computing Education. – 2025. – Article. – DOI: 10.1145/3733840.

7. Fernandes S., Dinis-Carvalho J., Ferreira-Oliveira A., Oliveira A. T. Improving the Performance of Student Teams in Project-Based Learning with Scrum // Education Sciences. – 2021. – Vol. 11, № 8. – Article 444.

8. ISO/IEC 25010:2011. Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models. – Geneva : ISO, 2011.

9. Bacchelli A., Bird C. Expectations, outcomes, and challenges of modern code review // Proceedings of the 35th International Conference on Software Engineering (ICSE 2013). – 2013. – P. 712–721. – DOI: 10.1109/ICSE.2013.6606617.

10. Gousios G., Zaidman A., Storey M. A., Van Deursen A. Work practices and challenges in pull-based development: The integrator's perspective // Proceedings of the 37th IEEE/ACM International Conference on Software Engineering (ICSE 2015). – 2015. – P. 358–368. – DOI: 10.1109/ICSE.2015.55.

11. Salleh N., Mendes E., Grundy J. Empirical Studies of Pair Programming for CS/SE Teaching in Higher Education: A Systematic Literature Review // IEEE Transactions on Software Engineering. – 2011. – Vol. 37, № 4. – P. 509–525. – DOI: 10.1109/TSE.2010.59.

12. Лучкевич М. The impact of DevOps methodologies on the development of IT students' digital competencies // Information Technologies and Learning Tools. – 2025. – Vol. 108, № 4. – P. 53–63. – DOI: 10.33407/itlt.v108i4.6057.

References

1. Kuzminska O. H., Mazorchuk M. S. Hrupova dynamika i komunikatsii: dobir zasobiv pidtrymky proiektnoho navchannia studentiv // Vidkryte osvitnie e-seredovyshche suchasnoho universytetu. – 2023. – № 14. – S. 26–39. – DOI: 10.28925/2414-0325.2023.143.

2. Rantiuk I. I., Vakaliuk T. A. Development of the model for using ICT in IT companies specialists non-formal education // Information Technologies and Learning Tools. – 2023. – Vol. 98, № 6. – P. 164–176. – DOI: 10.33407/itlt.v98i6.5288.

3. Moskvin D. V., Vakaliuk T. A. Agile and Scrum: Increasing the Productivity of Software Development Teams // Tezy KhIV Mizhnarodnoi naukovo-tekhnichnoi konferentsii «Informatsiino-kompiuterni tekhnolohii – 2024» (Zhytomyr, 28–29 bereznia 2024 r.). – Zhytomyr : Zhytomyrska politekhnika, 2024. – S. 7–8.

4. Stray V., Stabell I., Sæter G. E., Barbala A. M., Lindsjørn Y. Teamwork in agile software development: A mixed-method study of gender diversity and collaboration dynamics // Information and Software Technology. – 2025. – Vol. 187. – Article 107840. – DOI: 10.1016/j.infsof.2025.107840.

5. Nasir N., Usman M., Börstler J., Dzamashvili Fogelström N. Software engineering team project courses with industrial customers: Students insights on challenges and lessons learned // Journal of Systems and Software. – 2025. – Vol. 226. – Article 112441. – DOI: 10.1016/j.jss.2025.112441.

6. Dorić L. et al. Evaluating Teamwork Components in Large Undergraduate Software Engineering Teams // ACM Transactions on Computing Education. – 2025. – Article. – DOI: 10.1145/3733840.

7. Fernandes S., Dinis-Carvalho J., Ferreira-Oliveira A., Oliveira A. T. Improving the Performance of Student Teams in Project-Based Learning with Scrum // Education Sciences. – 2021. – Vol. 11, № 8. – Article 444.

8. ISO/IEC 25010:2011. Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models. – Geneva : ISO, 2011.

9. Bacchelli A., Bird C. Expectations, outcomes, and challenges of modern code review // Proceedings of the 35th International Conference on Software Engineering (ICSE 2013). – 2013. – P. 712–721. – DOI: 10.1109/ICSE.2013.6606617.

10. Gousios G., Zaidman A., Storey M. A., Van Deursen A. Work practices and challenges in pull-based development: The integrators perspective // Proceedings of the 37th IEEE/ACM International Conference on Software Engineering (ICSE 2015). – 2015. – P. 358–368. – DOI: 10.1109/ICSE.2015.55.

11. Salleh N., Mendes E., Grundy J. Empirical Studies of Pair Programming for CS/SE Teaching in Higher Education: A Systematic Literature Review // IEEE Transactions on Software Engineering. – 2011. – Vol. 37, № 4. – P. 509–525. – DOI: 10.1109/TSE.2010.59.

12. Luchkevych M. The impact of DevOps methodologies on the development of IT students digital competencies // Information Technologies and Learning Tools. – 2025. – Vol. 108, № 4. – P. 53–63. – DOI: 10.33407/itlt.v108i4.6057.