

МЕТОДИЧНІ ЗАСАДИ ФОРМУВАННЯ НАВИЧОК КОМАНДНОЇ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ У СТУДЕНТІВ ЧЕРЕЗ УЧАСТЬ У РЕАЛЬНИХ ІТ-ПРОЄКТАХ УНІВЕРСИТЕТУ

METHODOLOGICAL PRINCIPLES OF DEVELOPING SOFTWARE TEAM DEVELOPMENT SKILLS IN STUDENTS THROUGH PARTICIPATION IN REAL UNIVERSITY IT PROJECTS

У статті обґрунтовано методичні засади формування навичок командної розробки програмного забезпечення у студентів закладів вищої освіти через їхню участь у реальних ІТ-проєктах університету. Актуальність теми зумовлена тим, що сучасна програмна інженерія вимагає від випускника не лише технічної підготовки, а й здатності до командної взаємодії, розподілу ролей, колективного ухвалення рішень, комунікації із замовником, роботи зі змінами, спільної відповідальності за результат і рефлексії командного процесу. Водночас у практиці професійної підготовки майбутніх фахівців з комп'ютерних наук значна частина навчальних завдань зберігає індивідуальний характер, через що командна складова майбутньої професійної діяльності формується фрагментарно.

У роботі уточнено сутність навичок командної розробки програмного забезпечення як інтегративного результату професійної підготовки, що поєднує технічну, організаційну, комунікативну та рефлексивну складові. Показано, що реальні ІТ-проєкти університету виступають педагогічно доцільним середовищем для формування зазначених навичок, оскільки забезпечують автентичність завдань, наявність реального користувача або замовника, необхідність взаємодії в команді, обмеження у часі, узгодження технічних рішень та відповідальність за якість кінцевого продукту. Теоретичною основою запропонованого підходу визначено компетентнісний, діяльнісний, проєктний, контекстний і колаборативний підходи.

Запропоновано модель організації формування навичок командної розробки програмного забезпечення, яка включає підготовчий, проєктно-виконавчий та рефлексивно-оцінювальний етапи. Визначено провідні методичні принципи: автентичність завдання, рольова організація команди, інструментальна наближеність до професійного середовища, поєднання педагогічного супроводу, поєднання командного та індивідуального оцінювання, обов'язкова рефлексія. Окремо охарактеризовано педагогічні умови ефективності: наявність реального університетського ІТ-проєкту, структурованих ролей у команді, використання цифрових інструментів спільної роботи, прозорих критеріїв оцінювання, наставницького супроводу викладача та регулярного зворотного зв'язку.

Практичний внесок статті полягає у розробленні цілісної методичної рамки, придатної для впровадження в освітні компоненти з програмування, програмної інженерії, веброзробки, командних курсів і курсового проєктування. Запропоновані положення можуть бути використані для модернізації професійної підготовки студентів у напрямі наближення її до реальних умов ІТ-галузі. Подальші дослідження доцільно спрямувати на емпіричну перевірку ефективності запропонованої моделі, розроблення критеріїв оцінювання рівнів сформованості навичок командної розробки та вивчення особливостей її реалізації в різних форматах навчання.

Ключові слова: командна розробка програмного забезпечення, реальні ІТ-проєкти, проєктне навчання, колаборативне навчання, програмна інженерія, професійна підготовка, peer assessment, командна взаємодія.

The article substantiates the methodological principles of developing software team development skills in higher education students through their participation in real university IT projects. The relevance of the topic stems from the fact that modern software engineering requires graduates not only to possess technical training but also to be capable of teamwork, role distribution, collective decision-making, communication with stakeholders, change management, shared responsibility for results, and reflection on the team process. At the same time, in the professional training of future specialists in computer science, a significant proportion of learning tasks remains individual, leading to a fragmented development of the teamwork component of future professional activity.

The study clarifies the essence of software team development skills as an integrative outcome of professional training that combines technical, organizational, communicative, and reflective components. It is shown that real university IT projects constitute a pedagogically appropriate environment for developing these skills, since they ensure the authenticity of tasks, the presence of a real user or customer, the necessity of team interaction, time constraints, coordination of technical decisions, and responsibility for the quality of the final product. The theoretical foundation of the proposed approach is formed by the competency-based, activity-based, project-based, contextual, and collaborative approaches.

A model for organizing software team development is proposed, including preparatory, project-implementation, and reflective-evaluative stages. The leading methodological principles are task authenticity, role-based team organization, instrumental proximity to the professional environment, staged pedagogical support, the combination of team and individual assessment, and mandatory reflection. The pedagogical conditions for effective implementation are also characterized, including the presence of a real university IT project, structured team roles, the use of digital collaboration tools, transparent assessment criteria, mentoring support from the teacher, and regular feedback.

The article's practical contribution lies in developing a coherent methodological framework suitable for implementation in educational components related to programming, software engineering, web development, team-based courses, and course project work. The proposed provisions may be used to modernize students' professional training to better align it with the real conditions of the IT industry. Further research should focus on empirically verifying the effectiveness of the proposed model, developing criteria for assessing levels of software team development skills, and studying the specific features of its implementation across different learning formats.

Key words: software team development, software engineering education, real IT projects, project-based learning, collaborative learning, higher education students, peer assessment, teamwork skills.

УДК 004.4:378.147

DOI <https://doi.org/10.32782/ip/95.2.43>

Стаття поширюється на умовах ліцензії відкритого доступу CC BY 4.0



Сіциліцин Ю.О.,
orcid.org/0000-0002-3888-5575
Ph.D.,

доцент кафедри інформатики і кібернетики

Мелітопольського державного педагогічного університету імені Богдана Хмельницького (Запоріжжя, Україна)

Постановка проблеми у загальному вигляді та її зв'язок з важливими науковими чи практичними завданнями. Сучасна програмна інженерія функціонує переважно як командна діяльність, у якій створення програмного продукту пов'язане не лише з написанням коду, а й із погодженням вимог, плануванням, колективним обговоренням рішень, комунікацією із замовником, контролем версій, тестуванням, документуванням та оцінюванням внеску кожного учасника. У зв'язку з цим професійна підготовка майбутніх фахівців має бути орієнтована не лише на індивідуальне оволодіння технологіями, а й на формування досвіду спільної професійно-орієнтованої діяльності [1; 2; 3].

Разом з тим, у підготовці студентів спеціальностей, пов'язаних із комп'ютерними науками та програмною інженерією, зберігається суперечність між реальною логікою ІТ-практики та характером значної частини навчальних завдань. Лабораторні роботи, тренувальні вправи й типові індивідуальні проекти формують окремі технічні вміння, але не відтворюють повною мірою середовище колективної розробки. Унаслідок цього випускник може володіти мовою програмування чи певним стеком технологій, але не мати достатньої готовності до розподілу ролей, узгодження рішень, ефективної взаємодії та спільної відповідальності за результат [4; 5].

Одним із найбільш перспективних шляхів подолання цієї суперечності є залучення студентів до реальних ІТ-проектів університету: створення та вдосконалення внутрішніх сервісів, сайтів підрозділів, навчальних платформ, аналітичних модулів, електронних кабінетів, інформаційних систем і допоміжних цифрових інструментів. Такі проекти створюють автентичне середовище діяльності, поєднують освітні цілі з реальною користю для університету та природно формують ситуації командної взаємодії [1; 6; 7].

Аналіз останніх досліджень і публікацій. У сучасній педагогічній літературі проектно-орієнтоване навчання розглядається як одна з провідних моделей професійно орієнтованого навчання у вищій школі. Узагальнювальний огляд P. Guo, N. Saab, L. S. Post, W. Admiraal показав, що проектне навчання вищої школи позитивно впливає на академічні результати, мотивацію, розвиток навичок співпраці та формування зв'язку між навчанням і реальними проблемами [1]. У праці E. Ceh-Varela, C. Canto-Bonilla, D. Duni доведено, що застосування проектно-орієнтоване навчання у курсі програмної інженерії у гібридному середовищі сприяє розвитку як технічних, так і soft skills, а також посилює включеність студентів у навчальний процес [6]. Додатково актуальні дані про спільне проектне навчання онлайн у вищій освіті підтверджують, що поєднання проектної логіки,

співпраці й системного супроводу позитивно пов'язане з мотивацією, спільним навчання та сприйнятими результатами навчання [7].

Окремий напрям досліджень стосується командних курсів програмної інженерії, пов'язаних із роботою над проектами для реальних замовників.

N. Nasir, M. Usman, J. Börstler, N. Dzamashvili Fogelström показали, що участь студентів у team project courses with industrial customers дає їм змогу застосовувати дисциплінарні знання у професійно наближеному середовищі, а більшість труднощів і засвоєних уроків стосується саме командної роботи, координації, комунікації, співпраці з користувачами і роботи в гібридному чи дистанційному форматі [4]. Подібні висновки узгоджуються з роботами

B. Bruegge, S. Krusche, L. Alperowitz та D. Broman, K. Sandahl, M. Abu Baker, де реалістичні проектні курси трактуються як ефективний механізм наближення університетської підготовки до умов професійної розробки [2; 3].

Значну увагу дослідники приділяють і проблемі оцінювання командної роботи. У статті N. Francis, C. Pritchard, Z. Prytherch, S. Rutherford наголошено, що командна робота є однією з найбільш затребуваних компетентностей, але саме командні заходи нерідко викликають у студентів незадоволення через нерівномірний внесок, конфлікти та відчуття несправедливості оцінювання [8]. У цьому контексті важливими є роботи, присвячені оцінюванню колег і діагностиці командної ефективності. B. Sridharan, J. McKay, D. Voud окреслюють чотири опори якісного оцінювання колег у командному навчанні [9].

M. L. Loughry, M. W. Ohland, D. D. Moore пропонує теоретичну оцінку ефективності члена команди [10], а M. Freeman, J. McKenzie демонструють переваги інструменту SPARK для самої взаємооцінювання внеску студентів у командні проекти [11].

Помітну роль у сучасному осмисленні командної підготовки відіграють і дослідження групової динаміки в software engineering education. V. Isomöttönen і T. Taipalus показали, що в студентських програмних групових проектах статус і вплив учасника визначаються не лише технічною компетентністю, а й відповідальністю, залученістю, здатністю підтримувати команду та брати на себе організаційні функції [12]. S. A. Licorish, M. Galster, G. M. Kapitsaki, A. Tahir встановили зв'язок між зусиллями студентів, якістю співпраці, задоволеністю й адекватністю створених результатів у проектах з розробки програмного забезпечення [13]. Додатково J. Navío-Marco, D. Ardura, A. Galán та Z.-B. Xie зі співавторами підтверджують, методи педагогіки, орієнтовані на командну роботу та засновані на

командній роботі і якість навчальних результатів [14; 15].

Виділення невіршених раніше частин загальної проблеми. Попри значний масив праць з проєктного навчання, спільного навчання та оцінювання командної роботи, недостатньо розкритим залишається питання саме методичного забезпечення формування навичок командної розробки програмного забезпечення через участь студентів у реальних ІТ-проєктах університету. У наукових джерелах добре представлені або загальні переваги командного навчання, або окремі кейси програмної інженерії, або підходи до оцінювання командної роботи, проте бракує цілісного опису моделі, у якій університетський ІТ-проєкт виступає автентичним педагогічним середовищем для системного формування готовності студентів до командної розробки [6; 4; 8].

Недостатньо визначеними залишаються також критерії добору проєктів для студентської участі, межі допустимого втручання студентів у реальні цифрові системи університету, баланс між освітньою доцільністю та ризиком, а також роль викладача у супроводі такої діяльності.

Мета статті полягає в теоретичному обґрунтуванні методичних засад формування навичок командної розробки програмного забезпечення у студентів через участь у реальних ІТ-проєктах університету.

Для досягнення поставленої мети передбачено розв'язання таких завдань: уточнити сутність і структуру навичок командної розробки програмного забезпечення; визначити педагогічний потенціал реальних університетських ІТ-проєктів для формування таких навичок; обґрунтувати провідні методичні підходи, принципи та педагогічні умови їх формування; запропонувати трьохетапну модель організації цієї діяльності; окреслити критерії добору проєктів, до яких доцільно залучати студентів; сформулювати практичні рекомендації щодо впровадження запропонованої моделі в освітній процес.

Виклад основного матеріалу. Під навичками командної розробки програмного забезпечення у цьому дослідженні розуміємо інтегровану сукупність умінь, способів дії та моделей професійної поведінки, що забезпечують ефективну участь студента у спільному створенні програмного продукту. До їх структури доцільно віднести технічний, організаційний, комунікативний та рефлексивний компоненти. Технічний компонент охоплює розроблення, тестування, документування, роботу з репозиторіями та спільне редагування артефактів. Організаційний пов'язаний із плануванням, дотриманням дедлайнів, веденням завдань, координацією дій і прийняттям рішень. Комунікативний включає здатність до фахового обговорення, аргументації, прийняття та надання

зворотного зв'язку. Рефлексивний передбачає самооцінку внеску, усвідомлення труднощів і корекцію власної поведінки в команді [1; 5].

Теоретико-методичною основою формування зазначених навичок виступає поєднання кількох підходів. Компетентнісний підхід орієнтує освітній процес на здатність діяти у професійно значущих ситуаціях. Діяльнісний підхід забезпечує формування вмінь через включення студента у реальну або наближену до реальної практику. Проєктний підхід створює цілісне завдання з кінцевим продуктом, часовими рамками й відповідальністю. Контекстний підхід наближує навчальну діяльність до професійного середовища. Колаборативний підхід акцентує, що знання й рішення у команді виникають через взаємодію, координацію та спільне конструювання результату [1; 14; 7].

Педагогічний потенціал реальних ІТ-проєктів університету полягає передусім в автентичності. На відміну від умовного навчального завдання, реальний проєкт має конкретного користувача, практичну функцію, обмеження, пов'язані з ресурсами й термінами, а також наслідки для якості виконаної роботи. У такому середовищі студент працює не просто над вправою, а над цифровим продуктом, який має бути придатним до використання. Це підсилює мотивацію, відповідальність, необхідність узгодження дій і формує реальну потребу в командній роботі.

Запропонована модель формування навичок командної розробки програмного забезпечення у студентів через участь у реальних ІТ-проєктах університету включає три взаємопов'язані етапи: підготовчий, проєктно-виконавчий та рефлексивно-оцінювальний. Їх зміст, послідовність і взаємозв'язки подано на рисунку 1.

На підготовчому етапі визначається зміст реального університетського ІТ-проєкту, формуються команди, проводиться первинне ознайомлення з вимогами, обираються інструменти, встановлюються правила комунікації, оцінювання та звітності. На цьому ж етапі доцільно вводити короткий модуль підготовки до teamwork: принципи ролей, правила зустрічей, способи прийняття рішень, базові техніки розв'язання конфліктів, етичні норми командної взаємодії.

Проєктно-виконавчий етап передбачає ітеративну реалізацію продукту. Робота організується у коротких циклах: постановка задач, планування, виконання, рев'ю, демонстрація проміжного результату, корекція. За такого підходу студентська команда функціонує як мала модель професійної розробницької групи з розподілом ролей і відповідальностей. Роль викладача змінюється: він виступає не лише джерелом знань, а й наставником, модератором і фасилітатором процесу, допомагає структурувати взаємодію, вчасно виявляти дисбаланс внеску та підтримувати освітню цінність проєкту.

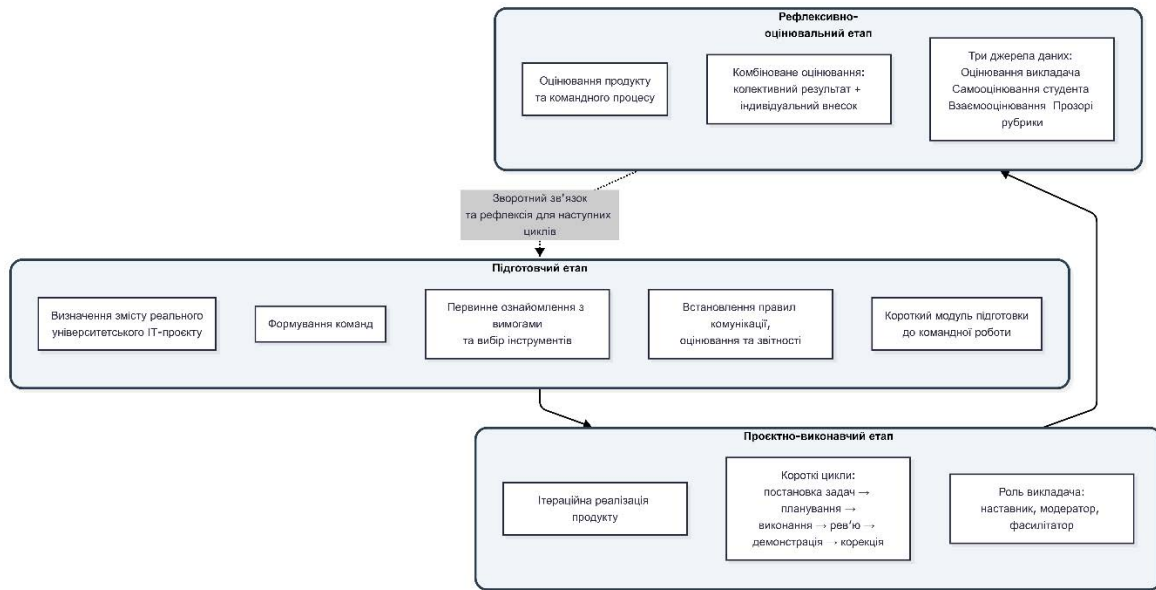


Рис. 1. Структурно-логічна модель формування навичок командної розробки програмного забезпечення у студентів через участь у реальних ІТ-проектах університету

Рефлексивно-оцінювальний етап спрямований на аналіз не лише програмного продукту, а й самого командного процесу. У межах цього етапу доцільно застосовувати комбіноване оцінювання: частина результату визначається якістю колективного продукту, частина – індивідуальним внеском студента, підтвердженим виконаними задачами, участю в обговореннях, документацією, рев'ю та дотриманням ролі. Ефективність такого підходу підсилюється поєднанням оцінювання викладача, самооцінювання й взаємооцінювання студентів, а також використанням прозорих рубрик.

Важливою умовою реалізації запропонованої моделі є обґрунтований добір реальних ІТ-проектів, до яких залучаються студенти. Не кожен університетський цифровий продукт є педагогічно доцільним об'єктом студентської участі. Доцільно залучати студентів до проектів із низьким або помірним рівнем ризику, що мають освітню або організаційну цінність для університету та не передбачають прямого втручання у критичну інфраструктуру, обробку чутливих персональних даних чи виконання незворотних операцій у бойовому середовищі. Натомість системи, пов'язані з безпекою, автентифікацією, фінансовими операціями, кадровими даними, юридично значущими записами або повноцінним доступом до продуктивних баз даних, не повинні виступати первинним середовищем студентської роботи.

З огляду на це до робіт, до яких студентів доцільно допускати насамперед, належать інформаційні сайти підрозділів, навчальні сервіси, тестові або пілотні модулі, аналітичні панелі на деперсоналізованих даних, допоміжні інструменти внутрішньої автоматизації, UI/UX-прототипи, документація, тестування й удосконалення нефінансових модулів.

Натомість складніші або ризикованіші задачі можуть бути доступні лише старшим студентам, лише у тестових чи захищених-середовищах, з анонімізованими даними, поетапним допуском, обов'язковим code review і постійним наставницьким контролем. Такий підхід дозволяє поєднати автентичність навчання з вимогами безпеки, відповідальності та педагогічної доцільності.

Педагогічними умовами ефективності запропонованої моделі є: наявність реального, а не штучно змодельованого університетського ІТ-проекту; значущість кінцевого результату для конкретного підрозділу або користувача; чітка організація ролей; використання цифрових інструментів, що забезпечують прозорість процесу; регулярні короткі зустрічі або контрольні точки; поступове ускладнення проектних завдань; наставницький супровід викладача; поєднання технічного та поведінкового оцінювання; обов'язкова письмова або усна рефлексія. Без цих умов командна робота ризикує залишитися звичайним поділом задач між студентами, а не повноцінним середовищем формування навичок командної розробки.

Практично доцільним є дотримання кількох рекомендацій. По-перше, реальний проєкт не слід розпочинати без короткого модуля підготовки до командної роботи. По-друге, критерії оцінювання мають бути оголошені до старту роботи та містити окремі показники командного процесу. По-третє, проєкт потрібно розбивати на короткі етапи з проміжними демонстраціями, а не оцінювати лише фінальний продукт. По-четверте, студентам слід надавати не лише технічні консультації, а й підтримку щодо організації взаємодії. По-п'яте, доцільно поєднувати зворотний зв'язок викладача, оцінювання колег та нотатки для рефлексії. По-шосте, у разі виявлення дисбалансу

внеску необхідно мати механізм корекції індивідуального балу.

Отже, запропонована трьохетапна модель – підготовчий, проєктно-виконавчий і рефлексивно-оцінювальний етапи – дозволяє розглядати реальний університетський ІТ-проєкт не як додаткову позааудиторну активність, а як керований освітній механізм формування готовності студентів до роботи в сучасному середовищі програмної інженерії.

Висновки. Формування навичок командної розробки програмного забезпечення у студентів є одним із ключових завдань сучасної професійної підготовки майбутніх фахівців з комп'ютерних наук і програмної інженерії. Такі навички мають інтегративний характер і поєднують технічну, організаційну, комунікативну та рефлексивну складові. Їх розвиток неможливо забезпечити повною мірою лише за допомогою індивідуальних навчальних завдань, оскільки вони потребують систематичної спільної діяльності у професійно наближеному контексті.

Установлено, що участь студентів у реальних ІТ-проєктах університету є педагогічно доцільним середовищем для формування навичок командної розробки програмного забезпечення. Методичними засадами цього процесу виступають компетентнісний, діяльнісний, проєктний, контекстний і колаборативний підходи, а провідними принципами – автентичність завдання, рольова організація команди, інструментальна наближеність до професійного середовища, поетапний педагогічний супровід, поєднання командного й індивідуального оцінювання та обов'язкова рефлексія.

Практичне значення статті полягає в тому, що її положення можуть бути використані для модернізації курсів з програмування, програмної інженерії, веброзробки, проєктування програмних систем і курсового проєктування. Перспективою подальших досліджень є емпірична валідація моделі, розроблення критеріїв і показників сформованості навичок командної розробки, а також визначення оптимальних форматів її впровадження у різних умовах організації навчального процесу.

БІБЛІОГРАФІЧНИЙ СПИСОК:

1. Guo P., Saab N., Post L. S., Admiraal W. A review of project-based learning in higher education: Student outcomes and measures. *International Journal of Educational Research*. 2020. Vol. 102. Article 101586. DOI: 10.1016/j.ijer.2020.101586.
2. Bruegge B., Krusche S., Alperowitz L. Software engineering project courses with industrial clients. *ACM Transactions on Computing Education*. 2015. Vol. 15, No. 4. Article 17. DOI: 10.1145/2732155.
3. Broman D., Sandahl K., Abu Baker M. The Company Approach to Software Engineering Project Courses. *IEEE Transactions on Education*. 2012. Vol.

55, No. 4. P. 445–452. DOI: 10.1109/TE.2012.2187208.

4. 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.

5. Licorish S. A., Galster M., Kapitsaki G. M., Tahir A. Understanding students' software development projects: Effort, performance, satisfaction, skills and their relation to the adequacy of outcomes developed. *Journal of Systems and Software*. 2022. Vol. 186. Article 111156. DOI: 10.1016/j.jss.2021.111156.

6. Ceh-Varela E., Canto-Bonilla C., Duni D. Application of Project-Based Learning to a Software Engineering course in a hybrid class environment. *Information and Software Technology*. 2023. Vol. 158. Article 107189. DOI: 10.1016/j.infsof.2023.107189.

7. Admiraal W., Saab N., Guo P. Online Collaborative Project-Based Learning in Higher Education: Students' Motivation, Collaborative Learning, and Perceived Outcomes. *Active Learning in Higher Education*. 2026. OnlineFirst. DOI: 10.1177/14697874261419448.

8. Francis N., Pritchard C., Prytherch Z., Rutherford S. Making teamwork work: enhancing teamwork and assessment in higher education. *FEBS Open Bio*. 2025. Vol. 15, No. 1. P. 35–47. DOI: 10.1002/2211-5463.13936.

9. Sridharan B., McKay J., Boud D. The Four Pillars of Peer Assessment for Collaborative Teamwork in Higher Education. In: Noroozi O., De Wever B. (eds). *The Power of Peer Learning*. Cham: Springer, 2023. P. 3–24. DOI: 10.1007/978-3-031-29411-2_1.

10. Loughry M. L., Ohland M. W., Moore D. D. Development of a Theory-Based Assessment of Team Member Effectiveness. *Educational and Psychological Measurement*. 2007. Vol. 67, No. 3. P. 505–524. DOI: 10.1177/0013164406292085.

11. Freeman M., McKenzie J. SPARK, a confidential web-based template for self and peer assessment of student teamwork: Benefits of evaluating across different subjects. *British Journal of Educational Technology*. 2002. Vol. 33, No. 5. P. 551–569. DOI: 10.1111/1467-8535.00291.

12. Isomöttönen V., Taipalus T. Status indicators in software engineering group projects. *Journal of Systems and Software*. 2023. Vol. 198. Article 111612. DOI: 10.1016/j.jss.2023.111612.

13. Licorish S. A., Galster M., Kapitsaki G. M., Tahir A. Understanding students' software development projects: Effort, performance, satisfaction, skills and their relation to the adequacy of outcomes developed. *Journal of Systems and Software*. 2022. Vol. 186. Article 111156. DOI: 10.1016/j.jss.2021.111156.

14. Navío-Marco J., Ardura D., Galán A. Teamwork skills development in hybrid and online universities: the perspective of the future teachers. *Journal of New Approaches in Educational Research*. 2025. Vol. 14. Article 13. DOI: 10.1007/s44322-025-00032-1.

15. Xie Z. B., Cheng X. Y., Li X. Y. et al. Team based learning pedagogy enhances the education quality: a systematic review and meta-analysis. *BMC Medical Education*. 2025. Vol. 25. Article 580. DOI: 10.1186/s12909-025-07175-x.

Дата першого надходження статті до видання: 30.04.2026

Дата прийняття статті до друку після рецензування: 20.05.2026

Дата публікації (оприлюднення) статті: 29.05.2026