

DOI: <https://doi.org/10.32782/2524-0072/2025-77-74>

УДК 33.02:338.24:004.738.5

ВПЛИВ ІНТЕГРАЦІЇ JIRA, SLACK ТА ІНШИХ ІНСТРУМЕНТІВ НА ЕФЕКТИВНІСТЬ МЕНЕДЖМЕНТУ КОМАНДНОЇ ВЗАЄМОДІЇ В SCRUM-ПРОЄКТАХ

IMPACT OF INTEGRATING JIRA, SLACK, AND OTHER TOOLS ON THE EFFECTIVENESS OF TEAM INTERACTION MANAGEMENT IN SCRUM PROJECTS

Ревенко Артем Володимирович

здобувач вищої освіти ступеня «доктор філософії»,

Сумський державний університет

ORCID: <https://orcid.org/0000-0002-8854-2386>**Матвєєва Юлія Анатоліївна**

кандидат економічних наук, доцент,

Сумський державний університет

ORCID: <https://orcid.org/0000-0002-3082-5551>**Сулим Вікторія Василівна**

кандидат економічних наук, доцент,

Сумський державний університет

ORCID: <http://0000-0001-9114-068X>**Revenko Artem, Matvieieva Yuliia, Sulym Viktoriia**

Sumy State University

У статті досліджено вплив інтеграції Jira, Slack та інших цифрових інструментів на ефективність управління взаємодією в Scrum-проєктах. Актуальність зумовлена потребою у прозорості процесів, зменшенні координаційних витрат і підвищенні передбачуваності командної роботи. Методологія базувалася на аналізі показників «до/після» впровадження інтеграцій та кейс-дослідженні розподіленої Scrum-команди. Оцінювалися швидкість реагування на інциденти, кількість нарад, рівень інформаційних втрат і навантаження Product Owner. Результати показали: час реакції скорочено до 15 хв, кількість координаційних зустрічей зменшено на 30-40 %, інформаційні втрати – більш ніж на 80 %, а моніторинг – до 10 хв/день. Практична цінність полягає у рекомендаціях щодо створення інтегрованої цифрової екосистеми, яка підвищує ефективність планування, прискорює постачання інкрементів і забезпечує стабільний контроль завдань.

Ключові слова: методологія Scrum, Jira, Slack, управління IT-проєктами, інтеграція програмних інструментів, менеджмент командної взаємодії, безперервна інтеграція/безперервне постачання (CI/CD), прозорість процесів, розподілені та віддалені команди.

The article analyzes the impact of integrating Jira, Slack and complementary digital tools on the efficiency of managing team interaction in Scrum projects. The relevance is determined by the growing complexity of IT products, the geographic distribution of teams and the demand for transparency, rapid coordination and predictable outcomes in dynamic development environments. The purpose is to identify how integrated ecosystems enhance managerial decisions, reduce communication gaps and increase team productivity. The methodology combines comparative analysis of performance indicators before and after integration with a case study of a distributed Scrum team. Managerial metrics included response time to critical incidents, preparation and duration of stand-ups, frequency of unplanned meetings, the level of information losses and the Product Owner's daily monitoring effort. The results show significant improvements: response time to defects decreased to about fifteen minutes, coordination meetings were reduced by 30-40 percent, information losses dropped by more than 80 percent, and monitoring was cut to around ten minutes per day. Qualitative findings demonstrate that threaded Slack discussions linked to Jira



tasks enabled asynchronous collaboration across time zones, streamlined communication flows, reduced context switching and allowed earlier identification of blockers. Automated notifications and task updates increased collective awareness and improved the quality of managerial decisions. Overall, the integrated setup functioned as a socio-technical workflow in which Jira provided structure and traceability while Slack ensured immediacy and continuity. This synergy enhanced feedback loops, predictability of sprint outcomes, discipline in task execution and reduced managerial risks. The practical value lies in recommendations for configuring notifications, structuring channels, linking tasks and integrating CI/CD tools. These measures can help organizations accelerate incident resolution, stabilize workflow cadence, optimize resources and strengthen transparency. Although the focus is on IT teams, the approach can also be applied to other project environments that require disciplined task management and rapid decision-making.

Keywords: Scrum methodology, Jira, Slack, IT project management, integration of software tools, team collaboration management, continuous integration/continuous delivery (CI/CD), process transparency, distributed and remote teams.

Постановка проблеми. Сучасна ІТ-індустрія розвивається високими темпами: очікування коротких термінів постачання та потреба швидкої адаптації посилюють вимоги до команд. Agile-методології, зокрема Scrum, забезпечують адаптивність, прозорість і взаємодію з клієнтом [1]. Водночас практика показує, що впровадження Scrum саме по собі не усуває типові проблеми: ускладнення координації між ролями, зниження прозорості зі зростанням обсягів робіт, затримки управлінських рішень та зростання адміністративного навантаження у розподілених командах. Втрата або несвоєчасне оновлення інформації про статуси завдань підвищує технічний борг, зменшує передбачуваність результатів та відтерміновує релізи, водночас створюючи ризики управлінської дезінформації.

Оптимізація можлива через інтегровані інструменти, що поєднують централізоване управління завданнями, безперервний обмін актуальними даними, автоматизовані сповіщення та прозорість KPI. Ефективне застосування таких інструментів сприяє не лише підвищенню продуктивності, але й зміцненню дисципліни виконання, зменшенню управлінських ризиків і підвищенню якості планування. Практичною основою такого підходу є зв'язка Jira (спринти, моніторинг ключових показників) [2] і Slack (структурована комунікація, підтримка командної синергії) [3], яка прискорює ухвалення рішень, стабілізує ритм роботи та зменшує ризики невиконання цілей спринту.

Аналіз останніх досліджень і публікацій. Протягом останніх двох десятиліть методології гнучкої розробки програмного забезпечення стали об'єктом інтенсивних наукових досліджень і водночас невід'ємною складовою практики управління ІТ-проектами. Від моменту оприлюднення Agile Manifesto [4] у 2001 році підходи, засновані на принципах Agile, зокрема Scrum, зарекомендували себе

як ефективні інструменти організації командної роботи в умовах швидких змін вимог і скорочених життєвих циклів продуктів. Актуальна редакція Scrum Guide [5] визначає ролі, артефакти та події, водночас залишаючи простір для вибору конкретних інструментів, що підтримують процеси планування, комунікації та моніторингу прогресу.

Дослідження К. Дікерта, М. Паасіваарі та К. Лассеніуса [6] свідчать, що успішне масштабування Agile-підходів значною мірою залежить від якості комунікаційної інфраструктури команди та рівня прозорості робочих процесів.

Серед інструментів, що найчастіше згадуються в наукових і галузевих джерелах, провідні позиції займають Jira та Slack. За визначенням GoIT [2], Jira є комплексною платформою для управління беклогом, планування спринтів, відстеження завдань і формування аналітичних звітів. Slack, у свою чергу, позиціонується як гнучка комунікаційна платформа, що забезпечує структуровані обговорення, інтеграцію з іншими інструментами та автоматизацію повторюваних комунікаційних процесів.

Роботи М. Друрі-Гроган [7] підтверджують, що своєчасний доступ до актуальних даних про виконання завдань сприяє прискоренню ухвалення рішень у Scrum-командах. Інтеграція таких рішень, як Jira та Slack, формує безперервний інформаційний потік між учасниками процесу, що узгоджується з висновками 17th Annual State of Agile Report [8], де вказано, що близько 75% Agile-команд застосовують інтегровані цифрові платформи для підтримки прозорості та координації.

Водночас низка досліджень [9] акцентує увагу на потенційних недоліках інтегрованих підходів – надмірній кількості сповіщень, дублюванні інформації та потребі у додатковому налаштуванні інтеграцій відповідно до

специфіки конкретної команди. Це свідчить, що, попри очевидні переваги інтеграції Jira та Slack, залишається відкритим завдання розроблення рекомендацій щодо їх оптимального використання в контексті Scrum із урахуванням розміру команди, типу проєкту та організаційної культури.

Проведений аналіз літератури та практичних прикладів підтверджує, що поєднання систем управління завданнями з комунікаційними платформами не є новою ідеєю. Проте аналіз впливу інтеграції Jira та Slack на ефективність командної взаємодії у межах Scrum-проєктів є недостатньо розробленим у науковій площині, що й визначає актуальність даного дослідження.

Формулювання цілей статті. Метою статті є вироблення науково обґрунтованих підходів до підвищення ефективності менеджменту Scrum-команд через інтеграцію Jira, Slack та суміжних сервісів. Для цього необхідно відстежувати вплив інтеграції на продуктивність і прозорість процесів, визначати сценарії її використання у різних типах команд, використовувати цифрові інструменти для покращення планування, синхронізації та контролю, а також розробити практичні рекомендації для впровадження інтегрованої екосистеми у сучасних IT-проєктах.

Виклад основного матеріалу дослідження. Scrum [10] є сучасним гнучким фреймворком управління проєктами, розробленим для ефективної роботи в умовах високої невизначеності, швидкої зміни вимог та обмежених ресурсів. Його методологічна основа передбачає інкрементальне створення продукту шляхом реалізації серії коротких ітерацій (спринтів), у межах яких здійснюються планування, виконання визначеного обсягу робіт та аналіз отриманих результатів [11].

На відміну від традиційних каскадних моделей, Scrum акцентує увагу на забезпеченні прозорості процесів, управлінській гнучкості та підтримці безперервної взаємодії із замовником. В управлінському контексті цей підхід спрямований на формування адаптивних командних структур, ефективне використання наявних ресурсів, своєчасне виявлення ризиків і мінімізацію управлінських затримок.

З позицій проєктного менеджменту Scrum виконує функцію інтегрованої системи організації робіт, що передбачає:

- моніторинг і контроль прогресу через зустрічі та аналіз артефактів;
- своєчасну ідентифікацію проблем та блокерів, які можуть вплинути на досягнення цілей спринту;

Таблиця 1

Структурні елементи Scrum-середовища

Компонент	Елемент	Опис
Ролі	Product Owner (PO)	Формує та пріоритизує <i>Product Backlog</i> , орієнтуючись на максимізацію цінності продукту.
	Scrum Master (SM)	Забезпечує дотримання принципів Scrum, усуває перешкоди та виконує фасилітаційні функції.
	Development Team	Кросфункціональна група фахівців, яка безпосередньо реалізує інкремент продукту.
Артефакти	Product Backlog	Впорядкований перелік функцій, покращень, виправлень та інших завдань, необхідних для продукту.
	Sprint Backlog	Підмножина завдань із <i>Product Backlog</i> , визначена для реалізації в поточному спринті.
	Increment	Готовий до використання результат роботи, створений у межах спринту.
Події	Sprint Planning	Планування завдань та визначення способів їх виконання в межах спринту.
	Daily Scrum	Щоденна коротка нарада для синхронізації роботи та обговорення прогресу.
	Sprint Review	Презентація інкременту та збір зворотного зв'язку від стейкхолдерів.
	Sprint Retrospective	Аналіз процесу роботи та визначення шляхів його вдосконалення.

Джерело: складено авторами на основі [1; 2]

- підвищення відповідальності учасників команди за результати виконаних завдань;
- гнучке коригування планів відповідно до змін зовнішнього та внутрішнього середовища проєкту;
- підтримання високого рівня залученості стейкхолдерів у процес прийняття управлінських рішень.

У межах проведеного дослідження структурні елементи Scrum систематизовано за трьома ключовими управлінськими компонентами – ролями, артефактами та подіями, що наведено у таблиці 1.

З наведених у таблиці 1 даних випливає, що Scrum забезпечує чітко структуровану організаційну модель, яка створює умови для ефективної координації командної взаємодії, своєчасного виявлення відхилень та підтримання високого рівня прозорості процесів.

В умовах сучасних розподілених мультидисциплінарних команд, де часові зони, різнорідність компетенцій та високий темп змін збільшують ризики втрати контексту й виникнення затримок, практична реалізація Scrum дедалі частіше ґрунтується на використанні інтегрованих цифрових інструментів управління завданнями та комунікацій. Поєднання системи Jira як централізованої платформи планування та відстеження прогресу із сервісом Slack як середовищем оперативної комунікації створює інтегровану управлінську екосистему, що мінімізує інформаційні розриви, підвищує швидкість ухвалення рішень та сприяє стабільності ритму виконання проєктних робіт.

Jira у Scrum-процесах, розроблена компанією Atlassian, еволюціонувала від інструмента відстеження помилок до комплексної гнучкої системи управління робочими процесами, яка широко використовується у фреймворках Scrum та Kanban [12]. В інтегрованому Scrum-середовищі вона виконує роль центрального управлінського вузла, забезпечуючи: планування завдань і спринтів; контроль виконання та дотримання термінів моніторинг ключових показників ефективності (KPI); формування звітності для стейкхолдерів.

Система Jira забезпечує прозоре ведення Product Backlog, підтримуючи різні типи завдань (Story, Task, Bug, Epic), їх пріоритизацію та оцінювання у story points або людиногодинах. Епіки зберігають стратегічну цілісність продукту, а детальні описи завдань із посиланнями на Confluence перетворюють беклог на структуровану знаннєву базу, доступну для всієї команди.

Планування спринтів здійснюється шляхом формування Sprint Backlog із візуалізацією навантаження та автоматичною фіксацією дат, що дозволяє зіставляти обсяг робіт із пропускну здатністю команди та зменшувати ризик невиконання цілей.

Відстеження прогресу охоплює повний життєвий цикл завдання з гнучкими статусами. Призначення відповідальних, додавання коментарів, вкладень та ведення історії змін підвищують прозорість процесів і скорочують час узгоджень, зокрема під час code review або переходу до тестування. Налаштовувані Scrum-дошки візуалізують стан робіт, а додаткові колонки, фільтри за епіками чи виконавцями та гнучке сортування роблять інструмент ефективним у мультикомандних середовищах [13].

Аналітичний модуль Jira підтримує інспекцію та адаптацію процесів, надаючи дані для систематичного аналізу ефективності. Використання цих метрик підвищує передбачуваність виконання завдань на 20–25 % у порівнянні з командами, що не здійснюють аналогічний моніторинг. Водночас ізольоване використання інструменту сповільнює поширення контексту; інтеграція Jira зі Slack та іншими сервісами забезпечує безперервний обмін інформацією і зменшує інформаційні втрати, зміцнюючи прозорість управлінських процесів. Таким чином, Jira виступає не лише як трекер завдань, а як ядро цифрової екосистеми Scrum-команди.

Slack є корпоративною платформою робочих комунікацій, спроектованою для структурованої взаємодії в команді та мінімізації інформаційного шуму [14]. У межах Scrum він функціонує як основний канал оперативного обміну, у якому проводяться щоденні узгодження, технічні дискусії та вирішення поточних питань.

Структурування комунікацій досягається через тематичні канали, організовані за командами, спринтами або напрямками робіт. Така організація дозволяє фільтрувати нерелевантні повідомлення та підтримувати фокус на пріоритетних завданнях. Ключова перевага Slack полягає в інтеграції з Jira: події щодо завдань (створення, зміни, коментарі) автоматично надходять до відповідних каналів; зі Slack можна ініціювати створення завдань та здійснювати посилання на них у процесі обговорення. У практичній роботі автора впровадження окремого каналу технічних оновлень скоротило час між виявленням проблеми та її реєстрацією в Jira з годин до кількох хвилин.

Функція гілок обговорень (threads) дає змогу зберігати контекст дискусій без перевантаження загального каналу, що особливо корисно для асинхронних стендапів і консультацій у розподілених командах. Гнучкі налаштування сповіщень (за ключовими словами, «тихі години», розділення критичних і звичайних оновлень) дозволяють регулювати інтенсивність комунікацій і знижувати ризик професійного вигорання.

Окремої уваги заслуговує автоматизація за допомогою ботів: нагадування про Scrum-ритуали, короткі опитування щодо планів і результатів, а також автоматичне створення завдань у Jira за певними позначками повідомлень. Таким чином, якщо Jira виконує роль «центрального сховища» робочої інформації, то Slack можна розглядати як «нервову систему» команди, яка забезпечує швидке поширення контексту та мінімізує лаг між подією і дією. При цьому Slack не замінює інструментів управління завданнями; максимальний ефект досягається саме в комплексній зв'язці з Jira та іншими сервісами. Ефективність взаємодії Jira-Slack суттєво зростає при інтеграції з інструментами документації, контролю версій та автоматизації, які формують єдиний інформаційний простір і забезпечують команді своєчасний доступ до необхідних даних. Зокрема, Confluence виконує функцію централізованого сховища знань: сторінки безпосередньо пов'язуються з Jira-тикетами, а документація може містити динамічні звіти з Jira. Такий підхід прискорює процес онбордингу та зменшує кількість повторних запитів, оскільки достатньо надати посилання у відповідному тикеті.

GitHub та GitLab забезпечують повноцінну підтримку роботи з репозиторіями та процесами погодження змін (Pull Request/Merge Request), із можливістю прив'язки комітів до ключів завдань у системі управління проєктами [15]. Інтеграція зі Slack дозволяє автоматично надсилати сповіщення у спеціалізовані канали рев'ю, що істотно скорочує час погодження змін та зменшує тривалість циклу PR → merge, у практичних сценаріях – до менш ніж 24 годин.

Системи безперервної інтеграції та розгортання (CI/CD), зокрема Jenkins, GitHub Actions і GitLab CI/CD, автоматизують процеси збірки програмного забезпечення [16; 17], виконання тестів та розгортання продукту. Результати автоматично фіксуються у Jira, а Slack оперативно інформує команду про завершення відповідних етапів. Такий підхід забезпечує мож-

ливість негайного початку тестування з боку фахівців QA після успішної збірки без необхідності додаткових запитів, що оптимізує часові витрати та підвищує узгодженість роботи. За необхідності в інтегрованому середовищі застосовуються й інші інструменти, функціональне призначення яких визначається специфікою проєктних процесів. Зокрема, Trello використовується для підтримки спрощених робочих потоків, характерних для завдань у сфері маркетингу та дизайну; Azure DevOps – для комплексного управління вихідним кодом та завданнями в межах корпоративної інфраструктури; Miro і FigJam – для фасилітації дизайн-спринтів із подальшою структурованою фіксацією результатів у Confluence [18].

Таким чином, поєднання Jira та Slack з інструментами Confluence, GitHub/GitLab та CI/CD формує інтегровану цифрову екосистему управління проєктами, у якій завдання структуровані, комунікація контекстуалізована, знання впорядковані, код перебуває під контролем із проходженням рев'ю, а процеси розгортання – автоматизовані. Це забезпечує узгодженість дій, підтримує прозорість процесів і підвищує керованість проєктного циклу.

Інтеграція Jira та Slack створює ефект синергії: Jira виконує роль «центрального мозку» управління, тоді як Slack виступає «нервовою системою» оперативної комунікації. Така взаємодія знижує бар'єри у комунікації, прискорює реакцію на зміни та підвищує прозорість процесів. Автоматизовані сповіщення з Jira у Slack повідомляють про створення завдань, зміну їх статусів і появу коментарів; завдяки гнучким фільтрам надходять лише релевантні події. На практиці це дозволяє скоротити середній час реакції на критичні дефекти з 60 до приблизно 15 хвилин. Синхронізація також усуває необхідність дублювання інформації: зміни, внесені у Jira, миттєво відображаються у Slack, а фрагменти обговорень із чату можуть бути приєднані безпосередньо до завдань.

Публікація змін у відкритих каналах комунікацій забезпечує прозорість прогресу для всіх учасників проєкту незалежно від їх ролей, що дозволяє зменшити кількість статусних нарад на 30-40 %. Використання потоків обговорень (threads) сприяє підтриманню безперервного зворотного зв'язку та є ефективним інструментом для асинхронних консультацій у командах, що працюють у різних часових поясах. Зокрема, розробники можуть оперативно отримувати коментарі від фахівців з контр-

олю якості, а власник продукту – отримувати відгуки після проведення демонстрацій.

Ілюстративним прикладом є кейс розподіленої команди з 12 учасників, що працювала над SaaS-продуктом. До впровадження інтеграції Jira та Slack команда стикалася з затримками у реагуванні на критичні помилки, нерегулярним оновленням статусів завдань, а власник продукту витрачав близько 40 хвилин щодня на ручний моніторинг прогресу. Після впровадження каналу #dev-updates та налаштування автоматичного відображення карток завдань при згадуванні ключа тикета середній час реакції на критичні дефекти зменшився на 65 %, випадки пропуску оновлень були повністю усунені, а час моніторингу для Product Owner скоротився до приблизно 10 хвилин на день. Інтеграція Jira та Slack не лише підвищує зручність використання інструментів, а й трансформує саму модель взаємодії Scrum-команди. Процеси стають більш синхронізованими та прозорими, що важливо у розподілених командах та проєктах із високою динамікою змін.

Для обґрунтування ефективності інтеграції проведено порівняння з аналогічним, але неінтегрованим середовищем. Аналіз базувався на даних реальних проєктів та узагальненій статистиці команд зі схожими характеристиками. Оцінювання здійснювалося за такими показниками: середній час реакції на нові завдання та критичні дефекти, тривалість підготовки до щоденних стендапів, кількість позапланових зустрічей для уточнення статусів, частка пропущених оновлень у завданнях, а також час, який Product Owner витрачає на моніторинг прогресу (таблиця 2).

Отримані результати свідчать, що автоматизовані сповіщення у Slack істотно скорочують час реакції на завдання та дефекти, що особливо важливо для розподілених команд.

Підвищення прозорості процесів підтверджується зниженням частки пропущених оновлень у Jira, а зменшення кількості позапланових зустрічей вказує на те, що Slack частково бере на себе функції синхронізації команди. Скорочення часу підготовки до стендапів зумовлене доступністю актуальних даних з Jira у зручному для роботи форматі безпосередньо у Slack. Для Product Owner це означає істотне зниження витрат часу на контроль завдань і можливість зосередитися на стратегічних аспектах розвитку продукту.

Висновки. У роботі досліджено вплив інтеграції Jira, Slack та суміжних інструментів на ефективність взаємодії та управління в Scrum-командах. Аналіз охоплював не лише функціональні можливості окремих сервісів, а й результативність їхнього поєднання у єдину інтегровану екосистему. Проведено детальний аналіз можливостей Jira і Slack для планування, ведення Product Backlog, управління комунікаціями та інтеграційних сценаріїв. Визначено ключові сценарії використання: автоматизовані сповіщення, прив'язка обговорень у Slack до задач Jira, відображення статусів завдань у чаті. Додатково розглянуто роль Confluence, GitHub/GitLab, CI/CD та Miro у формуванні єдиного інформаційного простору для команди. Для оцінки ефективності інтеграції проведено порівняльний аналіз показників «до» і «після» впровадження інтегрованого середовища.

Результати дослідження свідчать, що інтеграція Jira, Slack та допоміжних інструментів забезпечує значне підвищення ефективності роботи Scrum-команд: середній час реакції на нові задачі та критичні дефекти скорочується на 65-75 %, кількість координаційних зустрічей зменшується на 30-40 %, прозорість процесів зростає, а пропущені оновлення у Jira знижуються більш ніж на 80 %. Крім того,

Таблиця 2

Порівняння ключових показників до та після інтеграції Jira і Slack

Показник	До інтеграції Jira + Slack	Після інтеграції Jira + Slack	Зміна, %
Середній час реакції на критичний дефект	~60 хв	~15 хв	-75%
Середній час реакції на нове завдання	~2 год	~35 хв	-71%
Час підготовки до щоденного стендапу	~20 хв	~15 хв	-25%
Кількість позапланових зустрічей на місяць	10–12	6–7	-40%
Відсоток пропущених оновлень завдань	12%	2%	-83%
Час, витрачений Product Owner на моніторинг прогресу	~40 хв/день	~10 хв/день	-75%

Джерело: складено авторамт на основі власних спостережень та даних [6; 7]

щоденний час моніторингу для Product Owner скорочується до приблизно 10 хвилин.

Практична цінність дослідження полягає у тому, що висновки можуть бути використані Scrum-майстрами та Project Manager'ами для підвищення продуктивності команд, особливо в умовах розподілених та мультидисциплінарних колективів, а також для розроблення політик автоматизації процесів у межах Agile-трансформацій. Перспективи подальших досліджень включають інтеграцію AI-агентів у Jira та Slack для класифікації задач, генерації

звітів та надання рекомендацій; оцінку впливу гейміфікації у Slack на мотивацію команди; аналіз взаємозв'язку між швидкістю комунікації та якістю управлінських рішень; а також розроблення уніфікованої системи KPI для різних типів IT-проектів.

Таким чином, інтеграція Jira, Slack і суміжних інструментів не лише оптимізує управління завданнями та комунікаційні процеси, а й суттєво підвищує ефективність Scrum-команд, роблячи їх більш узгодженими, прозорими та результативними.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ:

1. Atlassian. Scrum. URL: <https://www.atlassian.com/agile/scrum> (дата звернення: 02.08.2025).
2. Goit. Що таке Jira і для чого вона потрібна. URL: <https://goit.global/ua/articles/shcho-take-jira-i-dlia-chohova-potribna/> (дата звернення: 02.08.2025).
3. Speka. Гілова Г. Що таке Slack. URL: <https://speka.media/korporativnii-mesendzer-slack-gaid-dlya-menedzera-ta-laifxaki-p1y8xp> (дата звернення: 02.08.2025).
4. Zhezherau, A. (Wrike). What Is the Agile Manifesto? Wrike Agile Guide. URL: <https://www.wrike.com/agile-guide/agile-manifesto/> (дата звернення: 03.08.2025).
5. Schwaber K., Sutherland J. Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game. Scrum.org, 2020. – 20 с.
6. Dikert K., Paasivaara M., Lassenius C. Challenges and Success Factors for Large-Scale Agile Transformations: A Systematic Literature Review. *Journal of Systems and Software*. 2016. № 119. P. 87–108.
7. Drury-Grogan M. L. Performance on Agile Teams: Relating Iteration Objectives and Critical Decisions to Project Management Success Factors. *Information and Software Technology*. 2014. № 56(5). P. 506–515.
8. VersionOne. 17th Annual State of Agile Report. Digital.ai, 2023. 28 p.
9. Moe N. B., Smite D., Agerfalk P. J. Understanding Shared Leadership in Agile Development: A Case Study. *Journal of Systems and Software*. 2012. № 85(12). P. 2812–2826.
10. Hossain E., Babar M. A., Paik H. Using Scrum in Global Software Development: A Systematic Literature Review (ICGSE 2009)
11. Scrum.org. What is Scrum? URL: <https://www.scrum.org/resources/what-scrum-module> (дата звернення: 07.08.2025)
12. Atlassian. Jira Software overview. URL: <https://confluence.atlassian.com/jirasoftwareserver/jira-software-overview-938845024.html> (дата звернення: 07.08.2025).
13. Malsam, W. Scrum Methodology: An Introduction to the Scrum Process. ProjectManager.com. URL: <https://www.projectmanager.com/blog/scrum-methodology> (дата звернення: 08.08.2025).
14. Slack Technologies. Slack Help Center. URL: <https://slack.com/help> (дата звернення: 08.08.2025).
15. Білик, K. GitHub vs GitLab. Обираємо сервіс під потреби проекту. DOU. URL: <https://dou.ua/lenta/articles/github-vs-gitlab/> (дата звернення: 09.08.2025).
16. Vivesh. Jenkins vs. GitHub Actions vs. GitLab CI. DEV Community. URL: <https://dev.to/574n13y/jenkins-vs-github-actions-vs-gitlab-ci-2k35> (дата звернення: 09.08.2025).
17. Forsgren N., Humble J., Kim G. Accelerate: The Science of Lean Software and DevOps (2018).
18. StackShare. Azure DevOps vs Trello. What are the differences? URL: <https://www.stackshare.io/stackups/azure-devops-vs-trello> (дата звернення: 10.08.2025)

REFERENCES:

1. Atlassian. Scrum. Available at: <https://www.atlassian.com/agile/scrum> (accessed August 2, 2025).
2. Goit. What is Jira and what is it used for. Available at: <https://goit.global/ua/articles/shcho-take-jira-i-dlia-chohova-potribna/> (accessed August 2, 2025).
3. Speka. Hilova, H. What is Slack. Available at: <https://speka.media/korporativnii-mesendzer-slack-gaid-dlya-menedzera-ta-laifxaki-p1y8xp> (accessed August 2, 2025).

4. Zhezherau, A. (Wrike). What Is the Agile Manifesto? Wrike Agile Guide. Available at: <https://www.wrike.com/agile-guide/agile-manifesto/> (accessed August 3, 2025).
5. Schwaber, K., & Sutherland, J. Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game. Scrum.org, 2020. – 20 p.
6. Dikert, K., Paasivaara, M., & Lassenius, C. (2016) Challenges and Success Factors for Large-Scale Agile Transformations: A Systematic Literature Review. *Journal of Systems and Software*, no. 119, pp. 87–108.
7. Drury-Grogan, M. L. (2014) Performance on Agile Teams: Relating Iteration Objectives and Critical Decisions to Project Management Success Factors. *Information and Software Technology*, no. 56(5), pp. 506–515.
8. VersionOne. 17th Annual State of Agile Report. Digital.ai, 2023. 28 p.
9. Moe, N. B., Smite, D., & Agerfalk, P. J. (2012) Understanding Shared Leadership in Agile Development: A Case Study. *Journal of Systems and Software*, no. 85(12), pp. 2812–2826.
10. Hossain, E., Babar, M. A., & Paik, H. Using Scrum in Global Software Development: A Systematic Literature Review (ICGSE 2009).
11. Scrum.org. What is Scrum? Available at: <https://www.scrum.org/resources/what-scrum-module> (accessed August 7, 2025).
12. Atlassian. Jira Software overview. Available at: <https://confluence.atlassian.com/jirasoftwareserver/jira-software-overview-938845024.html> (accessed August 7, 2025).
13. Malsam, W. Scrum Methodology: An Introduction to the Scrum Process. ProjectManager.com. Available at: <https://www.projectmanager.com/blog/scrum-methodology> (accessed August 8, 2025).
14. Slack Technologies. Slack Help Center. Available at: <https://slack.com/help> (accessed August 8, 2025).
15. Bilyk, K. GitHub vs GitLab: Choosing a Service for Project Needs. DOU. Available at: <https://dou.ua/lenta/articles/github-vs-gitlab/> (accessed August 9, 2025).
16. Vivesh. Jenkins vs. GitHub Actions vs. GitLab CI. DEV Community. Available at: <https://dev.to/574n13y/jenkins-vs-github-actions-vs-gitlab-ci-2k35> (accessed August 9, 2025).
17. Forsgren, N., Humble, J., & Kim, G. Accelerate: The Science of Lean Software and DevOps. 2018.
18. StackShare. Azure DevOps vs Trello. What are the differences? Available at: <https://www.stackshare.io/stackups/azure-devops-vs-trello> (accessed August 10, 2025).