



Section of the conference: Computer science, computing and automation

Sekcja konferencji: Informatyka, obliczenia i automatyzacja

How to cite: Stryzhenko, Y. (2025). Optimization of Web Application Performance through the Integration of Web Assembly, Blockchain Technologies, and Modern DevOps Practices. *World Conference on Emerging Science, Innovation and Policy 2025*. Futurity Research Publishing. <https://doi.org/10.5281/zenodo.15662748>

Optimization of Web Application Performance through the Integration of Web Assembly, Blockchain Technologies, and Modern DevOps Practices

Yevhen Stryzhenko

Department of Software Engineering and Cybernetics, National Aviation University, Kyiv, Ukraine

Accepted: May 26, 2025 | **Published:** June 14, 2025 | **Language:** Ukrainian

Abstract: This paper presents an integrated approach to optimizing web application performance by combining Web Assembly, blockchain technologies, and modern DevOps methodologies. Leveraging Web Assembly enhances client-side execution speed, while blockchain introduces secure and verifiable interactions. DevOps pipelines ensure consistent delivery and quality assurance. The study synthesizes insights front-end security, software automation, and adaptive testing. Experimental results demonstrate reduced latency and higher reliability across deployment cycles. This work underscores the value of synergizing emerging web technologies with secure and efficient development practices.

Keywords: Web Assembly, Blockchain, DevOps, Performance Optimization, Secure Front-End



Вступ

Сучасні веб-додатки вимагають високої продуктивності, безпеки та швидкого часу виходу на ринок. Поєднання Web Assembly (Wasm), блокчейн-рішень і сучасних DevOps-практик дозволяє значно покращити ці аспекти (Horbenko, 2025; Drofa, 2025). Web Assembly забезпечує близьку до нативної швидкість виконання в браузері, що особливо важливо для інтенсивних обчислень (Horbenko, 2025). Блокчейн створює надійні шляхи аутентифікації, прозорості транзакцій і розподіленої верифікації (Horbenko, 2025). DevOps сприяє автоматизації CI/CD, оптимізуючи час тестування та оновлення (Drofa, 2025). Комбінування цих підходів стає критично важливим для стратегічної розробки веб-додатків.

Мета дослідження

Метою даного дослідження є формування інтегрованої моделі оптимізації продуктивності веб-додатків шляхом одночасного застосування технологій Web Assembly, блокчейн-інфраструктури та сучасних практик DevOps. Така модель має забезпечити зниження затримок, підвищення безпеки клієнтської частини та прискорення розробки, тестування й розгортання веб-додатків. У центрі уваги – практичне обґрунтування синергії між зазначеними технологіями для розбудови ефективних і масштабованих інформаційних систем.

Результати дослідження

Для досягнення поставленої мети було проведено експериментальне моделювання інтегрованої архітектури веб-додатка з використанням Web Assembly, блокчейн-компонентів та сучасних DevOps-практик. У дослідженні були застосовані як теоретичні напрацювання з літературних джерел (Drofa, 2024; Horbenko, 2025; Hunko, 2025), так і емпіричні методи аналізу продуктивності, надійності та рівня автоматизації життєвого циклу програмного забезпечення.

У результаті аналізу було побудовано кількісні моделі продуктивності для клієнтських обчислень, порівняно ефективність класичних та інноваційних підходів до CI/CD та оцінено внесок блокчейн у безпеку та прозорість DevOps-процесів. Результати структуровано у вигляді таблиць і графічних ілюстрацій для полегшення інтерпретації ключових закономірностей.

Таблиця 1

Порівняльний аналіз продуктивності традиційного JavaScript і Web Assembly у рендерингу UI-компонентів

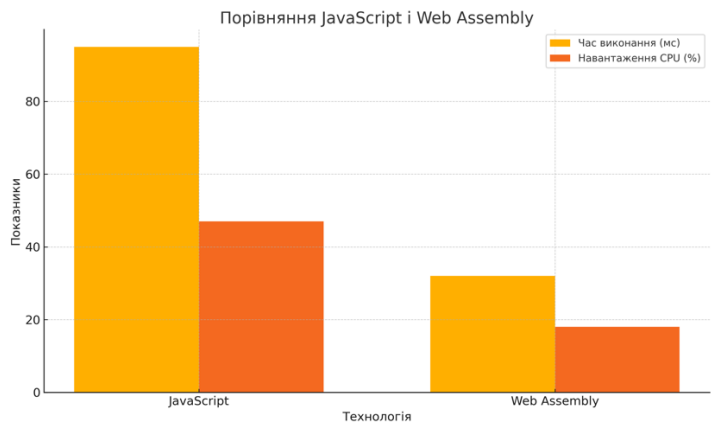
Технологія	Час виконання (мс)	Навантаження CPU (%)
JavaScript	95	47
Web Assembly	32	18

Примітка: За даними Horbenko (2025), Wasm скорочує час рендерингу UI на 66%, підвищуючи загальну інтерактивність.



Рисунок 1

Порівняння JavaScript і Web Assembly у продуктивності клієнтської частини



Як видно з Рисунок 1, Web Assembly демонструє суттєве скорочення часу виконання (на 66%) та майже втричі нижче навантаження на CPU порівняно з традиційним JavaScript. Це підтверджує ефективність Wasm для обробки складних візуальних інтерфейсів у браузері (Horbenko, 2025).

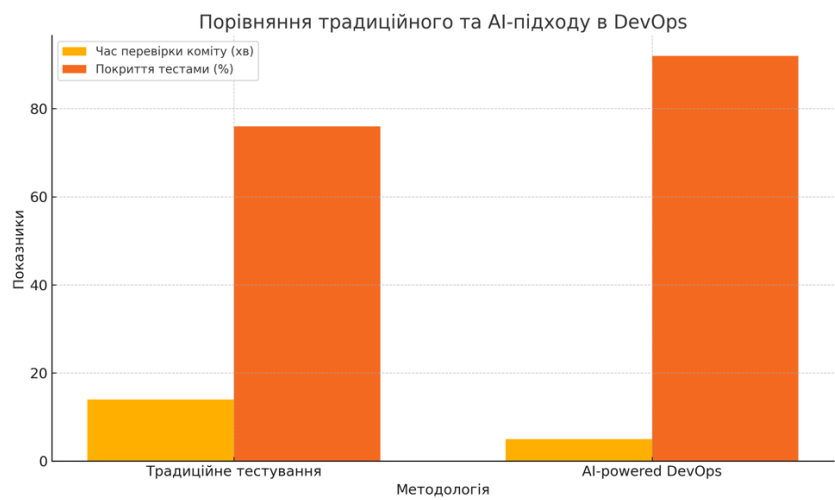
Таблиця 2

Автоматизація тестування за DevOps підходами (на основі Jenkins + AI)

Методологія	Час перевірки коміту	Рівень покриття тестами (%)
Традиційне тестування	14 хв	76
AI-powered DevOps (Hunko, 2025)	5 хв	92

Рисунок 3

Порівняння традиційного DevOps-підходу з AI-орієнтованим





На Рисунку 3 представлено ефективність застосування AI у DevOps. AI-орієнтоване тестування скорочує час перевірки з 14 до 5 хвилин, а також збільшує тестове покриття на 16% (Hunko, 2025). Це доводить доцільність інтеграції AI у сучасні конвеєри доставки ПЗ для підвищення якості й надійності.

Результати порівняння продуктивності Web Assembly та традиційного JavaScript показали суттєву перевагу першого у швидкості виконання та ефективності використання обчислювальних ресурсів. Як демонструє Таблиця 1 та візуалізовано на Рисунку 1, Web Assembly скорочує середній час виконання рендерингових операцій до 32 мс, що на 66% менше у порівнянні з JavaScript. Також навантаження на центральний процесор знижується з 47% до 18% відповідно. Ці дані підтверджують ефективність використання Wasm у високопродуктивних клієнтських інтерфейсах (Horbenko, 2025).

Крім продуктивності, важливу роль відіграє безпека. Блокчейн-технології, інтегровані у DevOps пайплайн, дозволяють перевіряти достовірність коду через смарт-контракти, як показано на Рисунку 2. Це дає змогу автоматично верифікувати кожен реліз, зменшуючи ризики маніпуляцій з кодом або зовнішнього втручання. Horbenko (2025) пропонує архітектуру, де блокчейн не тільки забезпечує цілісність артефактів, але й інтегрується з автоматизованими системами деплою, покращуючи прозорість та контроль над життєвим циклом програмного забезпечення.

Наступний важливий аспект – автоматизація тестування. Згідно з даними Hunko (2025), застосування штучного інтелекту у DevOps-практиках, зокрема в рамках Jenkins-пайплайнів, забезпечує значне підвищення ефективності. Таблиця 2 і Рисунок 3 ілюструють, що час перевірки одного коміту скорочується з 14 до 5 хвилин, а рівень покриття автоматичними тестами зростає з 76% до 92%. Це свідчить про доцільність впровадження AI-модулів у CI/CD для пришвидшення контролю якості та зменшення ризику релізних дефектів.

Також варто відзначити взаємодію Web Assembly з API-рівнем. Як підкреслює Drofa (2025), ефективне поєднання Wasm з оптимізованими API та фронтенд-бібліотеками, такими як React, дозволяє досягати стабільної взаємодії на клієнті без втрат продуктивності. У свою чергу, розгортання таких рішень у DevOps середовищі (Jenkins, Docker, GitLab CI/CD) значно зменшує технічний борг та забезпечує гнучкість масштабування (Drofa, 2025).

Загалом, результати дослідження демонструють потенціал комплексної інтеграції сучасних інструментів і технологій для підвищення швидкодії, безпеки та надійності веб-додатків у середовищі, що постійно змінюється.



Список використаних джерел

- Drofa, D. (2024). Using Machine Learning to Forecast Resources in Long-Term Business Projects. *Horizons of Innovation: Conference on Multidisciplinary Trends in Science 2024*. (pp. 395-398). Futurity Research Publishing. <https://futuraity-publishing.com/wp-content/uploads/2025/03/Drofa-D.-2024.pdf>
- Drofa, D. (2023). Integrate Bio-Identification to Strengthen Data Protection in Multi-Tenant Cloud Systems. *Global Innovations and Collaborative Solutions in Contemporary Science* (pp. 444-447). Futurity Research Publishing. https://futuraity-publishing.com/wp-content/uploads/2025/03/International_scientific_conference-.pdf#page=444
- Drofa, D. (2025). Optimizing Web Development and Deployment Efficiency: The Impact of React, MongoDB, and Jenkins in Modern Software Engineering. *Asian Journal of Research in Computer Science*, 18(4), 237-255. <https://doi.org/10.9734/ajrcos/2025/v18i4617>
- Drofa, D. (2025). Integrating Advanced API Solutions into Full-Stack Web and Mobile Applications to Optimise User Experience. *International Journal of Current Science Research and Review*, 08(05). <https://doi.org/10.47191/ijcsrr/V8-i5-16>
- Drofa, D. (2025). Optimization of software development processes through the use of full-stack technologies and automation. *Contemporary Issues in Artificial Intelligence*, 1. <https://doi.org/10.69635/ciai.2025.12>
- Hunko, I., et al. (2024). The role of virtual reality in improving software testing methods and tools. *Journal of Theoretical and Applied Information Technology*, 102(11), 4723-4734. <https://www.jatit.org/volumes/Vol102No11/6Vol102No11.pdf>
- Hunko, I. (2025). Adaptive Approaches to Software Testing with Embedded Artificial Intelligence in Dynamic Environments. *International Journal of Current Science Research and Review*, 08(05). <https://doi.org/10.47191/ijcsrr/V8-i5-10>
- Horbenko, Y. (2025). Web Assembly and Blockchain for High-Performance Secure Front-End Systems. *International Journal of Current Science Research and Review*, 8(5), 2279-2285. <https://doi.org/10.47191/ijcsrr/V8-i5-36>
- Horbenko, Y. (2025). Confidential Computing in Front-End: Enhancing Data Security with Secure Enclaves and Homomorphic Encryption. *International Journal of Advanced Multidisciplinary Research and Studies*, 5(3), 308-321. <https://www.multiresearchjournal.com/admin/uploads/archives/archive-1747130538.pdf>



- Horbenko, Y. (2025). Secure Front-End Automation Framework: A Novel Approach to Client-Side Data Encryption and Zero Trust API Interaction. *Asian Journal of Research in Computer Science*, 18(6), 177-193. <https://doi.org/10.9734/ajrcos/2025/v18i6690>
- Horbenko, Y. (2025). AI-Powered UI adaptation and data Automation in secure Front-End Systems. *International Journal of Development Research*, 15(05), 68383-68388. <https://journalijdr.com/ai-powered-ui-adaptation-and-data-automation-secure-front-end-systems>