

Список використаних джерел:

1. Chacon S., Straub B. Pro Git. 2nd ed. New York : Apress, 2014. 456 p.
2. Git internals – Git objects . URL: <https://git-scm.com/book/en/v2/Git-Internals-Git-Objects> (дата звернення: 16.03.2026).
3. O’Sullivan B. Mercurial: the definitive guide . URL: <https://book.mercurial-scm.org/read/concepts.html> (дата звернення: 16.03.2026).
4. Fossil SCM technical overview. URL: https://fossil-scm.org/home/doc/trunk/www/tech_overview.wiki (дата звернення: 16.03.2026).
5. Darcs theory . URL: <https://darcs.net/Theory> (дата звернення: 16.03.2026).
6. Git commit-graph documentation . URL: <https://git-scm.com/docs/commit-graph> (дата звернення: 16.03.2026).
7. Git partial clone . URL: <https://git-scm.com/docs/partial-clone> (дата звернення: 16.03.2026).
8. Mercurial wire protocol. URL: <https://www.mercurial-scm.org/wiki/WireProtocol> (дата звернення: 16.03.2026).
9. Git pack objects . URL: <https://git-scm.com/docs/git-pack-objects> (дата звернення: 16.03.2026).

УДК 004.4:004.2

РЕТРО-ТЕХНОЛОГІЇ: ЧИ МОЖЛИВО ВІДТВОРИТИ ПРОГРАМУВАННЯ ДЛЯ ЗАСТАРІЛИХ ПЛАТФОРМ

Собчук Є. О.

yehor.sobchuk@gmail.com

Черкаський державний фаховий бізнес-коледж

Люта М. В.

м. Черкаси, Україна

Актуальність роботи зумовлена широким використанням застарілих інформаційних технологій у критично важливих галузях, таких як фінансовий сектор, державне управління, енергетика та охорона здоров'я. Незважаючи на стрімкий розвиток сучасних ІТ-рішень, значна частина інформаційних систем

продовжує функціонувати на базі застарілих програмних платформ, що обумовлено їх надійністю, високою вартістю модернізації та складністю міграції.

Водночас використання таких технологій супроводжується низкою проблем, зокрема підвищеними ризиками кібербезпеки, обмеженою сумісністю з сучасними системами та дефіцитом кваліфікованих фахівців для їх підтримки. У цих умовах особливої актуальності набуває питання оцінювання доцільності збереження, модернізації або заміни застарілих технологій, а також можливості відтворення програмування для таких платформ.

Застарілі технології (obsolete technologies) визначаються як програмні та апаратні засоби, що втратили масове застосування або офіційну підтримку виробників, однак продовжують функціонувати в окремих галузях. Відповідно до сучасних досліджень, їх існування є природним наслідком швидкого розвитку інформаційних технологій та обмеженої можливості повної заміни складних інформаційних систем [1].

Життєвий цикл технологій передбачає етапи впровадження, активного використання та поступового витіснення більш ефективними рішеннями. Проте, на відміну від класичної моделі, у сфері програмного забезпечення спостерігається явище «технологічної інерції», коли застарілі системи продовжують експлуатуватися через високу вартість їх заміни або критичність для бізнес-процесів [2].

Критичний аналіз показує, що твердження про повну неактуальність старих технологій є спрощеним. Наприклад, мова програмування COBOL, створена у 1959 році, досі широко використовується у фінансовому секторі та державних установах. За оцінками аналітиків, значна частина банківських транзакцій у світі обробляється системами, написаними на COBOL [3]. Це свідчить про високу стабільність та надійність таких систем, але водночас створює проблему дефіциту фахівців.

З іншого боку, використання застарілих технологій має суттєві недоліки. Основними з них є:

- відсутність регулярних оновлень безпеки;
- обмежена сумісність із сучасними стандартами;
- складність інтеграції з новими системами.

Ці фактори значно підвищують ризики кіберзагроз. Зокрема, відсутність оновлень безпеки робить системи вразливими до відомих експлойтів, що підтверджується звітами міжнародних організацій у сфері кібербезпеки [4].

Водночас слід зауважити, що не всі сучасні технології є принципово новими. Більшість концепцій програмування (алгоритми, структури даних, принципи управління пам'яттю) були сформовані ще в середині ХХ століття. Сучасні мови програмування переважно розвивають ці підходи, додаючи нові рівні абстракції та покращуючи зручність використання [5].

Критичний аналіз також показує, що проблема застарілих технологій часто полягає не стільки в їх віці, скільки в якості реалізації програмних систем. Низька якість коду, відсутність документації та застаріла архітектура є більш значущими факторами ризику, ніж сама технологія [2].

У сучасних умовах доцільно застосовувати комбінований підхід, що передбачає:

- поступову модернізацію існуючих систем;
- використання методів міграції даних;
- впровадження сучасних засобів захисту інформації;
- підготовку фахівців для підтримки критично важливих застарілих систем.

Таким чином, ретро-технології не слід розглядати виключно як застарілий та неактуальний інструмент. У багатьох випадках вони залишаються важливою складовою інформаційної інфраструктури, хоча й потребують модернізації та додаткових заходів безпеки.

Застарілі технології є невід'ємною частиною сучасного ІТ-середовища. Їх використання обумовлене економічними, технічними та організаційними чинниками. Незважаючи на наявні ризики, повна відмова від таких систем є часто недоцільною. Оптимальним підходом є їх поетапна модернізація та інтеграція із сучасними технологіями.

Список використаних джерел:

1. Laudon K. C., Laudon J. P. Management Information Systems. New York: Pearson, 2021. 560 p.
2. Sommerville I. Software Engineering. 10th ed. Boston: Pearson, 2021. 816 p.
3. IBM Corporation. COBOL Systems and Business Applications. URL: <https://www.ibm.com> (дата звернення: 15.03.2026).
4. ENISA. Threat Landscape 2023. Luxembourg: Publications Office of the European Union, 2023. 120 p.
5. Sebesta R. W. Concepts of Programming Languages. 12th ed. Boston: Pearson, 2022. 720 p.

УДК 004.42

МЕХАНІЗМИ УНАОЧНЕННЯ АЛГОРИТМІВ ТА ЕФЕКТИВНІ СПОСОБИ ЇХ ВІЗУАЛІЗАЦІЇ ЗАСОБАМИ ВЕБТЕХНОЛОГІЙ

Близнюк В. П.

iterbleiker1@gmail.com

Черкаський державний фаховий бізнес-коледж

Марченко С. В.

м. Черкаси, Україна

Унаочнення алгоритмів використовується як засіб підтримки розуміння обчислювальних процесів, налагодження програм і дослідження поведінки структур даних. На відміну від статичних описів алгоритмів, інтерактивні візуальні представлення відображають зміну станів під час виконання, що може бути корисним як у навчальних, так і в інженерних задачах. Сучасні вебзастосунки дають змогу поєднати в одному середовищі код, візуальну модель, історію виконання та керування сценаріями взаємодії. Автори [1] вказують, що корпус емпіричних досліджень у цій галузі все ще є відносно обмеженим, тобто наявність відкритих питань щодо того, які саме механізми унаочнення є справді результативними.