

О. Є. ОГНЄВА

кандидат технічних наук, доцент,  
завідувач кафедри програмних засобів і технологій  
Херсонський національний технічний університет  
ORCID: 0000-0001-6206-0285

Д. В. ЗЯРКО

здобувач другого (магістерського) рівня вищої освіти  
кафедри програмних засобів і технологій  
Херсонський національний технічний університет  
ORCID: 0009-0002-7103-2838

## ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ МОДЕЛЕЙ РЕІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ЦИФРОВОЇ ТРАНСФОРМАЦІЇ ТУРИСТИЧНОГО СЕРВІСУ

У статті досліджено ефективність моделей реінжинірингу програмного забезпечення як ключового інструменту цифрової трансформації туристичних інформаційних систем. Об'єктом дослідження є програмне забезпечення туристичної платформи, а предметом – процеси оптимізації її архітектури, продуктивності, безпеки та якості обслуговування користувачів. Особливу увагу приділено глибокому аналізу вихідного коду, реверс-інжинірингу, редизайну архітектури та впровадженню багаторівневих моделей із використанням сучасних технологій, таких як PHP, MySQL, OAuth, хешування паролів. Практична частина ґрунтується на реінжинірингу веб-застосунку Travel-website, що включає детальний аналіз коду, розробку оновленої архітектури, інтеграцію з Google OAuth, підвищення безпеки зберігання паролів та впровадження адаптивного інтерфейсу для покращення користувацького досвіду.

У статті обґрунтовано необхідність адаптації програмних систем до вимог цифрової трансформації, що є особливо актуальним в умовах жорсткої конкуренції та воєнного стану. Результати дослідження свідчать, що реінжиніринг сприяє значному підвищенню безпеки, гнучкості та продуктивності системи, що є критично важливим для сучасних туристичних сервісів. Впровадження багатофакторної автентифікації, а також інтеграція з CRM і аналітичними системами розширюють функціональні можливості платформи, забезпечуючи її конкурентоспроможність на ринку. Запропонована модель реінжинірингу є ефективним інструментом цифрової трансформації, що дозволяє підвищити якість та надійність туристичних інформаційних систем. Крім того, стаття розглядає питання оптимізації витрат і часу розробки за рахунок використання перевірених стратегій і технологій реінжинірингу, що робить запропонований підхід економічно вигідним та практично застосовним у сучасних умовах розвитку IT-індустрії.

**Ключові слова:** реінжиніринг, програмне забезпечення, цифрова трансформація, архітектура, безпека, адаптивний інтерфейс.

О. YE. OHNIEVA

Candidate of Technical Sciences, Associate Professor,  
Head of the Software Tools and Technologies Department  
Kherson National Technical University  
ORCID: 0000-0001-6206-0285

D. V. ZIARKO

Master's Degree Student at the Software Tools and Technologies Department  
Kherson National Technical University  
ORCID: 0009-0002-7103-2838

## RESEARCH OF THE SOFTWARE REENGINEERING MODELS EFFECTIVENESS FOR THE TOURISM SERVICES DIGITAL TRANSFORMATION

The paper investigates the effectiveness of software reengineering models as a key tool for the digital transformation of tourism information systems. The object of the study is the software of a tourism platform, while the subject is the processes of optimizing its architecture, performance, security, and quality of user service. Special attention is given to

*in-depth analysis of source code, reverse engineering, architecture redesign, and the implementation of multi-layered models using modern technologies such as PHP, MySQL, OAuth, and password hashing. The practical part is based on the reengineering of the Travel-website web application, which includes detailed code analysis, development of an updated architecture, integration with Google OAuth, enhancement of password storage security, and implementation of an adaptive interface to improve user experience.*

*The paper substantiates the necessity of adapting software systems to the requirements of digital transformation, which is especially relevant under conditions of intense competition and martial law. The research results indicate that reengineering significantly improves system security, flexibility, and performance, which is critically important for modern tourism services. The introduction of multi-factor authentication, as well as integration with CRM and analytical systems, expands the platform's functional capabilities, ensuring its competitiveness in the market. The proposed reengineering model is an effective tool for digital transformation, enabling the improvement of quality and reliability of tourism information systems. Furthermore, the article addresses the optimization of development costs and time through the use of proven reengineering strategies and technologies, making the proposed approach economically advantageous and practically applicable in the current IT industry environment.*

**Key words:** reengineering, software, digital transformation, architecture, security, adaptive interface.

### Постановка проблеми

Воєнний стан створює значні виклики для економіки, інфраструктури та соціальної сфери. Тому модернізація та оптимізація інформаційних систем стають критично важливими для підтримки та розвитку туристичної галузі. Реінжиніринг інформаційної платформи для продажу турів має особливе значення для України в умовах воєнного стану, оскільки забезпечує адаптивність і стійкість туристичного бізнесу в кризових умовах.

Реінжиніринг інформаційної платформи виступає стратегічним інструментом, що забезпечує технологічні зміни, підвищення ефективності бізнес-процесів і, як наслідок, поліпшення якості туристичних послуг, що відповідає сучасним вимогам цифрової економіки та очікуванням споживачів. Об'єктом дослідження є програмне забезпечення туристичної платформи, а предметом – процеси реінженерії, спрямовані на оптимізацію архітектури системи, підвищення продуктивності, покращення заходів безпеки та підвищення загальної якості обслуговування користувачів.

В умовах стрімкої цифрової трансформації та зростання вимог до якості інформаційних сервісів у туристичній галузі особливої актуальності набуває питання реінженерії програмного забезпечення (ПЗ). Програмні системи туристичних платформ часто мають застарілу архітектуру, низьку масштабованість та обмежені можливості інтеграції з сучасними сервісами, що ускладнює впровадження нових функцій та підтримку високого рівня безпеки. Тому виникає потреба у застосуванні сучасних моделей реінженерії ПЗ, які дозволяють оптимізувати архітектуру, підвищити продуктивність, забезпечити захист даних та покращити користувацький досвід.

### Аналіз останніх досліджень і публікацій

Українські науковці досліджують особливості реінжинірингу та розробляють концептуальні засади здійснення реінжинірингу бізнес-процесів у програмних системах. Так в праці [1] розглядаються засоби редокументування успадкованого програмного забезпечення, в праці [2] підкреслюється ключова роль інформаційних технологій у реінжинірингу бізнес-процесів. Автором в [3] досліджується генезис реінжинірингу бізнес-процесів як елемент формування ефективної системи управління на промислових підприємствах. Автори [4] аналізують особливості реінжинірингу в організаціях, що розробляють програмні продукти, з акцентом на витрати, переваги та труднощі, а в [5] – розробляють концептуальні засади реінжинірингу бізнес-процесів у програмних системах. В роботі [6] були розглянуті теоретично-методичні основи реінжинірингу бізнес-процесів. Авторами [7] реінжиніринг представляється як інноваційна основа реконструкції системи управління підприємством.

Іноземні науковці також заклали теоретичні та практичні основи реінжинірингу ПЗ, охоплюючи питання модернізації застарілих систем, радикальної перебудови процесів та стратегічних аспектів впровадження інформаційних технологій. Робота [8] описує методи рефакторингу для покращення існуючого коду, в [9] пропонується модель процесу реінжинірингу ПЗ, що поєднує традиційний підхід зі спіральною моделлю, в [10] розглядаються принципи перепроєктування ПЗ для гетерогенних платформ, в [11] досліджується реінжиніринг застарілих систем у напрямку нових технологій, в [12] описується впровадження інновацій у процеси через реінжиніринг роботи за допомогою інформаційних технологій, в [13] надаються практичні поради щодо ефективної роботи з успадкованим кодом. Авторами в праці [14] надають огляд концепцій, моделей і підходів до реінжинірингу ПЗ, в [15] – викладають маніфест революції в бізнесі через реінжиніринг корпорацій, а в [16] – описують метод синтезу специфікацій і вимог у процесі реінжинірингу ІТ-проектів. Автор [17] пропонує модель процесу реінжинірингу ПЗ. Різні аспекти інженерії ПЗ, включно з реінжинірингом розглядаються в [18]. Методи реверсивного інжинірингу описуються в [19].

Дослідники розглядають різноманітні підходи до реінженерії ПЗ: від реверс-інжинірингу до впровадження мікросервісної архітектури, використання CI/CD, DevOps, автоматизованого тестування, а також застосування UML-діаграм для моделювання систем. Проте, недостатньо досліджено питання комплексної реінженерії туристичних платформ із врахуванням специфіки галузі та сучасних вимог до якості ПЗ.

### Формулювання мети дослідження

Обґрунтування ефективної моделі реінжинірингу бізнес-процесів туристичної платформи, спрямованої на підвищення якості обслуговування клієнтів, оптимізацію процесу продажу турів та впровадження сучасних цифрових технологій для покращення взаємодії з користувачами, що досягається за рахунок переходу до тришарової архітектури з MySQL, хешуванням паролів, валідацією даних та захистом від SQL-ін'єкцій.

### Викладення основного матеріалу дослідження

Реінжиніринг у цьому дослідженні передбачав повний цикл: від аналізу вихідного коду до впровадження сучасної архітектури та нових функцій. Реінжиніринг інформаційних платформ для продажу турів є ефективним фактором підвищення якості туристичних послуг через комплексне удосконалення бізнес-процесів, застосування інноваційних технологій і оптимізацію маркетингових комунікацій [1, 4, 6, 10–12, 15].

Для дослідження була обрана власна програма – веб-додаток Travel-website, розміщений у відкритому доступі на GitHub за посиланням: <https://github.com/EienZ/Travel-website>.

Реверсивна інженерія охопила аналіз архітектури, функціоналу та структури бази даних веб-додатку. Результати аналізу веб-додатку Travel-website наведено в табл. 1. Архітектура додатку реалізована за класичною схемою клієнт-сервер: в клієнтській частині браузер користувача взаємодіє із сервером через HTML-форми, в серверній частині Apache з PHP обробляє запити через register.php, login.php та інші скрипти, база даних MySQL зберігає дані користувачів, дані передаються у вигляді POST-запитів, обробляються на сервері, і результат передається назад клієнту у вигляді HTML. Для моделювання нової архітектури було створено UML-діаграми класів та варіантів використання, що дозволило чітко визначити взаємодію між модулями системи.

Таблиця 1

Результати аналізу веб-додатку Travel-website

| Елемент аналізу       | Стан                                                                    |
|-----------------------|-------------------------------------------------------------------------|
| Безпека               | Відсутнє хешування паролів, немає захисту від SQL-ін'єкцій              |
| Документація          | Відсутня                                                                |
| База даних            | Мінімальна, відсутні обмеження на email, немає індексації               |
| Структура коду        | Модулі login і register реалізовані в одному файлі кожен                |
| Валідація даних       | Відсутня перевірка на стороні сервера і клієнта                         |
| Структура проекту     | Простий файл-орієнтований підхід без розділення на логіку/представлення |
| Вихідні тести         | Ручне тестування                                                        |
| Інтерфейс користувача | Присутній базовий HTML, з проведеним ручним тестуванням.                |

В результаті дослідження додатку було прийняте рішення про покращення основних функцій модуля логіну та реєстрації користувачів. Зміни сприяють підвищенню безпеки, зручності та масштабованості системи. Паролі зберігаються в зашифрованому вигляді, що значно покращує захист даних. Реалізація входу через сторонні сервіси, такі як Google, спрощує процес авторизації для користувачів, підвищуючи зручність використання. Оновлена структура бази даних забезпечує гнучкість та можливість легко додавати нові методи авторизації. Всі ці зміни зробили систему більш надійною та зручною для кінцевих користувачів.

Впровадження хешування паролів забезпечує надійний захист збережених даних користувачів, зменшуючи ризик несанкціонованого доступу навіть у разі витоку бази даних. Поєднання хешування з протоколом OAuth дозволяє делегувати аутентифікацію надійним зовнішнім провайдерам (Google, Facebook), що додатково знижує ризики фішингових атак і крадіжки облікових даних. OAuth також забезпечує безпечний обмін інформацією між сервісами без передачі паролів [8–10, 13–14].

Інтеграція входу через Google одним кліком значно спрощує процес реєстрації та авторизації для користувачів, усуваючи необхідність запам'ятовувати додаткові паролі та вводити багато полів форми. Це підвищує конверсію реєстрацій і зменшує кількість незавершених сесій, що позитивно впливає на користувацький досвід і лояльність клієнтів. Крім того, використання соціальних мереж і єдиних акаунтів сприяє швидшому доступу до персоналізованих пропозицій.

Архітектура системи, побудована на стандарті OAuth, дозволяє легко додавати нових провайдерів аутентифікації, таких як Facebook, GitHub, Apple ID тощо, без суттєвих змін у базовому коді платформи. Це забезпечує гнучкість розвитку системи, адаптацію до потреб різних сегментів користувачів і підтримку мультиплатформеності. Масштабованість також включає можливість інтеграції з CRM-системами та аналітичними інструментами для покращення маркетингових комунікацій і персоналізації [8–10, 17].

Після аналізу успадкованої програми (модулів логіну та реєстрації з підтримкою OAuth) виявлено такі основні вимоги до відновленої системи:

– система повинна дозволяти гостеві користувачу: зареєструватися через email/пароль; увійти через email/пароль; увійти через сторонні сервіси (Google, Facebook); відновити пароль через email; після входу користувач отримує доступ до захищених сторінок;

– нефункціональні вимоги: система повинна відповісти на запит входу/реєстрації не пізніше ніж за 2 секунди; інтерфейс має бути адаптивним для мобільних пристроїв; всі паролі мають зберігатися у хешованому вигляді; серверний API – HTTPS із сертифікатом.

Після аналізу моделі вимог було встановлено, що оновлена програма повинна забезпечувати автентифікацію користувачів через email/пароль та сторонні сервіси (Google, Facebook), мати просту навігацію, безпечно зберігання даних та зручний інтерфейс.

Архітектура була трансформована у класичну тришарову модель:

- фронтенд: HTML/CSS/JS-інтерфейс для взаємодії з користувачем;
- бекенд: PHP-скрипти для обробки запитів, логіки автентифікації та інтеграції з OAuth-сервісами;
- БД: MySQL для зберігання облікових записів користувачів, логів і токенів сесій.

В системі реалізовано шифрування паролів, валідацію введених даних, обробку помилок та захист від SQL-ін'єкцій. У ході дослідження було успішно трансформовано архітектуру старої автентифікаційної системи у сучасну реалізацію з використанням HTML, CSS, JavaScript, PHP та MySQL. Вона підтримує як класичний вхід через email/пароль, так і авторизацію через Google та Facebook. Архітектура є масштабованою, безпечною та гнучкою, що дозволяє легко її модифікувати чи розширювати у майбутньому.

Таблиця 2

## Критерії якості та завершення тестування

| Критерій              | Очікувана поведінка                                                  |
|-----------------------|----------------------------------------------------------------------|
| Безпека               | Паролі не зберігаються у відкритому вигляді                          |
| Відсутність збоїв     | При неправильному введенні не відбувається аварійне завершення       |
| Відповідність вимогам | Реалізовано всі сценарії, передбачені моделлю варіантів використання |
| Сумісність            | Код працює у середовищі Nginx + PHP + MySQL                          |
| Розширюваність        | Можливість додати Facebook OAuth або двофакторну автентифікацію      |

Таблиця 3

## Сценарії тестування

| № | Назва сценарію                | Вхідні дані                              | Очікуваний результат                   |
|---|-------------------------------|------------------------------------------|----------------------------------------|
| 1 | Реєстрація нового користувача | Email: test@mail.com, Password: 12345678 | Перенаправлення на сторінку входу      |
| 2 | Вхід з правильними даними     | Email: test@mail.com, Password: 12345678 | Вхід у систему (сесія встановлена)     |
| 3 | Вхід з неправильним паролем   | Email: test@mail.com, Password: wrong    | Повідомлення про помилку входу         |
| 4 | Вхід через Google             | Google акаунт (валідний)                 | Вхід у систему через Google            |
| 5 | SQL-ін'єкція                  | Email: ' OR 1=1 --, Password: anything   | Невдалий вхід, захист спрацював        |
| 6 | Перевірка шифрування паролю   | Перевірка запису в БД                    | Пароль збережено як хеш                |
| 7 | Повторна реєстрація           | Email: вже існуючий                      | Повідомлення про існуючого користувача |
| 8 | Доступ без входу              | Перехід на dashboard без сесії           | Перенаправлення на форму входу         |

Таблиця 4

## Результати тестування

| Сценарій | Статус   | Примітка                                        |
|----------|----------|-------------------------------------------------|
| 1        | Пройдено | Реєстрація працює                               |
| 2        | Пройдено | Вхід з коректними даними                        |
| 3        | Пройдено | Повідомлення про помилку при невірному паролі   |
| 4        | Пройдено | Авторизація через Google успішна                |
| 5        | Пройдено | SQL-ін'єкція заблокована                        |
| 6        | Пройдено | Перевірено: bcrypt-хешування збережено у БД     |
| 7        | Пройдено | Повідомлення про існуючого користувача виведено |
| 8        | Пройдено | Перенаправлення при відсутності сесії працює    |

У процесі перепроєктування програми було здійснено деталізацію компонентів та реалізацію логічної структури, алгоритмів взаємодії й структури розгортання (рис. 1–3). Побудована модель чітко відображають взаємозв'язки між компонентами, підтримують цілісність архітектури.

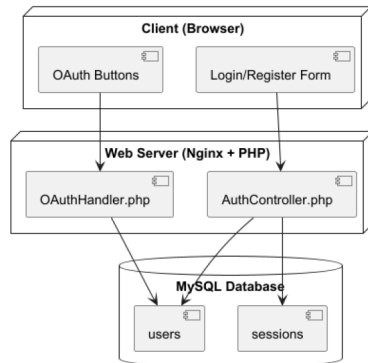


Рис. 1. Модель розгортання

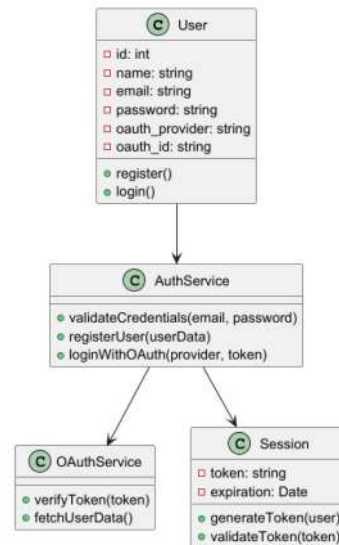


Рис. 2. Структурно-логічна модель (класова діаграма)

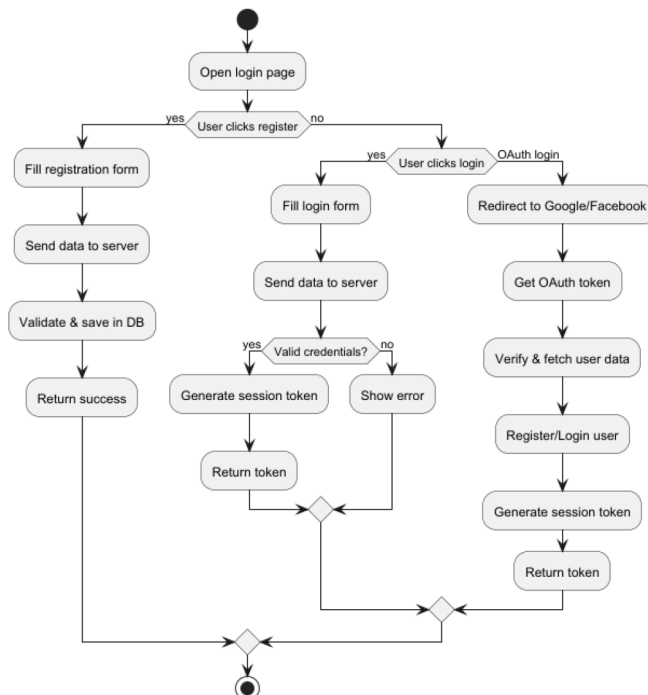


Рис. 3. Процедурна (функціональна) модель

У дослідженні реалізовано оновлення системи логі в додатку туристичного агентства:

- багатофункціональну систему автентифікації;
- шифрування паролів із використанням password\_hash;
- інтеграцію з Google OAuth 2.0;
- MVC-структурування логіки.

Це дозволяє користувачам мати гнучкий та безпечний спосіб входу. Оновлена система автентифікації повністю відповідає заявленим вимогам, є безпечною, стійкою до атак, а також підтримує автентифікацію зі сторонніх ресурсів (табл. 2–4).

Впровадження хешування паролів забезпечує надійний захист збережених даних користувачів, зменшуючи ризик несанкціонованого доступу навіть у разі витоку бази даних. Поєднання хешування з протоколом OAuth дозволяє делегувати автентифікацію надійним зовнішнім провайдерам (Google, Facebook), що додатково знижує ризики фішингових атак і крадіжки облікових даних. OAuth також забезпечує безпечний обмін інформацією між



Рис. 4. Старий дизайн

Рис. 5. Новий дизайн

сервісами без передачі паролів. Паролі тепер шифруються за допомогою password\_hash, який використовує bcrypt для генерації безпечного хешу. Валідація забезпечує очищення даних під час створення.

Схема реінжинірингу [16–19], яка була використана в роботі, може бути описана наступним чином:

1. Реверсивна інженерія (Reverse Engineering):
  - аналіз і документування існуючого функціоналу модулів реєстрації та логіна веб-додатка Travel-website;
  - виявлення архітектурних недоліків (відсутність безпеки, погана структура коду);
  - документування структури бази даних;
  - фіксація наявного інтерфейсу та сценаріїв взаємодії користувача.
2. Аналіз вимог:
  - визначення функціональних та нефункціональних вимог до оновленої системи;
  - функціональні вимоги включали можливість реєстрації/входу через email/пароль та сторонні сервіси (Google, Facebook), відновлення паролю, доступ до захищених сторінок після входу;
  - нефункціональні вимоги включали швидкість відповіді системи, адаптивність інтерфейсу, безпечне зберігання паролів та захищений API (HTTPS).
3. Перепроєктування (Re-design):
  - трансформація архітектури у класичну тришарову модель (фронтенд, бекенд, база даних);
  - використання HTML/CSS/JS для інтерфейсу, PHP для обробки запитів та MySQL для зберігання даних;
  - реалізація шифрування паролів, валідації даних, обробки помилок та захисту від SQL-ін'єкцій.
4. Реалізація (Re-implementation):
  - реалізація нового дизайну інтерфейсу (рис. 5);
  - розробка нового функціоналу автентифікації з підтримкою OAuth;
  - деталізація компонентів, реалізація логічної структури, алгоритмів взаємодії та структури розгортання (рис. 1–3).
5. Тестування:
  - визначення критеріїв якості та завершення тестування (безпека, відсутність збоїв, відповідність вимогам, сумісність, розширюваність);
  - проведення тестування за різними сценаріями (табл. 3);
  - аналіз результатів тестування та виправлення виявлених дефектів (табл. 4).

Ця схема поєднує елементи реверсивної інженерії, аналізу вимог, перепроєктування, реалізації та тестування, що дозволяє комплексно модернізувати застарілий веб-додаток Travel-website та підвищити його безпеку, зручність і масштабованість.

### Висновки

Реінжиніринг веб-додатку Travel-website включав аналіз існуючої архітектури, структури бази даних та функціоналу модулів реєстрації й логіну. Основними напрямками модернізації стали впровадження тришарової архітектури, використання хешування паролів, інтеграція з OAuth для авторизації через Google та підвищення адаптивності інтерфейсу. Це дозволило забезпечити безпечну автентифікацію, гнучкість розширення та інтеграцію з зовнішніми сервісами. Оновлена система відповідає сучасним вимогам до продуктивності, безпеки та зручності для користувачів.

Така модель реінженерії ПЗ для туристичної платформи відповідає сучасним вимогам інженерії програмного забезпечення, забезпечує високу якість, підтримуваність і конкурентоспроможність системи. Результати можуть бути використані для модернізації IT-систем у туристичній галузі та інших сферах.

Перспективи досліджень в даному напрямку у подальшому будуть спрямовані на практичне впровадження та тестування запропонованої моделі реінженерії у різних предметних галузях, зокрема в умовах специфіки українського ринку. Важливим перспективним напрямом є вивчення інтеграції запропонованої моделі з сучасними технологіями штучного інтелекту та машинного навчання для подальшого підвищення інтелектуальності та адаптивності програмних систем.

## Список використаної літератури

1. Авраменко О. А. Архітектура засобів редокументування успадкованого програмного забезпечення. *Вісник НАУ*. 2020. № 3–4 (33). С. 58–62.
2. Гвоздь М. Я. Ключова роль інформаційних технологій під час реінжинірингу бізнес-процесів. *Вісник Національного університету «Львівська політехніка». Серія: Проблеми економіки та управління*. 2015. № 815. С. 202–206. URL: [http://nbuv.gov.ua/UJRN/VNULPP\\_2015\\_815\\_27](http://nbuv.gov.ua/UJRN/VNULPP_2015_815_27)
3. Гриценко Л. Л. Генезис реінжинірингу бізнес-процесів як елемента формування ефективної системи управління в діяльності промислових підприємств. *Механізм регулювання економіки*. 2015. № 2. С. 31–37.
4. Каризська А. Р., Ситник Н. І. Особливості реінжинірингу в організаціях, що займаються розробленням програмного продукту. *Актуальні проблеми економіки та управління: збірник наукових праць молодих вчених*. 2017. Вип. 11. URL: <https://ela.kpi.ua/server/api/core/bitstreams/c5472d16-13b6-4432-89e6-25b716d0b68a/content>
5. Ковальов А. І., Шевченко В. В. Концептуальні засади здійснення реінжинірингу бізнес-процесів. *Вісник соціально економічних досліджень: зб. наук. пр. Одеський держ. екон. ун-т., Одеса*. 2009. Вип. 35. С. 365–371.
6. Продіус О. І. Теоретично-методичні основи реінжинірингу бізнес-процесів. Економіка: реалії часу. *Науковий журнал*. 2016. № 6 (28). С. 79–87. URL: <http://economics.opu.ua/files/archive/2016/n6.html>
7. Прохорова В. В., Божанова О. В., Юрман Я. В. Реінжиніринг як інноваційна основа реконструкції системи управління підприємством. *Проблеми економіки*. 2020. № 4. С. 210–216. URL: <http://jnas.nbuv.gov.ua/article/UJRN-0001219373>
8. Фаулер М. та ін. Рефакторинг. Поліпшення існуючого коду. «Діалектика». 2019. 448 с.
9. Abbas A. S., Jeberson W., Jeberson D. W., Klinsega V. V. Proposed Software Re-engineering Process That Combine Traditional Software Re-engineering Process With Spiral Model. *International Journal of Advanced Research in Computer Science*. 2013. Vol. 4 (2). Pp. 75–83.
10. Andrade H., Berger C., Crnkovic I., Bosch J. Principles for Re-architecting Software for Heterogeneous Platforms. *2020 27th AsiaPacific Software Engineering Conference (APSEC)*. 2020. Pp. 405–414. DOI: 10.1109/apsec51365.2020.00049
11. Bouchiha D. Reengineering Legacy Systems Towards New Technologies. *Encyclopedia of Information Science and Technology, Fifth Edition, IGI Global*. 2020. Pp. 1214–1230. DOI: 10.4018/978-1-7998-3479-3.ch084
12. Davenport T. Process Innovation: Reengineering Work through Information Technology. Harvard Business School Press, Boston. 1993. 337 p.
13. Feathers M. Working Effectively with Legacy Code. Person. 2020. 464 p.
14. Fung L., Tan S. Overview of Software Re-Engineering Concepts, Models and Approaches. *JOIV: International Journal on Informatics Visualization*. 2025. № 9. Pp. 46. DOI:10.62527/joiv.9.1.3034
15. Hammer M., Champy J. Reengineering the Corporation: Manifesto for Business Revolution. Zondervan. 2009. 332 p.
16. Kernytsky O., Kernytsky A., Teslyuk V. The Synthesis Method for Specifications and Requirements in the Process of IT Project Reengineering. *International Scientific and Technical Conference on Computer Sciences and Information Technologies, Institute of Electrical and Electronics Engineers Inc.* 2023. № 5. Pp. 1–8. DOI:10.1109/CSIT61576.2023.10324175
17. Kumar A. Software Re-engineering Process Model. *International Journal of Science and Research*. 2019. Pp. 2319–7064. DOI:10.21275/SR21101152601
18. Sommerville I. Software engineering. Pearson Education. 2007. 792 p.
19. Team N. Reverse Engineering. *Handbook for CTFers, Singapore: Springer Nature Singapore*. 2022. Pp. 295–427. DOI: 10.1007/978-981-19-0336-6\_5

## References

1. Avramenko, O. A. (2020). Architecture of Re-documentation Tools for Legacy Software. *Proceedings of the National Aviation University*, 3–4 (33), 58–62.
2. Hvoz, M. Ia. (2015). Key Role of Information Technology in Business Process Reengineering. *Bulletin of the Lviv Polytechnic National University. Series: Problems of Economics and Management*, 815, 202–206. URL: [http://nbuv.gov.ua/UJRN/VNULPP\\_2015\\_815\\_27](http://nbuv.gov.ua/UJRN/VNULPP_2015_815_27)
3. Hrytsenko, L. L. (2015). The Genesis of Business Process Reengineering as an Element of Forming an Effective Management System in the Activities of Industrial Enterprises. *Mechanism of Economic Regulation*, 2, 31–37.
4. Karyzska, A. R., & Sytnyk, N. I. (2017). Features of Reengineering in Organizations Engaged in Software Development. *Current Issues in Economics and Management: Collection of Scientific Papers by Young Researchers*, 11. URL: <https://ela.kpi.ua/server/api/core/bitstreams/c5472d16-13b6-4432-89e6-25b716d0b68a/content>
5. Kovalov, A. I., & Shevchenko, V. V. (2009). Conceptual Foundations of Business Process Reengineering. *Bulletin of Socio-Economic Research: Collection of Scientific Papers of Odesa State Economic University*. Odesa, 35, 365–371.

6. Prodius, O. I. (2016). Theoretical and Methodological Foundations of Business Process Reengineering. Economics: Realities of the Time. *Academic Journal*, 6 (28), 79–87. URL: <http://economics.opu.ua/files/archive/2016/n6.html>
7. Prokhorova, V. V., Bozhanova, O. V., & Iukhman, Ya. V. (2020). Reengineering as an Innovative Basis for the Reconstruction of Enterprise Management Systems. *Problems of Economics*, 4, 210–216. URL: <http://jnas.nbuv.gov.ua/article/UJRN-0001219373>
8. Fauler, M. (2019). Refactoring. Improving Existing Code, “Dialektyka”, 448.
9. Abbas, A. S., Jeberson, W., Jeberson, D. W., & Klinsega, V. V. (2013). Proposed Software Re-engineering Process That Combine Traditinal Software Re-engineering Process With Spiral Model. *International Journal of Advanced Research in Computer Science*, 4 (2), 75–83.
10. Andrade, H., Berger, C., Crnkovic, I., & Bosch, J. (2020). Principles for Rearchitecting Software for Heterogeneous Platforms. *2020 27th AsiaPacific Software Engineering Conference (APSEC)*, 405–414. DOI: 10.1109/apsec51365.2020.00049
11. Bouchiha, D. (2020). Reengineering Legacy Systems Towards New Technologies. *Encyclopedia of Information Science and Technology, Fifth Edition, IGI Global*, 1214–1230. DOI: 10.4018/978-1-7998-3479-3.ch084
12. Davenport, T. (1993). *Process Innovation: Reengineering Work through Information Technology*. Harvard Business School Press, Boston, 337.
13. Feathers M. (2020) *Working Effectively with Legacy Code*. Person, 464.
14. Fung, L., & Tan, S. (2025). Overview of Software Re-Engineering Concepts, Models and Approaches. *JOIV: International Journal on Informatics Visualization*, 9, 46. DOI:10.62527/joiv.9.1.3034
15. Hammer, M., & Champy, J. (2009). *Reengineering the Corporation: Manifesto for Business Revolution*. Zondervan, 332.
16. Kernytskyy, O., Kernytskyy, A., & Teslyuk, V. (2023). The Synthesis Method for Specifications and Requirements in the Process of IT Project Reengineering. *International Scientific and Technical Conference on Computer Sciences and Information Technologies, Institute of Electrical and Electronics Engineers Inc*, 5, 1–8. DOI:10.1109/CSIT61576.2023.10324175
17. Kumar, A. (2019). Software Re-engineering Process Model. *International Journal of Science and Research*, 2319–7064. DOI:10.21275/SR21101152601
18. Sommerville, I. (2007). *Software engineering*. Pearson Education, 792.
19. Team, N. (2022). Reverse Engineering. *Handbook for CTFers, Singapore: Springer Nature Singapore*, 295–427. DOI: 10.1007/978-981-19-0336-6\_5

Дата першого надходження рукопису до видання: 19.09.2025  
Дата прийнятого до друку рукопису після рецензування: 15.10.2025  
Дата публікації: 28.11.2025