

Технічні науки

УДК 004.415.2

Мілованов Володимир Валентинович

студент

Національного аерокосмічного університету

"Харківський авіаційний інститут"

Milovanov Volodymyr

Student of the

National Aerospace University «Kharkiv Aviation Institute»

ORCID: 0009-0004-9962-5293

АВТОМАТИЗАЦІЯ ПРОЄКТУВАННЯ WEB-ЗАСТОСУНКІВ:

ПРИНЦИПИ, ЕТАПИ, ПЕРСПЕКТИВИ

AUTOMATION OF WEB APPLICATION DESIGN: PRINCIPLES, STAGES, PROSPECTS

***Анотація.** Автоматизоване проєктування веб-застосунків є важливим напрямом у сучасній розробці програмного забезпечення, що дозволяє зменшити час та ресурси, необхідні для створення складних систем. У статті розглядаються основні підходи до автоматизації процесу проєктування веб-застосунків, включаючи використання генеративного програмування, шаблонів архітектури та сучасних інструментів автоматизації. Аналізуються переваги та недоліки автоматизованого підходу порівняно з традиційними методами розробки. Розглянуто питання стандартизації процесу автоматизованого проєктування, що сприяє підвищенню ефективності взаємодії між командами розробників та забезпечує відповідність найкращим практикам галузі.*

Особливу увагу приділено методам автоматизованої генерації коду на основі моделей, що дозволяють мінімізувати людські помилки та підвищити продуктивність команди розробників. Також розглянуто використання штучного інтелекту та машинного навчання для оптимізації процесу проєктування, що дозволяє створювати адаптивні системи, здатні до самоналаштування. Важливим аспектом є забезпечення узгодженості між вимогами замовників і кінцевим результатом, що досягається за допомогою моделей автоматизованого контролю якості та тестування.

У статті пропонуються методи формалізації вимог до веб-застосунків, що дозволяють автоматично генерувати їх архітектурні моделі. Розглядається інтеграція автоматизованих інструментів у процес розробки, що забезпечує гнучкість та масштабованість застосунків. Проведено порівняльний аналіз існуючих підходів до автоматизації та визначено перспективи їх подальшого розвитку. Додатково розглянуто аспекти кібербезпеки та забезпечення відповідності сучасним стандартам захисту даних при використанні автоматизованого підходу до проєктування.

Результати дослідження свідчать про значний потенціал автоматизованого проєктування веб-застосунків у прискоренні процесу розробки, підвищенні якості коду та зменшенні витрат. Автоматизація дозволяє зосередитися на розв'язанні складних завдань, залишаючи рутинні операції алгоритмам, що в перспективі може сприяти розробці ще більш інтелектуальних та самонавчальних систем.

Ключові слова: автоматизоване проєктування, веб-застосунки, генерація коду, архітектура ПЗ, штучний інтелект.

Summary. Automated web application design is a crucial area in modern software development, allowing for a reduction in time and resources required

to create complex systems. This article examines the main approaches to automating the web application design process, including the use of generative programming, architectural templates, and modern automation tools. The advantages and disadvantages of automated approaches compared to traditional development methods are analyzed. The issue of standardizing the automated design process is considered, contributing to increased efficiency in developer team collaboration and ensuring compliance with industry best practices.

Special attention is given to methods of automated code generation based on models, which help minimize human errors and improve developer team productivity. The use of artificial intelligence and machine learning for design process optimization is also explored, allowing for the creation of adaptive systems capable of self-adjustment. An important aspect is ensuring alignment between customer requirements and the final product, achieved through automated quality control and testing models.

The article proposes methods for formalizing web application requirements, enabling the automatic generation of their architectural models. The integration of automated tools into the development process is discussed, ensuring flexibility and scalability of applications. A comparative analysis of existing automation approaches is conducted, and perspectives for their further development are identified. Additionally, cybersecurity aspects and compliance with modern data protection standards when using automated design approaches are examined.

The research results indicate a significant potential for automated web application design in accelerating development processes, improving code quality, and reducing costs. Automation enables developers to focus on solving complex tasks, leaving routine operations to algorithms, which in the long term may contribute to the development of even more intelligent and self-learning systems.

Key words: *automated design, web applications, code generation, software architecture, artificial intelligence.*

Постановка проблеми. Стрімкий розвиток веб-технологій та зростаюча складність веб-застосунків створюють значні виклики для розробників. Традиційні методи розробки часто вимагають значних витрат часу та ресурсів, що призводить до неефективності та непослідовності у процесі проєктування програмного забезпечення. Оскільки веб-застосунки стають дедалі функціональнішими та динамічнішими, зростає потреба у більш ефективних і точних підходах до проєктування, зокрема автоматизованих, які дозволяють мінімізувати людський фактор і знизити ймовірність помилок. Автоматизація проєктування дає змогу не лише прискорити процес розробки, а й забезпечити вищий рівень підтримуваності та масштабованості проєктів завдяки чітко формалізованим алгоритмам і шаблонам, що позитивно впливає на якість кінцевого продукту.

Попри переваги автоматизації, багато сучасних інструментів і методологій не мають достатньої гнучкості для адаптації до змінних вимог проєкту. Забезпечення безперебійної інтеграції між автоматизованими процесами проєктування та різними середовищами розробки залишається складним завданням. Крім того, необхідно ретельно збалансувати автоматизацію та контроль з боку розробників, щоб зберегти високу якість програмного забезпечення при зменшенні ручної роботи.

Аналіз останніх досліджень і публікацій. Останні дослідження у сфері автоматизованого проєктування веб-застосунків вказують на суттєві покращення ефективності розробки та якості програмного забезпечення завдяки впровадженню сучасних технологій. Дослідження автоматизації окремих елементів розробки та тестування проводили Івануса А. І., Ткачук Р. Л., Брич Т. Б., Балацька В. С., Ткаченко А. М. та інші науковці [3]. Проте більшість сучасних досліджень спрямовані не стільки на

проєктування, скільки на тестування веб-застосунків, автоматизований пошук вразливостей тощо, тобто на автоматизацію роботи з готовими Web-застосунками, а не їх проєктування. Сергій Теленик та інші [5] розробили теоретичну основу для формалізації вимог проєктування та розробки веб-застосунків. Дослідженням методів оптимізації навантажень на хмарні сервіси при використанні веб-застосунків займались Р.М. Минайленко, В.А. Резніченко, О.К. Коноплицька-Слободенюк, Л.І. Поліщук, О.А. Бабенко [1; 2; 4] та інші фахівці, проте вони розглядали саме автоматизацію окремих елементів, а не повноцінну автоматизовану розробку або проєктування. Тому актуальним є дослідження сучасних систем та технології автоматизації проєктування в цілому і застосування методик автоматизованого створення окремих елементів застосунку.

Метою статті є дослідження та аналіз сучасних методів автоматизованого проєктування веб-застосунків, формулювання загальних принципів методології автоматизації, а також визначення критеріїв ефективності таких підходів.

Виклад основного матеріалу. Проєктування — це процес створення і опису майбутньої системи, що враховує її функціональні цілі, структуру, взаємодію елементів і середовище експлуатації. У загальному вигляді проєктування охоплює перехід від вимог до рішень, що їх задовольняють.

Проєктування програмного забезпечення — це конкретизація цього процесу в ІТ-сфері. Воно включає вибір архітектури, визначення алгоритмів, побудову інтерфейсів та підготовку до подальшої реалізації і підтримки програмного продукту.

Проєктування Web-застосунків — це підмножина проєктування ПЗ, яка орієнтується на створення інтерактивних, розподілених систем з клієнт-серверною архітектурою, які функціонують у середовищі браузера.

Воно враховує специфіку фронтенду, бекенду, мережевої взаємодії, хостингу, масштабування, та супроводу.

Шевель, Крицький, Попов подають процес проєктування як побудову моделі з трьох описів(F/M/N) [7].

Для системного охоплення всіх аспектів Web-проєктування доцільно використовувати модель із чотирьох взаємопов’язаних описів (F/M/N/E): функціонального, математичного (алгоритмічного), фізичного та експлуатаційного.

Швидке зростання складності Web-додатків вимагає впровадження методів, які зменшують витрати часу та ресурсів. Традиційні підходи до розробки часто є неефективними через велику кількість рутинних завдань, що ускладнює як проєктування, так і реалізацію застосунків. Автоматизація саме процесу проєктування дає можливість спростити етапи планування архітектури, визначення функціональних характеристик та структурування майбутнього додатку за допомогою генерації коду, шаблонізації й візуального моделювання. Це дозволяє оптимізувати створення програмного продукту загалом, оскільки чітке та ефективне проєктування є передумовою успішної технічної реалізації.

Проєктування Web-застосунку доцільно розглядати як процес побудови його повного опису, що включає чотири взаємопов’язані складові: функціональний (F-), математичний (M-), фізичний (N-) та експлуатаційний (E-) образи. Такий підхід дозволяє структуровано охопити всі етапи створення застосунку — від формулювання функцій до забезпечення його підтримки в умовах реального використання(рис.1):

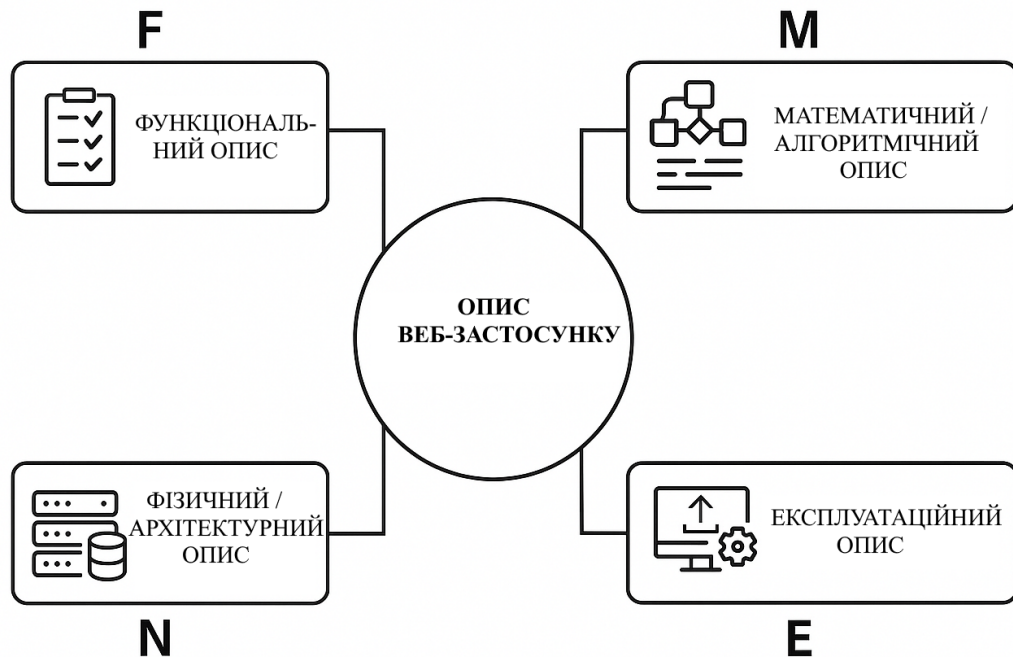


Рис.1. Складові опису Web-застосунку

Джерело: авторська розробка

- F-опис (функціональний) — опис набору функцій, які має виконувати Web-застосунок відповідно до вимог користувача й цілей системи.
- M-опис (математичний або алгоритмічний) — формалізація функцій у вигляді алгоритмів, моделей і структур даних, реалізованих за допомогою коду.
- N-опис (фізичний або технічний) — опис елементів архітектури, інтерфейсів, структур баз даних, що реалізують функції на технічному рівні.
- E-опис (експлуатаційний) — опис механізмів розгортання, супроводу, моніторингу, CI/CD-процедур, а також забезпечення безперервної роботи застосунку в експлуатаційному середовищі.

Усі чотири типи описів тісно пов’язані між собою і мають бути враховані вже на етапі автоматизованого проєктування. Це дозволяє уникнути невідповідностей між архітектурою, реалізацією та подальшим розгортанням Web-застосунку, забезпечуючи підвищену якість, адаптивність і підтримуваність розробленого продукту.

Таким чином, головне завдання автоматизованого проєктування полягає у спрощенні та стандартизації етапу планування та моделювання Web-додатків, що суттєво впливає на швидкість і якість їх подальшого створення. Для успішного використання автоматизованих методів необхідно чітко визначити їхні принципи та обмеження. Важливими характеристиками автоматизованого проєктування є гнучкість, масштабованість і підтримуваність системи. Визначення основних складових саме проєктної діяльності дозволяє ефективно структурувати процес розробки та забезпечує кращу адаптивність створеного застосунку до майбутніх змін.

Перший етап автоматизованого проєктування полягає у визначенні змісту майбутнього Web-застосунку на основі попередньо сформованих специфікацій. На цьому етапі формується F-опис Web-застосунку — визначається загальна функціональна мета, набір підфункцій, що мають бути реалізовані. На основі вимог користувачів та бізнес-логіки уточнюються функції, які надалі будуть формалізовані у вигляді M-опису (алгоритмів) і реалізовані у N-описі (архітектурі). Також варто вже на цьому етапі передбачити загальні вимоги до E-опису — очікуваний рівень доступності, масштабованості та супроводу. Отримані специфікації стають вихідною інформацією для безпосереднього проєктування, під час якого здійснюється визначення архітектури, структури інтерфейсу, моделі даних та основних алгоритмів роботи застосунку. При цьому частина характеристик є універсальною для більшості Web-застосунків (наприклад, компоненти користувацького інтерфейсу), а деякі – залежать

виключно від специфіки проєкту (зокрема, структура бази даних). Автоматизоване проєктування використовує шаблони, побудовані на базі отриманих специфікацій, що дозволяє стандартизувати та прискорити безпосередньо процес технічного проєктування. Завдяки цьому забезпечується ефективне структурування роботи, знижується ризик пропусків і помилок на подальших етапах, а також зменшується ймовірність неточностей при переході від специфікацій до технічної реалізації Web-додатків.

Другий етап автоматизованого проєктування Web-застосунків – архітектурне проєктування, яке передбачає створення узагальненої структури майбутнього програмного продукту. Архітектура програмного продукту — це сукупність ключових рішень щодо організації підсистем, способів їхньої взаємодії, технологій, які використовуються для побудови логіки та інтерфейсів. У контексті автоматизованого проєктування архітектура розглядається як опис структури системи через компоненти та зв’язки між ними. Такі зв’язки охоплюють взаємодії між модулями, виклики API, залежності між сервісами та напрямки передачі даних. Отже, проєктування архітектури — це, насамперед, проєктування взаємозв’язків між функціональними частинами застосунку, що забезпечують виконання його цілісної логіки.

Архітектурне проєктування реалізує M-опис та N-опис Web-застосунку. На основі сформованих функцій (F-опису) визначаються алгоритми та компоненти, які реалізовуватимуть ці функції. Вибір архітектурної моделі (наприклад, MVC чи мікросервіси) формує основу N-опису — технічної реалізації застосунку. Одночасно формуються базові вимоги до E-опису, зокрема інтеграція з DevOps-процесами. Саме на цьому етапі приймаються рішення щодо вибору серверних і клієнтських технологій, визначаються шаблони проєктування (наприклад, MVC-

модель чи мікросервіси) та формалізуються правила взаємодії компонентів застосунку.

Автоматизовані інструменти значно спрощують архітектурне проєктування, дозволяючи автоматично генерувати основні структурні елементи Web-застосунків. Наприклад, фреймворки на основі MVC-моделі або мікросервісна архітектура можуть бути створені за допомогою спеціалізованих генераторів коду, які використовують заздалегідь визначені шаблони. Вибір правильної архітектури є критично важливим, оскільки він безпосередньо впливає на масштабованість, продуктивність і підтримуваність Web-застосунку. Автоматизоване архітектурне моделювання дозволяє мінімізувати ризики, пов'язані з помилками під час створення складних систем. Грамотно спроектована архітектура забезпечує зменшення витрат на підтримку застосунку та покращує його стабільність.

Третій етап автоматизованого проєктування передбачає реалізацію та тестування компонентів Web-застосунку відповідно до архітектури, визначеної на попередньому етапі. На цьому етапі матеріалізується М-опис (у вигляді коду, логіки, моделей даних) та N-опис (впровадження компонентів, інтерфейсів, баз даних). Функціональні вимоги (F-опис) трансформуються у готові модулі. Також здійснюється підготовка до E-опису: створення автотестів, інструментів моніторингу та розгортання, які забезпечують підтримку в експлуатації.

Автоматизоване модульне тестування дозволяє перевіряти відповідність реалізованих компонентів заданим специфікаціям, виявляючи та виправляючи помилки ще до етапу інтеграції компонентів у цілісну систему. Це значно знижує ризики виникнення критичних дефектів у фінальній версії Web-застосунку. Крім того, автоматичні інструменти для аналізу коду та швидкого виправлення помилок забезпечують стабільність роботи застосунку та високу якість кінцевого

продукту. Таким чином, створення й тестування компонентів є логічним продовженням попереднього архітектурного проєктування та нерозривно пов'язані з його результатами.

Четвертий етап автоматизованого проєктування передбачає підготовку Web-застосунку до майбутнього розгортання та подальшого супроводу. Цей етап відповідає за формування Е-опису Web-застосунку — його експлуатаційної частини. Проєктуються засоби CI/CD, масштабування, моніторингу, що забезпечують безперебійну роботу застосунку. Водночас зберігається узгодженість з N-описом (технічною архітектурою) та з алгоритмічною логікою, закладеною в М-описі. Підтримка функцій користувача (F-опис) відбувається шляхом стабільної роботи й оперативного оновлення компонентів. Хоча саме розгортання й підтримка не належать безпосередньо до проєктування, на етапі проєктування необхідно передбачити відповідні механізми, інструменти й стандарти, які дозволять автоматизувати ці процеси в майбутньому. Вибір конкретних DevOps-інструментів і визначення процедур Continuous Integration/Continuous Deployment здійснюються саме під час проєктування, що суттєво впливає на подальший супровід.

Таким чином, хоча впровадження, моніторинг та підтримка є окремими етапами життєвого циклу програмного продукту, саме на етапі автоматизованого проєктування необхідно передбачити архітектурні та технологічні рішення, які забезпечать ефективність цих процесів. Це дає змогу мінімізувати ризики, пов'язані з розгортанням і подальшою експлуатацією Web-застосунку, оптимізувати витрати часу й ресурсів розробників, а також забезпечити довгострокову стабільність та масштабованість програмного продукту.

Методологія автоматизованого проєктування Web-застосунків базується на чіткій структуризації процесу проєктування, що передбачає послідовне визначення специфікацій, архітектури, основних компонентів

та механізмів їх взаємодії. Ця методологія передбачає використання сучасних технологій і автоматизованих інструментів для спрощення процесу проєктування, що є важливою передумовою ефективної технічної реалізації застосунків. Завдяки такому підходу скорочуються витрати часу на створення архітектури, мінімізуються ризики помилок та підвищується якість підготовленої до реалізації технічної документації, яка потім використовується на етапі розробки.

Ефективність автоматизованого проєктування значною мірою залежить від коректного вибору відповідних інструментів, врахування особливостей майбутнього Web-застосунку та стратегічного планування етапів проєктної діяльності. Подальший розвиток методології автоматизованого проєктування спрямований на інтеграцію штучного інтелекту для оптимізації архітектурних рішень, впровадження автоматизованого аналізу продуктивності вже на етапі проєктування, а також глибшу адаптацію проєктних рішень до потреб кінцевих користувачів. Удосконалення проєктних процесів дозволить компаніям, що займаються розробкою Web-застосунків, створювати продукти з високим рівнем конкурентоспроможності завдяки більш якісній архітектурі та зниженню ризиків на етапі технічної реалізації.

Не менш важливим аспектом автоматизованого проєктування Web-застосунків є створення проєктних рішень щодо автоматизованих тестів, які надалі дозволять ефективно перевіряти працездатність застосунку. На етапі проєктування формуються тестові сценарії, визначаються необхідні фреймворки (наприклад, Selenium, Cypress або Jest) і закладаються принципи організації функціональних, регресійних та інтеграційних тестів. Грамотно спроектовані тестові сценарії забезпечують можливість раннього виявлення потенційних помилок, що значно полегшує їх виправлення вже під час розробки.

Проектування автоматизованих тестів передбачає також визначення механізмів їх інтеграції у процеси CI/CD, що дозволяє надалі здійснювати постійний контроль якості створюваного програмного продукту. Водночас ефективність такого підходу залежить від чіткої структуризації тестових сценаріїв, ретельного планування їх використання та забезпечення актуальності в умовах змін архітектури чи функціоналу. Таким чином, саме ретельне проектування автоматизованих тестових сценаріїв є важливою умовою успішного впровадження автоматизації процесу тестування на подальших етапах життєвого циклу Web-застосунку.

Важливу роль у методології автоматизованого проектування Web-застосунків відіграють штучний інтелект (AI) та машинне навчання, які допомагають у прийнятті ефективних проєктних рішень. Інструменти на основі AI можуть аналізувати специфікації та запропонувати оптимальні варіанти архітектури, структури даних і вибору технологій відповідно до поставлених вимог. Зокрема, штучний інтелект дозволяє прогнозувати потенційні слабкі місця в архітектурі, автоматично аналізувати якість проєктних рішень і попереджати виникнення помилок ще до початку етапу реалізації програмного продукту. Це забезпечує підвищення продуктивності та стабільності Web-застосунку вже на етапі проектування.

Крім того, алгоритми машинного навчання можуть бути використані для автоматизованого аналізу поведінки потенційних користувачів, що дозволяє краще спроектувати інтерфейс і логіку взаємодії з Web-застосунком. Це відкриває нові можливості для створення більш персоналізованих, зручних і адаптивних інтерфейсів, які відповідають очікуванням цільової аудиторії. Водночас, важливо ретельно налаштовувати й контролювати використання AI-інструментів у процесі проектування, щоб уникнути некоректних або неефективних рішень. Таким чином, штучний інтелект безпосередньо впливає на якість

проектних рішень та дозволяє суттєво підвищити ефективність і точність процесу автоматизованого проектування Web-застосунків.

Автоматизоване проектування Web-застосунків ґрунтується на певних принципах, які дозволяють стандартизувати та спростити процес розробки. Вони забезпечують узгодженість між різними етапами життєвого циклу програмного забезпечення та сприяють ефективному використанню ресурсів. Застосування цих принципів допомагає зробити розробку гнучкішою, забезпечує можливість масштабування системи та покращує якість кінцевого продукту. Основна мета формулювання таких принципів полягає у визначенні ключових підходів, що дозволяють досягти максимальної ефективності та стабільності. Нижче у таблиці 1 наведено основні принципи автоматизованого проектування Web-застосунків, що напряду пов'язані з реалізацією відповідних аспектів F-, M-, N- та E-описів Web-застосунку.

Таблиця 1

Принципи методології автоматизації проектування

Принцип	Опис
Модульність	Поділ системи на окремі автономні компоненти, що можуть розроблятися, тестуватися та оновлюватися незалежно один від одного.
Стандартизація	Використання загальноприйнятих форматів та архітектурних шаблонів для забезпечення сумісності компонентів та їхньої масштабованості.
Гнучкість	Можливість адаптації до змін у вимогах проекту завдяки використанню параметризованих конфігурацій та модульних підходів.
Автоматизація тестування	Інтеграція тестових фреймворків для безперервного контролю якості та виявлення помилок на ранніх етапах розробки.
Оптимізація ресурсів	Використання ефективних алгоритмів і розподілених обчислень для зменшення навантаження на серверну та клієнтську частини застосунку.
Безперервна інтеграція та розгортання (CI/CD)	Впровадження автоматизованих процесів доставки оновлень для зменшення часу між розробкою нових функцій та їхнім впровадженням.
Безпека	Врахування загроз безпеці на всіх етапах проектування, зокрема шифрування даних, автентифікація користувачів та захист від атак.

Формулювання цих принципів дозволяє забезпечити узгодженість між різними етапами автоматизованого проєктування Web-застосунків. Вони є основою для ефективної організації процесу розробки, що сприяє прискоренню впровадження нових функцій, зменшенню кількості помилок та підвищенню продуктивності системи. Особливу увагу слід приділяти питанням безпеки та стандартизації, оскільки вони забезпечують довготривалу підтримку та масштабованість розроблених рішень. Дотримання цих принципів сприяє створенню надійних Web-застосунків, які можуть адаптуватися до змінних умов ринку та вимог користувачів.

Для оцінки ефективності автоматизованого проєктування необхідно визначити критерії, що дозволять порівнювати різні методи та технології. Визначення критеріїв ефективності є ключовим для вибору оптимальних підходів до автоматизації та їхнього впровадження в практику. Вони дають змогу аналізувати якість, продуктивність і адаптивність розроблених систем, що є важливими факторами для прийняття управлінських рішень. У таблиці 2 наведено основні критерії ефективності автоматизованого проєктування Web-застосунків, які використовуються для об'єктивної оцінки процесу розробки та його результатів.

Таблиця 2

Критерії ефективності автоматизованого проєктування

Критерій	Показник ефективності
Швидкість розробки	Скорочення часу на створення прототипу, запуск MVP та вихід на ринок.
Якість коду	Мінімізація помилок у коді, відповідність стандартам програмування.
Гнучкість	Можливість внесення змін у функціональність без значних переробок архітектури.
Масштабованість	Простота розширення системи без втрати продуктивності.
Продуктивність	Мінімізація затримок при обробці запитів та оптимальне використання ресурсів.
Автоматизоване тестування	Частка тестів, що виконуються без ручного втручання, їхня ефективність у виявленні помилок.

Критерій	Показник ефективності
Витрати на підтримку	Скорочення витрат на усунення дефектів та оновлення застосунку після релізу.

Запропоновані критерії дозволяють оцінити ефективність автоматизованого проєктування Web-застосунків через кількісні та якісні показники, які відображають вплив автоматизації саме на етапі проєктування щодо подальших результатів розробки програмного продукту. Під ефективністю у даному випадку розуміють ступінь досягнення поставлених цілей (скорочення часу, підвищення якості, зменшення витрат) завдяки використанню автоматизованих інструментів та методів проєктування порівняно з традиційними підходами. Наприклад, якщо зафіксовано суттєве скорочення часу на створення прототипу, зростання частки автоматичних тестів чи зниження кількості помилок у коді після реалізації проєкту, то це свідчить про високу ефективність застосованого автоматизованого підходу. Таким чином, ці критерії допомагають зробити об'єктивні висновки про доцільність і успішність впровадження автоматизованого проєктування у конкретному Web-проєкті.

Застосування наведених критеріїв дозволяє здійснювати об'єктивний аналіз ефективності автоматизованих методів розробки. Висока швидкість розробки дає змогу швидше виводити продукт на ринок, що особливо важливо для конкурентного середовища. Водночас, якість коду та автоматизація тестування впливають на стабільність і надійність застосунку, що є визначальним фактором для його довгострокової підтримки. Дотримання цих критеріїв допомагає оптимізувати розробку Web-застосунків, покращити їхню продуктивність та зробити систему зручнішою для користувачів.

Незважаючи на переваги, автоматизація проєктування Web-застосунків має і певні виклики. Одним із них є складність інтеграції

автоматизованих інструментів у вже існуючі системи. Багато компаній стикаються з проблемами сумісності нових технологій із застарілим програмним забезпеченням. Крім того, автоматизовані інструменти вимагають відповідної кваліфікації розробників, що може сповільнити їхнє впровадження. Одним із можливих рішень є поетапний перехід на автоматизовані процеси із залученням відповідного навчання персоналу. Це допоможе мінімізувати ризики та підвищити ефективність автоматизації. Збалансований підхід до впровадження нових технологій є запорукою успішного їх використання.

Перспективи автоматизованого проєктування Web-застосунків включають подальше підвищення рівня автоматизації. Використання хмарних платформ, AI та DevOps-підходів дозволить ще більше оптимізувати процеси розробки. Крім того, зростає популярність безсерверних (serverless) технологій, які дозволяють автоматизувати управління інфраструктурою Web-застосунків. Це сприяє зниженню витрат на підтримку серверів та підвищенню масштабованості систем. У майбутньому можна очікувати розвиток інтелектуальних систем аналізу безпеки та продуктивності, які ще більше спростять роботу розробників. Таким чином, автоматизація відіграватиме ключову роль у формуванні майбутнього програмної інженерії.

Загалом, методологія автоматизованого проєктування Web-застосунків дозволяє значно підвищити ефективність розробки та якість кінцевого продукту. Використання сучасних інструментів, таких як генератори коду, AI-асистенти та автоматизоване тестування, дає змогу прискорити життєвий цикл програмного забезпечення. Водночас, впровадження автоматизації вимагає ретельного планування, навчання персоналу та поступового переходу на нові технології. Незважаючи на виклики, автоматизоване проєктування стає стандартом у сучасній розробці Web-застосунків. Успішне впровадження цих методів

забезпечить конкурентоспроможність компаній та високий рівень задоволеності користувачів.

Висновки. Під проєктуванням Web-застосунку доцільно розуміти побудову чотирьох взаємопов’язаних описів (F — функціонального, M — математичного, N — фізичного, E — експлуатаційного). В ході проєктування необхідно виконати низку процедур: визначити зміст і функції застосунку, спроектувати архітектуру та компоненти, створити тестові сценарії, а також передбачити механізми автоматичного розгортання, моніторингу та підтримки.

При виконанні цих процедур у «ручному» режимі виникають проблеми, пов’язані з великими витратами часу та ресурсів, складністю стандартизації рішень, частими помилками та низькою підтримуваністю проєктів. Для підвищення ефективності доцільно автоматизувати найбільш рутинні та трудомісткі процеси проєктування, такі як генерація базової архітектури, створення стандартних компонентів, автоматичне формування шаблонів коду, а також проведення початкового тестування компонентів. Водночас, стратегічні рішення щодо архітектурного вибору, адаптації інтерфейсу та прийняття складних проєктних рішень доцільніше залишати для «ручного» опрацювання.

Автоматизоване проєктування Web-застосунку можна розглядати як процес побудови чотирьох взаємопов’язаних образів (F, M, N, E), що дозволяє системно організувати як створення, так і подальшу підтримку цифрового продукту. Упорядковане проєктування за цією схемою забезпечує узгодженість між вимогами користувачів, реалізацією функціоналу, технічною структурою та інфраструктурною підтримкою.

Для автоматизації процедур проєктування ефективно використовувати моделі на основі шаблонів (MVC, мікросервісна архітектура), а також методи візуального моделювання, генерації коду та автоматизованого тестування. Запропонована модель автоматизованого

проєктування Web-застосунків має враховувати послідовність визначення специфікацій, формування архітектури, створення тестових сценаріїв та їхньої інтеграції з CI/CD. Використання такої моделі дозволяє оцінювати якість майбутнього застосунку на основі конкретних критеріїв (швидкість розробки, гнучкість, масштабованість, продуктивність та витрати на підтримку), що дозволяє заздалегідь прогнозувати ефективність проєктних рішень.

Подальший розвиток методологій автоматизованого проєктування Web-застосунків буде спрямований на глибшу інтеграцію штучного інтелекту, хмарних технологій та DevOps-підходів. Використання AI-асистентів, інтелектуального аналізу продуктивності та автоматизованого управління інфраструктурою сприятиме ще більшій оптимізації процесів. Окрім цього, безсерверні (serverless) архітектури дозволять мінімізувати витрати на управління серверами та забезпечать високу масштабованість. Разом із цим, важливо враховувати виклики, пов'язані з безпекою та підтримкою автоматизованих рішень, що потребує ретельного підходу до їхнього проєктування. Впровадження нових технологій повинно супроводжуватися поступовими змінами у процесах розробки, навчанням персоналу та тестуванням ефективності автоматизованих методів. Таким чином, автоматизоване проєктування продовжуватиме розвиватися, забезпечуючи ефективніші та більш адаптивні Web-застосунки для сучасного цифрового світу.

У межах автоматизованого проєктування вже існують ефективні засоби для реалізації частини задач функціонального, математичного та експлуатаційного описів: наприклад, шаблони інтерфейсів, генератори коду, CI/CD платформи. Незважаючи на окреслену модель автоматизованого проєктування, залишаються відкритими низка проблем, які потребують подальшого вивчення. Зокрема, складність адаптації шаблонів до специфічних вимог різних доменів, інтеграція

автоматизованих засобів із застарілими системами, а також забезпечення гнучкості при змінах архітектури або бізнес-вимог. Не менш важливими є питання автоматизованого врахування вимог безпеки вже на етапі проєктування та адаптації архітектури до нестабільних навантажень у хмарному середовищі. Водночас значна частина задач — особливо в частині архітектурного планування (N-опису) та адаптивної підтримки в умовах зміни вимог — потребують подальшої автоматизації. Ці аспекти є пріоритетними напрямками майбутніх досліджень.

Література

1. Бабенко О.А., Гіоргізова-Гай В.Ш. Балансування навантаження «хмарних» додатків. *Системний аналіз та інформаційні технології: матеріали 17-й Міжнародної науково-технічної конференції SAIT 2015*, Київ, 22-25 червня 2015 р. 2015. С. 184.
2. Захарова І. В., Філіпова Л.Я. Основи інформаційно-аналітичної діяльності. К.: ЦУЛ, 2013. 336 с.
3. Івануса А. І., Ткачук Р. Л., Брич Т. Б., Балацька В. С., Ткаченко А. М. Методи та моделі проєктування системи автоматизованого пошуку вразливостей у веб-додатках. *Вісник Львівського державного університету безпеки життєдіяльності*. 2024. № 30. С. 110–122.
4. Ільченко М.Ю., Кравчук С.О. Телекомунікаційні системи. Київ: Наукова думка, 2017. 730 с.
5. Telenyk S., Nowakowski G., Zharikov E., Vovk Y. Інформаційна технологія для розробки та впровадження web-застосунків. *Адаптивні системи автоматичного управління: міжвідомчий науково-технічний збірник*. 2019. № 1(34). <https://doi.org/10.20535/1560-8956.1.2019.178242>.
6. Mitchell R. Web Scraping with Python. Boston: O'Reilly Media, 2015. 255 с.

7. Shevel V., Kritskiy D., Popov O. Toward Building a Functional Image of the Design Object in CAD. *Computation*. 2022. 10. 134. <https://doi.org/10.3390/computation10080134>.

8. Balsam S., Mishra D. Web application testing—Challenges and opportunities. *Journal of Systems and Software*. 2025. Vol. 219. <https://doi.org/10.1016/j.jss.2024.112186>.