УДК 004.45
DOI https://doi.org/10.15589/znp2025.1(499).20

ANALYSIS AND FORMALIZATION OF CRITERIA FOR SELECTING A SOFTWARE DEVELOPMENT ECOSYSTEM

АНАЛІЗ І ФОРМАЛІЗАЦІЯ КРИТЕРІЇВ ДЛЯ ВИБОРУ ЕКОСИСТЕМИ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Roman V. Olijnyk
oliynykroma@gmail.com
ORCID: 0009-0008-4967-4926
Vasyl S. Kostyrko
vkostyrko@lute.lviv.ua
ORCID: 0000-0002-6366-8695

Р. В. Олійник,
канд. техн. наук, доцент
В. С. Костирко,
канд. ф-м. наук, доцент

Lviv University of Trade and Economics, Lviv
Львівський торговельно-економічний університет, м. Львів

Abstract. Modern software development requires careful selection of tools for creating a software product, and must consider criteria such as functionality, performance, scalability, and security. *The purpose* of this article is to investigate the criteria for selecting tools, platforms, and technologies that enable the transition from choosing limited infrastructure to selecting a comprehensive ecosystem for developing software products. *The research methodology* involves the examination of various tools and frameworks, particularly for web and mobile applications, such as React Native, Flutter, and Ionic, focusing on the characteristics that determine platform selection (performance, access to device features, development tool support, etc.). It is shown that the choice of infrastructure significantly impacts the success of projects, as technical limitations can lead to additional costs and reduced development efficiency. *The scientific novelty* of the research lies in the systematization of the criteria for selecting a software development ecosystem, which contributes to the effectiveness of creating software solutions capable of meeting the requirements of the developed product. For the first time, the rationale for choosing an ecosystem is proposed not only based on technological characteristics but also considering the needs of teams, their experience, and development costs. *The practical significance* of the research is that the developed recommendations allow professionals in the field of software engineering to optimize the selection of ecosystems for software development, taking into account the specifics of projects and requirements for the final product. The results can be used as a guideline for startups and companies engaged in mobile development to select the highest quality tools and platforms, as well as to effectively manage the development process while maximizing resource conservation. In summary, the selection of an ecosystem for software development is a complex process that takes into account technical, business, and user requirements, and the formalization of platform and tool selection criteria is crucial for ensuring development efficiency and achieving optimal results in projects.
Key words: framework; ecosystem; development criteria; platform; Ionic; Native; Web App; Hybrid.

Анотація. Сучасна розробка програмного забезпечення вимагає ретельного підбору інструментів для створення програмного продукту, та повинна враховувати такі критерії, як функціональність, продуктивність, масштабованість та безпеку. *Мета* цієї статті полягає в дослідженні критеріїв вибору інструментів, платформ і технологій, що дозволяє здійснити перехід від вибору обмеженої інфррастурктури до вибору комплексної екосистеми розробки програмних продуктів. *Методика дослідження* передбачає розгляд різноманітних інструментів і фреймворків, зокрема для web– і мобільних застосунків: React Native, Flutter та Ionic, з фокусом на характеристики, що визначають вибір поатформи (швидкодія, доступ до функцій пристроїв, підтримка інструментів розробки тощо.). Показано, що вибір інфраструктури має великий вплив на успішність проєктів, оскільки технічні обмеження можуть призвести до додаткових витрат і зниження ефективності розробки. *Наукова новизна* дослідження полягає у систематизації критеріїв вибору екосистеми розробки програмного забезпечення, що сприяє ефективності створення програмних рішень, здатних задовольнити вимоги розроблюваного прогармного продукту. Вперше запропоновано обґрунтування вибору екосистеми не лише на основі технологічних характеристик, але й з огляду на потреби команд, їх досвід і витрати на розробку. *Практична значимість* дослі-

дження полягає в тому, що розроблені рекомендації дозволяють фахівцям у галузі програмної інженерії оптимізувати вибір екосистеми для розробки програмного забезпечення, враховуючи специфіку проектів і вимоги до кінцевого продукту. Результати можуть бути використані як орієнтир для стартапів і компаній, що займаються мобільною розробкою, для вибору найбільш якісних інструментів і платформ, а також для ефективного управління процесом розробки з максимальним збереженням ресурсів. Узагальнення вибору екосистеми для розробки програмного забезпечення є комплексним процесом, який враховує технічні, бізнесові та користувацькі вимоги, а формалізація критеріїв вибору платформ та інструментів є важливою для забезпечення ефективності розробки та досягнення оптимальних результатів у проєктах.
Ключові слова: фреймворк; екосистема; критерії розробки; платформа; Ionic; Native; Web App; Hybrid.

ПОСТАНОВКА ЗАДАЧІ

Розвиток інформаційних технологій, мережі інтернет, зростання обчислювальних потужностей та можливостей сучасних пристроїв, а також запит суспільства на нові сервіси [1], пришвидшує еволюцію різноманітних платформ для розробки як вузькоспеціалізованого програмного забезпечення для певних галузей господарства і універсальних рішень, які часто складаються з кількох технологій, що обмінюються даними за допомогою прикладного інтерфейсу API. Таким чином, інформаційні технології стають рушійною силою розвитку бізнесу. Разом з тим, успішність програмних проєктів значною мірою залежить від обраної інфраструктури розробки, яка визначає функціональні можливості, продуктивність, масштабованість, підтримку і безпеку розроблюваних програмних продуктів. Окрім того обрана інфраструктура повинна забезпечувати підтримку інструментів інтегрованого середовища розробки (IDE).

Перехід від інфраструктури до екосистеми розробки програмного забезпечення означає розширення цієї інфраструктури до більш комплексної структури, яка включає в себе не тільки інструменти та технології, але й методології, стандарти, платформи для тестування та розгортання. Екосистема розробки програмного забезпечення охоплює весь життєвий цикл продукту від початкового проектування та розробки до підтримки, створюючи умови для ефективної, безпечної та масштабованої розробки.

АНАЛІЗ ОСТАННІХ ДОСЛІДЖЕНЬ І ПУБЛІКАЦІЙ

Дослідження процесу вибору платформи для розробки програмного забезпечення є важливим аспектом програмної інженерії. Як показує проведений аналіз, сьогодні можна виокремити різні напрямки досліджень та підходів до вибору платформи. У своїй роботі «Управління якістю програмних веб-систем засобами розробки», О. М. Шинкарук, О. М. Яшина, О. Г. Онишко [2], досліджують теоретичні аспекти застосування сучасних фреймворків для розробки програмного забезпечення, а також обґрунтування доцільності та впливу варіативності вибору на якість розроблюваних програмних додатків з використанням популярних фреймворків, а Н. Ю. Філь і О. С. Волошин у роботі «Модель вибору програмного забезпечення для розробки мобільного додатку в умовах нечіткої інформації» [3], аналізують особливості сучасного програмного забезпечення для розробки мобільних додатків. Економічно-технічний вплив вибраної платформи досліджується у роботах, що відносяться до вибору lowcode та nocode bpms, як сучасних трендів до автоматизації бізнес процесів підприємства [4]. Також зустрічаються дослідження де аналізується вплив методологій Agile і Waterfall, щоб визначити, яка методологія є найбільш підходящою для різних програмних проектів де враховується практичний досвід розробників програмного забезпечення [5]. Проведений аналіз показує, що аналіз та дослідження критеріїв вибору екосистеми для розробки програмного забезпечення, а також формалізація критеріїв вибору інтегрованого середовища розробки, є своєчасною та актуальною.

ВІДОКРЕМЛЕННЯ НЕВИРІШЕНИХ РАНІШЕ ЧАСТИН ЗАГАЛЬНОЇ ПРОБЛЕМИ

Попередні дослідження переважно фокусувалися на виборі окремих інструментів чи технологій для розробки програмного забезпечення, однак недостатньо уваги було приділено системному підходу до вибору цілісної екосистеми для розробки програмних продуктів. Крім того, існуючі дослідження здебільшого не враховують потреби команд розробників, що може значно впливати на вибір платформи та інструментів.

МЕТА ДОСЛІДЖЕННЯ

Визначення критеріїв вибору інструментів, платформ та технологій для розробки програмного забезпечення, з акцентом на перехід від вибору окремих інструментів до комплексних екосистем розробки.

МЕТОДИ ДОСЛІДЖЕННЯ

Використовувались методи порівняльного аналізу, аналітичного і системного підходу.

Об'єкт дослідження: екосистеми для розробки програмного забезпечення, зокрема платформи та фреймворки для мобільних і веб-застосунків.

Предмет дослідження: критерії вибору екосистеми для розробки програмного забезпечення, зокрема на основі технічних характеристик платформ, доступу до функцій пристроїв, витрат на розробку, підтримки інструментів розробки, а також потреб і досвіду команд розробників.

ОСНОВНИЙ МАТЕРІАЛ

Сучасний етап розвитку інформаційних технологій, засобів проектування, зокрема в галузі web-розробки, призвів до швидкого зростання кількості підходів до створення мобільних застосунків, засобів їх розробки та підтримки. Такі засоби призначені для виконання свого комплексу задач і мають свої переваги та недоліки, а також потребують спеціалізованих інструментів для спрощення їх використання, масштабування та підтримки. З урахуванням того, що головною ціллю будь-якого фреймворку або бібліотеки є полегшення написання коду і запуску продукту, перед вибором відповідних інструментів необхідно визначити тип програмного забезпечення, для розробки якого вони будуть застосовуватися. Станом на сьогодні, у випадку мобільної розробки, найбільш поширеними можна вважати три типи застосунків [6]: нативні (Native), web- (Web app) та гібридні (Hybrid) застосунки (рис. 1).

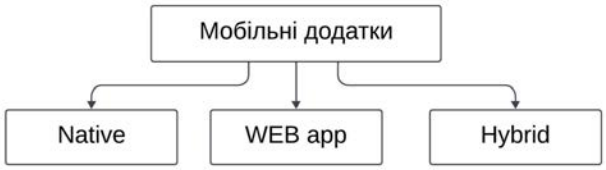

Рис. 1. Типи мобільних застосунків

Нативні застосунки створюються спеціально для конкретної платформи, як-от iOS чи Android, із використанням відповідних засобів та мов програмування, таких як Swift чи Kotlin. Вони забезпечують найкращу продуктивність, оскільки мають прямий доступ до апаратного забезпечення пристрою, наприклад до web-камери, GPS чи сенсорів. Завдяки цьому гарантується застосування виняткового користувацького досвіду (UX) та використання всіх можливостей платформи. Однак їх розробка є затратною, оскільки для кожної платформи потрібно створювати окрему версію застосунку, що може ускладнювати подальше масштабування та оновлення.

Загально відомим підходом є використання замість мобільного додатку web-застосунку. Web-застосунок працює через web-переглядач і є незалежним від операційної системи пристрою. Це значно зменшує витрати на розробку, адже достатньо створити одну версію програми, яка буде працювати на будь-якому пристрої з підключенням до Інтернету. Оновлення web-застосунків відбувається централізовано, що спрощує їх масштабування. Проте їх продуктивність обмежена можливостями web-переглядача, а доступ до функцій пристрою, таких як web-камера чи GPS, стає значно більш обмеженим. Також користувацький досвід у web-застосунках стає менш гнучким і персоналізованим через відсутність тісної інтеграції з операційною системою пристрою. На відміну від нативних та web-застосунків, гібридні є спробою поєднати їх переваги. Вони створюються з використанням фреймворків, таких як React Native, Flutter чи Ionic, що дозволяє розробникам використовувати один код для різних платформ. Це значно знижує витрати на розробку і спрощує підтримку застосунків. Гібридні застосунки мають кращий доступ до функцій пристрою, ніж web-застосунки, але поступаються в цьому аспекті нативним. Продуктивність гібридних застосунків також є компромісною та залежить від обраного фреймворку. Користувацький досвід у гібридних рішеннях близький до нативного, хоча іноді можуть виникати обмеження, пов'язані з адаптацією інтерфейсу. Таким чином, головними критеріями вибору типу застосунку можна вважати продуктивність, користувацький досвід (UX), доступ до функцій пристрою або його API, масштабованість, а також витрати на розробку (рис. 2).

Критерії вибору типу застосунку тісно пов'язані з вибором відповідних програмних інструментів та середовища розробки. Сьогодні, що все більшої популярності отримує розробка з використанням frontend-, backend- та fullstack-фреймворків. Фреймворк визначає наскільки ефективною та простою буде розробка, а також, які можливості платформи стануть доступними. Так, при розробці нативних застосунків, обирати фреймворк для роботи відносно просто, оскільки тут є набір засобів, що забезпечать високу продуктивність і підтримку особливостей конкретної платформи. Наприклад, для iOS це Swift і Xcode, а для Android – Kotlin та Android Studio. Основний критерій тут – глибока інтеграція з платформою та доступ до всіх API, але при цьому слід враховувати додаткові витрати на підтримку окремого фреймворку для кожної операційної системи. У свою чергу фреймворки для розробки web-застосунків визначається здатністю забезпечувати швидку розробку та високий рівень сумісності. Основними критеріями вибору тут стають: продуктивність фреймворку, наявність документації та активність спільноти, яка може допомогти у вирішенні технічних проблем. Гібридні застосунки повинні забезпечувати максимальну продуктивність на всіх платформах і одночасно дозволяти повторне використання коду. Такі фреймворки, як React Native, Flutter або Ionic, створюють оптимальні умови для одночасної розробки застосунків для iOS і Android [7]. Вибір між ними залежить від таких критеріїв, як продуктивність, простота інтеграції з нативними компонентами, витрати на навчання команди, а також здатність масштабувати застосунок. Наприклад, Flutter виділяється високою продуктивністю завдяки власному механізму рендерингу, тоді як React Native та Ionic пропонують велику гнучкість завдяки можливості використання нативних модулів.

Оскільки всі фреймворки покликані спростити розробку [8], то одним з головних критеріїв

Рис. 2. Критерії вибору екосистеми

вибору можна вважати продуктивність, яка є критичним показником, особливо для застосунків із високими вимогами до швидкодії. Наприклад, Flutter забезпечує високу продуктивність завдяки прямому доступу до графічного інтерфейсу пристрою, тоді як React Native демонструє гарну продуктивність, але залежить від API для взаємодії з нативними компонентами. У випадку Ionic продуктивність може бути меншою через використання web-технологій, таких як HTML, CSS і JavaScript, що впливає на роботу застосунків на деяких пристроях.

Наступним важливим критерієм можна вважати гнучкість і адаптивність фреймворку. Бібліотека React і фреймворк Vue є гнучкими інструментами, які легко інтегруються з іншими бібліотеками чи проєктами. Водночас, Angular, попри свою комплексність, пропонує великий набір вбудованих інструментів та чітко визначені практики, хоча може бути практики менш гнучким для деяких специфічних задач. У свою чергу, гібридний застосунок, створений з використанням Ionic, вирізняється підтримкою Angular, React або Vue, що додає йому гнучкості. Ще одним важливим критерієм є крива навчання, яка визначає наскільки швидко команда зможе освоїти новий інструмент, оскільки бізнес може мати своє бачення застосування фреймворку і команда повинна швидко освоїти нові інструменти. Оскільки React і Vue мають порівняно просту структуру, то це дозволяє розпочати розробку практично одразу. Angular, навпаки, має складнішу екосистему, через що може вимагати більше часу на навчання. В силу того, що Ionic є своєрідним «мостом» між фронт-енд фреймворками і нативною платформою, він легкий для освоєння, особливо для розробників, які знайомі з web-технологіями. До значущих критеріїв вибору фреймворку слід віднести підтримку та спільноту, оскільки популярність обраного програмного комплексу може дозволити швидко обмінюватись досвідом з іншими розробниками і часто вирішувати спільні задачі в рамках кількох розрізнених проектів. Набір інструментів та екосистема плагінів є критеріями, які визначають вбудований функціонал. Так, Angular CLI, RxJS та інші інструменти надають зручні інтерфейси для роботи реактивністю, а React і Vue надають широкий вибір сторонніх бібліотек і розширень для параметризації проєкту. Ionic, своєю чергою, забезпечує розробників інструментами для створення мобільних застосунків і можливістю інтеграції з різними фреймворками. Довготривала підтримка також може вважатися критерієм вибору фреймворку.

Нарешті, важливою є підтримка мов програмування. Angular базується на TypeScript, що забезпечує статичну типізацію й додаткові можливості розробки. React і Vue також підтримують JavaScript та TypeScript, що дозволяє гнучко налаштовувати середовище розробки. Flutter використовує мову

Dart, розроблену спеціально для високопродуктивних застосунків, тоді як React Native та Ionic працюють з мовами JavaScript та TypeScript.

Після вибору типу застосунку та відповідного фреймворку наступним важливим етапом стає визначення середовища розробки (IDE), яке забезпечить максимально ефективну роботу над проєктом. IDE є ключовим інструментом розробника, який значно впливає на продуктивність команди, зручність створення та тестування коду, а також швидкість інтеграції з іншими інструментами, такими як системи штучного інтелекту Copilot, Chat GPT, тощо [9]. Вибір середовища розробки залежить від декількох важливих факторів, зокрема від розширень, а також інструментів для налагодження й тестування. Так для кожного типу додатку (нативний, веб або гібридний) існують спеціалізовані IDE, які забезпечують необхідний функціонал та оптимізують процес розробки. Наприклад, для розробки нативних застосунків Xcode є основним середовищем для iOS, тоді як Android Studio ідеально підходить для створення Android-застосунків завдяки вбудованим інструментам інтеграції з платформою. Для web-розробки популярними середовищами є Visual Studio Code, WebStorm і SublimeText, які пропонують розширену підтримку web-технологій і численні розширення для інтеграції з Angular, React або Vue. Варто зауважити, що гібридні фреймворки, такі як Flutter чи React Native, також мають свої рекомендації щодо середовища розробки. Для Flutter зазвичай використовуються Android Studio або Visual Studio Code, які забезпечують підтримку мови Dart і потужні інструменти для тестування. React Native добре працює з Visual Studio Code завдяки великій кількості розширень.

ВИСНОВКИ

Комплексний підхід до вибору платформи, заснований на багатокритеріальному аналізі, дозволяє враховувати як технічні, так і бізнес-потреби розроблюваного продукту. Вибір типу додатку допомагає на початковому етапі конкретизувати вимоги до майбутнього програмного засобу. Вибір програмних інструментів, таких як frontend-, backend- або fullstack-фреймворк, базується на комплексній оцінці технічних, бізнесових і користувацьких вимог проєкту, що забезпечує оптимальний результат для кожного конкретного випадку. Обрання середовища розробки, яке повною мірою підтримує обраний фреймворк, впливає не лише на зручність роботи з проєктом, а й на швидкість та якість розробки. Таким чином, важливість критеріїв вибору IDE, фреймворків і типу додатку полягає в забезпеченні ефективної інтеграції всіх компонентів розробки. Тип додатку, програмні інструменти та середовище розробки формують цілісну екосистему, здатну забезпечити ефективну реалізацію проєкту, враховуючи специфічні вимоги та цілі розробки. Така екосистема не лише сприяє оптимізації процесу створення програмного продукту, а й гарантує його високу якість, масштабованість і відповідність сучасним стандартам. Подальші дослідження можуть бути спрямовані на уточнення критеріїв, зокрема в контексті нових технологій, наприклад, штучного інтелекту та великих даних.

Дослідження виконувались у межах наукової теми «Моделювання та застосування хаотичних динамічних систем для оптимізації алгоритмів штучного інтелекту, машинного навчання та веб-технологій» (державний реєстраційний номер 0124U004447).

REFERENCES

[1] Ranjan, P. (2022, March 15). An anthology of insights, for a more inclusive internet. Google. Retrieved from: *https://blog.google/technology/next-billion-users/anthology-insights-more-inclusive-internet/*.

[2] Shynkaruk, O., Yashyna, O., & Onyshko, O. (2020). Upravlinnia yakistiu prohramnykh veb-system zasobamy rozrobky [Quality management of software web systems using development tools]. *Visnyk Khmelnytskoho Natsionalnoho Universytetu. Tekhnichni Nauky,* (6), 39-44 [in Ukrainian].

[3] Fil, N., & Voloshyn, O. (2019). Model vyboru prohramnoho zabezpechennia dlia rozrobky mobilnoho dodatku v umovakh nechitkoi informatsii [Model for selecting software for mobile application development under fuzzy information]. *Tekhnolohiia Pryladobuduvannia. DP Naukovo-Doslidnyi Tekhnolohichnyi Instytut Pryladobuduvannia*, (2), 33–36 [in Ukrainian].

[4] Korzachenko, O. V. (2022). LOW-CODE ta NO-CODE BPMS: Suchasni trendy avtomatyzatsii biznes-protsesiv pidpryiemstva [LOW-CODE and NO-CODE BPMS: Modern trends in business process automation]. *Modeliuvannia ta Informatsiini Systemy v Ekonomitsi. Natsionalnyi Ekonomichnyi Universytet imeni Vadyma Hetmana*, (102), 102–114 [in Ukrainian].

[5] Asieieva, A., & Kulakovska, I. (2019). Analiz problem vyboru tekhnolohii dlia rozrobky prohramnoho zabezpechennia [Analysis of problems in choosing technology for software development]. *Kompiuterno-Intehrovani Tekhnolohii: Osvita, Nauka, Vyrobnytstvo*, (37), 10–18 [in Ukrainian].

[6] Shtanova, A. L. (2024). Mobilni zastosunky v systemi mobilnoho marketynhu [Mobile applications in the mobile marketing system]. *Problemy Suchasnykh Transformatsii. Seriia: Ekonomika ta Upravlinnia,* (13). https://doi.org/10.54929/2786-5738-2024-13-04-05 [in Ukrainian].

[7] Farooq, M. S., Riaz, S., Alvi, A., Ali, A., & Rehman, I. U. (2022). Cross-Platform Mobile Development Approaches and Frameworks. *VFAST Transactions on Software Engineering*, *10* (2), 79–93. https://doi.org/10.21015/vtse.v10i2.978.

[8] Jeel Patel.(2023). *Detailed Web Framework Comparison With Pros And Cons, 2023*. Retrieved from https://www.monocubed.com/blog/web-development-framework-comparison.

[9] Sotnik, S., Lyashenko, V., & Schakurova, T. (2021). Modern Integrated Software Development Environments. *International Journal of Academic and Applied Research (IJAAR),* Vol. 5 Isue 10, 157–161.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

[1] Ranjan, P. (2022, March 15) An anthology of insights, for a more inclusive internet. Google. URL: *https://blog.google/technology/next-billion-users/anthology-insights-more-inclusive-internet*.

[2] Шинкарук, О., Яшина, О., & Онишко, О. (2020). Управління якістю програмних веб-систем засобами розробки. *Вісник Хмельницького національного університету. Технічні науки*. (6), 39–44.

[3] Філь, Н., & Волошин, О. (2019). Модель вибору програмного забезпечення для розробки мобільного додатку в умовах нечіткої інформації. *Технологія приладобудування. ДП Науково-дослідний технологічний інститут приладобудування.* (2), 33–36.

[4] Корзаченко, О. В. (2022). LOW-CODE та NO-CODE BPMS: Сучасні тренди автоматизації бізнес-процесів підприємства. Моделювання та інформаційні системи в економіці. *Нац. екон. ун-т ім. Вадима Гетьмана*. (102), 102–114.

[5] Асєєва, А., & Кулаковська, І. (2019). Аналіз проблем вибору технології для розробки програмного забезпечення. *Комп'ютерно-інтегровані технології: освіта, наука, виробництво*. (37), 10–18.

[6] Штанова, А. Л. (2024). Мобільні застосунки в системі мобільного маркетингу. *Проблеми сучасних трансформацій. Серія: економіка та управління*. (13). https://doi.org/10.54929/2786-5738-2024-13-04-05

[7] Farooq, M. S., Riaz, S., Alvi, A., Ali, A., & Rehman, I. U. (2022). Cross-Platform Mobile Development Approaches and Frameworks. *VFAST Transactions on Software Engineering*. 10 (2), 79–93. https://doi.org/10.21015/vtse.v10i2.978

[8] Jeel Patel.(2023). *Detailed Web Framework Comparison With Pros And Cons, 2023*. https://www.monocubed.com/blog/web-development-framework-comparison.

[9] Sotnik, S., Lyashenko, V., & Schakurova, T. (2021). Modern Integrated Software Development Environments. *International Journal of Academic and Applied Research (IJAAR),* Vol.5 Isue 10, 157–161.

Дата надходження статті до редакції: 03.02.2025
Дата затвердження статті до друку: 15.02.2025