



## ОЦІНКА ЕФЕКТИВНОСТІ ФРЕЙМВОРКІВ ЗГІДНО З КРИТЕРІЯМИ ПРОЄКТУ

*Поляков А.О., доцент, кафедра ІС, ХНЕУ ім. С. Кузнеця*  
*Литнів В.С., студент, кафедра ІС, ХНЕУ ім. С. Кузнеця*

В сучасному світі інформаційних технологій, використання фреймворків є невід'ємною складовою розробки програмного забезпечення. Фреймворки надають розробникам потужний інструментарій для швидкої і ефективної реалізації різноманітних проєктів, сприяючи збереженню часу і ресурсів. Однак, вибір і використання фреймворку повинно бути обґрунтованим та суттєвим для конкретного проєкту.

*Критерії вибору фреймворку.* При виборі фреймворку для веб-розробки слід враховувати такі фактори: тип додатку (фокусує на необхідних функціях що повинен мати фреймворк, односторінковим додаткам (SPA) необхідна компонентна архітектура; додаткам веб-ігор – високо продуктивність [1]), розмір і складність додатку (великим і складним додаткам потрібна масштабованість і надійність, малим і простим додаткам достатньо фреймворків з постим налаштування та використанням), компетенції команди розробників (достатня експертиза у використанні фреймворку).

*Критерії оцінки фреймворків.* Щоб оцінити різні фреймворки, можна використовувати такі критерії: кількість користувачів (чим більше тим вище ймовірність що фреймворк надійний і добре документований), стабільність розробки (це відповідь на такі питання: Як часто випускаються нові версії фреймворку? Як організована підтримка для старих версій?), суспільство (Який розмір спільноти розробників, яка підтримує фреймворк? Яка кількість навчальних матеріалів, статей на StackOverflow.com, блогів?).

Аналізуючи фреймворки що представлено у Frontend розробці [2], то домінують три з них, це Angular, React, Vue, які ми будемо аналізувати у роботі.

*Практичний аналіз вихідного коду фреймворків.*

*Якість коду.* Для його оцінки використовувався інструмент SonarQube, що отримує різні метрики для оцінки якості коду, також має свою шкалу оцінки.

*Валідність коду.* Для його оцінки використовувався інструмент ESLint (інструмент для виявлення помилок у кодї JavaScript).

Аналіз ефективності Front-End фреймворків згідно з критеріями проєкту було проведено за допомогою наступних методів: статичним аналізом коду Scitools Understand 4.0, утиліти Plato аналіз (структуру коду та взаємодію між компонентами), утиліта complex-report (складність коду), тестуванням продуктивності у реальному середовищі.

Таблиця 1 – Результати оцінки якості коду фреймворків React, Angular, Vue.js отримані SonarQube 9.0 CE

Фреймворк	Шкала SonarQube
React	8.6
Angular	7.4
Vue.js	8.2

Таблиця 2 – Результати оцінки валідності коду фреймворків React, Angular, Vue.js отримані ESLint v8.55.0

Фреймворк	Кількість помилок
React	20
Angular	30
Vue.js	10



Таблиця 3 – Результати розрахунку показників коду фреймворків

Показник програмного забезпечення	Angular	Vue	React
Кількість рядків у коді	36595	11965	2333
Кількість виразів	14775	2655	875
Кількість рядків коментарів	8839	996	329
Співвідношення кількості рядків коментарів та рядків коду	0,24	0,08	0,14
Цикломатична складність	3544	1173	415
Глибина вкладеності	10	9	9
Кількість попереджень	19	176	100
Середня важкість функцій згідно методу Холстеда	1323	5333	4356

### *Висновки дослідження ефективності Front-End фреймворків.*

На основі результатів порівняльного аналізу метрик продуктивності, якості і валідності коду трьох популярних Front-End фреймворків React, Angular і Vue.js можна зробити наступні висновки:

React має найкращі показники продуктивності, якості і валідності коду. Його цикломатична складність є найнижчою, а важкість за методом Холстеда – середньою. Це робить React більш простим у підтримці та налаштуванні. Крім того, React має найвищий відсоток рядків коментарів, що свідчить про високу документованість вихідного коду.

Angular має найгірші показники продуктивності, якості і валідності коду. Його цикломатична складність є найвищою, а важкість за методом Холстеда – середньою. Це ускладнює підтримку та налаштування Angular. Крім того, Angular має найнижчий відсоток рядків коментарів, що свідчить про низьку документованість вихідного коду.

Vue.js має середні показники продуктивності, якості і валідності коду. Його цикломатична складність є нижчою, ніж у Angular, але вище, ніж у React. Важкість за методом Холстеда також є середньою. Vue.js має середній відсоток рядків коментарів, що свідчить про середню документованість вихідного коду.

Була розроблена методика оцінки фреймворків, яка включала в себе порядок проведення експериментів, процедуру збору даних та аналізу результатів. Ця методика враховувала вимоги та обмеження конкретних проєктів, на яких проводилася оцінка.

Загалом, ця методологія дозволила систематично та об'єктивно дослідити та оцінити ефективність різних фреймворків відповідно до конкретних критеріїв, що були сформульовані для даного дослідження.

### Список літератури

1. Knupp, J. (2019). What is a Web Framework? Python Programming. <https://jeffknupp.com/blog/2019/03/03/what-is-a-web-framework/>.
2. Greif, S. (2022). State of JavaScript 2022. State of JavaScript 2022. <https://2022.stateofjs.com/en-US>.