



ДОСЛІДЖЕННЯ МЕТОДІВ ТА АРХІТЕКТУРНИХ РІШЕНЬ ДЛЯ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ВЕБ-РОЗРОБКИ НА ПЛАТФОРМІ .NET

Валенда Н.А., к. т. н., доцент, кафедра ПІ, ХНУРЕ

Мітряєв С.С., студент, кафедра ПІ, ХНУРЕ

Дана робота присвячена оцінці ефективності різних типів архітектур розробки програмного забезпечення для вдосконалення розробки веб-додатків. Розглянуто основні типи програмних архітектур, такі як мікросервісна, багаторівнева, клієнт-серверна та сервіс-орієнтована. Для їх оцінки використовувався метод адитивної лінійної згортки з ваговими коефіцієнтами. Експеримент проводився на спеціально створеному для цього дослідження програмному комплексі з використанням платформи Windows 11 і .NET Framework версії 4.8.1. Критерії порівняння альтернатив були обрані з огляду на їх важливість під час розробки.

Майже кожна людина у світі у наш час стикається із веб-системами. Вони оточують нас повсюди, коли ми замовляємо продукти, купуємо квитки, перевіряємо банківські акаунти та при інших взаємодіях з інформаційним світом.

Існує багато різних мов та методів програмування, які дозволяють створювати потужні системи, які вміють не тільки презентувати інформацію, а й обробляти та зберігати її. Існує безліч критеріїв, за якими можна обирати ті чи інші архітектури та методи. Основними критеріями при розробці програмних систем є легкість користувацької взаємодії, безпека даних, можливість масштабування системи та ціна розробки [1].

Першим кроком у описі задачі багатокритеріального вибору варіанту архітектури веб-системи є визначення критеріїв, які відображають вимоги та обмеження проекту. Основні критерії, які можуть мати вплив на вибір архітектури, можуть бути наступними:

- вартість: загальна вартість розробки та підтримки проекту за обраною архітектурою;
- масштабованість: можливість легкого внесення змін у кількість користувачів, що одночасно можуть користуватися системою;
- місце зберігання даних: можливість зберігати дані віддалено або локально та створювати резервні копії;
- надійність: можливість запобігати критичним помилкам та відмовам системи та швидко відновлювати її роботу;
- гнучкість системи: можливість легко вносити зміни у систему без необхідності у значних змінах системи;
- безпека: підтримка системою захисту персональних даних від витоків.

Окрім даних критеріїв, ще можна розглядати інші, менш суттєві критерії, такі, як легкість користування, продуктивність та інші.



Отже, задача вибору архітектури для розробки інформаційної програмної веб-системи відноситься до класу багатокритеріальної задачі вибору, де кожен критерій має певну вагу, яка відображає його важливість для проекту. Завдання даного дослідження полягає у тому, щоб обрати архітектуру розробки програмного забезпечення, яка забезпечує найкраще поєднання даних критеріїв, що будуть задовольняти вимогам та обмеженням даного проекту.

Існують різні способи розв'язання задач багатокритеріального вибору. В рамках даного дослідження, ми розглянемо метод, який дозволяє вирішити дану задачу, враховуючи найважливіші критерії розробки програмної веб-системи та враховуючи їх вагу [2]. Для цього завдання ми будемо використовувати метод адитивної лінійної згортки із ваговими коефіцієнтами.

Ідея даного метода адитивної лінійної згортки із ваговими коефіцієнтами полягає у тому, щоб ефективно оцінити та вибрати оптимальний варіант з набору альтернатив, приділяючи вагомий внесок кожному критерію відповідно до його важливості для прийняття рішення.

Кожен критерій має свій ваговий коефіцієнт, який відображає його важливість у прийнятті рішення. Ці вагові коефіцієнти можуть визначатися експертами або іншими методами, які враховують пріоритети та представлення індивідуальних критеріїв [2].

Після створення шкал вимірювання критеріїв, зважаючи на доступну інформацію про альтернативи, ми можемо оцінити їх за даними шкалами [3]. Для цього кожна із альтернатив необхідно оцінити за кожною із шкал.

Важливим етапом пошуку оптимального варіанту є обрання підмножини альтернатив за принципом Парето. Обрання підмножини Парето – важливий етап у багатокритеріальному виборі, спрямований на визначення оптимальних рішень, які не мають собі рівних за всіма критеріями. Процес починається з оцінки рішень за різними критеріями та побудови матриці порівнянь. Визначення відносної важливості критеріїв та їхніх вагових коефіцієнтів є ключовим етапом. Після ранжування рішень за кожним критерієм вибирається підмножина Парето – рішення, які не мають інших, що б краще за них за будь-яким з критеріїв. Остаточний етап включає оптимізацію та аналіз вибраних рішень враховуючи конкретні умови задачі та вимоги. Цей процес дозволяє визначити оптимальні рішення та побудувати простір Парето – набір всіх можливих оптимальних рішень [2].

Після проведення експерименту та розрахунків з використанням вищенаведеного методу, було отримано наступне ранжування видів архітектури від найбільш корисної до найменш корисної:

- мікросервісна;
- подієво-орієнтована;
- клієнт-сервер;
- мультирівнева;
- сервісно-орієнтована;
- монолітна.



На основі розглянутих питань та проведеного дослідження можна зробити наступні висновки: кожна система має свої певні переваги та недоліки, які розробники повинні враховувати при використанні певної методології. Найбільш ефективною для розробки веб-додатків було визначено мікросервісну архітектуру, яка базується на розподілі додатку на сервіси, які взаємодіють між собою. Найменш ефективною було визначено монолітну архітектуру, що ставить за основу розташування усіх елементів у одному кодовому монолітному базисі.

При виконанні дослідження, було проаналізовано різні способи вирішення задач багатокритеріального вибору та обрано метод адитивної лінійної згортки із ваговими коефіцієнтами, що дозволяє легко проаналізувати багато альтернатив, враховуючи різні критерії та важливість цих критеріїв.

Іншим важливим висновком є те, що різні архітектури розробки програмного забезпечення мають повністю протилежні недоліки та переваги. Тому не можна сказати, що, наприклад, монолітна архітектура є повністю неефективною. Вона може бути ефективною при розробці програмного забезпечення іншого типу, аніж веб-застосунків [3].

Отже, у результаті дослідження було виявлено, що мікросервісна архітектура дозволяє найкраще за інші отримати безпечну, легко масштабовану, дешеву у розробці, просту у підтримці веб-систему при використанні платформи .Net. Але при використанні іншої важливості критеріїв оцінювання, інший тип архітектур програмного забезпечення може мати перевагу над використанням мікросервісної архітектури, але це потребує іншого дослідження, використовуючи конкретні вимоги та пріоритети.

Список літератури

1. Мартін, Р. (2019). Чиста архітектура: мистецтво розробки програмного забезпечення. Харків: Фабула.
2. Волошин, О.Ф., & Машенко, С.О. (2018). Моделі та методи прийняття рішень. Київ: Видавництво Людмила.
3. Мартін, Р. (2019). Чистий код. Створення і рефакторинг. Харків: Фабула.