

ЕВОЛЮЦІЙНА АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ: РОЗРОБКА ФУНКЦІЙ ПРИДАТНОСТІ ДЛЯ МАСШТАБУВАННЯ ТА ОПТИМІЗАЦІЇ

Макєєв О. С.

Науковий керівник – к.т.н., доцент Кравець Н. С.

Харківський національний університет радіоелектроніки, каф. ПІ

м. Харків, Україна

e-mail: oleksii.makieiev@nure.ua

The purpose of the study is to develop and validate fitness functions for the evaluation and optimization of various software architectures, for the experiment, it was decided to choose the monolithic architecture of a web application to use fitness functions to evaluate and optimize the architecture. The key fitness function was picked the scalability that provides the ability to evaluate an architecture from the monolithic to microservices architecture. The study highlighted the importance of evaluation architecture as a key element in supporting and developing software programs.

Компанії часто зіштовхуються з викликами, пов'язаними зі старінням їх програмного забезпечення. У зв'язку з цим, регулярно набирають нових програмних інженерів для розширення та підтримки існуючих систем. Основною проблемою є те, що із зростанням функціональності програмного забезпечення, його архітектура стає складнішою для модифікації та підтримки. Це посилюється появою нових бізнес-вимог, що змушує компанії адаптувати та переосмислювати свої архітектурні підходи.

Відповіддю на ці виклики є інновації, зокрема перехід до хмарних рішень [1], як-от AWS, Microsoft Azure, чи Google Cloud. Проте, перенесення існуючих систем у хмару може бути складним через їх недостатню оптимізацію для хмарного середовища, що призводить до збільшення витрат.

Метою дослідження є розробка та валідація функцій придатності для оцінки та оптимізації різноманітних архітектур програмного забезпечення. Дані функції дозволяють не лише вдосконалювати монолітні структури, але й ефективно працювати з мікросервісними та іншими гнучкими архітектурними підходами. Мета дослідження забезпечити адаптивність та ефективність систем у відповідь на динамічні зміни бізнес-вимог та технологічних змін, незалежно від їх первісної архітектури.

Існує чимало рішень для створення оптимальної архітектури програмного забезпечення для хмарних систем, зокрема мікросервіси та лямбда-функції. Однак, з часом можуть виникнути проблеми з масштабуванням, інтеграцією нових функцій або підтримкою існуючих через зміни ключових елементів архітектури. Наприклад, може

збільшитися час відгуку на запити з однієї секунди до тридцяти секунд. Еволюційна архітектура [2] застосовується для забезпечення стабільності та підтримки програмного забезпечення у процесі його розвитку.

Еволюційна архітектура – це підхід у проектуванні програмного забезпечення, який дозволяє системі гнучко адаптуватися та розвиватися у відповідь на зміни в бізнес-вимогах, технологіях та інших факторах, що впливають на розвиток програмного продукту.

Еволюційну архітектуру використовують для того щоб адаптація до нових змін. Швидке реагування на нові бізнес-вимоги та технологічні тренди. Система працює стабільно та інженери не втрачають швидкість додавання нових змін чи підтримки системи.

Еволюційна архітектура складається на принципах інкрементних змін та функцій придатності[3], а також включає розгляд додаткових аспектів, таких як модульність, адаптивність до змін і інтеграція з іншими системами і сервісами, що разом формують комплексний підхід до розвитку стійких та гнучких програмних рішень.

Функція придатності – це міра пристосованості рішення до змінюваного контексту, в якому вона використовується. Виступає у якості керівництва та обмеження, в рамках якого еволюційно розвивається архітектура. Функції придатності діляться на категорії, а саме: атомарні, комплексні, тригерні, безперервні, статичні та динамічні.

Архітектурна функція придатності надає оцінку цілісності деяких архітектурних характеристик.

У дослідженні було застосовано функції придатності для оцінки та оптимізації монолітної архітектури веб-додатку, з метою її переходу до мікросервісної архітектури. Важливо підкреслити, що даний випадок фокусується на конкретному застосуванні, функції придатності є загальним інструментом, придатним для різних архітектурних контекстів, включаючи мікросервісні, гібридні та інші системи, які вимагають адаптації та розвитку. Це дозволяє забезпечити адаптивність та ефективність програмного забезпечення у відповідь на зміни бізнес-вимог та технологічного середовища.

Визначившись з функціями придатності, було обрано ключову функцією придатності – масштабованість, яка дасть можливість еволюціонувати архітектурі з моноліту до мікросервісної архітектури, дасть можливість оптимізувати запити та впровадити кешування.

Веб додаток було написано на Java, використовуючи Spring Framework. Для перевірки архітектурних функцій придатності, а саме: циклічної залежності та перевірки шарів, було використано ArchUnit [4] та JDepend. Запуск unit та інтеграційних тестів проходить використовуючи GitHub Actions. База даних була обрана PostgreSQL та Redis для кешування.

Функцію придатності, яка відповідає за час відгуку, було описано формулою $y(x) \leq 5000$, де 5000 – це максимально допустимий час відгуку в мілісекундах, $y(x)$ – це значення, яке буде отримане, коли виконається запит.

На основі всіх функцій придатності було прораховано загальну функцію придатності за формулою $F(x) = f_1(x) + f_2(x) + f_3(x) \dots + f_n(x)$, де $F(x)$ – функція придатності всієї системи, $f_1(x)$, $f_2(x)$, $f_3(x)$ – функції придатності, які відображають вимірювання певних характеристик системи.

У дослідженні було оцінено важливість і необхідність розвитку та оптимізації програмного забезпечення за допомогою функцій придатності, що дозволяє забезпечити адаптивність і ефективність систем в умовах швидкого розвитку технологій та змінних бізнес-вимог.

Список використаних джерел:

1. Макеєв, О., Кравець, Н. Дослідження методів створення сервісно-орієнтованих програмних систем у AZURE. 2023. *Computer Systems and Information Technologies*, (2), 38–47. DOI: <https://doi.org/10.31891/csit-2023-2-5>.

2. Ford, N., Parsons, R., Kua, P., Sadalage P. *Building Evolutionary Architectures: Automated Software Governance*. O'Reilly Media, 2022. 269 p.

3. Chondamrongkul, N., Sun, J. Architectural Refactoring for Functional Properties in Evolutionary Architecture. 2022. *IEEE 19th International Conference on Software Architecture (ICSA)*, Honolulu, HI, USA, 2022, 146-156. DOI: 10.1109/ICSA53651.2022.00022.

Richards, M., & Ford, N. *Fundamentals of Software Architecture: An Engineering Approach*. O'Reilly Media, 2020. 285 p.