Abstract

The rapid growth of backend technology correlates positively with the number of problems that arise. One of the issues is the negligence of following an ideal backend system architecture. It causes various problems such as huge numbers of code duplication in many locations of the backend system codebase leading to a low maintainability level. This research aims to investigate the impacts of applying clean architecture on the backend codebase by creating several layers in the codebase according to the business logic hierarchy. To analyze the impacts, this research utilizes several metrics such as Cyclomatic Complexity, Weighted Method Count, Kan's Defects, Halstead's score, and Maintainability Index. The results showed that all maintainability metric scores improved after refactoring a selected backend codebase by applying the clean architecture principle. The improvements range from 21% to 61% for various maintainability metrics. This study validates that the implementation of a clean architecture in the backend codebase could increase the maintainability, reduce its complexity, and reduce developer effort to modify the codebase.

Keywords: clean architecture, maintainability, backend