Abstract

The increasing complexity of modern software and the demand for more functionality, making performance a critical component of application engineering. VIPER is an architectural paradigm based on the principles of Clean Architecture and is preferred for creating complex software systems. The strength of VIPER is its strong separation of code responsibilities, which encourages a distinct division of responsibilities. At the same time, modularization, a process to dividing an application into separate modules—has been adopted to improve component reusability and development process. However, the most essential point is how such a modular approach affects the applications' performance, especially when using the VIPER structure. In order to do this, the research conducts an empirical investigation on the impact of modularization on the functionality of iOS applications organized using VIPER. This study uses an experimental framework to examine nine iOS applications, all of which have different levels of complexity, but share a joint code base and technology stack. The Xcode Instruments testing suite was used to interrogate the performance. This study shows that non-modular apps perform better than their modular counterparts in every performance criterion. The higher overhead caused by intermodule communication is likely the cause of this performance disparity, which may impair the responsiveness and resource efficiency of systems with modular architectures.