

STRATEGIES OF MODULARIZATION FOR ANDROID APPLICATIONS

Pelypets O. S., Dvukhhlavov D. E.

National Technical University «Kharkiv Polytechnic Institute», Kharkiv

The growing complexity of mobile applications, with large codebases, necessitates modularization to enhance maintainability, scalability, and development efficiency. Transitioning from monolithic to modular architectures benefits in testing, bug detection, and project support.

The study investigates possible ways to break projects to modules and states that Gradle modules are the most convenient and beneficial. The study proposes three modularization strategies:

- horizontal (layer-based, e.g., data, domain, UI);
- vertical (feature-based, e.g., cart, profile);
- combined (integrating both).

All strategies follow to Clean Architecture and SOLID principles to ensure code isolation and reusability. As can be seen from Fig. 1, horizontal modulation can also be considered as architectural modulation, because the modules are divided similarly to the division in the multilayer architecture of applications. Vertical modulation can also be called functional, because the division is based on typical functions that are implemented in the development of Android applications.

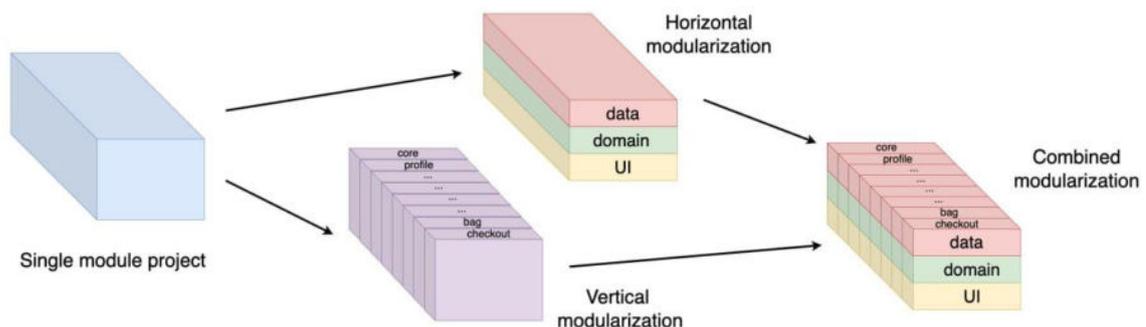


Fig. 1. – Strategies of modularization for Android applications

The transformation from a monolithic to a modular structure affects such parameters as the cohesion of the project modules $Coh(M_i)$ (the degree of internal consistency of the modules) and the coherence of the project modules $Cohes(M)$ (the degree of intermodular dependence). To compare the options for dividing into modules, it is proposed to use the expression:

$$V(M) = \alpha \cdot \sum_{i=1}^n Coh(M_i) - \beta \cdot Cohes(M).$$

In the general case, the higher its value, the better. But this thesis requires experimental confirmation.

Benefits of modularization are reducing merge conflicts, accelerating build time via Gradle parallel and incremental building, and streamlining testing of isolated modules. This improves developer productivity and project flexibility and maintainability.