

Collaborative Discussion 1: Factors which Influence Reusability

Refer to the article by Padhy et al. (2018), specifically Table 1, where the authors present a list of factors which they consider influence the reusability of a piece of object-oriented software.

In this collaborative discussion, you are required to prioritise this list, presenting your argument for the priorities assigned.

Summary Post:

In my initial discussion post on the paper by Padhy et al. (2018) on software reusability, I ranked their proposed reusability assets based on their potential and value in enhancing software reuse. Software reusability is defined as “capability of a product to be used as assets in more than one system, or in building other assets” (ISO25010, 2023). The primary assets in this context are the reusable products or by-products of software development.

Several factors have been identified in literature that influence software reusability, including Coupling, Cohesion, Complexity, Inheritance, Size (code base), and Documentation. These factors are measured using various metrics to determine the degree of reusability, although quantifying such a context-dependent quality aspect remains challenging.

In my analysis, I prioritised software assets based on their reusability potential. The Architecture-Driven Approach (ADP) remains at the top of the ranked list due to its comprehensive nature, which allows for the reuse of entire system designs. This is also in line with the rankings of my peers and wider literature in software design and development. Design Patterns (DPs) and Modules in the Program (MIP) are also highly reusable, although certain DPs vary in their reusability with some argued even

hinder reusability (Feitosa et al., 2019; Wedyan & Abufakher, 2020), and MIPs may require additional work to manage module coupling.

Test cases (TCTDs) and Requirement Analysis (RA) are integral to software development, with RA particularly noted for its high cohesion. Additionally, some of my peers have argued that I have undervalued RA's potential for reuse with respect to other assets, suggesting it should be ranked higher. Given how much time and effort organisations spend on RA and the opportunity of reusing RA within the same, I have reconsidered RA and placed it higher in the ranking. Knowledge Requirements (KR) and Algorithms (AP) are valuable but complex, while Data (UD) is less suitable for reuse due to its variability.

I've also discussed Documentation (DIP) and Service Contracts (SC), noting that DIP is more of a factor than an asset in reusability, and SC is an extension of software development rather than a reusable asset. This nuanced view on prioritising assets for software reusability reflects the ongoing debates and highlights the need for unified approaches to effectively measure and improve reusability in software development.

References:

Feitosa, D., Ampatzoglou, A., Avgeriou, P., Chatzigeorgiou, A. & Nakagawa, E.Y. (2019) What can violations of good practices tell about the relationship between GoF patterns and run-time quality attributes?. *Information and Software Technology*, 105: 1-16.

ISO25010 (2023) ISO/IEC 25010:2023(en). *ISO Online Browsing Platform (OBP)*. Available at: <https://www.iso.org/obp/ui/#iso:std:iso-iec:25010:ed-2:v1:en> [Accessed 24 March 2024].

Padhy, N., Satapathy, S., Singh, R.P. (2018) State-of-the-art object-oriented metrics and its reusability: a decade review. In Smart Computing and Informatics: Proceedings of the First International Conference on SCI 2016, Volume 1, 431-441. Springer Singapore.

Wedyan, F. & Abufakher, S. (2020) Impact of design patterns on software quality: a systematic literature review. IET Software, 14(1): 1-17.