

Initial Post

◀ Initial Post

Initial Post ▶

Display replies in nested form

Settings ▾



Initial Post

by [Lauren Pechey](#) - Thursday, 7 August 2025, 11:35 AM

Prioritising Factors that Influence Reusability in Object-Oriented Software

Reusability is a critical quality attribute in object-oriented software development, contributing to reduced development time, improved maintainability, and cost efficiency. Based on the analysis by Padhy et al. (2018), key factors influencing reusability include modularity, complexity, cohesion, coupling, documentation, inheritance, and size. These factors, while interdependent, can be prioritised based on their relative impact.

Modularity is arguably the most significant factor. A modular system, characterised by discrete and self-contained components, facilitates reuse by allowing developers to isolate and repurpose functionality with minimal modification (Koti et al., 2024). Closely related to this is coupling. Low coupling—representing minimal interdependence between modules—is essential for reuse across different systems. High coupling, by contrast, limits flexibility and complicates integration (Koti et al., 2024).

Cohesion, the extent to which a module performs a single, well-defined task, further enhances reusability. High cohesion improves readability and clarity, which are necessary for effective reuse (Mehboob et al., 2021). Documentation also plays a crucial role, as even well-structured components are unlikely to be reused if their usage is not clearly communicated (Mehboob et al., 2021).

Although complexity, inheritance, and size influence reusability, they are generally secondary. Their impact is often mitigated when the primary factors—modularity, coupling, cohesion, and documentation—are well addressed (Padhy et al., 2018).

In conclusion, prioritising modularity, low coupling, high cohesion, and comprehensive documentation offers the most robust foundation for enhancing reusability in object-oriented software.

References:

Koti, A., Koti, S.L., Khare, A., & Khare, P. (2024) *Multifaceted approaches for data acquisition, processing & communication*. 1st ed. CRC Press. Available at: <https://doi.org/10.1201/9781003470939> [Accessed 7 Aug. 2025].

Mehboob, B., Chong, C., Lee, S., & Lim, J. (2021) Reusability affecting factors and software metrics for reusability: A systematic literature review. *Software: Practice and Experience*, 51(6): 1259–1286. DOI: <https://doi.org/10.1002/spe.2961>

Padhy, N., Satapathy, S. and Singh, R.P. (2018) State-of-the-Art Object-Oriented Metrics and Its Reusability: A Decade Review. In: Satapathy, S.C., Bhateja, V. and Das, S. (eds.) *Smart Computing and Informatics*. Springer, Singapore, pp. 431–441. https://doi.org/10.1007/978-981-10-5544-7_42

Maximum rating: -

[Permalink](#) [Reply](#)



Re: Initial Post

by [Ruben Marques](#) - Friday, 8 August 2025, 4:28 PM

Hey there, Lauren :)

Your prioritisation of modularity, low coupling, high cohesion and documentation as the most significant factors for reusability is strongly supported in the literature. As Koti et al. (2024) note, modularity allows discrete components to be developed, tested and reused independently, reducing the risk of unwanted side effects when components are being adapted for new systems, since high interdependence between modules can make reuse impractical and increase maintenance costs, a concern also highlighted



by Padhy et al. (2018).

Your point on cohesion is equally important. As Mehboob et al. (2021) argue, high cohesion not only improves readability but also makes it easier for future developers to understand and safely reuse components. Likewise, documentation is often underestimated in practice but remains critical for effective reuse, as even well-designed modules can fail to be adopted if their usage is not clearly explained (Mehboob et al., 2021).

I would add to your point that requirement analysis also plays a vital role in enabling reusability. Bennett et al. (2010, Ch. 6) emphasise that capturing overall system requirements before developing use cases allows developers to identify common functions and design modular components with clear boundaries and future reuse in mind. This early planning directly supports your priorities of modularity, low coupling and cohesion.

Overall, your prioritisation offers a well-balanced perspective that combines structural and process-related factors, providing a strong basis for building reusable object-oriented systems.

References

Bennett, S., McRobb, S. and Farmer, R. (2010) Object-oriented systems analysis and design using UML. 4th edn. Maidenhead: McGraw-Hill Higher Education.

Koti, A., Koti, S.L., Khare, A. and Khare, P. (2024) Multifaceted approaches for data acquisition, processing & communication. 1st edn. CRC Press. doi:10.1201/9781003470939.

Mehboob, B., Chong, C., Lee, S. and Lim, J. (2021) 'Reusability affecting factors and software metrics for reusability: A systematic literature review', *Software: Practice and Experience*, 51(6), pp. 1259–1286. doi:10.1002/spe.2961.

Padhy, N., Satapathy, S. and Singh, R.P. (2018) 'State-of-the-Art Object-Oriented Metrics and Its Reusability: A Decade Review', in Satapathy, S.C., Bhateja, V. and Das, S. (eds.) *Smart Computing and Informatics*. Singapore: Springer, pp. 431–441. doi:10.1007/978-981-10-5544-7_42.

[Permalink](#)

[Show parent](#)

[Reply](#)



Re: Initial Post

by [Stelios Sotiriadis](#) - Tuesday, 12 August 2025, 10:47 AM

Hi Lauren,

Thank you for your thoughtful post and thorough analysis. You've made excellent points, and your selection of references is very strong.

Stelios

[Permalink](#)

[Show parent](#)

[Reply](#)

◀ Initial Post

Initial Post ▶

You are logged in as Lauren Pechey (Log out)

[Policies](#)



