

---

### **Unit 3: Unified Modelling Language (UML)**

---

#### **Summary Post on Discussion Topic: Factors which Influence Reusability**

The concept of software reuse, originating back to Douglas McIlroy's (1968) proposal in the late 1960s, has significantly evolved and gained traction within the software engineering community. It encompasses the efficient utilization of reusable components, design skeletons, and processes to enhance development speed, reliability, and cost-effectiveness, as corroborated by studies from Frakes and Kang (2005) and Haefliger et al. (2008).

In my initial post, I highlighted various factors influencing code reusability, emphasizing the positive impact of modularity, understandability, and other best practices identified by Buse & Weimer (2010). Modularity, in particular, emerged as a fundamental prerequisite for effective code reuse, allowing developers to break down software into manageable and cohesive units. Practices like naming conventions, code comments, and readable code also enhance source code quality and reusability.

Object-oriented software development stresses the importance of prioritizing reusable properties to improve efficiency, maintainability, and overall software quality across projects. The systematic review and prioritization of reusability factors, as proposed by Padhy et al. (2018), including architecture-driven approaches, design patterns, and modularization, provides a structured framework for developers to optimize software quality and efficiency.

Peer feedback highlighted valuable insights into refining the discussion on software reuse. John Heart Ojabo emphasized the need for greater clarity and conciseness, particularly in breaking down the prioritized factors influencing reusability. This feedback underscores the importance of effectively communicating complex concepts clearly and concisely to enhance understanding. Aleksandr Vygodchikov appreciated the historical overview and emphasis on modularity, however, the feedback suggested incorporating practical examples and real-world scenarios to demonstrate the impact of modularity and other factors on software development today.

Reflecting on these peer reviews, I acknowledge the importance of enhancing clarity and conciseness in discussing the prioritized factors influencing reusability. To achieve this, I have strived to provide succinct explanations of each factor in the initial post, however, more emphasis on the factor's significance and its practical implications for software development was needed. Moreover, integrating real-world examples and case studies would enrich the discussion by illustrating the benefits of software reuse in various contexts. For instance, showcasing how modular design and reusable components led to substantial time savings and enhanced maintainability in specific software projects would resonate with readers and highlight the practical impact of these principles.

In conclusion, exploring software reuse and its impact on development efficiency and quality is an ongoing journey. Refining the discussion by clarity and conciseness, and incorporating practical examples and case studies, the discussion would offer a comprehensive improvement. This reflective process underscores the dynamic nature of software engineering and the importance of continuous improvement in communicating complex concepts effectively.

## References:

- Buse, R.P. & Weimer, W.R. (2010) 'Learning a metric for code readability', *IEEE Transactions on Software Engineering*, 36(4), pp. 546–558.  
doi:10.1109/tse.2009.70.
- Frakes, W.B. & Kang K., (2005) 'Software reuse research: Status and future', *IEEE Transactions on Software Engineering*, 31(7), pp. 529–536.  
doi:10.1109/tse.2005.85.
- Haefliger, S., von Krogh, G. & Spaeth, S. (2008) 'Code reuse in open source software', *Management Science*, 54(1), pp. 180–193.  
doi:10.1287/mnsc.1070.0748.
- McIlroy, D. (1968) 'Mass Produced Software Components'. *Proceedings of NATO Software Engineering Conference*, Garmisch, Germany, October 1968, 138-155.
- Padhy, N., Satapathy, S. and Singh, R.P. (2017) 'Utility of an object oriented reusability metrics and estimation complexity', *Indian Journal of Science and Technology*, 10(3). doi:10.17485/ijst/2017/v10i3/107289.
- Padhy, N., Satapathy, S., & Singh, R.P. (2018) 'State-of-the-Art Object-Oriented Metrics and Its Reusability: A Decade Review', in: Satapathy S., Bhateja V., Das S. (eds) *Smart Computing and Informatics. Smart Innovation, Systems and Technologies*. 77. Springer.