

Making use of existing software code and design is called Reusability (Bisht, 2021). Mehboob et al. (2021) further describes reusability of software as its utilisation in another form or another place, and cites that the advantage of reusing and maintaining software instead of creating one from scratch, is that it lowers the production price. Van Atteveldt et al. (2019) argue that both codes and data are accessible if they are recorded properly, making them readily understood, which results in it being easily recreated as well.

Capilla et al. (2019) gathered responses from software researchers and professionals which assets they often reuse. The results will partly be used as the basis in the prioritised list of Reusability Factors referred to in Padhy et al. (2019).

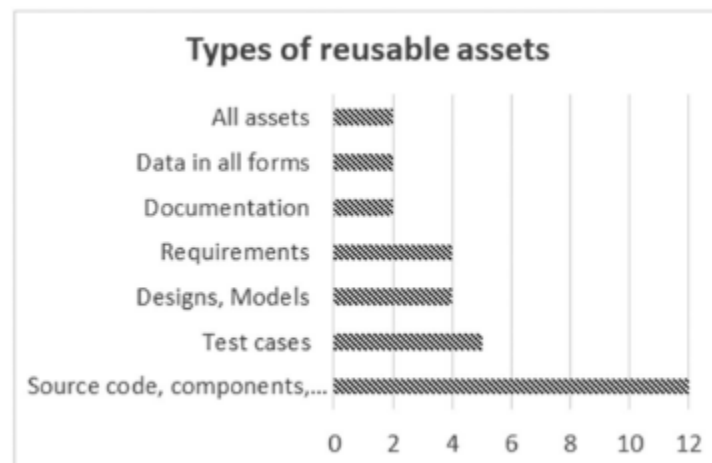


Figure 1: Reusable asset types according to researchers and specialists (Capilla et al., 2019).

As seen on Figure 1, 'Source code, components' lead the preferred reusable assets by specialists, with 'Test Cases' coming in at second, and tied at third are 'Designs, Models' and 'Requirements'. With these in mind, the prioritised list is presented in Figure 2.

Top Priorities

1. Architecture driven approach (ADP)	3. Test cases/test design (TCTD)	4. Knowledge requirement (KR)	6. Models in the project (MP)
2. An algorithm used in the program (AP)		5. Requirement analysis (RA)	7. Design patterns (DP)

Next Priorities

8. Used in the data project (UD)	10. Service contracts (SC)	11. Modules in the program (MIP)
9. Documentation in project (DIP)		

Figure 2: Reusability Factors according to Padhy et al. (2019), arranged in a priority list.

Each factor was explained by Padhy et al. (2019), and basing on that, the reason for their order in the priority list is discussed in the next statements. ADP and AP are in the top 2 because they are the architectural blueprint and algorithm foundation of the software. TCTD is third as it enables multiple testing of designs and cases. RA is fifth because the standard quality is established, but only if the software expert can implement KR, which is fourth in the list. Sixth is MP as it is the part where relevant code is interpreted. DP, which is seventh in the list, can easily be reused if it matches specs. The eighth is UD, since after the 7 earlier listed factors, it can denote reusability.

Ninth is DIP, where the development of software is logged and recorded. Tenth is SC, since after all the previous Reusability Factors are met, only then can service contracts be established. Last is MIP, where instructions are bundled with the application, which should only be given once all the previous properties are tested and executed.

References:

- Bisht, B. (2021) Metric Approach to Anticipate Reusability of Object-Oriented (O-O) Software Systems. *Turkish Journal of Computer and Mathematics Education* 12(6): 2456-2467. Available from: <https://login.uniessexlib.idm.oclc.org/login?url=https://www.proquest.com/scholarly-journals/metric-approach-anticipate-reusability-object/docview/2623922172/se-2?accountid=10766> [Accessed 15 November 2023].
- Capilla, R., Gallina, B., Cetina, C. & Favaro, J. (2019) Opportunities for software reuse in an uncertain world: From past to emerging trends. *J Softw Evol Proc* 31(e2217):1-16. DOI: <https://doi.org/10.1002/smr.2217>.
- Mehboob, B., Chong, C.Y., Lee, S.P. & Lim, J.M.Y. (2021) Reusability affecting factors and software metrics for reusability: A systematic literature review. *Softw: Pract Exper* 51(6): 1416-1458. DOI: <https://doi-org.uniessexlib.idm.oclc.org/10.1002/spe.2961>.
- Padhy, N., Satapathy, S. & Singh, R.P. (2019) 'State-of-the-Art Object-Oriented Metrics and Its Reusability: A Decade Review', in Satapathy, S.C., Bhateja, V. & Das, S. (eds) *Smart Computing and Informatics: Proceedings of the First International Conference on SCI 2016, Volume 1*. Singapore: Springer. Available from: <https://ebookcentral.proquest.com/lib/universityofessex-ebooks/reader.action?docID=5217051&ppg=405> [Accessed 8 November 2023].
- van Atteveldt, W., Strycharz, J., Trilling, D. & Welbers, K. (2019) Toward Open Computational Communication Science: A Practical Road Map for Reusable Data and Code. *International Journal of Communication* 13: 3935-3954. Available from: <https://ijoc.org/index.php/ijoc/article/viewFile/10631/2765> [Accessed 14 November 2023].