Reusability Factors of Object-Oriented Software (Initial Post)

Reusing software components provide many benefits for software development, such as less expenditure, time-saving, reduced cost, less use of resources, and increased productivity and quality.

If I go by the importance of reusability, I will sort the reusability factor list by N. Padhy et al. (2018) in this order:

1. Requirement analysis (RA)
2. Architecture driven approach (ADP)
3. Design patterns (DP)
4. Test cases/test design (TCTD)
5. Documentation in project (DIP)
6. Knowledge requirement (KR)
7. Used in the data project (UD)
8. Modules in the program (MIP)
9. An algorithm used in the program (AP)
10. Models in the project (MP)
11. Service contracts (SC)

Firstly, without the requirements analysis, developers can not find out which software components for software development exist. The architecture driven is second since the architecture structure connects all the software components, and it should be mostly standard for quick changes. Similarly, if the requirements have been changed, reusable design patterns help to save time for modifying the software. Furthermore, test cases/designs are very time-consuming; thus, reusable test cases are essential to developing software efficiently.

In my opinion, the remaining lower factors are difficult to judge since all of them are important to increase the software quality.

References:

Shamil, F. R. (2022) Software reuse and software reuse oriented software engineering. Available from: https://t4tutorials.com/software-reuse-and-software-reuse-oriented-software-engineering/ [Accessed 15 June 2022].

Padhy, D. N., S., Satapathy, S. &. S. R. Singh (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, vol 77. Springer, Singapore. https://doi.org/10.1007/978-981-10-5544-7_42 431-441.