

7. Knowledge requirement (KR): The knowledge generated during the software development life cycle (SDLC) is valuable for promoting reusability. It represents experiences, ideas, and reasoning that can guide future development efforts.

8. Models in the project (MP): Models that represent project tasks, solutions, and insights can provide reusable representations. They enable sharing and reuse of meaningful code and design artifacts.

9. Requirement analysis (RA): Gathering clear and comprehensive requirements is important for understanding and meeting the needs of the system. While it is crucial for project success, its impact on direct code or solution reusability may be relatively lower.

10. Service contracts (SC): Service contracts facilitate reusable interfaces and communication channels between developers and users. While important for effective collaboration, their impact on direct code reusability may be lower.

11. Test cases/test design (TCTD): Reusable test cases and test designs contribute to efficient testing. While important for quality assurance, their impact on overall reusability may be relatively lower compared to other factors.

It's important to note that the ranking can vary depending on the specific context and requirements of a software development project. The provided ranking is a general guideline based on the information provided.