

e-Portfolio Activity:

Reflection on Software Quality. Review the article by McCall, Richards & Walters (1977). Select a timelier academic article on software quality. Discuss, in 300 words, the major differences in relation to software quality between the two articles.

The article by McCall et al. (1977) lays the groundwork for our understanding of software quality. It identifies three key aspects: product operation, adaptation, and transition. Initially, the focus was on product operation, but the study also recognised the importance of long-term considerations, such as maintainability, flexibility, and portability. The article revealed a crucial insight: the costs of maintaining and redesigning a product over its lifecycle often exceed the initial development costs. Additionally, it was noted that rectifying errors after deployment is significantly more expensive than during the testing phase. These findings underscore the importance of considering these factors in software development, as Westland (2002) explains.

On the other hand, as discussed in Fitzpatrick's (1996) report, a more recent perspective on software quality expands and updates these concepts to reflect advancements in software development and the increasing complexity of software systems. Fitzpatrick (1996) revisits McCall's model, acknowledging its foundational role, and integrates newer quality models and standards such as the ISO/IEC 25010. This modern standard introduces more detailed quality characteristics and sub-characteristics, emphasising functional suitability, performance efficiency, compatibility, usability, reliability, security, maintainability, and portability (Estdale & Georgiadou, 2018). These additions reflect the changing landscape of software development, where security and compatibility have gained paramount importance, providing a clear picture of the evolution of software quality perspectives.

The significant differences between the two perspectives underscore the evolution in understanding software quality and illuminate the progress made in the field. McCall et al. (1977) model focused on the essential yet critical attributes necessary for early software systems, stressing the importance of internal and external qualities. On the other hand, the updated models like ISO/IEC 25010 reflect a broader and more nuanced view, incorporating elements like security and compatibility, which have become crucial in today's interconnected and multifaceted software ecosystems.

Additionally, while McCall et al. (1977) model was primarily a theoretical framework, modern quality models are often implemented with practical tools and methodologies that empower real-world software development, testing, and maintenance processes. This shift signifies a move from purely conceptual models to actionable frameworks integrating quality assurance throughout the software lifecycle (Autumn & Hurry, 2023).

In summary, the key difference lies in software quality models' expansion and practical applicability over time. The foundational principles McCall et al. (1977) established have been built upon to create more comprehensive and detailed standards that address the complexities of modern software systems.

References:

Autumn, H. & Hurry, R. (2023) *Quality Assurance in Software Engineering: Best Practices for Success*. Available from: <https://doi.org/10.13140/RG.2.2.35963.28965> [Accessed 30 May 2024].

Estdale, J. & Georgiadou, E. (2018) Applying the ISO/IEC 25010 Quality Models to Software Product. *Communications in Computer and Information Science*, pp.492–503. Available from: https://doi.org/10.1007/978-3-319-97925-0_42.

Fitzpatrick, R. (1996) *Software quality: definitions and strategic issues*. Available from: <https://arrow.tudublin.ie/scschcomrep/1/> [Accessed 30 May 2024].

McCall, J.A., Richards, P.K. & Walters, G.F. (1977) *Factors in Software Quality. Volume I. Concepts and Definitions of Software Quality*. Available from: <https://apps.dtic.mil/sti/citations/ADA049014> [Accessed 30 May 2024].

Westland, J.C. (2002) An Introduction to Software Testing. *Electronic Notes in Theoretical Computer Science*, 148(1), pp.89–111. Available from: <https://doi.org/10.1016/j.entcs.2005.12.014>.