References:

Agrawal, T., Walia, G.S. and Anu, V.K. (2024) Development of a software design error taxonomy: A systematic literature review. *SN Computer Science, 5*: 467. DOI: **https://doi.org/10.1007/s42979-024-02797-2**

Nizam, A. (2022) Software project failure process definition. *IEEE Access*, 10: 34428–34441. DOI: **https://doi.org/10.1109/ACCESS.2022.3161036**

Sallai, D., & Pepper, A. (eds.) (2025) *Navigating the 21st Century Business World: Case Studies in Management*. LSE Press, London.

Scheibner, J., Ienca, M., Sleigh, J. and Vayena, E. (2021) Benefits, challenges and contributors to success for national eHealth systems implementation: A scoping review. arXiv. Available at: **https://arxiv.org/abs/2106.08737**

Verner, C.M. and Sarwar, D. (2021) Avoiding project failure and achieving project success in NHS IT system projects in the United Kingdom. *International Journal of Strategic Engineering (IJoSE), 4*(1): 1–22. DOI: **https://doi.org/10.4018/IJoSE.2021010103**

Permalink **Reply**

Re: Initial Post

by Ruben Marques - Sunday, 25 January 2026, 9:58 AM

I found your point about insufficient risk management particularly relevant. While you focused on the technical side, some authors like Nizam (2022) argue that the "failure process" often starts much earlier with a lack of organisational readiness. If the internal culture does not support transparency, even the most rigorous testing procedures can be bypassed or ignored to meet a deadline.

To build on your discussion of Agrawal's taxonomy, I looked for some other perspectives on these failure points:

While Agrawal (2024) focuses on the "definition" of requirements, other recent studies suggest that the speed of change in modern software environments makes any initial plan nearly obsolete by mid-development. This suggests that the issue might not be "poor planning" but rather a lack of "adaptive planning" (Zhu, 2025).

Regarding your third point, research indicates that teams often fall victim to "optimism bias" during the testing phase. This leads them to underestimate the impact of technical debt until it causes a major breakdown like the RBS incident you mentioned (Arnold et al., 2023).

Given that the Agrawal paper emphasizes a "design error taxonomy," do you think the RBS failure was a result of a fundamental design flaw in the batch processing logic, or was it purely an operational

breakdown during the release management stage?

References

Agrawal, T., Walia, G.S. and Anu, V.K. (2024) ‘Development of a software design error taxonomy: A systematic literature review’, SN Computer Science, 5(7), p. 467.

Arnold, M., Goldschmitt, M. and Rigotti, T. (2023) ‘Dealing with information overload: a comprehensive review’, Frontiers in Psychology, 14.

Zhu, H.N. et al. (2025) ‘From Bugs to Benchmarks: A Comprehensive Survey of Software Defect Datasets’, arXiv.

Maximum rating: -

Permalink Show parent Reply

Initial Post

You are logged in as Ruben Marques (Log out)

Policies

Powered by Moodle

Site Accessibility Statement
Privacy Policy