Unit 1 Collaborative Discussion 1

Project Failures Study

The comprehensive paper by Lehtinen et al. (2014) identifies causal relations between types of software project failures and the causes that lead to each failure. The article considers that the three main reasons for project failure are linked directly to ignoring the importance of software quality, low test scope or coverage, and poor communication between affected stakeholders. In addition to the three stated points of failure, Verner et al. (2008) also identify the impact of project delivery dates that accounts for seventy-five percent of all project failures and poor estimations that account for eighty-one percent of project failures.

A lack of software quality controls for software delivery contributes to an ever-increasing technical debt burden (Besker et al., 2020) which in turn cannot support the development of new features given low-quality underlying implementations. Technical debt impacts developer morale and productivity and leads to wasted time effort, impacting project delivery. Low software quality can be improved through good techniques like unit testing by software developers or automation testing by qualified testers. Lack of testing or low-test scope leads to rework of delivered software implementations which directly impacts the costs of project delivery. Given the two points addressed, we can argue that communication is the most critical key to successful project delivery (Utama & Purwandari, 2020), given that people are involved in all aspects of software projects and require instructions, guidance, and support. Communication must occur regardless of geography, with management leading the way to maintain regular project meetings and reviews. Communication is vital regarding customers, as often projects fail because they do not meet customer requirements.

Overall, the literature for software project failures, unfortunately, does not pin the failure on a single cause, but rather on a multitude of areas such as staffing issues, low morale, inexperienced staff, unrealistic estimates, or lack of risk assessment (Verner et al., 2008).

Example 1: London Stock Exchange, "Taurus"

This example highlights poor estimation, unrealistic delivery dates, technical complexity, unrealistic expectations, poor project planning, scope creep, unstable organisational environment (Abdullah & Verner, 2021). The LSE project was abandoned after ten years of development and a cost of four hundred million British pounds— the original budget was six million pounds for delivery in six months (Thomson Reuters, 2022). The failed project was quoted by a newspaper (The Independent) as "one of the outstanding examples of technological ineptitude of our times". Reasons for its failure was complexity (the manner of

handling share certificates), ambitious requirements ("aimed to achieve many purposes and to keep many groups happy") and failing to negotiate a solution acceptable to all stakeholders.

Example 2: Sydney Opera House

This example highlights the lack of communication plan among stakeholders, unrealistic requirements, and scope creep. The designer, Jørn Utzon of Denmark, did not work closely with the engineering firm Ove Arup and Partners on design decisions such as designing a heavy roof for the foundations or even using poor materials for the roof. Utzon chose to withhold his design thoughts until he had a satisfactory solution. Lack of communication impacted the design process, frustrated relevant stakeholders, and led the project to run over budget to A$102M—estimated initially at A$7M—and took 14 years to complete. Because neither Arup nor Utzon had project management skills, the project is considered a project management failure.

References

Abdullah, L.M. & Verner, J.M. (2012). Analysis and application of an outsourcing risk framework. Journal of Systems and Software, 85(8):1930-1952.

Besker, T., Ghanbari, H., Martini, A. & Bosch, J. (2020). The influence of Technical Debt on software developer morale. *Journal of Systems and Software*, (167):110586.

Lehtinen, T.O., Mäntylä, M.V., Vanhanen, J., Itkonen, J. & Lassenius, C. (2014). Perceived causes of software project failures–An analysis of their relationships. *Information and Software Technology*, *56*(6):623-643.

May, L.J. (1998). Major causes of software project failures. CrossTalk: The Journal of Defense Software Engineering, 11(6):9-12.

Thomson Reuters (2022). Taurus: Learning lessons from failure. Available from https://uk.practicallaw.thomsonreuters.com/1-100-3790?contextData=(sc.Default)&transitionType=Default&firstPage=true [Accessed 15 Marc. 2022]

Utama, D. & Purwandari, B. (2020). Critical Success Factors of Software Projects: The Perspective A Software Company from Indonesia. 2020 6th International Conference on Computing Engineering and Design (ICCED):1-5.

Verner, J., Sampson, J. & Cerpa, N., 2008, June. What factors lead to software project failure? In 2008 second international conference on research challenges in information science (pp. 71-80). IEEE.