

Software Engineering Project Management May 2023

[Home](#) / / [My courses/](#) / [SEPM_PCOM7E May 2023](#) / / [Unit 1](#) /
/ [Collaborative Discussion 1: Project Failures Study](#) / / [Initial Post](#) /



Collaborative Discussion 1: Project Failures Study

Initial Post

⚙ Settings ▾

[◀ Initial Post](#)

[Summary Post ▶](#)

Display replies flat, with oldest first



Initial Post

by [Michael Botha](#) - Monday, 8 May 2023, 11:20 AM

It is very difficult to clearly attribute specific isolated reasons for the failure of software projects (Lehtinen et al., 2014). This due to their complexity as it pertains to the various process areas which need to work correctly together to produce an outcome that meets the specified Key Performance Indicators (KPIs) (Lehtinen et al., 2014). Therefore, 'most common' is a broad term as there are different activity areas where more specific failures need to occur, which would then chain together across process areas to cause an ultimate failure (Lehtinen et al., 2014). Furthermore, 'failures' within a project are common however ultimately the project should accommodate such and still produce success (Goatham, 2020). Therefore, an approach has been to create a list of more generic isolated high-level classes of common failures (Goatham, 2020). Within such a model three common areas identified and which stand out to me are as follows (Goatham, 2020):



- Requirements analysis which is a form of estimation.
- Testing to ensure quality.
- Governance and processes which presents the controls and dashboard of the project vehicle.

An example of a software project collapse where poor initial scoping and overall lack of governance played a pivotal role, was that of the replacement of the Australian State Queensland's health payroll system by IBM (Beyond Software, 2017). With regards to where better testing may have caught some design flaws, the recall of many Model S Tesla vehicles due to flash failures is a good reference (Mlot, 2021).

References:

Lehtinen, A., Mäntylä, 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.

Goatham, R. (2020) Why Projects Fail. Available from: https://calleam.com/WTPF/?page_id=2213 [Accessed 6 May 2023].

Beyond Software. (2017) The Queensland Health Payroll Fiasco. Available from: <https://blog.beyondsoftware.com/the-queensland-health-payroll-fiasco> [Accessed 8 May 2023].

Mlot, S. (2021) Tesla Asked to Recall 158,000 Vehicles Over Flash Memory Failure. <https://au.pcmag.com/cars-auto/84875/tesla-asked-to-recall-158000-vehicles-over-flash-memory-failure> [Accessed 8 May 2023].

Maximum rating: 👍 (1)

[Permalink](#) [Reply](#)



Re: Initial Post

by [Wang Wang](#) - Monday, 15 May 2023, 5:16 PM

Peer Review

Hi Michael, thank you for your insightful post.

Lehtinen's argument resonates well - there is rarely a single, isolated reason leading to project failure. Instead, it is typically a culmination of various mistakes that result in a project's downfall (Lehtinen et al., 2014). For instance,



the analysis of the failed Obamacare Website project highlighted multiple reasons for its failure. It was not a single error but a series that led to its downfall (Anthopoulos et al., 2016).

However, how does this series of errors start? Insufficient quality control can result in a low-quality product. However, poorly designed software could inherently have low testability, which might be the natural starting point of the problems.

Humphrey's data illustrates that as the project size increases, the chances of its success decrease. In the paper, Humphrey pointed out that "the hierarchical management style does not work well for managing large software projects", and in further he concluded as "if we want to have successful large-scale software projects, we must develop a project management system that is designed for this purpose." (Humphrey, 2005)

If we agree with his arguments, can we then deduce that the root cause behind many project failures is a mismatch between the management style and the nature of software development? Alternatively, there is no one-size-fits-all solution for all software projects. We might need to adapt our approach based on each project's unique characteristics to guide it appropriately.

Reference

Anthopoulos, L., Reddick, C.G., Giannakidou, I. and Mavridis, N., 2016. Why e-government projects fail? An analysis of the Healthcare. gov website. Government information quarterly, 33(1), pp.161-173.

Humphrey, W.S., 2005. Why big software projects fail: The 12 key questions. The Journal of Defense Software Engineering, 18, pp.25-29.

Lehtinen, A., Mäntylä, 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.

[Permalink](#) [Show parent](#) [Reply](#)



Re: Initial Post

by [Andrea Paolo Gussoni](#) - Tuesday, 16 May 2023, 10:31 AM

Hello Micheal, thank you for posting. I agree that finding an ultimate failure cause is not possible and should not be, neither is possible to carry out a project without a single failure (Ashkenas, 2003). the aim should be to minimize them as much as possible. Do you think that modern approaches to project management, especially in the engineering field, lack connection with



the development/engineering team? You mentioned a fail in the Tesla project, one of the many they faced, but we can see how the project of the Tesla owner is successful in the long run along with his other major companies (paypal, spacex etc.), a major part of the success, he often states, is having the final objective clear and basing every decision around that (pmconsulting, 2023), furthermore, the management hired to manage projects is often as qualified as the engineers working with them in terms of technical knowledge, shortening the distance between management and engineering team. Given the success rate of the companies mentioned, do you think these could be the two base principles for a successful large project?

N. Ashkenas, (2003). Harvard business review. Available at:

<https://hbr.org/2003/09/why-good-projects-fail-anyway>

PM360 consulting (2003). HOW TO GET PROJECTS DONE WITH ELON

MUSK. Available at: <https://pm360consulting.ie/how-to-get-projects-done-with-elon-musk/>

[Permalink](#) [Show parent](#) [Reply](#)



Re: Initial Post

by [Etkin Getir](#) - Tuesday, 16 May 2023, 10:31 PM

Hi Michael,

Thanks for your interesting post.

I also agree that poor requirements analysis often leads to project failure. I think one of the most efficient mitigations to this is the "Vision and Scope Document" as suggested by Stellman & Greene (2006). Preparing this document as the very first step definitely helps to lock in the requirements, share them with the development team and show the stakeholders in the company that their needs are well understood. This way it also helps to overcome the communication problems. The document also lays the foundation of the project plan.

What other tools do you think may be used to carry out the requirement analysis in an efficient and accurate way?

References:

Stellman, A. & Greene, J. (2006) Applied Software Project Management. Sebastopol: O'Reilly.

[Permalink](#) [Show parent](#) [Reply](#)





Re: Initial Post

by [Michael Botha](#) - Sunday, 21 May 2023, 7:44 AM

Hi Etkin,

I agree that laying out the scope and vision of a project is a good starting point for decreasing scope creep and ensuring all stakeholder priorities are aligned. However, due to various changing variables such as technical challenges, business requirements, and governmental laws a more dynamic approach than keeping the original scope and vision one hundred percent fixed is required (Bigelow, 2019). Agile software development practices do allow for changes in the requirements specified by the product requester (Bigelow, 2019). Therefore, part of the solution would definitely be to adopt such a model.

References:

Bigelow, S. (2019) How to Tame Ever-Changing Requirements in Software Development. Available from:

<https://www.techtarget.com/searchsoftwarequality/tip/How-to-tame-ever-changing-requirements-in-software-development> [Accessed 21 May 2023].

Maximum rating: -

[Permalink](#)

[Show parent](#)

[Reply](#)



Re: Initial Post

by [Michael Botha](#) - Sunday, 21 May 2023, 8:52 AM

Hi Wang,

I would say that the starting point of a project failure is not fixed, just as the sequence of errors is not (Lehtinen et al., 2014). Picture a nodal system where the weighting of specific failures would add up to produce ultimate failure.



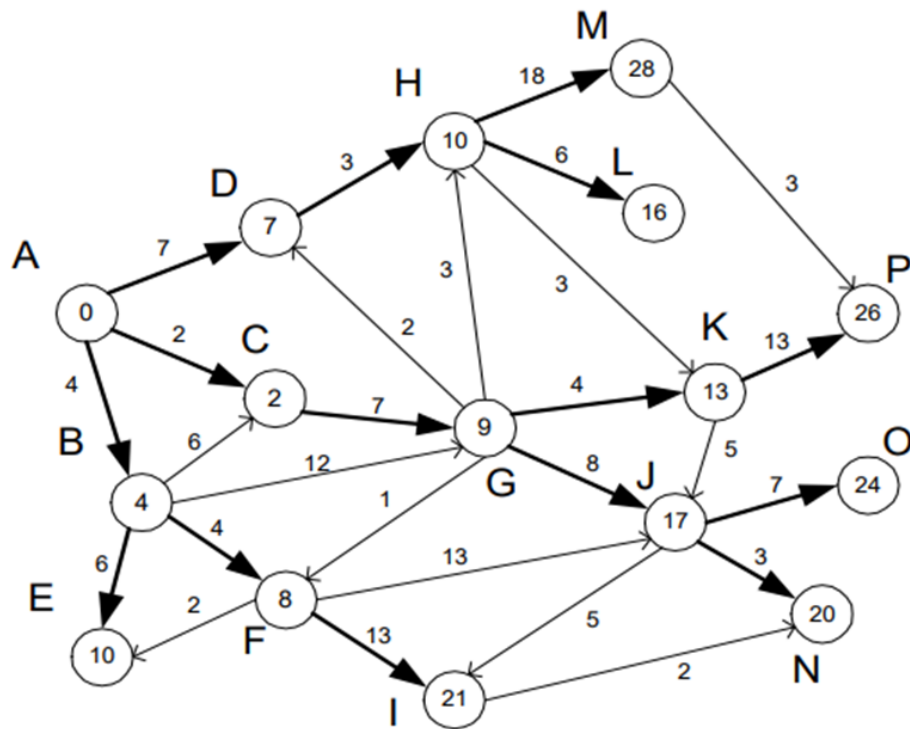


Figure 1 – Nodal System with Weighted Severity (Xiao et al.)

In such a system a multitude of start and end points would exist. Furthermore, perhaps a root cause analysis may be singular in many isolated projects, but if one were to amalgamate multiple of the graphs mentioned it would not be the same outcome. Additionally, a probability graph would be even more complex when using an iterative software development lifecycle.

References:

Lehtinen, A., Mäntylä, 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.

Xiao, B., Zhuge, Q., Sha, E. (2001). Minimum dynamic update for shortest path tree construction. 1. 126 - 130 vol.1. 10.1109/GLOCOM.2001.965092.

Maximum rating: -

[Permalink](#)

[Show parent](#)

[Reply](#)





Re: Initial Post

by [Michael Botha](#) - Sunday, 21 May 2023, 10:07 AM

Hi Andrea,

It would be difficult for me to comment on all the modern frameworks for project management. However, I can say that no approach to project management can prevent a disconnect with any teams members if the following is done by the management:

- Infrequently providing direction (Pallister, 2022).
- Not communicating regularly with the team (Pallister, 2022).
- Disregarding the importance of any member (Pallister, 2022).
- Disregarding feedback from programmers (Pallister, 2022).

Having a clear objective throughout a project definitely seems to be a commonly mentioned goal in literature. For instance, Stellman & Greene (2006) advise on creating scope and vision documents to ensure agreed ultimate goals are achieved. I don't believe it is a realistic target to have all project managers be as competent as specialist engineers in a project team, however, they definitely need to have a high level understanding of all the technical aspects of the project in order to have a real understanding of the needs at any point (Stellman & Green, 2006).

References:

Pallister, B. (2022) Transgressions of a New Product Development Project Manager. Available from: https://www.linkedin.com/pulse/transgressions-new-product-development-project-bradley-pallister-1e?trk=public_post [Accessed 21 May 2023].

Stellman, A., Greene, J. (2006) Applied Software Project Management. Sebastopol: O'Reilly.

Maximum rating: -

[Permalink](#)

[Show parent](#)

[Reply](#)

[◀ Initial Post](#)

[Summary Post ▶](#)

