

Unit 1 Reflection

Project Managers

This week, I considered the differences between traditional project management and software engineering project management. Considering the role of a project manager (PM), I cannot help but see a correlation between their role and that of an enterprise architect: they share similar interests in software projects. For instance, a project manager must be transparent and trust their teams, they also focus on:

- project vision and scope document,
- communicate with relevant stakeholders,
- create project plans,
- risk planning and mitigation,
- act as project leaders,
- establishing teams,
- resource allocation,
- knowledge of the entire software development lifecycle,
- manage project dependencies and prioritise each project, and
- requirements management.

All the tasks above (except for project plans, although I would argue it is still relevant) apply to enterprise architects too. Though, I consider project managers more focused on project *planning* and *scheduling* more than another other task. Project management of software projects is a complex role, and I think it is important for a PM to at least show their competency by taking one or more certification courses such as [CompTIA Project+](#) (great for introduction), [PRINCE2](#) (globally recognised), or even [Certified ScrumMaster](#).

Software Engineering Management

I found quite enlightening the idea that software engineering management is far more complex than traditional project management given rapidly changing requirements and technology, level of creativity, and software complexity. All of which make software engineering management a challenging task, however as Borque and Fairley (2014), state “effective management requires a combination of both measurement and experience” where measurements assign value to software engineering work products and related items together with software development lifecycle models.

Collaborative Discussion: Project Failures

This week I investigated the underlying causes for why software projects often fail. While literature does not point to a single cause, initial investigation shows that stood out to me were related to communication and requirements management. Since requirements form the backbone of any system—there must exist a need—one would suspect that requirements engineering would be a suitable approach which PMs may use to resolve conflicts, vague requirements, or even unrealistic requirements. However, Tamburri et al. (2020) show that “success and failure in software projects is mediated by more than 500 factors”, that requirements engineering is the lowest occurring core concept, and the idea of best practice is the highest occurring core concept related to preventing project failures.

Other

I am thrilled once more to work with fellow students I had the pleasure of working with previously. After the first seminar, I was thoroughly pleased to have a tutor with a sense of enthusiasm and real-world work experience for the topic that will be addressed in this module.

References

Borque, P. & Fairley, R. (2014). Guide to the Software Engineering Body of Knowledge Version 3.0. *IEEE Computer Society Staff*.

Tamburri, D.A., Palomba, F. & Kazman, R. (2020). Success and failure in software engineering: a followup systematic literature review. *IEEE Transactions on Engineering Management*, 68(2):599-611.