Unit 3 Reflection

This unit consideration was given to agile and waterfall in project management. I found interesting the role a project manager plays within the software development lifecycle (SDLC). In this module, I learned about the Volere Template (Robertson and Robertson, 2019). This template is extremely in-depth and easily runs into ninety pages long. Considering this template, I do not think (in agile work teams), the entire template is feasible or should be recommended because it will severely hamper timely delivery in short development cycles. However, I think that Volere template is a brilliant starting point of topics for project managers to consider when applying an appropriate SDLC. For instance, one of the critical requirements for project manager is to provide well-defined customer requirements, both functional and non-functional and for this I consider the Volere template quite supportive.

Considering that requirements are at the heart of any software system, the SMART definition goes a long way to thinking more deeply about customers' requirements in that they requirements need to be *Specific, Measurable, Actionable, Realistic* and *Timely*. To me, SMART requirements shape the way we think about *what* needs to be delivered and *how*. In tandem with SMART, I learnt about the Gherkin syntax for delivering requirements in a behaviour-driven approach. I previously had exposure to the BDD method of developing user stories and find the syntax to flow well with natural thinking of how a given interaction must unfold in the system. Although this unit introduced various alternate SDLC models such as Fish Model, V-Shaped Model, or the Incremental Model (Saravana et al., 2020) I do not find any real-world exposure to these models since Agile and Waterfall seem to be the most well-known, I've encountered through my professional life.

In the concept of SDLC and project management, security must never be considered an afterthought, something to tack onto a project after it's delivered. To support this, I enjoyed reading the paper by Mohammed et al. (2017) in which they consider which stage of the software development lifecycles incorporates software security. They conclude that security is mostly considered in the coding phase. This is a shocking reflection of the software development industry because I would have assumed the outcome to be security-oriented from project planning, scoping, requirements to development and testing.

Lastly, I considered the triple constraint of project management to be an interesting balancing act because customers—were it possible—want delivery of the best quality, widest scope

within the shortest time. It is realistically impossible to deliver on all three constraints, so something will eventually have to give way; compromise in paradise, as it were. I think that, despite the difficulty of balancing these criteria, the overarching decision must always be driven by the concept of Minimum Viable Product (MVP), what can be delivered that has the most functionality which can be demonstrated to customer?

This week I contributed my thoughts to the first collaborative discussion and come away with consideration that requirements and communication as two key factors to prevent project failure. We continued our team discussions this week to determine the content of the customer report. We focused on better understanding the customer's requirements and I drafted the first iteration of the SMART requirements. The team dynamics works well, and each member contributes where possible. I enjoy listening to the ideas expressed by other members as we attempted to address topics such as what is meant by this requirement, how do we deliver on it or the process we took to resolve ambiguity.

References

Mohammed, N.M., Niazi, M., Alshayeb, M. & Mahmood, S. (2017). Exploring software security approaches in software development lifecycle: A systematic mapping study. *Computer Standards & Interfaces*, 50:107-115.

Robertson, J. & Robertson, S. (2019). Volere Requirements Specification Template. Available from https://www.volere.org/templates/volere-requirements-specification-template/ [Accessed 5 April 2022]

Saravanan, T., Jha, S., Sabharwal, G. & Narayan, S. (2020). Comparative Analysis of Software Life Cycle Models. *2020 2nd International Conference on Advances in Computing, Communication Control and Networking* (ICACCCN):906-909.