ePortfolio URL

https://micjustus.github.io/essex-eport2/modules/m05-sepm.html

---

End of Module 5 Reflection

Total word count (excluding headers and captions): 1043

Software Engineering Project Management

I enjoyed studying the content of this module. Standout topics were a project manager's influence on project success (risk assessment, project planning and resource estimation) and the impact of software quality on users' experiences. These two topics have impacted my professional outlook on software delivery and given me a deeper appreciation for the role of a project manager. Before taking this module, I had little understanding of project management in the context of software engineering. However, I now realise the vital role they play in guiding successful projects (see *Unit 1 Reflection: Project Managers*).

In addition, our collaborative discussions identified that project failures are multi-pronged, with no single source identified as the root cause of such failures. For example, offshore development teams introduce culture and communication risks that project managers need to manage. But risks such as intellectual property rights or political instability are out of a project manager's control. Despite this, based on Verner et al. (2014), communication (and management thereof) stands out as the single biggest concern in software engineering project management.

Team Project – Communication Management

The first half of this module was enjoyable as we worked well together: discussing experiences, Agile practices and how to refine customer requirements using Gherkin and SMART (specific, measurable, actionable, realistic, timely) techniques. Team morale was upbeat, and I believe this attitude helped each member perform excellently. However, the latter half was more challenging as the focus switched to software development. When we started the implementation phase (after requirements analysis), the team got caught up with analysis paralysis, which impacted my collaboration level because it was mentally

exhausting. I observed the impact of neverending monologues on team members and how we drifted into a stupor, effectively ending the free flow of ideas. When a gap arose, I raised a topic question to other team members to combat the sense of drift.

I contemplated the role of project managers in managing stakeholder communications to ensure project success. Manata et al. (2022) state that project managers are responsible for high levels of collaboration within teams. Since our study team lacked a designated project manager, we did not consider the risk of unbounded discussions: we merely continued talking regardless. However, even without a designated project manager, I think a form of leadership would have led to more fruitful conversations. Indeed, some studies reveal that leadership is performed by a Scrum Master (Moe et al. 2010), while other studies suggest that the whole team should take leadership (Srivastava and Jain, 2017). Understandably, the lack of experience among the team members led to none being fully prepared to act as Scrum Master or project manager.

Team Project – Experience

Along with communication concerns, developer-oriented team members did not focus on delivering a minimum viable product (MVP)—useful for startups and study groups (Lee and Geum, 2021). Opting instead into the notion that we were developing a paid-for product. This idea was troubling. Why? We were operating in a study environment, and, in my opinion, the need was to focus on *what* is software engineering project management rather than the *act of writing code*.

Considering the *what*, we used a requirements management tool (ClickUp) which no team member (save one) had prior experience with. This posed a problem: it was free and easy to use, but the assigned Product Owner did not fully understand operating it. We held a team meeting (6 April 2022) to work collaboratively with ClickUp and better understand how requirements management worked. Once settled, we were able to create stories and schedule development tasks.

Another experience-related issue concerned developing data structures. I proposed to use the most straightforward design possible (MVP principle). MVP favours *working solutions* above *technically perfect solutions* which take longer to deliver and cost more. Given our time and experience constraints, I thought this approach was sound.

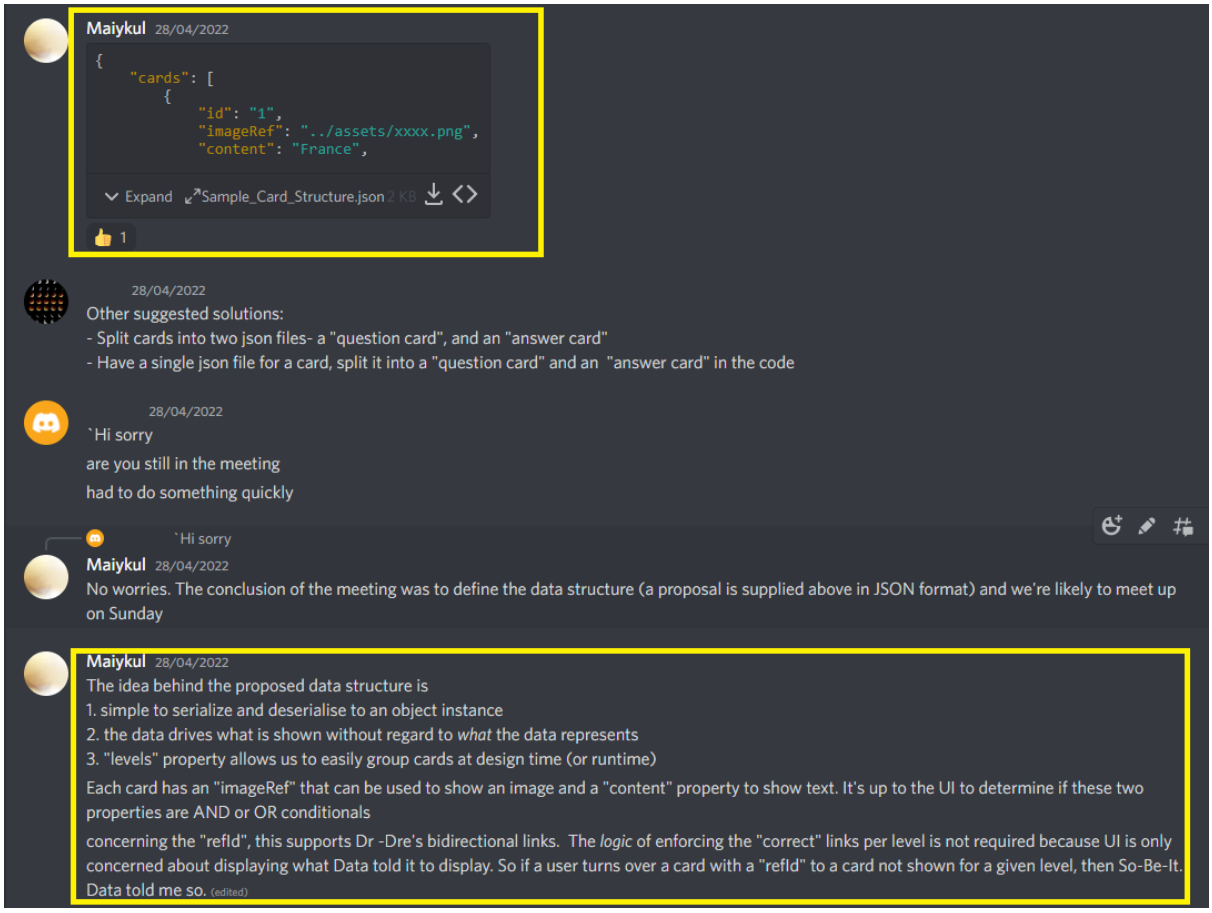

Figure 1 Discussion of data structures (Michael Justus, 2022)

However, this simplistic approach was not favoured. Why? Because the team raised concerns related to single responsibility, data quality management, presentation concerns mixed with data, and scalability. These are excellent points for consideration in light of software quality, which I advocate for in real-world environments.

Our discussions evidenced that relational thinking is a powerful concept in software development teams. This is both good and bad. Good, because it demonstrates that the team considered data normalisation and single data concerns upfront, ensuring clean data structures. Bad, because based on expert judgement, (1) data can be stored as documents (without schema), and (2) database technology must not dictate the ability to meet requirements. Suitable data structures are only one part of successful solutions; the other relates to good software quality.

Software Quality

This module introduced several facets of software quality, such as usability, modularity or usability (McCall et al., 1977), including considering various factors affecting users' experiences with software systems. These topics shaped my view of what good quality

software means, that it is not just visual appeal. Certainly, aesthetics play a role, but our collaborative discussions highlighted that the influence of aesthetics diminishes over time—functionality over form, as it were. Despite this, human emotions are also influenced by the entire product journey (Holbrook and Hirschman, 1982), which helps form users' opinions about a product's quality.

Therefore, I considered my own user experience in the ePortfolio, specifically its usability. So I enhanced unit areas, improved the hamburger menu and sidebar, and updated colour themes. And now? I enjoy this layout more than previous modules because the user experience feels better: beautiful and still functional. After all, I acted as a customer of the ePortfolio.

Figure 2 Update styled for module 5

Closing Remarks

This module opened my mind to the broader scope of project management and its role in initiating, governing, communicating and managing quality software delivery. It also gave me the experience of working with team members with differing views and experiences.

Our use of a modified version of the Scrum process was suitable given constraints such as time and roles. I was more involved in documentation content and review in the team—ensuring consistent layout, message, and tone— than software development. I considered the *what* (process, abstractions) more important than the *how* (code, physical). Customer engagement was a weak point in this project due to the experience level of the Product Owner role.

In summary, I believe that focus on software quality factors, teamwork and stakeholder communication go hand in hand with delivering successful projects.

References

Holbrook, M.B. & Hirschman, E.C. (1982). The experiential aspects of consumption: Consumer fantasies, feelings, and fun. Journal of consumer research, 9(2):132-140.

Lee, S. & Geum, Y. (2021). How to determine a minimum viable product in app-based lean startups: Kano-based approach. *Total Quality Management & Business Excellence*, 32(15-16):1751-1767.

Manata, B., Miller, V.D., Mollaoglu, S. & Garcia, A.J. (2022). Documenting the Interactive Effects of Project Manager and Team-Level Communication Behaviors in Integrated Project Delivery Teams. *Project Management Journal*, *53*(1):33-48.

McCall, J.A., Richards, P.K. & Walters, G.F. (1977). *Factors in software quality: concepts and definitions of software quality*. Rome air development center, Air force Systems command.

Moe, N.B., Dingsøyr, T. & Dybå, T. (2009). Overcoming barriers to self-management in software teams. *IEEE Software*, *26*(6):20-26.

Srivastava, P. & Jain, S. (2017). A leadership framework for distributed self-organised scrum teams. *Team Performance Management: An International Journal*.

Verner, J.M., Brereton, O.P., Kitchenham, B.A., Turner, M. & Niazi, M. (2014). Risks and risk mitigation in global software development: A tertiary study. Information and Software Technology, 56(1):54-78

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