ePortfolio: https://michaelbotha-repos.github.io/ePortfolio-MSc/#

Reflections

Week 1 and 2

It was helpful to gain insight into the various aspects of a software project through the prescribed readings, as now I better know what to expect when performing the duty of a Software Engineering Project Manager. Software Engineering Project Management (SEPM) differs from other engineering project management as the requirements and deliverables are much harder to calculate. Software development can often incorporate solving problems whilst figuring out what the problems actually are. Hence the success of the iterative agile development methodologies.

The immensity of some of the historical software project failures in terms of money, and other resources is astounding. It was interesting yet understandable to learn that software projects don't have certain isolated and well-defined causes of failure, yet various issues chained across differing process areas produce ultimate failure. Good overarching processes which measure well defined and agreed upon Key Performance Areas (KPAs) and Key Performance Indicators (KPIs) at suitable intervals will definitely go a long way in keeping a 'finger on the pulse' of any project, as well as help to pick up issues as early as possible. However, various aspects need to function correctly to ensure success. Therefore, the buy-in and ownership of all stakeholders will be key for me to achieve when working as a Software Engineering Project Manager.

Meeting with my team was frustrating as no one has read the assignment brief or the required reading. In a corporate setting this would not be acceptable as it wastes time and does not invoke confidence in stakeholders. However, I acknowledge in a corporate setting there are structured roles and responsibilities prior to any engagement which would foster a more structured environment early on.

Even though a good Software Engineering Project Manager does not need to be the strongest technically, a good understanding of Version Control Systems (VCS) is pivotal when wanting to leverage modern SEPM tools. Additionally, to be better prepared for my current employment as a Software Engineer a firm grasp of VCSs is pertinent. Therefore, I studied the “Git and GitHub for Beginners” course offered by Google, as I noted that VCSs are not currently covered by any of my MSc modules. Most modern engineering fields have their equivalent to Software Engineering’s VCSs. The course has better equipped me to manage large coding projects which have schedules, tasks, and many collaborators working across the world in different time zones.

Requirements analysis is certainly a skilful activity as it requires drawing out what is required from the customer, as well as better informing the customer by providing options in terms of what is technically achievable within the given constraints. Moreover, it is important to not over commit to various features. Therefore, the agile product lifecycle makes more sense ass it iterates and grows each time. However, it may add complexity to project planning if too many engagements are made with a customer.

I like that Behaviour Driven Development (BDD) tries to establish a ubiquitous language between all stakeholders, non-technical and technical, which can be used to drive development with executable specifications. Thereby, preventing miscommunication between “customers” and “providers”. For instance, using the Gherkin language to create a set of clear requirements for a specific scenario using a certain feature of the software. Furthermore, testing that the requirements are met with a behavioural driven focus rather than purely technical outputs.

Unit 3

Estimation is a very difficult task, because it requires the consideration of various variables, many of which are not known in the early stages of a project. It would be beneficial if a database of common tasks was created that could be referenced by the software engineering community. But even then, there is still a lot of variability which needs to be accounted for. I have experienced the concerns mentioned in the “Mythical Man Month”, where trying to assign more people to a task can create a lot more complexity and inevitably slow the entire project down. However, communication media and project management tools have improved significantly since the book was written. Therefore, I believe that the management of such a requirement is much more possible should good onboarding processes exist.

Unit 4

Considering the various factors which attribute to failed projects it becomes obvious that a risk mitigation strategy is required. Although Enterprise Risk Management

(ERM) is the current trend, Anton & Nucu (2020) conclude in the prescribed reading that the determinants of such are not yet clear. Therefore, making an ERM difficult to formulate. However, using structured risk management frameworks like OCTAVE will certainly cover many requirements of risk management. My experience is always the difficulty of amalgamating the various risk frameworks into current project governance structures as many engineers are not interested in the 'administrative' aspects of projects and business.

I find the Planning Poker approach to estimation to be the most attractive estimation methodology as it incorporates many stakeholders. However, I acknowledge that it might not be the most sound technique, considering that every project is unique.

It was required to allocate team members in the assignment group to different tasks as it is not efficient to have everyone work on the same tasks. However, this adds complexity as it requires the oversight of some sought of authority to ensure deliverables are met, and a certain level is achieved. This in a normal setting would be the Project Manager (PM) and the high-level managerial staff he/she reports to. Moreover, it is difficult to assign tasks in a university team setting as nobody wants to be perceived as usurping any other. I find it helpful in these sought of situations to suggest tasks to individuals or subgroups, thereby not being perceived as assuming leadership.

Unit 5

Learning about the complexities of User Experience reiterates the importance of good requirements analysis and keeping all stakeholders involved throughout the development process. However, this still does not entirely resolve the difficulties of producing software products which satisfy the range of human backgrounds and expectations which can be very dynamic. Therefore, it is necessary to ensure strong contractual scope and clarity to prevent never-ending cycles of development. Furthermore, because one’s emotions and perceptions change it would be wise to encourage users to spend time with a system to overcome any misjudgements due to aesthetic characteristics of such.

As previously covered in the first discussion – the article on governance reiterated the obvious requirement of such, however in the context of an open-source environment which lends itself to a very different project setting than one would find in a typical business environment (Gardler& Hanganu, 2013).

Completing the Scope of Work (SoW), Work Breakdown Structure (WBS), and Scheduling for the hardware section of the assignment was challenging because one needs to take into account various aspects. Furthermore, there are multiple unknowns when considering manufacturing which through personal experience is a massive aspect to account for. Additionally, there are many facets such a tooling which is assumed during the assignment and forms an important part of any project.

Unit 6 & 7

Through the assignment I have been able to understand how much work a large-scale project would require, particularly if there are other key business outcomes to balance simultaneously with the task of project managing, or if a team does not have full investment from all stakeholders. Additionally, I increasingly see the importance of predefined governance structures and project roles.

Wyngaard et al. do well by stating that “the triple constraint constitutes one of the primary building blocks of the project plan”. More recently I have experienced this firsthand at work in a startup company, where various pressures exist that push one to review projects scrupulously in terms of where an upper hand may be gained by affecting “scope”, “time”, and “cost”. Generally, I believe that scope in terms of documentation and security are neglected. Additionally, a fair bit of technical debt is gained as there is just not enough time to sit for weeks to plan designs thoroughly, especially because the Research and Development (R&D) environment is one where there are various uncertainties and expending too much energy on uncertain capital investments could be harmful to the business. A Technical Debt management approach like that suggested by Ramasubbu and Kemerer may prove important once the organisation settles.

Testing software requires a skilful balancing of scope and time with cost. Furthermore, scenario prediction is typically a difficult facet particularly when dealing with complex systems. The Pytest module can help testing, however requires well-designed test cases.

Unit 8

Revising data structure concepts reiterated the foundational importance they form in the field of Computer Science (CS). To make a program secure, efficient, modular, maintainable, readable, and so on, it is important to select appropriate data structures and their interfaces. However, I have found at a practical level the more complex structures taught and spoken about like graphs and trees are not commonly required in application design. Their workings may be below the high level of abstraction found in the many popular programming languages and environments. Generally, the inefficiencies of selecting less optimal data representations are less impactful than in the past, due to the superior processors and Random Access Memory available today. Therefore, one can typically gain a lot more time during projects by not overly focussing on the efficiency of the user-defined structures that will be used. Especially, if one is certain that the program will not need to scale much higher and process large amounts of information.

Summarising the user experience discussion exposed how diagrams in scientific literature are not always capable of conveying all necessary points without becoming complicating and messy. I will endeavour to make diagrams simple and clear, capturing the main points and essence of the concepts to be presented.

Unit 9 and 10

Setting up a shared repository and working with Andrea on the application portion of the assignment reiterated the necessity of VCSs, and how they can improve collaboration during a project. Additionally, they provide a way to measure activity and keep records of the development of the project. I continue to learn the broad functionality of Git.

After noting the lack of input from two of my group members, I suggested to the rest of the team that we reach out to the module tutor. Leaving problems unattended cause bigger issues down the line, therefore it is always better to resolve concerns as soon as they are noted as such. I sent out an email to Doug who informed everyone that he would attend our weekly meeting. All members attended the meeting this time, and it worked out well, as many questions could be discussed with Doug. It is always very difficult to bring a project to satisfactory conclusion without the buy-in of all stakeholders. To improve project commitment governance structures will need to be well enforced by the PM and upper management.

It was interesting for me to learn that the definition of quality is subjective in the sense that it is related to how a product can fulfil a set of requirements. Therefore, what is deemed as quality for one customer is not so for another. Consequently, requirements analysis bares more importance than just functional scope, but quality necessities too. However, there would always be a need for a baseline quality requirement.

Technical debt is a consideration of great importance within the start-up sector which I work in. One needs to carefully consider timeframes around release dates as customer confidence relies on a company’s ability to deliver. Therefore, it is often necessary to provide a solution which is not optimum, however, provides a minimum viable product.

I find using built-in linters within the Integrated Development Environment (IDE) very useful as it decreases the time required to organise code. Furthermore, it can decrease cognitive load whilst debugging, as often clearly shows errors. However, not all languages and environments have the same level of support. For example, many JavaScript developers have reverted to using TypeScript which provides static typing to JavaScript, allowing better linting capabilities. When setting up a development environment for a new project I will need to consider these facets very carefully, as they affect development time drastically.

Unit 11 and 12

During this week's studying it has become even more apparent that there is difficulty in assigning metrics to software project performance. This may be due to the fact that software engineering is a newer engineering discipline, however, one also needs to consider how producing software relies on hard-to-measure human traits such as creativity and ingenuity. Trying to better estimate and track software projects will certainly continue to be an important future topic. Personally, I am still surprised at how long certain programming tasks take me, especially when I had estimated a much shorter required time to complete. I will continue to fine-tune this self-management skill, however, I acknowledge that some tasks are just very difficult to predict, and planning in and of itself may also consume unnecessary time for smaller tasks.

Tool chains are well known by programmers to drastically increase developer performance. Some programming sectors have advanced support mechanisms for development, such as web development. I continue to learn about appropriate tooling within specific development fields. For instance, in web development Vite is very popular as it helps set up software directory structures and hosts a website without the need for a webserver. Furthermore, it updates the frontend automatically so that your development is not hindered by having to keep building a project and rendering it in the browser. With regards to embedded programming many chip manufacturers provide a suitable IDE with built-in features supporting debugging and building code for the relevant platform. However, I have found that the linting is not as easy to setup and may not be as helpful due to how the code ties into the hardware, which slows development down. Thorough thought needs to go into what would be the most applicable development environment and support tools, as this will make a massive difference on the success of a software project, as well as future maintenance options.

Through my preparation for the last seminar, I was convinced that secure software development will be one of the most influential future SEPM trends. This due to various factors, with some of the pertinent ones listed below:

- Government requirements
- Loss of revenue from Cybercrime
- Increased Cybercrime
- Pervasiveness of interconnected devices

Secure development requires a paradigm shift, where just getting the job done is not enough. I will need to focus on the various ways an application can be susceptible.

Furthermore, a SEPM methodology which focuses on security throughout the lifecycle of a project will need to be chosen.

The main thread throughout the Assignment case study was the adaptability of a project team to the problem landscape. It is important to be able to accommodate changes and come up with solutions which can benefit all stakeholders. Therefore, one needs to keep an open mind when brainstorming solutions, trying to envisage how the solution may affect all parties.

References

Anton, G. & Nucu, A. (2020). Enterprise Risk Management: A Literature Review and Agenda for Future Research. *Journal of Risk and Financial Management* 13(11): 281.

Gardler, R. & Hanganu, G. (2013) Governance Models. Available from: http://oss-watch.ac.uk/resources/governancemodels [Accessed 28 May 2023].

Ramasubbu, N. & Kemerer, C. (2018) Integrating Technical Debt Management and Software Quality Management Processes: A Framework and Field Tests. IEEE/ACM 40th International Conference on Software Engineering (ICSE). 883-883.

Wyngaard, V., Pretorius, C., & Pretorius, L. (2012) Theory of the triple constraint — A conceptual review. IEEE International Conference on Industrial Engineering and Engineering Management. 1991-1997.