



Module Reflection

Course: MSc Computer Science

Module: Software Engineering Project Management

Assignment: ePortfolio

Date: Sunday 20th February 2022

Student ID: 126853

E-Portfolio Links:

E-Portfolio: <https://eportfolio.kieronholmes.me/modules/software-engineering-project-management>

GitHub Branch: <https://github.com/KieronHolmes/UoEoPortfolio/tree/sepm-build>

Reflection:

Throughout this module, we have been introduced to various approaches to Project Management within the Software Engineering sector, contrasting these to the more traditional forms of project management that some of us were more familiar with. Within my workplace, we tend to use a waterfall-style approach to project management, using prior experience as a key factor in estimations provided. This module has helped me understand the benefits of using alternative project management styles, such as an Agile-based approach, due to its increased adaptability and overall reduced risk (Wrike, n.d.).

Within the Unit 11 assignment of this module, we were required to project manage and develop a prototype 'toy' for Group 2, acting as our clients in this scenario. As a group, we compared the positives and negatives of commonly used project management techniques, such as Agile, Scrum, Kanban and hybrid approaches such as Scrumban. In the end, it was decided that we would be using the Scrumban methodology, as it would provide us with a visual Scrumban board, combined with the flexibility provided with an agile-style approach to delivery (Kukhnavets, 2019). Compared to alternative methodologies, I believe this methodology allowed us to be more reactive to feedback from team members and the clients, making changes quickly where necessary. In general, as a team member, I found the

Scrumban methodology to be incredibly useful, as it allowed us to visually track our progress to achieving completion of our client's key requirements, whilst identifying areas that needed additional resources to achieve completion in the agreed timeframes.

Within my current role as a software developer, most of my project estimations are based upon prior experience, which can be compared to the Expert Judgement technique introduced within the Unit 7 lecture cast. When working as part of a larger team aiming to achieve a common goal, it became evident that this technique became less effective when working in areas where the developers had less experience. As such, we opted for a hybrid approach between Planning Poker and PERT which allowed us to draw upon the experience of all team members, identifying the most likely timescale for project fulfilment, which was calculated to be a total of 42 days (Holmes, 2021). Due to the perceived accuracy of the PERT/Planning Poker estimate, this is a technique that I am looking to use within future large projects I am part of.

One item covered within this module that I found to be very useful is the topic of Code Quality and the utilities available to help ensure that a standardised approach is always taken towards code development. One example of this is the Python PEP8 Style Guide which is a constantly-evolving style guide used within the main Python distribution (Van Rossum et al., 2001). To ensure that our code produced within the Unit 11 assignment benefitted from the items highlighted in the Code Quality aspects of this module, we opted to use the Black linting tool, automated within our GitHub repository using a series of automation available within the GitHub actions utility. Although team members had verbally agreed that this was the style guide of choice, enforcing it at the repository level had the benefits of ensuring that team members could not commit non-conforming code to the main repository branch. As this proved to be very beneficial, I will be transferring the knowledge

and tools gained from this module towards other projects and personal work I will be undertaking.

During this module, we were assigned to a team comprising of a total of 7 members (including myself) with different levels of experience within the industry (and located in different sections of the world). When acting as a Project Manager at sections of the module, I found this to be quite a difficult aspect to manage, specifically because of the language localisation differences and timezone differences. All of the previously mentioned factors were required to be taken into account at various points, including the organisation of team meetings & the delivery of the final presentation. However, I believe this is a valuable aspect of the module as this is a scenario that a software engineer is likely to experience when they break out of the SME sectors.

Although the techniques taught throughout this module provided an incredibly valuable insight into project management techniques, I believe that certain aspects won't be directly transferable into my existing working practice. For example, some of the projects I am currently working on within a professional environment contain an extremely limited number of team members (<5), often from different disciplines or organisations, which make estimations and planning more difficult than if it was all contained within the department. I feel as if introducing an element of multi-team collaboration for part of the module assignments would have helped alleviate this concern, providing a more representative example of how this knowledge could be transferred into a more realistic working environment. Despite this, I think the module as a whole has provided a valuable insight into the various estimation and planning techniques that are likely to be encountered throughout the industry.

Based on my findings in this particular module, if I were to undertake similar work in future, I would approach it in a slightly different manner. At the first opportunity, I will ensure that I have

fully understood every team members level of expertise and preferences. Earlier on within this module, there was a slight misunderstanding of the requirements of the assignments, as well as a minor disagreement in the overall choice of programming languages and frameworks we would use to develop a prototype for the opposing team. Despite this, we managed to come to a common conclusion and used a combination of the Python game framework, PyGame, combined with a simple SQLite3 database. In general, I was happy with the outcome of this project, as well as the overall quality of the prototype produced.

References:

Holmes, K. (2022) 'Development Team Project: Presentation'. Paper submitted to the University of Essex Online for Software Engineering Project Management.

Kukhnavets, P. (2019) Scrumban: How to Combine the Best of Two Agile Methods?.

Available From: <https://hygger.io/blog/scrumban-combining-the-best-of-two-methods/>

[Accessed 20th February 2022].

Van Rossum, G. Warsaw, B. Coghlan, N. (2001) PEP 8 - Style Guide for Python Code.

Available From: <https://www.python.org/dev/peps/pep-0008/> [Accessed 20th February

2022].

Wrike. (n.d.) The Benefits and Advantages of Agile. Available From:

<https://www.wrike.com/agile-guide/benefits-of-agile/> [Accessed 20th February 2022].