Reflective Assignment: Software Engineering Project Management

e-Portfolio link: https://rob-essex.github.io/portfolio/SEPM/

This course covered dozens of critical concepts over the span of twelve weeks. This has been one of the most intense curricula to date, as I have learned as many technical development skills as project management techniques. The Software Engineering Project Management (SEPM) module has provided me with a roadmap to continue my education, as well as essential resources to which I will refer throughout my career. Ultimately, this course taught me how project management and development integrate to produce effective software. I used Rolf et al.'s "What? So what? Now what?" framework (2001) to reflect upon the SEPM module.

What?

I began this course by reviewing common SEPM methods and software development life cycles (SDLCs) such as agile, waterfall, and V-model (Saravanan et al., 2020). I analysed the strengths and weaknesses of each SDLC, discovering how they could be combined and tailored to specific projects.

The module quickly pivoted to identifying the most common pathologies for project failure. Whilst underlying reasons for failure abound, literature suggested that the most common reasons for project failure were poor estimation, insufficient communication, and ill-defined requirements (Goatham, 2023; Lehtinen et al., 2014).

Through coursework, reading, and group assignments, I learned how to develop and track software requirements through use case modelling, personas, and user stories (Cohn, 2004; Löwgren & Stolterman, 2007). These lessons emphasised the importance of properly defining requirements; not only do requirements dictate development, but they also define a project's success criteria (Stellman & Greene, 2005). Additionally, I learned how to use the Gherkin language to automate requirements testing (Rice et al., 2017). This was an entirely novel concept to me.

This module emphasised the importance of employing sound estimation techniques to identify the time and resources required to satisfy a project's requirements. I learned that combining estimating techniques, such as bottom-up, top-down, and expert judgment (Nasir, 2006), can greatly improve accuracy. Similarly, tracking and managing risk is critical to managing technical debt (Ramasubbu & Kemerer, 2019).

Finally, the class examined the current state of SEPM, and future trends on the horizon. Open-source software, DevOps, and Enterprise Risk Management have become mainstays in SEPM, today (University of Essex, 2023). Reading and instructional materials from unit twelve introduced the next generation of possible game-changes for the field, such as Artificial Intelligence Ops (AIOps), increased use of secure programming languages such as Rust, and greater adoption of Robotic Process Automation (University of Essex, 2023).

So what?

Though I am an experienced project manager, I have never managed a software engineering project. This module showed me that the underlying principles of project management are somewhat universal, such as managing to the triple constraint (Van Wyngaard et al., 2012). However, this module showed that many aspects of SEPM extend far beyond general project management. For example, effective SEPM requires determining the most appropriate SDLC (or combination of SDLCs), selecting the best development methodology such as Behaviour-Driven Development, Test-Driven Development, or Feature-Driven Development (Anwer et al., 2017; Rice et al., 2017), and working with a variety of stakeholders to finalise requirements (Singh, 2020; Stellman & Greene, 2005). This course has demonstrated how SEPM presents a degree of complexity – and potential for failure – not found in other project management disciplines.

This course was also my first time working directly with a team of experienced developers on a software project. My teammates were all full-time software developers, and the group assignments afforded me the opportunity to witness professional software development first-hand. This experience proved that using industry standard language, processes, and standards could enable students from four different countries to work together on a complex project and deliver effective results.

I also learned that activities I had previously considered as afterthoughts – such as testing and quality assurance – belong in every step of the SDLC, from planning through implementation and sustainment. I also learned how the use of continuous integration/continuous delivery (CI/CD) pipelines can expedite development without incurring technical debt (Ramasubbu & Kemerer, 2019; Swaraj, 2022). This is vitally important when trying to manage project cost, schedule, and customer satisfaction.

DevOps – the philosophy of integrating developers and operations personnel to collaborate on software development (Vadapalli, 2018) – has continued to gain popularity with my clients. Until this course, my experience with DevOps has been squarely from the operations side. My postgraduate studies, and especially this module, have oriented me to DevOps from the developer's perspective. This will help me promote the merits of DevOps, increasing organisational involvement during development and minimising re-work after deployment.

As a closing take-away, I realised that doing the "boring" things right – requirements development, estimation, planning, testing, and quality assurance – creates the best environment for positive outcomes. I believe that effective SEPM requires discipline to develop, and adhere to, a project plan while resisting pressure to cut corners for near-term gains.

Now what?

This course has given me the most holistic view of software development, and SEPM, to date. My future projects will be more mature, benefitting from clearly defined requirements and a project management plan that aligns tasks and sprints to an SDLC with a defined schedule. Some of my previous projects have fallen victim to schedule and feature deviations because I did not have a plan. Instead of beginning a project by writing scripts, I will use an SDLC framework to properly initiate, plan, and execute software development.

My group showed me excellent practices for software development, such as automating CI/CD through GitHub Actions. I will continue to expand my knowledge with CI/CD, learning and integrating automated testing, quality, and security tools.

The group assignments also helped me learn, first-hand, how development teams like to communicate. My teammates were very well disciplined with frequent pull requests and commits to the GitHub version control system (VCS), updating the GitHub Project activity board, and frequent collaboration over Slack. I will apply these same practices to ongoing projects, and improve my habits of recording, documenting, and sharing my task progress with my team.

Finally, I will revisit the lessons and materials from this module and allow myself the luxury of perusing topics in depth. This course introduced a slew of fascinating concepts, and I had to keep my intellectual curiosity at bay to keep pace with the course.

References

Anwer, F., Aftab, S., Waheed, U. and Muhammad, S.S., 2017. Agile software development models TDD, FDD, DSDM, and Crystal Methods: A survey. *International journal of multidisciplinary sciences and engineering*, *8*(2), pp.1-10.

Cohn, M. (2004) *User stories applied: for agile software development*. 1st edition. Boston: Addison-Wesley.

Goatham, R. (2023). *Why Projects Fail – Why Do Projects Fail?* [online] Available at: https://calleam.com/WTPF/?page_id=2213.

Lehtinen, T.O.A., Mäntylä, M.V., Vanhanen, J., Itkonen, J. and Lassenius, C. (2014). Perceived causes of software project failures – An analysis of their relationships. *Information and Software Technology*, 56(6), pp.623–643. doi:https://doi.org/10.1016/j.infsof.2014.01.015.

Löwgren, J. & Stolterman, E. (2007) *Thoughtful interaction design: a design perspective on information technology*. Cambridge, Massachusetts: The MIT Press.

Nasir, M. (2006). *A Survey of Software Estimation Techniques and Project Planning Practices*. [online] IEEE Xplore. doi:10.1109/SNPD-SAWN.2006.11.

Ramasubbu, N. and Kemerer, C.F. (2019). Integrating Technical Debt Management and Software Quality Management Processes: A Normative Framework and Field Tests. *IEEE Transactions on Software Engineering*, 45(3), pp.285–300. doi:https://doi.org/10.1109/tse.2017.2774832.

Rice, B., Jones, R., and Engle, Jens. (2017) *Behavior Driven Development — behave 1.2.6 documentation*. [online] Available at: https://behave.readthedocs.io/en/stable/philosophy.html#the-gherkin-language [Accessed 18 Feb. 2023].

Rolfe, G., Freshwater, D., Jasper, M. (2001) *Critical reflection in nursing and the helping professions: a user's guide.* Basingstoke: Palgrave Macmillan.

Saravanan, T., Jha, S., Sabharwal, G. and Narayan, S. (2020). Comparative Analysis of Software Life Cycle Models. *2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*. doi:https://doi.org/10.1109/icacccn51052.2020.9362931.

Singh, N. K. (2020). *Behaviour-Driven Development (BDD) Helps in Optimizing the Value of Work Done by Scrum Team*. [online] Available at: https://www.scrum.org/resources/blog/behaviour-driven-development-bdd-helps-optimizing-value-work-done-scrum-team [Accessed 19 Feb. 2023].

Swaraj, N. (2022) *Accelerating DevSecOps on AWS : create secure CI/CD pipelines using Chaos and AIOps.* Birmingham, UK: Packt Publishing.

Stellman, A. and Greene, J. (2005). *Applied Software Project Management*. [Insert Publisher Location]: O'Reilly Media, Inc.

University of Essex. (2023) *Lecturecast: Future Trends.* [online]. SEPM_PCOM7E. Available at: https://www.my-course.co.uk/Computing/Computer%20Science/SEPM/SEPM%20Lecturecast%206/content/index.html#/ [Accessed 18 Feb. 2023].

Vadapalli, S. (2018) *DevOps: continuous delivery, integration, and deployment with DevOps : dive into the core DevOps strategies*. Packt Publishing.

Van Wyngaard, C.J., Pretorius, J.H.C. and Pretorius, L. (2012). Theory of the triple constraint — A conceptual review. *2012 IEEE International Conference on Industrial Engineering and Engineering Management*. [online] doi:https://doi.org/10.1109/ieem.2012.6838095.