Learning Journal

Student Name: Ujas Bhuva

Course: Software Project Management (SOEN 6841)

Journal URL: https://github.com/ujasbhuva/SOEN6841_SPM

Dates Rage of activities: 4th November - 9th November 2024

Date of the journal: 9th November 2024

Key Concepts Learned

Week 1 focused on **Project Closure**, as discussed in Chapter 8. We explored the final project phase and the activities necessary for successful project closure. Key tasks include delivering the final project components, managing source code versions, and archiving project metrics for future reference. A significant part of this process involves reflecting on project outcomes to document lessons learned, ensuring improvements for other projects.

Week 2 introduced **Software Lifecycle Management** from Chapter 9. This chapter covered the phases of software development and the importance of quality assurance at each stage. We examined lifecycle models, specifically the Waterfall model and various iterative approaches like SCRUM and Extreme Programming. While the Waterfall model suits projects with fixed requirements, iterative models allow flexibility for adapting to changes, especially for technology-driven products. This comparison provided insights into which model to select based on project requirements and goals.

By combining these two weeks' topics, we see that effective project closure and well-structured lifecycle management are essential for organized, high-quality software development.

Application in Real Projects

The insights from Project Closure and Software Lifecycle Management are directly applicable to real-world software development. And currently the majority of the organizations use these approaches. Project Closure methods are vital for organized project wrap-up and learning from past projects. This systematic documentation helps avoid mistakes in future projects. The choice of lifecycle model is essential in modern software engineering—iterative models like **SCRUM**

allow for adaptive planning and continuous improvement, beneficial in fast-evolving tech environments.

Peer Interactions

This week, I engaged with classmates to discuss the practical challenges of software lifecycle models in different project types, mainly talking about **AI projects**. These discussions clarified the flexibility and applicability of iterative models in projects with evolving requirements. Collaborating on project closure discussions also enhanced our collective understanding of metrics archiving and version control, important for long-term project sustainability.

Challenges Faced

One challenge was understanding the differences in risk management between lifecycle models. Waterfall's linear approach can lead to rework when requirements change, while iterative models, though adaptive, require frequent reviews that can impact timelines. Another challenge was in Project Closure, specifically in the **metrics data filtration** for archiving, which demands careful selection to ensure meaningful insights without excessive data storage.

Personal Development Activities

To enhance my understanding, I reviewed additional online resources on iterative models like **SCRUM** and watched instructional videos on project closure activities. I explored tools like **Jira** and **Confluence**, noting how they facilitate lifecycle management and closure documentation in **real-world** applications.

Goals for the Next Week

- Gain a deeper understanding of quality assurance techniques in iterative models by exploring case studies.
- Research further on the application of lessons learned documentation to improve project planning.
- Study best practices for implementing data filtration and archiving metrics for project closure.