

## Learning Journal - 4

**Student Name:** Bhrrugu Kothari [40270224]

**Course:** SOEN 6481 Software Project Management

**Journal URL:** [https://github.com/BKothari510/SOEN\\_6841\\_Software\\_Project\\_Management](https://github.com/BKothari510/SOEN_6841_Software_Project_Management)

**Date Range of Activities:** 04/11/2024 – 09/11/2024

**Date:** 09/11/2024

### Key Concepts Learned:

This week's explorations curial domains of project management, with a primary focus on Chapters 8, 9 and 10.

In Chapter 8:

- This chapter discusses the final actions required to complete a project properly. Key activities include assuring the smooth turnover of project resources, and documenting lessons learned. Documenting these lessons allows teams to record essential knowledge such as how to enhance work efficiency, overcome project problems, negotiate with clients, and successfully manage risks.
- Source code version management is an important aspect of software project completion. With numerous changes and problem fixes occurring throughout development, it is critical to determine the final version that will be sent to the customer. Proper data archiving guarantees that important information is available for future initiatives, without irrelevant details, and it may use for future reference.

In Chapter 9:

- This chapter explains software engineering (SE) and its impact on software projects to improve efficiency, reliability, and quality. It contains SDLC phases - requirements, design, implementation, testing, deployment, and maintenance and focuses on tracking metrics like code quality and defect density to monitor progress. The chapter also discusses Agile approaches such as Scrum and Extreme Programming (XP), where Scrum employs iterative sprints, and XP focuses on best practices for code (code quality and technical practices) and user feedback.

In Chapter 10:

- This chapter discusses the fundamentals of client requirements in software projects, with a focus on how they are collected, maintained, and classified. Customer requirements are end users' wants and expectations, as obtained through interactions to better understand their aims. Effective requirement management guarantees that any modifications are properly documented and applied into the design and development process with less rework. The chapter also differentiate between customer requirements (what users want) and system requirements (technical specifications that meet user need). Flowcharts, use cases, and data flow diagrams are the tools used to describe requirements and improve understanding among stakeholders.

**Applications in Real time projects:**

- Ensuring all deliverables are comprehensive and aligned with client expectations.
- Utilizing version control systems, such as Git, to manage code effectively.
- Selecting the appropriate project lifecycle model, such as Waterfall for stable requirements or Iterative for evolving project needs.
- Organizing projects with clear Software Development Life Cycle (SDLC) phases.
- Incorporating quality checkpoints for projects in critical industries.
- Leveraging iterative models in projects involving emerging or changing technology to streamline adjustments and minimize rework.

**Peer Interactions:**

- I participated in a debate about the differences between the Waterfall and iterative models, emphasizing the advantages of iterative models for dynamic technology projects.
- I talked with my peers about how we could integrate quality gates and automated testing to improve QA practices.
- I discussed with the professor how to reconcile quick iterations with project timeframes by categorizing features into critical and non-essential.

**Challenges Faced:**

**Complexity of Processes:** Understanding the step-by-step process for managing final deliverables and ensuring that all project results meet client expectations.

**Model Comparison:** Understanding the subtle distinctions between the Waterfall and iterative models, as well as when to use them in certain project contexts.

**Rework Management:** Adapting theoretical approaches for handling rework to real-world project situations, particularly when transitioning from Waterfall to iterative models.

**Personal Development Activities:**

Used tools like JIRA and automated testing frameworks for quality assurance and risk management, applied theoretical concepts to real-world project simulations, compared Waterfall and iterative models, and improved documentation practices through case study reviews and self-assessments.

**Goals for the Next Week:**

To prepare for the Quiz and go through the book to read Chapters 8 and 9. Moreover, to attend group meetings for project presentation discussions with Teaching Assistant and also to prepare for the final exam.