

## Learning Journal

**Student Name:** Bhругu 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:** 06/10/2024 – 02/11/2024

**Date:** 02/11/2024

### Key Concepts Learned:

This week's explorations curial domains of project management, with a primary focus on Chapters 5, 6 and 7.

In Chapter 5:

Configuration Management (CM) is known for the process of controlling and documenting change to a system in systematic manner for future reference. CM role is to maintaining project discipline. Sources of changes in a software project, including requirements, funding changes, and technological advancements. Risks associated with uncontrolled change and the need for a systematic approach to configuration management. Advantages of CM to a project, such as reducing confusion, establishing order, and ensuring correct product configurations. Elements of a Change Control Policy and the role of the Change Control Board (CCB) in decision-making. Configuration Management functions: identification, control, status accounting, and auditing.

In Chapter 6:

Project Planning is most time-consuming activity in Project management. Components of project planning, including project scheduling, budgeting, manpower planning, and quality planning. Techniques for project scheduling: Work Breakdown Structure (WBS), CPM, Goldratt's, Critical Chain Method. Collaborative aspects of project planning, including the role of peers and effective communication. Considerations for project budgeting and adjustments to goals based on progress and evolving understanding. Project planning in iterative software lifecycle models and the different approach compared to waterfall models.

In Chapter 7:

In this Chapter 7, they briefly explaining importance of Project Monitoring, what are different technique for it and How project monitoring is being carried out in iterative projects. EVM (Earned Value Management) is used to measure both schedule and budget progress for any project.

### Reflections on Case Study/Course Work:

The case studies in Chapter 5 serve as powerful, practical examples of theoretical ideas. From this examination, several important lessons emerge:

- The case study illustrates how a centralized configuration management system can effectively facilitate collaboration among various teams, including internal, external, and offshore groups.
- This real-world example highlights the crucial role of 24/7 availability and strong security protocols in ensuring seamless operations and system integrity.

- By implementing access control measures, the case study demonstrates how restricting permissions to authorized team members helps maintain document integrity, aligning with theoretical principles.
- The study supports the theory of version control best practices, showing how a primary branch simplifies management and streamlines development.
- Developers' practice of maintaining local builds and running tests reflects theoretical discussions, underscoring the importance of pre-commit validation to avoid disruptions in the central build.

**Collaborative Learning:**

Taking part in group discussions was helpful for understanding how to handle the unpredictable dynamics of projects while also building a collaborative learning atmosphere. Working with my peers in library sessions gave us the chance to bring theoretical concepts to life by applying them to real project scenarios, which helped us address practical aspects also. It also improves our estimation techniques. Informal talks during breaks and after class provided additional insights through shared real-world examples, allowing us to question our assumptions and explore solutions together.

**Further Research/Readings:**

In the coming weeks, I plan to dive deeper into advanced configuration management techniques and Agile project planning. I'm especially interested in exploring version control systems like Git, which will help me understand configuration management more thoroughly and how it ties into DevOps practices like CI/CD pipelines and monitoring. I hope my upcoming readings will give me more practical insights into industry best practices for modern software development. I also intend to look at additional resources beyond the course to gain a more comprehensive grasp of configuration management.