**Learning Journal 2**

**Student Name:** Harsh Tank

**Course:** Software Project Management (SOEN 6841)

**Journal URL:**<https://github.com/harsh-tank/SOEN-6481-SPM>

**Dates Rage of activities:** 23/09/2024 to 04/10/2024

**Date of the journal:** 05/10/2024

**Key Concepts Learned:**

**In Chapter 4 on Risk Management,** I delved into how risks can significantly influence project outcomes. I now understand the importance of systematically identifying, analyzing, prioritizing, and controlling risks. Project schedules should not be allowed to cross targeted release dates to avoid business opportunity loss. I also realized that risks stem from various sources, such as resource constraints, service interruptions, or outdated technologies. Risk is essentially a combination of its likelihood and the potential impact on the project. I explored key response strategies, including risk acceptance, transference, mitigation, and avoidance. The core lesson here is the need for proactive risk management to maintain project health.

**In Chapter 5 on Configuration Management (CM),** I learned the critical role CM plays in managing changes, version control, and ensuring the integrity of deliverables in software projects. CM involves several components: identification of configuration items, change control, status accounting, and regular audits. The change control board (CCB) designated for each project will decide which changes to implement. This structured approach helps maintain order amidst evolving project requirements, reducing the likelihood of errors and inefficiencies.

**Application in Real Projects:**

In real-world projects, risk management techniques are essential, particularly in high-uncertainty scenarios such as implementing new technologies. For instance, I would integrate risk assessments during the planning phase to identify potential risks and incorporate buffers in timelines. A creative challenge is the use of risk prioritization models to focus efforts and resources on the most critical risks, potentially averting project delays or quality issues.

Real projects conduct comprehensive risk assessments at the beginning of project development and iterations. This involves identifying and analyzing risks based on their likelihood of occurrence and impact on project, product, and business objectives.

These practices are especially important in large teams to ensure consistency and prevent conflicts across different project versions.

**Peer Interactions:**

We discussed the presentation of the group and the topics they had chosen for the project, what functionalities they though to add to the existing systems

A peer emphasized the importance of early risk identification, sharing their experience of how overlooked risks led to significant project issues. This helped me reinforce my understanding of early-stage risk prioritization and mitigation. Another peer shed light on how configuration management complements modern DevOps practices, particularly in continuous integration (CI).

**Challenges Faced:**

One challenge I encountered was understanding the mathematical aspects of risk prioritization, particularly in calculating risk exposure and translating those figures into actionable project strategies. Additionally, balancing strict configuration management processes with the need for project flexibility in fast-moving environments proved to be a nuanced concept that required further exploration.

**Personal Development Activities:**

To further enhance my learning, I watched a video on risk management techniques, which featured real-world case studies and practical examples of risk mitigation strategies. I also explored online courses on Git and version control systems to solidify my understanding of CM, learning practical techniques that align with real-world software development workflows.

**Goals for the Next Week:**

1. Learning Jira, Trello, Asana
2. Set up a small-scale configuration management system for a personal project to strengthen my practical understanding of CM.