**Learning Journal**

**Student Name: Alina Budhathoki**

**Course:** Software Project Management

**Journal URL:** https://github.com/aleenabudhathoki/SOEN6841/blob/main/LJ2.docx

**Dates Rage of activities:** 23 Sept – 5 Oct

**Date of the journal:** 5 October 2024

**Key Concepts Learned**

In this section, we learned what risk is, and how we can identify them. They can be identified as internal and external risks. The major risk types are resource, technology, budget, quality and time. Some of the major causes of these risks are scope creep, resource unavailability, cost constraints, bad negotiation etc. Risk can be assessed in 3 phases; identification, analysis and prioritization. Once the risks have been identified and prioritized, risk mitigation tasks can be carried out accordingly. We also discussed various risk response strategies such as avoidance, transference, and mitigation.

Another concept we learned in this section is configuration management. Configuration management is all about the changes we make in our system and how we can control and document them. Changes can be from anywhere like requirements, technology, budget and so on. CM is important so that we can track these changes. Some characteristics of good configuration management systems are version control, auditable, secure, centrally located etc. The key functions of CM are identification, change control, auditing, status accounting. We also discussed the elements of change control process and the flow of a change request.

**Application in Real Projects:**

**Risk Management** helps real-world projects navigate uncertainties by identifying, analyzing, and mitigating risks to minimize negative impacts. For example, in an e-commerce platform project, potential risks such as cybersecurity threats, payment gateway failures, and high traffic during sales events are identified and prioritized. To mitigate the risk of a server crash during peak sales, the team implements scalable cloud infrastructure. Contingency plans are created for payment system outages by integrating with multiple payment gateways.

**Configuration Management** ensures that changes to the project’s components are well-documented, controlled, and traceable, helping to maintain stability and consistency. In an automotive software project, configuration management ensures that different versions of embedded control software are tracked throughout the lifecycle. As the system evolves, changes to engine control modules, braking systems, and sensor algorithms are tracked using version control, and only authorized changes are deployed after extensive testing. Configuration management also ensures that compliance with automotive safety standards (e.g., ISO 26262) is maintained.

**Peer Interactions:**

In peer interactions, a few of us classmates had a discussion about how risk management is actually carried out in real life and how it differs in software development to any other tangible projects. W also discussed how important configuration management is in software development, especially when working with a team. As discussed by the professor, how a simple change in a document when not tracked properly can affect the whole development process or decision making.

**Challenges Faced:**

The main challenge that I faced in this section is identifying and analyzing risks in any project. As a software developer, I have been more involved in the development of the project and risk management is more of theoretical concept for me, and I am having some difficulties in understanding how those concepts apply to real life projects.

**Personal development activities:**

I tried to analyze how the risk identification, analysis and prioritizing activities fit into a software application that I am currently working on.

**Goal for the Next Week:**

* Better understand how risk management fits into software development project lifecycle.
* How we can apply risk mitigation techniques in projects.
* Look up real life projects in aspect with risk management.