Learning Journal - 2

Student Name: Priyansh Jagdishbhai Bhuva

Course: Software Project Management (SOEN 6841)

Journal URL: https://github.com/prjbhuva/SPM-Learning-journal

Dates Range of activities: 18/09/2024 – 30/09/2024

Date of the journal: 02/10/2024

Key Concepts Learned:

- Risk Management:

    - Definition of Risk: Learned that risk is the probability of an event impacting project success. Risks can be classified as internal or external based on their source.
    - Risk Categories: Risks include resource risks, technology risks, budget risks, and schedule risks, each affecting the project differently.
    - Risk Assessment Frameworks: The chapter detailed both qualitative (using scales like Low, Moderate, High) and quantitative (using probability-impact models) techniques for assessing risks.
    - Risk Mitigation Strategies: Explored various response strategies such as acceptance, avoidance, transference, and mitigation, each suited to different risk scenarios. For example, risk transference involves using warranties and contracts to shift the risk to a third party.

- Configuration Management (CM):

    - Definition of CM: Configuration Management (CM) ensures the systematic control of software changes across the development lifecycle. It covers configuration identification, configuration control, status accounting, and auditing.
    - Importance of CM: CM ensures version integrity, traceability, and quality assurance, minimizing chaos in multi-team environments and reducing rework due to version mismatches.
    - Key Functions and Practices: Learned the process of setting up a configuration management system (CMS) that integrates automated tools like Git for version control, Jenkins for continuous integration, and access control for secure collaboration across teams.

Application in Real Projects:

Risk Management in Agile Projects:

- During my previous internship, we faced a scenario where scope creep affected sprint planning due to frequent client-requested changes. Using the risk mitigation strategies,

specifically risk acceptance, I proposed creating a buffer in each sprint to handle unexpected changes. This approach was well-received and helped reduce overall rework, thereby improving the sprint success rate.

Configuration Management in Multi-Team Projects:

- In a recent personal project, I applied CM principles by setting up a centralized configuration system using GitHub, along with Jenkins for continuous integration. This setup ensured that my team had proper branching and merging strategies, reducing conflicts and ensuring a stable main build.
- A key insight was understanding the importance of configuration audits to verify system integrity after major code changes. Implementing regular audits caught a major defect early, avoiding a potential breakdown.

Peer Interactions:

During a group discussion on risk assessment techniques, I shared my insights on the practicality of using quantitative methods for high-impact risks. The team discussed potential challenges in gathering accurate data for these models. This exchange led us to explore Monte Carlo simulations, which we plan to apply in our future risk assessments. Additionally, in a collaborative exercise on configuration management, I suggested using Jenkins pipelines for automating integration tests.

Challenges Faced:

- Risk Identification and Prioritization:
  - Defining clear risk categories was challenging due to overlapping risk factors (e.g., technology risks affecting schedule). I addressed this by creating a detailed risk breakdown structure to segregate risks effectively.
- Implementing a Scalable CMS:
  - Setting up a centralized configuration system was harder than anticipated due to the complexity of managing permissions and access control for multiple teams. To overcome this, I researched role-based access models and used them to streamline access management.

Personal Development Activities:

- Attended a webinar on advanced risk management frameworks, where I learned about tools like RiskWatch for automated risk tracking and analysis. This activity helped me gain insights into leveraging technology for dynamic risk management.
- Reviewed case studies on real-world configuration management failures (e.g., NASA's Mars Climate Orbiter) to understand the consequences of poor CM practices.

Goals for the Next Week:

- Create a Comprehensive Risk Management Plan: Develop a full-fledged risk management plan for an ongoing project, including a dynamic risk matrix and regular risk review sessions.
- Explore Advanced Configuration Strategies: Implement a branching strategy for a larger project, using advanced Git techniques like feature branches and hotfixes.
- Long-Term Career Development: Begin researching certifications in project management (e.g., PMP) to align my learning with long-term career goals.