**Learning Journal 2**

**Student Name:** Akhilesh Ranjit Kanbarkar
**Course:** SOEN 6841
**Journal URL:** https://github.com/Akhilesh-Kanbarkar/Concordia/tree/main/SOEN%206841
**Dates Rage of activities:** 29 Jan 2025 – 08 February 2025
**Date of the journal:** 09 February 2025

**Summary of Learning Activities**:

During this period, I studied Chapters 5 and 6, focusing on Configuration Management and Project Planning. Through reading the textbook, lecture slides, and discussions, I developed an understanding of how these processes ensure software development efficiency and project success.
Configuration Management (Chapter 5): Explored key concepts including version control, configuration identification, change management, and status reporting. Gained insights into the significance of having a structured configuration management system to maintain consistency across project components.
Project Planning (Chapter 6): Learned techniques such as top-down and bottom-up planning, the Work Breakdown Structure (WBS), resource allocation, scheduling issues, and risk mitigation strategies. Learned how effective planning supports smooth project execution and delivery.

Additionally, I reviewed case studies demonstrating best practices in configuration control and iterative project planning.

**Key Concepts Learned:**

- Configuration Management: Helps track software changes, ensuring version integrity and preventing conflicts in project updates. Benefits include:

1. Reduces confusion and establishes order.
2. Ensures consistency and traceability across software versions.
3. Limits legal liability by maintaining a record of changes.
4. Enhances compliance with industry standards.
5. Reduces lifecycle costs by preventing unnecessary rework.

- Project Planning: Involves defining tasks, resource allocation, and scheduling. Methods like Work Breakdown Structure (WBS) help decompose tasks into manageable units.
- Top-Down vs. Bottom-Up Planning: Understanding how projects are structured differently based on initial planning approaches.
- Scheduling Challenges: Explored common project scheduling pitfalls and the importance of contingency planning.
- Read case study for Chapter 5 which refers to a U.S.-based SaaS vendor employs a centralized configuration management system to support incremental iteration development across global teams. The system ensures 24/7 availability, role-based access (admin/view-only/super-user), and automated smoke testing to maintain build integrity. Developers synchronize local builds with the central repository and test code locally before submission, minimizing failures. Escalation protocols resolve unresolved build issues within an hour, ensuring continuous productivity. This setup enables seamless collaboration among internal, external, and offshore teams while maintaining security and efficiency.

**Reflections on Learning:**

**Application in Real Projects:**

- In Agile development, configuration management tools like Git ensure seamless integration of incremental changes, preventing inconsistencies.

- A real-world example of poor project planning showed how unrealistic schedules led to missed deadlines and budget overruns. For example, a software firm underestimated the complexity of integrating a new payment gateway, leading to multiple delays and increased costs.
- The concept of iterative planning and feedback loops reinforced why adaptability is key to managing software projects.

**Peer Interactions:**

Engaged in discussions to compare project planning approaches, evaluating their effectiveness in different scenarios. As an interactive activity, we conducted a mock project planning exercise where each collogue was assigned a different planning methodology (Waterfall, Agile, and Hybrid). This helped us understand the strengths and weaknesses of each approach, and seeing Agile's flexibility in action was particularly insightful.

**Personal Development Activities**

- Created a visual timeline of project planning phases to better understand task interdependencies.
- Explored real-world configuration management tools, including GitHub and Jira, to understand their practical applications. Also came across a new project management tool ClickUp which is very versatile and is user-friendly and caters a variety of industry requirements.
- Reviewed industry case studies to identify best practices in project planning and risk management.
- I tried creating a Gantt chart and a PERT chart to understand the difference between the two, which helped me visualize project scheduling and organization techniques more effectively.

**Challenges Faced:**

- Initially found it difficult to understanding Configuration Baselines, the role of baselines in managing software changes.
- Realized that effort estimation varies greatly depending on team experience and historical data availability.
- Studied different approaches to mitigate project scheduling issues, including buffer time allocations.

To overcome these challenges, I referred to additional online tutorials and industry reports to bridge my knowledge gaps.

**Time Management:**

Allocated 5 hours to reading course material along with online tutorials, and 2 hours with peers to discuss topics and learing activities.

**Goals For Next Week:**

- Explore advanced risk management techniques used in software project planning.
- Analyze real-world failures due to poor configuration management.
- Try to learn and familiarise with project management tools like Jira, and to use it for personal projects.