Student Name: Devang Shah

Course: Software Project Management

Journal URL: https://github.com/devang-1910/SOEN6841/blob/main/Learning%20Journal%202.docx

Dates Rage of activities: 28th Jan, 2025 – 8th Feb, 2025

Date of the journal: 9th Feb,2025

Key Takeaways from the Course

Configuration Management (CM) ensures structured control over **software changes, preventing errors, lost documentation,** and **recurring defects**. Its core functions—**identification, control, status accounting**, and **auditing**—help maintain consistency and traceability.

Project planning organizes schedules, budgets, resources, and risks to ensure smooth execution. The **Work Breakdown Structure (WBS)** breaks projects into manageable tasks, while **top-down and bottom-up planning** help define realistic timelines. **Critical Path Method (CPM) and Gantt Charts** track key dependencies, ensuring task prioritization and risk mitigation. A structured **change control process** within CM supports project planning by maintaining version integrity, reducing rework, and ensuring efficient team collaboration.

By integrating CM with project planning, organizations can **avoid scope creep, manage project risks proactively, and maintain software quality**, leading to successful project execution. For instance, in agile development, CI/CD pipelines leverage CM principles to ensure controlled, iterative software releases.

Applying Concepts to Real-World Scenarios

Managing Versions in an E-Commerce Platform
Without **Configuration Management (CM)**, deploying the wrong code version can cause missing features or recurring bugs. Using **Git & CI/CD pipelines**, changes are tracked, approved, and audited, ensuring smooth deployments and rollback if needed.

Project Planning in a Mobile Banking App
Breaking tasks into **feature modules (WBS)** ensures efficient resource allocation. Using **Top-Down Planning**, the project timeline is set first, then tasks are scheduled. **Critical Path analysis** helps avoid delays in key areas like security audits and API integrations, with buffer time for unexpected risks.

Beyond these applications, **CM** can also be leveraged in **machine learning models**, where tracking dataset versions and model updates ensures reproducibility. Likewise, Agile methodologies integrate CM principles to maintain stable development environments in **rapid iteration cycles**. By combining CM with structured planning, projects achieve greater reliability, adaptability, and efficiency.

Learning Through Collaboration & Peer Discussions

Peer discussions provided valuable insights into how Configuration Management (CM) and Project Planning differ across industries. While finance and healthcare prioritize strict audit trails, e-commerce emphasizes rapid deployments, reinforcing the need for industry-specific CM strategies. One **discussion on Work Breakdown Structure (WBS)** and **scheduling** helped clarify task dependencies, where a peer's experience with software migration **illustrated the impact of misaligned critical path analysis**, prompting me to rethink how I approach

Obstacles Encountered During Learning

Understanding **Configuration Management (CM) workflows** was challenging, especially in **handling change control processes**. Differentiating between version control, configuration audits, and status accounting required extra effort to grasp how they interact in real projects. For example, during software development**, failing to track changes properly can lead to teams working on outdated versions**, causing rework and delays. In Project Planning, **estimating task durations and dependencies in Work Breakdown**

project scheduling. A breakthrough moment came when peer feedback highlighted **the risks of undocumented changes in CM**. Initially, I underestimated the **complexity of approval tracking**, but after discussing rollback strategies with my group, I realized the importance of **integrating automated version control and audit trails** to prevent disruptions. Applying this insight, I explored industry best practices for structured change control, improving my grasp of CM implementation. These interactions not only refined my understanding of structured planning but **also improved my ability to apply CM principles more effectively in real-world projects.**	**Structure (WBS) was difficult.** Theoretical models assume linear progress, but real projects face delays due to unforeseen complexities. An example is when integrating third-party APIs, where dependencies on external vendors can introduce unpredictable delays. Another challenge was **scheduling conflicts in Critical Path analysis.** Identifying which tasks directly impact project timelines and which have flexibility required deeper understanding. Peer discussions helped clarify these concepts, but practical application in real scenarios still needs improvement. To overcome these challenges**, I plan to review real-world case studies on CM implementation**, **practice task estimation using project management tools**, and explore **interactive scheduling exercises to enhance practical understanding**.
<u>**Additional Learning Activities & Self-Improvement**</u> To strengthen my understanding of **Configuration Management (CM) and Project Planning**, I explored real-world **case studies** on version control and change management, particularly in agile environments. Reading about DevOps practices helped me see how **CI/CD pipelines integrate CM principles for smoother deployments, reinforcing my interest in scalable software engineering processes**. I also watched **YouTube tutorials** on **Microsoft Project and Jira**, gaining hands-on experience with task dependencies, WBS, and critical path analysis. Exploring these tools helped me **visualize how scheduling adjustments impact project timelines**, a skill crucial for future project management roles. Additionally, I **researched best practices for change control policies**, focusing on **approval workflows and rollback strategies** to prevent mismanagement of software versions. This has deepened my understanding of structured change tracking, an essential aspect of maintaining software stability in enterprise environments. Moving forward, I plan to apply these learnings by **experimenting with scheduling exercises and analysing more case studies on CM failures and successes**. This will not only improve my practical application but also enhance my readiness for industry roles where CM and structured planning are key to software project success.	<u>**Goals For Next Week**</u> Next week, I aim to deepen my understanding of **advanced Configuration Management (CM) practices** and **practical project planning techniques**. My primary goal is to explore **real-world CM implementations**, particularly in DevOps environments, by studying how organizations **manage version control, audits, and change tracking** in large-scale software projects. For project planning, I plan to **practice creating Work Breakdown Structures (WBS)** and **Critical Path analyses** using tools like **Microsoft Project or Jira**. This hands-on approach will help me improve my ability to **estimate task durations, identify dependencies, and allocate resources effectively**. To align with long-term career goals, I will research **industry best practices for risk mitigation in software projects**, focusing on how **structured planning and CM policies** reduce project failures. Additionally, I plan to **experiment with scheduling case studies** to apply theoretical concepts in practical scenarios. By the end of the week, I aim to have a **stronger grasp of CM workflows, improved project scheduling skills, and deeper insights into risk management strategies**, ensuring I can apply these concepts effectively in future roles.

By documenting my progress, I have been able to track how theoretical concepts translate into practical applications, ensuring continuous improvement in the course of Software Project Management.