SOEN 6841 - Learning Journal 3

Student Name: Tharun Balaji

Course: Software Project Management – SOEN 6841

Journal URL: https://github.com/thxrun180/SOEN6841

Date Ranges of Activities: Feb 07, 2025 – Feb 19, 2025

Date of Journal: Feb 23, 2025

Key Concepts Learned

This week, my focus was on Chapter 5: Configuration Management and Chapter 6: Project Planning, which are essential for maintaining control over software development and ensuring efficient project execution. Configuration Management (CM) is a structured approach to managing changes in software, ensuring proper version control, and preventing inconsistencies.

Without CM, projects face issues such as lost updates, conflicting changes, and untracked modifications. The key functions of CM include configuration identification, which establishes system baselines, configuration control, which enforces structured change approval, configuration status accounting, which maintains records of modifications, and configuration auditing, which verifies compliance with project standards.

Project Planning, covered in Chapter 6, focuses on structuring a project efficiently from initiation to execution. It includes Work Breakdown Structure (WBS) for dividing work into manageable components, project scheduling to ensure timely execution, and resource allocation to optimize manpower and materials.

Scheduling methods such as Critical Path Method (CPM) help identify the longest sequence of dependent tasks, preventing bottlenecks that could delay the entire project. Risk management is another crucial aspect of planning, as unexpected challenges can derail progress. Effective risk planning includes contingency buffers and proper tracking mechanisms to mitigate potential delays.

Application in Real Projects

Our team is working on a system that tracks expiration dates of perishable food items and sends notifications to users to prevent food waste. Throughout the project, we encountered several challenges related to version control, task dependencies, and scheduling, making CM and Project Planning crucial to managing our workflow.

At the start, our team struggled with managing software versions, leading to overwritten changes and inconsistent updates across different modules. This was especially problematic when integrating the OCR feature for scanning expiry dates, as some team members worked on outdated versions of the database schema. After learning about Configuration Management, we implemented a structured Git branching strategy, where each team member worked on separate feature branches, changes were reviewed before merging, and proper commit messages were enforced. This improved code traceability and reduced integration issues.

From a Project Planning perspective, our initial lack of a structured schedule led to inefficiencies. Certain tasks, such as setting up the notification system and cloud storage for user data, took longer than expected, affecting dependent tasks. Using Work Breakdown Structure (WBS), we divided the project into clear, well-defined sections like database setup, UI development, and backend API integration. Applying CPM, we identified critical tasks that directly impacted the project deadline, ensuring that these received priority and sufficient resources.

Risk management also became a key focus after we encountered delays in testing our push notification feature, which needed adjustments based on different mobile platforms. Initially, we underestimated the testing effort required, but after reviewing risk planning strategies, we realized the importance of adding buffer time for tasks involving third-party integrations. Moving forward, we have allocated additional time in our schedule for unexpected debugging and testing phases.

Challenges Faced

One major challenge in Configuration Management was ensuring that database schema changes were properly versioned. Some team members modified the database without tracking changes, leading to compatibility issues when merging different modules. Implementing database migration tracking and maintaining a version log helped resolve these issues.

Estimating task durations accurately in Project Planning was also challenging. Some features, like real-time expiration tracking and user preference customization, took longer than expected, which affected the integration schedule. After applying CPM, we restructured our schedule by identifying which tasks were critical and which could be adjusted without affecting deadlines.

Another challenge was integrating CM practices into an Agile development approach. Since our team follows an iterative process, we had to balance flexibility with structured version control to prevent unnecessary delays while maintaining control over changes.

Peer Interactions

Throughout the week, discussions with team members and peers helped reinforce my understanding of best practices for version control, project scheduling, and risk assessment. Our team analyzed different Git workflows, debating between trunk-based development vs. feature branching and settling on a branching model that supports parallel development without conflicts.

I also participated in a CPM group activity, where we practiced breaking down projects into dependencies and identifying the critical path. This provided insights into how project schedules can be optimized by identifying bottlenecks early.

Additionally, reviewing case studies on failed software projects due to poor CM practices helped us understand the real-world consequences of inadequate version tracking, such as missing critical updates or deploying incorrect builds.

Personal Development Activities

To improve my project management skills, I explored real-world case studies of software failures caused by poor planning and version control, such as the Knight Capital trading disaster. I also experimented with project tracking tools like JIRA and Microsoft Project to simulate different scheduling strategies and visualize dependencies. Applying CPM to our team project's task structure allowed me to analyze our timeline more effectively.

Goals for the Next Week

- Study Chapter 7: Project Execution and Monitoring to learn how to track and adjust project progress.
- Apply CPM to refine our project's schedule, ensuring we allocate resources effectively.
- Research how Agile teams handle Configuration Management while maintaining rapid development cycles.
- Work on the Project and prepare for the next week's project pitches.