Learning Journal Template

Student Name: Sujith Kumaravel

Course: Software Project Management

Journal URL: https://github.com/Sujith-Kumar-2003/SOFTWARE_PROJECT_MANAGEMENT/blob/main/Learning_Journals/Learning%20Jorunal%203%2040281567.pdf

Week 3: Feb 07,2025 – Feb 2025

Date: Feb 23, 2025

Key Concepts Learned:

During this week, I focused on studying Configuration Management and Project Planning while also reviewing chapters 1 through 6 for my midterm. I learned that configuration management is crucial for software projects because client requests often prompt changes. A robust system lets you revert to earlier versions when needed and tracks important details—who made a change, why it was made, what was changed, and when. Without it, projects risk missing features, corrupted versions, or developers working on outdated code, all of which can delay progress and raise costs. This process includes identifying versions, controlling and approving changes, keeping detailed records of version history, and auditing to ensure the system meets requirements.

On the project planning side, establishing a clear plan is essential for guiding the project lifecycle. It covers schedule planning, effort estimation, resource allocation, quality management, and even planning for configuration management. There are two main scheduling approaches: top-down, where you first allocate time for the overall project and then break it down into tasks, and bottom-up, where you schedule smaller tasks and then combine them into a complete timeline. I've used the bottom-up method personally. Tools like a Work Breakdown Structure (WBS) help divide the project into manageable parts and pinpoint task dependencies, which improves resource allocation. It's also important to avoid overloading team members and to include buffer times, as adding new members late in the project can slow progress due to increased communication needs. Visual aids like Gantt chart further assist in managing and tracking tasks effectively.

Application in Real Projects

This week, I learned that even the best plans can fail without good resource management and clear change tracking. Our team is working on an emergency response app, and we ran into trouble when our resource estimates were off. Our task list was too vague, which led to scheduling issues and delays. To fix this, we held a meeting to review our plan. We found that our Work Breakdown Structure (WBS) didn't show enough detail about the work needed for each step, like gathering requirements, designing the UI, integrating the backend, and testing. We updated the WBS with clear tasks and dependencies, which helped us reassign work better and add extra time to handle delays.

We also improved our configuration management by tightening version control and automating parts of our deployment process. This ensured everyone worked on the most current version, reducing confusion and mistakes. We started regular team meetings to keep everyone informed and quickly solve any issues. This simpler approach has given us a clearer plan and a stronger project structure, turning a difficult situation into a valuable learning experience.

Peer Interactions

Throughout the week, our team stayed connected by meeting regularly on Google Meet, Zoom, and in person at the library. We worked together on project deliverables, brainstormed ideas, and planned our tasks, which helped us prepare for our midterm and refine our project pitches.

During one of these meetings, a female peer showed interest in joining our project team. After discussing the pros and cons, we decided to invite her to join us and are now waiting for her response.

Challenges Faced

This week, a major challenge was learning to track change requests and link them to approved versions. We set up a GitHub repository for our project, which helped us manage these changes more systematically. Initially, connecting requirements to modifications was confusing, and summarizing the benefits of Configuration Management (CM) concisely proved difficult. However, through iterative refinement and valuable feedback from teammates, I gained clarity. Concepts like identification, change control, and status accounting took time to grasp, but with team support, I learned to apply them effectively. This hands-on experience with our GitHub repo significantly improved our project management process.

Personal Development Activities

I engaged in self-study of project planning techniques by diving into online tutorials on the Work Breakdown Structure (WBS) to reinforce the concepts from Chapter 6. To deepen my understanding, I practiced creating Gantt charts using project management software, which helped me visualize task dependencies, sequencing, and slack time in a hands-on manner. Additionally, I completed a Udemy course on Git and configuration management, which further enhanced my practical skills and knowledge in managing changes and collaborating effectively on projects. This combined experience has significantly improved my ability to manage tasks and time efficiently.

Goals for Next Week

Next week, I plan to kick off the development of our app's prototype by diving deeper into configuration auditing and status accounting through case studies of successful implementations. I'll explore how these practices are applied in Agile environments, aiming to integrate them into our project. I also intend to refine my skills with task scheduling tools to better manage our timeline. During reading week, I'll meet with the team to finalize our deliverables, and we'll prepare for our project pitch on March 6, where we'll formally present who our app is being developed for and our overall development strategy.