**Learning Journal**

**Student Name:** Md Tazin Morshed Shad ( 40260834 )  
**Journal URL** : <https://github.com/TazinMorshed/SOEN-SPM>

**Course: SOEN 6841**  
**Date Range:** : 29-January-2025 to 9-February-2025  
**Date of Entry:** 09-February-2025

| **Key Concepts Learned** | This week, I explored the Configuration Management System and Software Project Plan. Key insights included Purpose of Configuration Management consisting of configuration identification, configuration control, configuration status accounting, configuration audits and recognizing that software is highly adaptable but prone to uncontrolled changes, leading to project chaos, delays, and quality issues. Also this week I learned about a software project plan, its parts and types. The inputs go in making a software project plan and the technique used. Learned that Project planning consists of project scheduling, project budgeting, manpower planning, communication planning, quality planning. I learned that the Critical Path Method (CPM) focuses on task dependencies and project duration, while Gantt charts visually track timelines, and that identifying the critical path helps prevent project delays. |
| --- | --- |
| **Application in Real Projects** | I used Configuration Management (CM) to track software versions, manage change requests, and prevent uncontrolled modifications. Version control tools ensured that developers worked on the latest code, while change control processes helped avoid project chaos. Also, I used Gantt Charts to track project timelines and keep the team aligned, while CPM helped identify critical tasks to prevent delays. Prioritizing bug fixes during testing ensured timely deployment, improving overall project efficiency.   All these are being done in my other project : <https://github.com/DvSoni/robocodeW25> |
|  | Challenging Component: Handling frequent and unplanned changes in software projects. In real-world projects, clients often request modifications mid-development, leading to scope creep and version conflicts. An approach could involve implementing a more robust change control process using tools like JIRA. Additionally, integrating automated conflict resolution tools such as Git, coupled with Continuous Integration/Continuous Deployment (CI/CD) pipelines, would help detect and resolve version conflicts early in the development process |
| **Peer Interactions & Collaboration** | This week, I had two productive meetings with peers where we discussed the design and structure of our chatbot for mental health support, brainstorming ideas on how to best approach its functionality and user interface to ensure it would be user-friendly and effective. These discussions provided valuable insights into structuring real-world projects and emphasized the importance of user-centered design. In addition to the chatbot project, I worked on other initiatives where I integrated CI/CD pipelines in Git, assigned issues to contributors on GitHub, and created labels to streamline the process, ensuring clear communication and efficient task tracking. These activities enhanced my understanding of version control and collaboration within development teams, improving both my technical and teamwork skills. |
|  | Challenging Component: Peer feedback during our meetings was particularly helpful in refining the project scope and identifying potential challenges early on. For example, a suggestion from a peer to prioritize mental health data security in our chatbot design led to a deeper exploration of encryption and data protection methods. |

### **Challenges Faced**

Setting up a github to store my learning journal. Previously I had placed a google docs link to the journal URL, however I have fixed this issue by replacing it with a github URL

Another problem was, integrating CI/CD pipelines into Git, particularly when assigning issues to contributors and organizing tasks on GitHub. While I was able to create labels and track progress, I encountered some difficulty in ensuring the smooth flow of communication among the contributors, which led to occasional misunderstandings about task priorities and deadlines.

### **Personal Development Activities**

This week, I focused on strengthening my technical and managerial skills by familiarizing myself with advanced Git features, including branching and conflict resolution, to improve version control in real-world projects. Additionally, I worked on enhancing my project management capabilities by exploring tools like Gantt charts and the Critical Path Method (CPM) to better track project timelines and dependencies. I also focused on improving my leadership and team coordination abilities, collaborating closely with teammates on our chatbot project, assigning tasks, and ensuring efficient communication.

### **Goals for the Next Week**

For the upcoming week, my goals are to read chapters 7 and 8 to deepen my understanding of the course material, particularly focusing on the topics that will aid in the completion of the upcoming project pitch. I also plan to collaborate closely with my teammates, discussing ideas, refining our approach, and preparing for the pitch to ensure we present a well-organized and compelling proposal.