Student Name: Sushmitha Tiwari Ganga

Course: Software Project Management

Journal URL: https://github.com/Sush0420/SPM

Dates Rage of activities: 16/01/2025 to 23/01/2025

Date of the journal: 28/01/2025

Key Concepts Learned:

The week's learning felt like a roadmap to becoming a better project manager. We started with understanding what makes a project and a software project unique. I found it fascinating how every project—whether building software or constructing a building—follows similar phases: initiation, planning, execution, and closure. What stood out was the emphasis on roles like Scrum Master and Project Manager; each has a clear focus, yet they all work towards the same goal.

In project initiation, the idea of creating a project charter felt like writing the project's "birth certificate," defining its purpose and boundaries. The SMART objectives framework was a useful takeaway—such a simple yet powerful way to set clear, actionable goals. Later, we dove into effort and cost estimation techniques. I realized how challenging it can be to predict effort for software projects, given how intangible the results can feel. Finally, we explored risk management, and it struck me how risks, if not managed, can turn the best plans into disasters. Identifying risks early and dealing with them through strategies like mitigation or avoidance felt like a life lesson, not just a project management principle.

Application in Real Projects:

Everything we learned this week felt directly applicable to real-world projects. For example, SMART objectives could streamline goal-setting in any project, not just software ones. The estimation techniques, especially the analogy-based one, seemed like a practical way to plan for tasks where historical data is available. It made me think about how I could apply similar techniques to better plan my own work or even group assignments.

The discussion on risk management reminded me of an app development project I worked on where we underestimated the timeline and hit roadblocks. If only we had spent more time on risk analysis and built contingency plans, we could have avoided the chaos. Now I see the importance of allocating buffers and monitoring risks continuously.

Peer Interactions:

Conversations with my peers added new dimensions to what I learned. We had a lively debate about whether iterative models are inherently better than waterfall models. While some argued for the flexibility of iterative models, others pointed out how they might not be suitable for projects with rigid timelines. Another interesting moment was during a group exercise on effort estimation—someone shared their experience of using COCOMO in an internship, which made the theoretical concepts feel much more real.

Challenges Faced:

Not everything was smooth sailing this week. Effort estimation felt like a puzzle, especially when we talked about COCOMO and function point analysis. The math wasn't hard, but understanding when to use which technique felt tricky. Another challenge was grasping the subtle differences between roles like Leader, Manager, and Scrum Master. Their responsibilities often overlap, and it took some time to clarify who does what.

Lastly, risk prioritization was harder than I expected. It's one thing to identify risks, but deciding which ones to tackle first—especially when everything feels important—required some deep thinking.

Personal Development Activities:

To make the concepts stick, I went beyond the lecture slides. I watched a few videos on project risk management, which gave me practical insights into handling risks. I also experimented with creating a mock project charter for a personal project—turning an idea into something structured made me appreciate its value. Additionally, I started reading a book on Agile project management to better understand Scrum, which I'm keen to use in the future.

Goals for the Next Week:

For next week, I want to focus on two things: mastering risk management strategies and sharpening my skills in effort estimation. I also plan to research more on the pros and cons of iterative vs. waterfall models, as it feels like a debate worth diving deeper into. Lastly, I'll practice drafting more project charters and using tools like Jira to simulate project planning in a hands-on way.