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%20Journal%203%2040281567.pdf

Week 3: March 07 2025 – March 16 2025

Date: March 16 2025

Key Concepts Learned:

This week, drawing on Chapters 7 (Project Monitoring & Control) and 8 (Project Closure), I focused on techniques that ensure a project is executed according to plan and wrapped up systematically. I learned how Earned Value Management (EVM) helps measure both schedule and budget progress by comparing baseline and actual values. Using EVM's schedule variance and cost variance can highlight deviations early. I also discovered how S-curves can track cumulative progress and expenses over time, giving a visual snapshot of whether the project is on track. Another important concept was resource loading and utilization, which helps avoid under- or over-allocation of team members. Finally, I studied the formal closure phase, including source code version management, archiving project metrics, and compiling lessons learned, all of which ensure future teams can reference and improve upon our outcomes.

Application in Real Projects

My current project is a Disaster Management Resource Allocation webapp. We initially struggled with vague task definitions and resource estimates, which caused schedule slips. By applying the monitoring techniques from Chapter 7, we restructured our Work Breakdown Structure (WBS) to capture tasks more accurately. We then used a simplified version of EVM calculations to gauge our schedule variance and discovered we needed extra time to finalize the backend integrations. This monitoring process allowed us to adopt schedule optimization measures (like moving noncritical tasks around) and to practice resource leveling. We also planned our closure phase more thoroughly, ensuring that once the webapp is complete, our source code is properly version-controlled and archived, and that we document any lessons learned about resource allocation for future disaster management solutions. 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

We continued to meet via Google Meet, Zoom, and in-person library sessions. These discussions provided clarity on applying EVM formulas and interpreting S-curves for schedule tracking. For example, during one of our library meetups, we realized our initial EVM calculations didn't account for a few unplanned tasks that arose mid-project, causing our schedule variance to look misleadingly small. Once we factored these tasks in, the data showed we were further behind

schedule than expected, which prompted us to reassign resources and revise our milestones. We also shared experiences on risk management and how controlling scope changes can reduce unnecessary complexity—one teammate mentioned how adding new features without updating the WBS led to confusion about who was responsible for integrating them. Our group also emphasized the importance of formal closure; several peers mentioned how skipping a structured closure phase in previous projects led to confusion about final deliverables, duplicated effort when reusing code, and hindered future maintenance efforts.

Challenges Faced

One challenge we faced was truly grasping the nuances of EVM metrics—especially in accurately calculating Planned Value (PV), Earned Value (EV), and Actual Cost (AC). We initially attempted to assign dollar amounts to each task without factoring in tasks that were only partially complete or running in parallel. This oversight led us to underreport our Earned Value. Additionally, when our scope changed mid-sprint to include an extra module for disaster data analytics, we struggled to update the baseline figures properly. It took several trial runs and cross-checking with peers before we felt confident that our calculations were capturing real progress.

Another hurdle was deciding how detailed our closure activities needed to be. During one of our group discussions, we realized we'd never standardized how to archive test results, documentation, and deployment scripts. We'd stored everything in a shared drive, but there was no consistent naming or folder structure. This disorganized approach made it difficult to retrieve earlier code versions or confirm final deliverables. Eventually, we designated a team member to organize the files and create a quick reference guide for others, which immediately cut down on confusion. Reflecting on this, we now see how critical a structured closure phase is for ongoing improvements—without it, future teams would likely waste time digging for information we could have easily recorded and labeled.

Personal Development Activities

I watched more tutorials on EVM and S-curves to see real-world examples of how they are used in projects with iterative delivery. I also practiced with a small test project to gain confidence in calculating and interpreting schedule and cost variances. For closure, I explored case studies on how organizations structure their final handover of deliverables, archive their metrics, and compile lessons learned. These examples helped me see the tangible benefits of a thorough wrap-up process.

Goals for Next Week

Next week, I aim to refine my resource utilization approach for the disaster management webapp, ensuring we maintain balanced workloads and realistic timelines. I plan to work through additional EVM and schedule optimization exercises to become more comfortable with these calculations. I also want to finalize a formal closure checklist, detailing exactly how our team will handle documentation, code archival, and a retrospective meeting. By doing so, I hope to solidify our project's final phase and ensure a smooth handover to any future team that might build upon our work.