

# Learning Journal 3

**Student Name:** Santhosh Dayakar

**Course:** SOEN 6481 Software Project Management

**Journal URL:** <https://github.com/S-15-77/Software-Project-Management-Journal>

**Dates Range of activities:** 27-10-2025 to 03-11-2025

**Date of the journal:** 09-11-2025

## **Key Concept Learned:**

Over the past few weeks, I explored two core areas of software project management: Project Monitoring and Control and Software Lifecycle Management. I learned that a project plan serves as the baseline for tracking cost, schedule, and scope performance. Monitoring focuses on gathering progress data, while control involves taking corrective actions when deviations occur. The four-step control framework establishing baselines, measuring performance, comparing results, and applying corrections ensures continuous alignment with project goals. The Earned Value Management technique integrates cost and schedule metrics using Planned Value, Earned Value, and Actual Cost to determine Schedule Variance and Cost Variance. Indicators such as defect density, resource utilization, and budget variance act as early signals for potential risks, while closure activities like verifying deliverables and documenting lessons learned promote accountability and improvement.

Software Lifecycle Management extended this understanding through the Software Development Life Cycle, which defines structured stages from requirements and design to testing and maintenance for predictable software delivery. The introduction of DevOps illustrated how automation and continuous feedback bridge development and operations, ensuring faster and more reliable releases. Finally, the Capability Maturity Model Integration framework presented five maturity levels from Initial to Optimizing that guide organizations in improving process efficiency and quality. Collectively, these concepts showed how structured monitoring, adaptive control, and mature lifecycle management align people, processes, and technology to achieve consistent, high-quality outcomes.

## **Application to Real Project:**

During my internship, I applied Project Monitoring and Control concepts by tracking task progress and resource allocation across multiple sprint cycles. Using tools similar to Earned Value Management, I compared planned timelines with actual completion data to identify delays early. This approach improved schedule visibility and helped our team reassign resources when

workloads became uneven. Implementing these monitoring techniques also revealed the importance of performance indicators such as task completion rate and defect count for evaluating productivity. The main challenge I faced was ensuring data accuracy, as inconsistent updates sometimes distorted variance calculations. However, once the tracking process was standardized, team coordination and delivery predictability improved significantly.

In my coursework projects, I applied Software Lifecycle Management principles to maintain structure and quality. While working on a team-based development assignment, we followed the Agile model, organizing tasks into short iterations with frequent peer reviews. Integrating DevOps practices such as automated testing and version control through GitHub improved deployment reliability and reduced integration issues. I also reflected on CMMI principles, using them informally to evaluate how our project evolved from ad-hoc collaboration to a defined, repeatable process. These experiences showed that applying lifecycle and control methods fosters accountability, consistency, and adaptability. Although balancing documentation with agility was sometimes challenging, the result was clearer communication, faster feedback, and higher-quality deliverables, reinforcing the value of structured yet flexible project management approaches.

### **Peer Interaction and Collaboration:**

Working with my teammates on our course project strengthened my understanding of Project Monitoring and Control and Software Lifecycle Management. During group discussions, we used EVM-based tracking to monitor task progress and identify schedule variances. For example, when one module of our project fell behind, our team analyzed the variance data and collaboratively redistributed workloads to stay on track. This experience showed how consistent reporting and transparency improve accountability and project control. Reviewing each other's deliverables also highlighted the importance of metrics like defect rate and milestone completion for maintaining quality.

Collaboration further deepened my grasp of Agile and DevOps practices. Through daily check-ins and Git-based version control, I observed how automation and continuous feedback streamlined workflow and reduced rework. Peer feedback during sprint reviews helped refine backlog priorities and iteration planning, reinforcing that communication, shared ownership, and structured reviews are essential for keeping projects aligned and adaptive.

### **Challenges Faced:**

One key challenge I encountered was understanding the practical application of Earned Value Management (EVM) in monitoring both schedule and cost performance. While the theoretical formulas for Planned Value (PV), Earned Value (EV), and Actual Cost (AC) were clear, translating them into meaningful project insights initially felt complex. To overcome this, I

revisited my internship data, created a small tracking sheet, and simulated EVM calculations to visualize cost and schedule variances. This exercise helped me see how minor data inconsistencies can distort performance indicators and impact decision-making. Another challenge was distinguishing how Agile and DevOps complement each other within the Software Development Life Cycle (SDLC). I resolved this by analyzing workflow case studies and applying automation tools during group projects. These steps improved my analytical understanding, enhanced my attention to data accuracy, and strengthened my ability to apply project control techniques in practical settings.

### **Personal development activities:**

To advance my professional growth, I focused on strengthening both my technical and project management competencies. I completed an online micro-course on Agile and DevOps practices, which deepened my understanding of continuous integration, automation, and iterative delivery. I also practiced Earned Value Management (EVM) calculations using real project data to enhance my analytical and reporting accuracy. In addition, I joined peer sessions to explore tools like Jira and Trello, gaining practical insights into collaborative project tracking. While preparing for interviews, I revisited key concepts such as SDLC models, Agile frameworks, and performance metrics, which not only improved my confidence but also reinforced my grasp of their real-world applications. These efforts collectively strengthened my ability to manage projects with clarity, discipline, and adaptability.

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

Looking ahead, I aim to:

- Apply project monitoring concepts from this week's lessons directly to my ongoing coursework project by tracking progress using Earned Value Management (EVM) metrics to measure schedule and cost performance.
- Integrate Agile and DevOps principles into our team workflow by setting up automated testing and version tracking to improve collaboration and delivery consistency.
- Evaluate process maturity within our project work, using insights from the Capability Maturity Model Integration (CMMI) to identify areas for improvement in planning and documentation.