DEPARTMENT OF COMPUTER SCIENCE AND SOFTWARE ENGINEERING CONCORDIA UNIVERSITY
SOEN 6841 Winter 2024
Shalvi Saxena (40220846)

# Learning Journal

**Student Name:** Shalvi Saxena

**Course:** Software Project Management [SOEN 6841]

**Journal URL:** https://github.com/Shalvi-Saxena/SPM-Journal

**Week 3:** Feb 18 - Mar 09

**Date:** 08 Mar 2024

**Key Concepts Learned:**

This week I explored chapter 7 and 8. Here are some of the key points I covered:

1. Software projects pose challenges in monitoring and control due to unclear specifications and assumptions made by the project team.

2. Clarity in specifications gradually improves as the project progresses, but managing work with vagueness remains a significant challenge.

3. Project monitoring and control become difficult due to this inherent vagueness in project work.

4. Various tools and techniques are available to aid project monitoring and control.

5. Project plans typically include schedule and budget buffers to accommodate risks and uncertainties.

6. When risks occur, adjustments are made by utilising these buffers.

7. Some techniques like resource levelling, schedule optimization, and corrective actions help in overcoming setbacks without consuming buffers.

8. Earned Value Management (EVM) technique is an effective method for taking corrective action and providing performance indicators through a project dashboard.

9. Deviations in performance indicators can prompt project managers to take timely actions to address issues.

10. Before project closure, various activities need completion, especially if project execution has been challenging.

11. Key tasks during closure include resource release, preparing lessons learned, source code management, and project data management.

12. Preparation of project data and lessons learned is essential for future reference and improvement.

13. Archiving project data and lessons learned ensures their availability for future projects.

14. Source code control becomes crucial due to numerous changes made during system testing and defect fixing.

15. Determining the appropriate version of the software for deployment at the customer site is a critical decision during this phase.

16. Care must be taken to ensure that project data does not contain any extraneous information.

17. Proper archiving procedures need to be followed to ensure the usefulness and accessibility of archived project data in the future.

**Reflections on Case Study/course work:**

**CHAPTER 7**

**1. Effective Contingency Planning:** The case study illustrates the importance of having contingency plans in place to address issues and risks that arise during project execution. The SaaS vendor's project team demonstrates preparedness by conducting weekly iteration review meetings and discussing both known and potential risks.

**2. Proactive Risk Management:** The project team's proactive approach to risk management is commendable. They analyse each issue or risk encountered, identify its root cause, and develop suitable solutions to mitigate its impact. This systematic approach helps in maintaining project momentum and minimising disruptions.

**3. Resource Considerations:** The case study highlights the challenge of managing risks within fixed resources. Unlike some projects where additional resources may be allocated to mitigate risks, the SaaS vendor's project team must consider the impact of risks on the project schedule without the luxury of extra resources. This constraint underscores the importance of efficient resource allocation and prioritisation.

**4. Schedule Adjustment and Adaptability:** The project team's ability to readjust schedules and adapt plans in response to encountered risks is crucial for maintaining project progress. They recognize the interconnectedness of tasks within an iteration and are willing to reschedule tasks if necessary to accommodate changes.

**5. Exploring Alternative Solutions:** In cases where risks threaten to significantly impact the entire iteration plan, the project team considers alternative solutions such as working overtime to cover for the extra time required. This demonstrates flexibility and a willingness to explore creative solutions to overcome challenges.

**CHAPTER 8**

**1. Resource Management:** The case study highlights the importance of effective resource management during project closure. Resources released from one project are seamlessly absorbed into subsequent projects, demonstrating efficient utilisation of human resources within the organisation. The role of the global program manager in planning resource allocation across projects is crucial for maintaining project continuity.

**2. Configuration Management:** The configuration manager's role in managing project documents and source code is emphasised in the case study. By saving all project documents and source code in a separate branch of the configuration management system, the organisation ensures version control and facilitates back integration with previous versions of the software product. This systematic approach to configuration management streamlines the closure process and supports future software development efforts.

**3. Lessons Learned:** The case study underscores the importance of knowledge management and lessons learned for continuous improvement. By utilising project documents available in the configuration management system, the organisation captures valuable insights and experiences from each project. The example of release 6.0 highlights the significance of learning from past challenges and adapting plans accordingly to mitigate risks.

**4. Adaptability and Risk Management:** The challenges faced during release 6.0 demonstrate the need for adaptability and proactive risk management in project execution. Despite elaborate planning, unforeseen complexities in the appointment scheduling functionality threatened the project's timeline and scope. The organisation's willingness to compromise on additional features and reallocate resources demonstrates agility in responding to project constraints and mitigating risks.

**5. Continuous Improvement:** The case study exemplifies the organisation's commitment to continuous improvement and iterative development. By leveraging lessons learned from past projects and adapting plans in response to challenges, the organisation strives to enhance the efficiency and effectiveness of its software development processes. This iterative approach fosters innovation and resilience in the face of evolving project requirements and constraints.

## Application in Real Projects:

The concepts we've delved into hold immense practical significance in real-world project scenarios. We thoroughly explored solution proposals, project plans, risk assessments, and budgeting documents as a team, ensuring a comprehensive understanding. Subsequently, we effectively distributed tasks among team members, accompanied by meticulous note-taking and constructive feedback exchanges to enhance collaboration and project outcomes.

## Peer Interactions/ Collaborative Learning:

We meticulously divided tasks among team members, taking into account individual strengths and expertise to optimise efficiency and productivity. This allocation process was informed by our thorough understanding of the project's requirements and objectives, ensuring alignment with organisational goals. In addition to task delegation, we prioritised effective communication and collaboration, fostering an environment where ideas flowed freely and feedback was actively sought and shared. Through note-taking and documentation, we captured valuable insights and lessons learned, facilitating continuous improvement and knowledge sharing within the team.

**Challenges faced:**

Sorting through research papers for the Sustainable Living Planner presented numerous challenges. Amidst a wealth of information, identifying the most pertinent content proved particularly arduous. Furthermore, delineating the deficiencies of existing similar applications and conceptualizing distinctive features to set ours apart posed a considerable obstacle. Additionally, assessing the feasibility and potential impact of these features proved challenging, given their absence in prior applications.

**Further Research/Readings:**

Next week, I am looking forward to immersing myself in the recommended readings from Chapter 8, focusing primarily on two key sources: "Learning Software Organizations: Methodology and Applications" by G. Ruhe (2001), presented at the 11th International Conference on Software Engineering and Knowledge Engineering (SEKE) and published by Springer in Berlin, Germany, and "Best Practice Tendering for Design and Build Projects" by A. Griffith and A. King (2003), published by Thomas Telford Ltd in London, U.K. These readings promise valuable insights into methodologies, applications, and best practices within the realms of software engineering and project tendering, respectively.

**Adjustment to Goals:**

Initially, I had intended to diligently complete the recommended readings for both Chapter 7 and Chapter 8. However, midterm exams demanded my immediate attention, causing me to postpone this task. Nonetheless, I am committed to catching up on these readings and aim to accomplish them by next week.