**Learning Journal**

**Student Name:** Sannidhi Shetty

**Course:** SOEN-6841 Software Project Management

**Journal URL:** [**https://github.com/sannidhi470/SPM**](https://github.com/sannidhi470/SPM)

**Week 4:** Feb 18 – Mar 8

**Date:** Mar 01

**Key Concepts Learned:**

**For the week of Feb 18 and Feb 25:**

Prepared for mid term examinations by going through the first 6 chapters.

**Below are the concepts I learnt from this week’s (March 9) session:**

**Chapter 6:** “Project Plan”

Chapter 6 discusses the crucial aspect of software project planning, which describes creating a comprehensive document outlining objectives, tasks, schedules, resources, and risks. This plan serves as a baseline throughout the project lifecycle, guiding project management activities to ensure the timely and efficient delivery of the software product. Key components of a software project plan include defining the project scope to establish boundaries, analyzing requirements to guide development, setting schedules to ensure timely completion, allocating budgets for expenses, assigning resources based on skills and availability, and managing risks through proactive identification and mitigation strategies. The chapter discusses top-down and bottom-up approaches of software project planning. There are various techniques in making a software project plan, such as Gantt charts for visualizing schedules, network diagrams for task sequencing, PERT/CPM charts for analyzing timelines.

**CHAPTER 7:** "Software Project Management & Software Engineering – Part I"

The chapter focuses on Project Monitoring & Control, which is an important step for the successful execution of any project. Below are the important topics mentioned:

Project Monitoring:

* A project plan serves as the baseline against which software project progress is measured.
* It is determined through Milestones in the project plan that helps track if target dates for completion are achieved as the project progresses based on the estimate done during initiation.

Monitoring Project Budget:

* Like monitoring the schedule, tracking the project budget becomes essential. Earned Value Management (EVM) is a reliable tool for measuring both schedule and budget progress. It will require accurate maintenance of baseline and actual data.

Task Dependencies and Critical Path:

* Understanding task dependencies and critical paths is important for project planning and monitoring.
* Tasks with dependencies must be completed sequentially, while the critical path determines the shortest duration for project completion.

Quality Assurance and Control:

* Project monitoring includes quality assurance and control processes to ensure deliverables meet predefined quality standards.
* Quality assurance focuses on proactive measures to prevent defects, while quality control involves detecting and correcting defects during project execution.

Purpose of Monitoring and Control:

* Monitoring involves observing project progress, while control involves taking corrective actions when necessary to ensure project progresses without any interruptions.

Designing a Monitoring and Control System:

* Establish baselines to monitor performance and compare it with baselines to take corrective action if needed.

Corrective Action Options:

* Options include adjusting schedules, reallocating resources, or revising plans.
* Performance indicators or project metrics should be used to help measure execution performance against the baseline plan.
* These will measure product quality, testing efforts, schedule variance, and budget variance.

Resource Management:

* Resource loading metrics help assess effective resource allocation to ensure enough resources are available during each phase of project development. Resource utilization metrics evaluate if resources are overloaded or underloaded leading to effective resource management.

Correcting Deviations:

* Deviations in project work can be addressed using available slack in the schedule or by resource leveling using the pipeline setup in the initial phase.
* Schedule optimization techniques can also help reduce unnecessary slack and shorten project durations.

In summary, the chapter describes the importance of closely monitoring project progress against the plan, employing effective tools and techniques to control deviations, and ensuring optimal resource utilization for successful project execution.

**CHAPTER 8:** “Project Closure”

Project closure marks the end of a project's lifecycle and explains various activities to wrap up the project and ensure a smooth transition.

* **Project Closure Process**: Project closure is the final phase of the software project development lifecycle, where all activities are formally concluded. It involves several key steps mentioned below:
* **Finalizing Deliverables**: This includes ensuring that all project deliverables, such as software modules, documentation, and reports, are completed and meet the expectations of stakeholders.
* **Client Acceptance**: It is important to obtain formal acceptance from the client or project sponsor to confirm that the project objectives have been met.
* **Resource Release**: Once the project is closed, resources such as team members, equipment, and facilities should be released so that it can be used in other projects.
* **Financial Closure**: Finalizing financial aspects like budget reconciliation, invoice settlements, and closure of contracts, is essential for proper project closure.
* **Lessons Learned**: For continuous improvement of software project, reflecting on the project's successes, challenges, and failures is important. Key activities include:
* **Post-Project Review**: Conducting a comprehensive review of the project to identify what went well, what didn't, and areas for improvement.
* **Documenting Lessons**: Capturing lessons learned in a structured manner ensures that valuable project knowledge is preserved for future projects. This documentation may include best practices, pitfalls to avoid, and recommendations for process improvements.
* **Knowledge Transfer**: Sharing lessons learned with relevant stakeholders, including project teams, management, and other project stakeholders, ensures that the knowledge gained is effectively utilized for further refinement in the project.
* **Version Management**: Managing source code versions is important for software development projects. Key aspects of version management include:
* **Version Control Systems (VCS)**: Utilizing VCS tools such as Git, SVN, or Mercurial to track changes, manage versions, and encourage collaboration among team members.
* **Branching and Merging**: Proper branching strategies and merging techniques help maintain code integrity and facilitate parallel development efforts.
* **Tagging Releases**: Tagging specific versions or releases of the software enables easy retrieval and reference, especially during project closure and future maintenance.
* **Metrics Data Filtration**: Filtering and archiving project metrics data is essential for several reasons:
* **Historical Analysis**: Archived metrics data provides valuable insights for historical analysis, trend identification, and performance evaluation.
* **Benchmarking**: Comparing current project metrics with historical data helps establish benchmarks and identify areas for improvement.
* **Compliance and Auditing**: Archiving metrics data ensures compliance with regulatory requirements and facilitates auditing processes.

In summary, Chapter 8 covers the importance of systematically closing out a project, documenting lessons learned, managing source code versions, and archiving project metrics data. These activities ensure that valuable knowledge is captured, resources are efficiently utilized, and future projects benefit from past experiences.

* Started working on project deliverable 2. Had team meetings to discuss feasibility study, solution proposal, project plan, risk assessment and budgeting documents. We collectively discussed on the solution that could be provided for the opportunity/problem selected for financial literacy app.
* Went through online resources available to help integrate API for our project solution. Here are some of them:
* **Plaid**: It is a service that helps app developers connect the applications to users' bank accounts securely. It offers tools that allow apps to access important financial information like verifying accounts, checking transaction history, and linking bank accounts together.
* **Yodlee**: It is a platform that helps developers build apps that can gather and analyze financial data from many different banks. This allows apps to provide services like helping people manage their money, create budgets, and even lend money, all using the financial data from users' accounts.
* **Finicity**: With Finicity, developers can do things like make sure a bank account is real, organize transactions into categories, verify income and assets, and help with credit decisions. This allows developers to create new and helpful financial apps and services for users.

We are planning to make use of few of these into our solution proposal to integrate financial institutions in financial literacy apps.

Top of Form

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

Through the first six chapters of the software project management course, I've gained an understanding of the best practices for effective project management.

Overall, the first half of the coursework has provided me with a solid foundation in software project management, providing me with the knowledge and skills necessary to navigate the complexities of project execution successfully. I feel confident in applying these concepts to real-world scenarios, ensuring timely and efficient project delivery while maintaining quality and adhering to budgetary constraints.

In Chapter 7, I learnt about monitoring and controlling project progress by setting a solid plan, managing resources effectively, ensuring quality standards, and tracking budget and schedule. Understanding task dependencies and critical paths helps in smooth project execution. Chapter 8 focuses on project closure, emphasizing formalizing project activities, documenting lessons learned, managing source code versions, and archiving project data for future reference. By applying these principles, project managers can ensure projects stay on track, meet quality standards, and conclude successfully, ultimately delivering value to stakeholders.

Top of Form

**Collaborative Learning:**

* Frequent project team meetings and discussions on project deliverable 2. The tasks are distributed amongst us and each of us make sure to give our updates in the meeting to ensure we all are in same page.
* Had team meetings to discuss and prepare for the pitch presentation.
* Had study session with few of the classmates to prepare for midterm during which we also shared how we used to apply these concepts on our individual software project experience. This collaborative approach allowed us to share insights, and clarify doubts, which enhanced our understanding of the subject matter.

**Further Research/Readings:**

* I came across a book "Software Project Management: A Unified Framework" by Walker Royce that offers an effective approach to managing software projects efficiently. The book covers essential topics such as project initiation, planning, execution, monitoring, and closure, providing practical techniques and strategies for each phase. The author emphasizes aligning project objectives with organizational goals and maintaining a unified approach to project management. Key topics in the book includes creating project charters, developing detailed plans, implementing projects, monitoring performance, and providing project closure. He also highlights the importance of collaboration, communication, and stakeholder engagement in achieving project success. As it offers practical insights and best practices, the book will be a valuable resource for software professionals to deliver successful software projects and effective project management for managers.
* "Project Management: A Systems Approach to Planning, Scheduling, and Controlling" by Harold Kerzner book provides practical insights into project management fundamentals, including monitoring and controlling project progress, managing resources, ensuring quality assurance, and closing out projects effectively. It covers various techniques and tools for project monitoring and control, such as Earned Value Management (EVM) and critical path analysis. Additionally, it addresses the importance of project closure, documenting lessons learned, and managing project archives. Overall, this book offers a well-rounded approach to project management.

**Adjustments to Goals:**

* Given the progress made in our team discussions and meetings, we will focus on further refining the feasibility study, solution proposal, project plan, risk assessment, and budgeting documents. This will involve going through additional research, gathering relevant data, and ensuring alignment with the project objectives and requirements.
* To read chapter 9 before the next class.