**Learning Journal**

**Student Name:** Neha Sudhir Himane

**Course:** Software Project Management

**Journal URL:** <https://github.com/nehahimane/SPM/blob/main/src/40219032_Neha_Himane_Learning_Journal.docx>

**Week 1:** Jan 15 – Jan 21

**Date:** 24 Jan 2024

**Key Concepts Learned:**

The focus was on the fundamental concepts of project management this week. A project is defined by its specific start and finish dates and seeks to achieve predetermined objectives. While software initiatives are similar to other types of projects, they provide unique issues such as invisibility, complexity, conformity, and adaptability. Project phases were introduced, including commencement, planning, monitoring and control, and closure, each with its own set of sub-processes.

SMART Objectives i.e. Specific, Measurable, Achievable, Relevant, Time-constrained objectives are introduced that provide a framework for defining project success criteria.

In addition, a technique called Project Division is introduced, which entails hiring an expert to evaluate effort and cost before accepting bids from software development businesses for project design and implementation.

**Application in Real Projects:**

Understanding project characteristics and software project issues will allow you to approach real-world projects more strategically. The importance of specialised skills in software project management, such as software engineering and testing, was stressed. Understanding the difference between project procedures and industry-specific processes, such as the software development life cycle, is critical for good project management.

**Peer Interactions:**

Discussions with peers provided a variety of opinions on project management. The investigation into how others viewed the challenges of software projects broadened my perspective and raised thought-provoking topics.

**Challenges Faced:**

Concepts like project stages and sub-processes required careful distinction. Making the distinction between project procedures and industry-specific processes was difficult, but necessary for a thorough understanding of the subject.

**Personal development activities:**

I participated in more readings, and I attended an online webinar about the most recent developments in project management. This external learning experience offered insights into developing techniques and tools, increasing my understanding.

**Goals for the Next Week:**

Next week, I aim to look more into software project management processes, namely requirements management, design management, and software testing. In addition, I intend to look into relevant case studies to apply theoretical knowledge to real-world scenarios.

**Week 2:** Jan 28 - Feb 3

**Date:** 2 Feb 2024

**Key Concepts Learned:**

This week, I concentrated on software project management's project effort and cost estimation. Understanding the inherent difficulties in estimating work because software development is an intangible process was one of the key ideas. In particular, I examined two main strategies: Algorithmic Cost Modelling and Experience-Based Techniques. Experience-based methods, such as analogous estimation, include applying multiplication factors, analysing estimates for subsystems, and comparing the new project with related previous projects. In contrast, algorithmic cost modelling estimates effort as a mathematical function by taking into account a number of attributes. This expanded on the groundwork established the week before, particularly with relation to project scope, objectives, project charter, and activities during initiation.

The key concepts learned about project effort and cost estimation in software project management are closely aligned with the fundamental principles introduced in the broader understanding of project management. Planning is one of the project phases that are highlighted in the curriculum, therefore the ideas that are acquired here are particularly relevant and provide budgeting and resource allocation insights. The implementation of SMART objectives aligns with the quantitative dimensions of effort and cost estimation, hence enhancing quantifiable and attainable project success criteria. Furthermore, the Project Division technique—which is covered in the content—directly depends on the proficiency in effort and cost estimation, highlighting the significance of these ideas in assisting with decision-making related to project design and implementation. All things considered, the concepts acquired improve the real-world implementation of project management principles, especially when it comes to handling the particular difficulties presented by software projects.

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

Engaging in the case study related to project effort and cost estimation provided valuable insights into the practical challenges of applying theoretical concepts. A notable takeaway was the significance of Experience-Based Techniques.   
The Software as a Service (SaaS) vendor's path is followed in the 3rd case study, which highlights the vendor's historical estimations, strategic staffing choices, and the challenges of a current project. The vendor used incremental development, initially projecting a product size of 500,000 SLOC. They looked into their possibilities and decided to go with offshore service providers for a staff of 50 because they needed a larger crew. Due to its sophisticated logic, the current project entails constructing an appointment scheduling engine, search capability, integration, and thorough testing. The project's estimated effort and cost of $300,000 SLOC highlights the careful preparation necessary for software project management. The case study emphasized that despite their subjective nature, these techniques, such as analogous estimation, play a crucial role in making initial project effort estimates.

**Collaborative Learning:**

The collaborative learning experience facilitated by the case study offered insightful knowledge about the real-world difficulties and decision-making procedures associated with software project management. Interacting with colleagues throughout the investigation provided a range of viewpoints, which improved our comprehension of the challenges the SaaS provider faced. Particularly insightful conversations included the importance of testing in a complex logic environment, offshore development, and hiring decisions. Group members' explanations and hearings of many points of view helped the group understand the case study more thoroughly, highlighting the collaborative element of learning in interpreting real-world project management circumstances.

**Further Research/Readings:**

I complemented the course material with further readings on project beginning approaches and

best practices. I delved into the book: " Software Project Management: A Process-Driven Approach" by Ashfaque Ahmed. This book provides comprehensive coverage of various aspects of software project management, including estimation techniques, planning, and execution. It offers practical insights and case studies that can complement the concepts discussed in your course.

Explored additional resource, such as upcoming webinar on “Mastering Effort and Cost Estimation in Software Projects” where professionals in the field will discuss the nuances of precise estimate methods. Find more about algorithmic cost modelling, experience-based approaches, and the difficulties in evaluating work for intangible software products. We'll examine real-world case studies that highlight effective tactics and lessons discovered. Regardless of your level of experience, this webinar will help you become a better decision-maker, allocate resources more efficiently, and make sure your software projects are successful. Don't pass up this chance to learn insightful advice and useful hints from professionals with a wealth of software development industry expertise.

**Adjustments to Goals:**

After giving the objectives from last week some thought, I saw that I needed to change course. I've made the decision to give a deeper understanding of effort and cost estimation methodologies priority because of the intricacy of the current software project management course material. As a result, I'm changing to dedicate more time to learning about algorithmic cost modelling and honing experience-based techniques. This change in emphasis will help me fully understand these difficult ideas, which is in line with my main objective of becoming an expert in software project management by the conclusion of the semester. This modification shows my dedication to modifying my learning approach to meet the changing needs of the course content.

**Week 3:** Feb 4 – Feb 10

**Date:** 10 Feb 2024

**Key Concepts Learned:**

This week, I have explored the fundamentals of project management, including project initiation, planning, execution, monitoring, and closure. Comprehending project management strategies and processes is crucial for the effective completion of projects. In addition to that, I have explored the significance of monitoring and recording system modifications, along with the range of capabilities offered by CM, including auditing, identification, control, and status accounting. A project management method called the Critical Path Method (CPM) can be used to find the longest chain of dependent activities and the shortest amount of time needed to finish a project. It shows the critical path, or the set of jobs that need to be finished on time in order to avoid delays in the project timeline.

Furthermore, the focus was on the project scheduling, budgeting, workforce planning, communication planning, and quality planning and learned in-depth methods like Goldratt's critical chain method, top-down planning, and bottom-up planning techniques. To optimise resource utilisation and reduce project time, Goldratt's Critical Chain Method is a project management technique that focuses on identifying and managing the critical chain of tasks in a project. In order to shorten the project completion time, superfluous buffers must be removed, and resource allocation must be optimised.

Connection to previous weeks:

Project management, risk management, and configuration management are interdependent concepts. Project management offers the structure for project supervision, risk management points out any roadblocks, and configuration management makes sure that modifications are properly controlled and documented along the project's lifecycle.

**Reflections on Case Study/Course Work:**

Upon reflection on the case studies in 4 and 5, it becomes evident that risk management and configuration control in software development projects are complex processes. The difficulties a SaaS seller encountered in Case Study 4 highlighted the vital significance of proactive risk mitigation techniques. Through the implementation of schedule buffers, feature prioritisation, efficient resource management, and strict quality assurance procedures, the development team successfully managed a number of risks associated with offshore collaboration, communication gaps, development expenses, schedule adherence, and software quality. These tactics reduced project risks while simultaneously promoting more efficient project management and improved product quality.

The case study 5 demonstrated how important a centralised configuration management system is for enabling smooth communication between scattered development teams. The software provider minimised build failures sped up the development cycle and ensured solid version control by promoting local build synchronisation, automating smoke testing, and setting up access rights management protocols. The case study serves as an example of how effective configuration management techniques support software development processes' stability, security, and integrity, which in turn produces fruitful project outcomes. All things considered, these case studies provide insightful information about the practical use of configuration control and risk management concepts in actual software development settings.

**Collaborative Learning:**

One of the collaborative learning experiences focused on the topic of configuration management. In a collaborative exercise, every team member was designated a particular facet of configuration management, including version control, access rights management, or automated testing, to investigate and report back to the group. We were able to effectively cover the breadth of configuration management procedures by breaking the issue down into smaller subtopics and giving each member the task of conducting in-depth research.

By sharing knowledge, materials, and best practices acquired from individual research, team members enabled us to explore every facet of configuration management in more detail. We were able to obtain a thorough understanding of the tactics and difficulties related to configuration management in software development projects by means of vigorous discourse and information sharing. Furthermore, collaborative learning provided a safe space for team members to explain ideas, pose questions, and come up with answers together.

**Further Research/Readings:**

Reading more about project management techniques like Agile, Scrum, and Kanban may provide you a better understanding of how contemporary project management techniques are used in software engineering projects.

Examining case studies of software projects that have succeeded and failed can provide insightful knowledge and real-world examples for comprehending the difficulties and ideal procedures in configuration management and project management. The paper "Effective Configuration Management in Software Engineering" by Roger S. Pressman is one that might be useful for further investigation. This paper offers insightful information about the role configuration management plays in software engineering processes. It covers all the important topics, including release management, version control, change control, and configuration identification. It also provides helpful advice on how to put an effective configuration management system in place.

**Adjustments to Goals:**

I have revised my goals in light of the information presented in chapters. via my initial emphasis on conceptual understanding, I've come to recognise the importance of combining academic knowledge with real-world application learned via case studies. To improve my ability to handle challenging project difficulties, this modification requires me to actively analyse real-world circumstances in order to derive actionable insights.

Furthermore, even though my initial objectives were to focus on individual learning, group projects highlighted the importance of peer interaction. As a result, I updated my goals to include more cooperative exercises like group talks and cooperative problem-solving. By interacting with peers, I hope to extend my horizons, enhance my comprehension, and develop the critical thinking abilities necessary for efficient software project management.

... Continue the Weekly Format for Weeks 3-13 ...

**Final Reflections:**

**Overall Course Impact:**

Summarize the overall impact of the course on your understanding.

Highlight key insights and transformations in your perspective.

**Application in Professional Life:**

Discuss how the knowledge gained in this course can be applied in your professional life.

Consider specific scenarios or projects where these skills would be valuable.

**Peer Collaboration Insights:**

Reflect on the value of peer collaboration throughout the course.

Consider how interactions with classmates contributed to your learning.

**Personal Growth:**

Share insights into your personal growth as a learner.

Identify areas where you have seen improvement or development.

Note: Ensure that the journal is updated weekly, at least twice a week, and that the publicly-accessible cloud service URL is provided for easy access by teaching assistants and for potential test-related inquiries.