**Learning Journal Template**

**Student Name:** Abhijit Banerjee

**Course:** Software Project Management

**Journal URL:** [Github Link](https://github.com/abhijitbanerjee05/SOEN-6841-Software-Project-Management.git)

**Dates Rage of activities:** 29.01.2025 – 09.02.2025

**Date of the journal:** 09.01.2025

**Key Concepts Learned:**

This week, I delved deep into two significant aspects of Software Project Management: Effort & Cost Estimation (Chapter 3) and Risk Management (Chapter 4). These topics are crucial because they form the foundation for making informed decisions regarding project planning, budgeting, and risk mitigation.

In Effort & Cost Estimation, I learned that software projects are inherently effort-driven due to their dependence on human labor. Since software development is largely an intellectual activity, estimating effort is challenging and requires structured methodologies. I explored different effort estimation techniques such as Function Point Analysis (FPA), Estimation by Analogy, the Delphi Method, and COCOMO models. Each technique has its advantages and limitations. For example, Estimation by Analogy compares a new project to a previous similar project, making it useful when past data is available. The Delphi Method relies on expert consensus, making it ideal when working with experienced professionals. The COCOMO model, on the other hand, is a mathematical approach that estimates project effort based on historical data and system complexity.

The second major topic was Risk Management, which focused on identifying, assessing, and prioritizing risks that could impact a software project. I learned that risks could arise from various factors, including technical issues, resource unavailability, budget constraints, and unexpected changes in project scope. Understanding risk categories, such as technical, legal, organizational, and economic risks, helped me see how different risks require different management strategies. Risk assessment involves both qualitative and quantitative analysis, where likelihood and impact are measured to determine the severity of a risk.

**Application in Real Projects:**

As I reflected on how these concepts apply to real-world projects, I realized that effort estimation is fundamental to my own software development experiences. In my previous work on a microservices architecture project using Spring Boot, Kafka, and Elasticsearch, having a well-defined estimation technique could have helped in determining project timelines and allocating resources more effectively. Function Point Analysis (FPA) would have been particularly useful for breaking down complex features into measurable components. Additionally, in projects involving emerging technologies, such as AI or machine learning, estimating effort can be even more challenging due to uncertainties in development time and training data requirements.

Risk management is another crucial aspect that applies to real-world projects. For instance, while working on web applications, one common risk is technology obsolescence. Selecting a framework that might become outdated in a few years poses a long-term risk, requiring mitigation strategies such as choosing well-supported technologies or designing modular architectures. Similarly, in collaborative projects, resource unavailability—such as team members leaving midway—can be a significant risk, necessitating proper documentation and knowledge-sharing practices to ensure continuity.

**Peer Interactions:**

Reflecting on how these concepts apply to real-world projects, I can see the importance of accurate effort estimation in setting realistic timelines and budgets. For instance, in my previous work on a microservices architecture using Spring Boot, Kafka, and Elasticsearch, applying function point analysis could have provided better estimates. Risk management strategies are also crucial in selecting third-party dependencies, ensuring they remain stable and do not become obsolete during development.

**Challenges Faced:**

During discussions with my peers, we debated the accuracy of COCOMO II models in modern software development. One of my classmates shared an experience where their team overestimated effort due to uncertainty in using AI-based tools. This made me realize the importance of refining estimates as the project evolves and how emerging technologies introduce new challenges in estimation accuracy.

**Personal Development Activities:**

To deepen my understanding, I engaged in several personal development activities. I reviewed case studies of companies that successfully implemented risk management strategies, which helped me see how theoretical concepts are applied in practice. Additionally, I worked with peers to apply the Delphi Method to estimate effort for a hypothetical project, allowing me to experience firsthand how expert-driven consensus can refine estimates. I also explored cost estimation tools used in the industry, such as Jira and Microsoft Project, to see how manual effort estimation compares to automated approaches. This gave me a clearer picture of how estimation techniques are used in professional settings.

**Goals for the Next Week:**

For the next week, I aim to deepen my understanding of effort estimation for complex software projects, particularly those involving highly unpredictable technologies like AI and cloud computing. I also want to study real-world cases of failed projects due to poor risk management to identify common pitfalls. Lastly, I plan to apply risk assessment and mitigation strategies to an upcoming academic or personal project, ensuring that I can put my theoretical knowledge into practical use. I believe that by doing so, I will gain more confidence in handling real-world software project challenges.