**Learning Journal Template**

**Student Name:** Purvesh Shailesh Sakhiya (Student Id: 40221354)

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

**Journal URL:** <https://github.com/purveshsakhiya9/COMP-6841---Software-Project-Management>

**Week 1:** 18/1/2024 – 24/1/2024

**Date:** 24/1/2024

Key Concepts Learned:

1. **Project Definition:** A project is a set of activities with a definite start and end time, aiming to achieve predefined goals. It consumes resources, time, and budget.
2. **Software Project Characteristics:** Software projects share similarities with other projects but are distinguished by factors like invisibility, complexity, conformity, and flexibility.
3. **Project Phases:** Software projects typically go through phases such as initiation, planning, monitoring & control, and closure.
4. **Software Project Management Fundamentals:** Specific tasks like requirement management, design management, source code building, testing, deployment, and maintenance are integral to software project management.
5. **Project Metrics:** Quality characteristics required for software project metrics include relevance, meaningfulness, practicality, calibration ability, and activity level.
6. **Project Initiation:** Involves creating a project charter, defining project scope and objectives, estimating budgets, and creating initial project schedules.
7. **Objectives and Goals:** Objectives should be SMART (Specific, Measurable, Achievable, Relevant, Time-constrained). Goals are steps toward achieving objectives.

Application in Real Projects:

1. **Practical Project Management:** Understanding project phases and initiation processes is crucial for effective project management. This knowledge can be applied in planning and executing real-world projects.
2. **Metrics for Evaluation:** Implementing relevant and meaningful metrics is vital for assessing the success of a project. It helps in making informed decisions and improvements.
3. **Effective Initiation:** Proper project initiation, with a well-defined charter, scope, and objectives, sets the foundation for a successful project. Real projects can benefit from clear direction and purpose.

Peer Interactions:

1. **Knowledge Exchange:** Discussions with peers provided insights into their experiences with project management, particularly in the software development field.
2. **Best Practices:** Shared best practices for project initiation and how certain metrics proved useful in different project scenarios.

Challenges Faced:

1. **Clarification on Metrics:** Some nuances of project metrics and their calibration need further clarification for a deeper understanding.
2. **Integration of Tools:** Understanding how various tools and technologies are integrated into project management processes requires further exploration.

Personal Development Activities:

1. **Research on Project Metrics:** Engaged in additional reading and research on project metrics to enhance understanding and application in real projects.

Goals for the Next Week:

1. **Deeper Understanding of Project Tools:** Focus on gaining a deeper understanding of the tools and technologies used in project management processes.
2. **Explore Advanced Project Management Concepts:** Research and study advanced project management concepts beyond the fundamentals covered this week.
3. **Apply Metrics in Simulated Projects:** Practice applying project metrics in simulated project scenarios to reinforce the understanding gained.

**Week 2:** 28/1/2024 – 3/2/2024

**Date:** 3/2/2024

Key Concepts Learned:

1. **Effort Estimation:** Understanding that software projects are effort-driven, and estimating effort is challenging due to the intangible nature of the results.
2. **Estimation Techniques:** Differentiating between experience-based techniques (like analogy and expert judgment) and algorithmic cost modelling.
3. **Function Point Analysis:** A method for effort estimation involving determining function count types, boundary and scope, unadjusted function point count, and applying value adjustment factors.
4. **COCOMO Cost Modelling:** An empirical model based on project experience, incorporating sub-models for different stages of software development.
5. **Resource Estimation:** Determining the number of people needed on a project, considering skills required, and accounting for variations in individual efficiency.

**Reflections on Case Study/Coursework:**

The case study emphasized the importance of accurate effort estimation at the project initiation stage. Insights gained reinforced the need for a flexible approach to estimation, considering the unique characteristics of each project. The practical application of function point analysis and COCOMO modelling was highlighted in real-world scenarios.

**Collaborative Learning:**

Engaged in group discussions about the challenges of effort estimation, particularly in the context of diverse software development lifecycle models. Peer interactions provided additional perspectives on the practicalities of applying estimation techniques in various industry settings.

**Further Research/Readings:**

Explored additional resources on algorithmic cost modelling and COCOMO 2 models. Readings provided deeper insights into the practical application of these models, their strengths, and limitations. The additional resources complemented the course material by offering real-world examples and case studies.

**Adjustments to Goals:**

Adjusted goals to focus on gaining a deeper understanding of function point analysis and its application in different software projects. Recognized the need to explore more examples of COCOMO modelling in diverse industry contexts. Updated goals to align with evolving interests and practical applications encountered during the week.