**Learning Journal**

**Student Name:** Umang Savla

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

**Journal URL: [https://github.com/umang232/SPM](https://github.com/sannidhi470/SPM)**

**Week 2:** Jan 23 – Jan 30

**Date:** Feb 9

**Key Concepts Learned:**

Below are the concepts I learnt from this week’s session:

**CHAPTER 3: Effort & Cost Estimation in Software Projects**

* In this chapter, we went through the process of estimating how much effort and money it takes to develop a software solution. In last class (chapters (1, 2)), we focused on the challenges of figuring out the human effort needed for software development.
* **Software Project Effort Estimation:**
* Software projects mostly rely on people's hard work. To guess how much effort is needed for software is tough because the released software is not something you can touch. So, we can adopt various estimation techniques that will help managing projects efficiently:
* **Estimation Techniques**:
* We use judgment to estimate effort directly or based on project details.
* Two estimation techniques: 1) Experience based techniques 2) Algorithmic cost modelling.
* **Experience-Based Estimation Approaches:**
* We can estimate the effort by Analogy: Compare the features of new projects to similar old ones.
* Estimation by expert judgement.
* **Function Point Analysis (FPA):** It is a software cost estimation method that focuses on quantifying the functionality of a system. It involves identifying and weighting different functions based on factors like inputs, outputs, inquiries, files, and interfaces to calculate Unadjusted Function Points (UFP). General System Characteristics (GSC), such as data communications and performance, are then considered to adjust the function points and derive Adjusted Function Points (AFP).   
  FPA classifies functions into Data Functions (Internal Logical Files, External Interface Files) and Transactional Functions (External Inputs, External Outputs, External Inquiries). It assigns complexity weights to these functions based on data elements and processing intricacy. This estimation process helps in determining a productivity factor based on historical performance to calculate effort estimation.

For projects that grow bit by bit, we use Function Point Analysis differently. It involves counting specific functions and adjusting for complexity. However, it has various challenges as we need to rely on past experiences that can be tricky if the new project is very different.

* **Algorithmic Cost Modeling:**

Estimate cost using a math formula based on project details. Challenges include accurately estimating factors like code size.

* **COCOMO Cost Modeling:**  
  COCOMO (Constructive Cost Model) is a popular method to estimate the time and effort needed to develop software. It does not belong to a specific software vendor and has evolved since its initial version in 1981 to COCOMO 2 in 1995. COCOMO 2 considers various software development approaches and includes sub-models like the application composition model, early design model, reuse model, and post-architecture model, tailoring estimates for different project stages. The basic COCOMO model considers lines of code for estimation, but COCOMO 2 adapts to the system's life cycle with models for different stages. The estimation formula involves constants, project size measured in kdsi, an exponent scale factor, and effort multipliers, ensuring more accurate predictions for different project scenarios.
* **Effort and Cost for Various Project Tasks:**

Understanding how effort and cost would be different for Iterative based development and waterfall model of development. It is important to think about the skills of these people, and the speed at which they work and complete the task, which affects how we estimate resources for the project.

* **Conclusion:**

This chapter gave us a deep dive into estimating the effort and cost of developing a software. We understood various methods and learned about the challenges in estimating the effort needed for software projects. This knowledge will help in better project management in the world of software engineering.

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

* Based on the concepts learned in the class, I understand that it can be applied to real world project like developing a new e-commerce platform. We can use Function Point Analysis (FPA) where the team can quantify the functionality by assessing the inputs, outputs, inquiries, files, and interfaces specific to the platform. This will help in estimating the effort required for tasks like implementing a secure payment gateway, tracking orders, or managing inventory. Additionally, Algorithmic Cost Modeling (e.g., COCOMO) can be used to calculate the development time and effort based on project size, lines of code, and other factors. Experience-based estimation comes into play when comparing features to similar e-commerce projects which will offer insights into potential challenges and required resources. These combined approaches enhance accuracy in estimating effort and cost helping in effective project management and resource allocation.
* Case Study 3 focuses on how the cost and effort estimation is done for the project of SaaS vendor. The SaaS vendor started on developing a software product with an estimated size of 500,000 source lines of code (SLOC) and chose to go for incremental development. Initially, a team of 22 members, with a quarterly cost of $400,000 was set up for two years, resulting in a development cost of $3,200,000. In order to accelerate the development process, they expanded to a team of 50, considering hiring locally but they found hiring offshore development member would be more cost-effective at $730,000 per quarter. Their ongoing project focuses on building an appointment scheduling engine, search functionality, integration, and performing thorough testing. Testing would be crucial due to the complexity of the newly implemented logic. The effort for the entire functionality was estimated to be around 300,000 SLOC which spans four iterations and a major release.

**Collaborative Learning:**

* Had project team meetings to discuss on the assigned topic “Financial Literacy App”. We shortlisted some potential opportunities in the domain. We distributed tasks amongst us to go through these problems/opportunities and in our next meet, we will be finalizing one and start working on the Market Analysis and Project Initiation.
* Had a study session with a classmate to go through chapter 3 and how can it be applied to our projects. As we have experience working in software industry, we shared our experiences as to how effort and cost estimation used to be done in our projects and if we were able to deliver the product within the estimated cost. We had a discussion on the challenges we faced. My friend faced issues with experimental based cost estimation as he had to develop a change request that didn’t resemble to any of the past developments done in the projects. We discussed how it can be a problem in large scale projects as the rapidly evolving nature of software development poses a challenge for analogical estimation.

**Further Research/Readings:**

* I came across this book "Software Estimation: Demystifying the Black Art" by Steve McConnell that provides practical insights into software estimation, covering various techniques on effort/cost estimation. McConnell talks about the challenges involved in estimation, emphasizing that estimating the cost at the beginning can be uncertain but become clearer as a project progresses. He calls this concept “Cone of Uncertainty”. He explains various techniques to estimate, including learning from past projects, using benchmarks, and considering risks. The book also covers on how to estimate in agile environments where there are continuous changes. McConnell suggests having good communication and negotiation during estimation. Overall, it gave me additional knowledge on how to perform cost estimation and various concepts around it, offering tips and strategies to make estimation more accurate and manageable.

**Adjustments to Goals:**

* The goal for last week was to start working on the project. We had team meetings and have shortlisted some problems/opportunities in financial literacy apps. We are focusing on understanding existing application as part of market analysis which will help us know the gap in market and provide effective software solution to it. We will begin with the project initiation by selecting one of the shortlisted problems.
* After attending the class for chapter 3 and reading extra materials, my knowledge on effort and cost estimation of software product has evolved. This will help me in the project and when working in real world projects.