

Software Project Management **Learning Journal**

Name: Adarsh Manojkumar Pawar

Course: Software Project Management

Journal link: <https://github.com/AdarshPawar/LearningJournal.git>

Date: 01/02/2024

Week 3: January 28 - February 3

Concepts Learned

Chapter 3 - Effort and Cost Estimation in Software Projects

In this chapter, we are introduced to effort estimation, the process of calculating amount of time and resources required to complete a project. It is said that tight budgets and tight schedules are the general norm for most projects today and this makes good and reliable effort, schedule and cost estimates for projects even more important.

The most important aspect here is that, more the information available about the project, the more accurate will be the estimate. Different situations require different estimation techniques but broadly they're classified into two types, **Experience based techniques** and **Algorithmic cost modelling**. It is also recommended to update project estimates regularly as we gather more information about the project. This helps ensure the estimates are as accurate as possible.

Functional Point Analysis

Function Point Analysis (FPA), an experience based technique is recommended when data available for both current as well as previous projects. It looks at two main things: **the size of the project and the productivity of the team**. The project size is figured out based on what the customer wants, like how many functions need to be built and how complex they are.

Then, **the number of function points (FPs) is determined** for each function. This is done by looking at things like the complexity of the functions and how many parameters they have. The total FPs for the whole project are added up to get the unadjusted FP count. After that, adjustments are made based on factors like the type of project and its integration needs. Finally, the complexity of each function is determined, and a value of FPs is assigned to each file based on that complexity.

Once the total number of function points (FPs) for the entire system is determined, we can calculate the **effort estimate for the project by multiplying this number with the team's productivity**. Productivity calculation is based on the experience and skills of the project team members and is often derived from statistical process control data from previous projects. An industry norm for productivity is typically expressed in terms of **effective lines of code (LOCs) generated per month per person**, accounting for work done in various phases of the development life cycle. For projects where statistical process control (SPC) data are unavailable, productivity may need to be calculated from scratch or based on an industry average figure.

COCOMO model

COCOMO model, an algorithmic cost modelling technique is recommended when good information is available for the current project but no data is available for previous projects. It employs a comprehensive set of project assumptions, definitions, and numerous cost factors to assess effort estimates for any given project.

At its core, COCOMO utilizes **source lines of code (LOCs)** as the primary measure of the volume of work required for the project, upon which the effort estimate is based. It also factors in cost drivers and scale drivers, both of which significantly influence the overall effort estimation process. Cost drivers are categorized into four main groups:

- **Personal factors** pertain to attributes such as analyst capability and programmer skill
- **Product factors** relate to characteristics like database size and software complexity.
- **Platform factors** consider constraints such as execution time and storage
- **Project factors** involve aspects like software tool usage and development schedule requirements

Scale drivers include attributes such as development flexibility, and team cohesion which play a crucial role in refining the effort estimate by accounting for various project-specific considerations.

COCOMO provides different variants to accommodate the diverse needs of software projects. **Basic COCOMO**, for instance, simplifies the effort estimation process by relying on a basic calculation equation that considers LOCs and adjustment factors. **Intermediate COCOMO**, on the other hand, offers a more

refined estimation approach by incorporating additional factors such as product size and a broader range of cost drivers.

Cost Estimation

After obtaining the effort estimate for the project, it becomes necessary to calculate the costs associated with the project where the typical approach involves converting the effort estimate into man-months followed by applying a standard man-month rate for the project. In project management, there are two main types of projects: fixed cost–fixed schedule based and time and material based.

Fixed cost–fixed schedule projects are those where requirements are clear and most project details are well-defined from the outset. Costing for such projects is straightforward as the expenses are determined at the beginning. However, not all projects start with such clarity. Many projects begin with uncertainties and ambiguities, making it challenging to establish precise costing and scheduling. In such cases, projects are often managed on a **time and material costs basis**, where the customer agrees to pay for the time spent by the project team, typically through monthly fees.

When it comes to estimating costs for iteration-based projects, the process remains similar to traditional waterfall model projects. Costs are determined for each iteration as well as for major software product releases. Additionally, costing for the entire product development can be computed by aggregating costs across all major releases, providing a comprehensive view of project expenditures at different levels.

Reflections on case study / coursework

I wanted to get a clearer picture of effort estimations in an enterprise when I came across this case study where a large financial institution worked on the development of a comprehensive online banking platform. The project scope included account management, fund transfers, bill payments, and personalized dashboard features. Due to the project's complexity and the criticality of delivering a robust and user-friendly application, the organization opted to utilize both FPA and the COCOMO model for effort estimation.

This was a shocking revelation to me as I had assumed that one had to use either of the methods to calculate their effort estimations and hence I decided to dig deeper. I found that FPA assesses the functional requirements of the project, providing a thorough understanding of its scope and complexity through function points. The COCOMO model complements FPA by considering a broader range of project attributes and cost drivers to refine the estimation process further.

When used separately, FPA excelled at evaluating the application's size and complexity but overlooked other critical factors affecting estimation accuracy. COCOMO, on the other hand provided a comprehensive approach but oversimplified estimation by focusing primarily on lines of code. Integrating FPA and COCOMO allowed the organization to leverage the strengths of both approaches while addressing their limitations. This integration resulted in more accurate estimates, aiding in better project planning, resource allocation, and risk management.

Collaborative Learning

Our project group was finally able to meet and we went over our project, Community Skill Exchange platform together. We brainstormed different ideas and perspectives for 3 hours and finalized on one. Previously, I used to have a hard time identifying the main problem amongst other problems, but after having had discussions, examining different examples with my teammates, I am now able to

identify the core problem in a given situation. For the stakeholder analysis, I was surprised to hear my teammate suggest government authorities as one of the stakeholders, but after further research we understood on why he was right, and what aspects make government authorities a stakeholder in our project.

My friend and I tried to understand effort estimations by thinking of real world scenarios and trying to plug realistic values into the estimation formulas. We then calculated what the effort would be. Eg - If five developers developed the application in 5 months, 2.5 months doing unit and integration testing and took 1 month for doing the system testing. Debugging for developers again took 15 days, system developed contained 30,000 SLOC etc. This gave us a better idea on how to perceive the above in real world scenarios

Further Research/ Readings

As a full stack developer, I wanted to know if there were any specific implications for web application development with regards to estimation techniques, and I came across the paper, "A Comparative Analysis of Software Cost Estimation Methods for Web Applications" by Zeyad Tareq Saeed Alwahedi and Ahmad Al-Omari, published in the Journal of Software Engineering and Applications which explores and compares various software cost estimation methods specifically tailored for web applications, which have unique characteristics and challenges compared to traditional software projects. The authors conduct a comprehensive review of existing cost estimation techniques, including parametric models, expert judgment-based methods, and hybrid approaches.

The study evaluates the effectiveness and accuracy of these estimation methods by analyzing their applicability to web application development projects. The article also discusses the importance of considering non-functional requirements, such as performance, scalability, and security, in cost estimation for web applications. It also emphasizes the need for empirical validation and continuous refinement of estimation models to improve their accuracy and reliability in real-world scenarios.

I was able to gain valuable insights into the challenges and opportunities associated with estimating the cost of web development projects and got a good

idea on how to select appropriate estimation techniques based on project characteristics and requirements.

Adjustment to goals

I was able to thoroughly understand the management metrics discussed in Chapter 1 as I had spent some time going through the coursework as well as a few YouTube videos to further enhance my understanding.

My goal was to go through both chapter 3 and 4, however I could find time to go through only chapter 3 and hence will be adjusting my goal to cover chapter 4 material before I attend the class the coming week. Apart from this, I plan to read about and get a good understanding on market research. This will be very helpful for the completion of the first deliverable of the project as well as its implementation in the real world .