Learning Journal

Student Name: Aswiin Ravi Prakash

Course: SOEN 6841 Software Project Management

Journal URL: https://github.com/aswiinraviprakash/SOEN6841-JOURNALS-ASWIINRAVIPRAKASH

Week 2: 28/01/2024 - 03/02/2024

Date: 03/02/2024

Key Concepts Learned:

The basic understanding of how the software projects are effort driven and how effort estimation can be done based on various factors and techniques. The lecture covered two main categories of effort estimation techniques they are.

Experience-based Techniques - Relies on past project experiences. Some of the techniques include Estimation by Analogy and Estimation by Expert Judgment.

Algorithmic Cost Modeling Techniques - Involves mathematical functions based on product, project, and process attributes. Some of the techniques include Function Point Analysis, Wide Band Delphi, and COCOMO.

Gained an understanding that all these estimation techniques depend on software projects being developed and project has a possibility of revising the technique when project grows. The lecture covered deep delve into all these techniques individually.

Estimation by Analogy:
This is one of the experience-based technique Involves comparing the current project with a similar past project to estimate effort and cost. The process unfolds in several steps, starting with the decomposition of previous projects into subsystems such as databases, tables, user interfaces, and business rules. In the second step, the size of the current project is compared to that of a comparable past project, focusing on individual subsystems. A multiplication factor (MF) is calculated for each subsystem by dividing the number of elements in the old project by the corresponding count in the new project. Subsequently, the sizes of each subsystem in the new project are determined by applying the multiplication factor. The total project size is then obtained by summing up the sizes of all subsystems. Finally, an effort estimate for the new project is derived based on its size. This technique leverages historical data and analogies to provide a structured and data-driven approach to project estimation.

Function Point Analysis:
This method relies on function point metrics to provide a standardized approach for measuring different functions in a software application. These metrics are centered on the user's perspective, focusing on what users expect and receive from the software.

The main objectives of this technique are.

- Measure functionality that the user requests and receives.
- Measure software development and maintenance independently of technology used for implementation.

The simplicity and consistency of the function points used across projects are crucial. The first step involves defining a boundary for measuring function points, followed by determining the Unadjusted Function Point (UFP) count. This count reflects the countable functionality offered to the user by the project and is calculated based on two data type functions (internal and external logic files) and three transaction type functions (external input, external output, and external inquiry). Each function type is ranked by complexity, and predefined weights are applied to compute the UFP. The Function Point (FP) is then calculated as the product of UFP and the Value Adjustment Factor (VAF). This technique has a disadvantage that FPA doesn't capture all functional characteristics of real-time software.

Wide band Delphi:
This technique involves team members independently estimate project components, meet to discuss, and compare these estimates, averaging them to establish a collective reference point. Through collaborative discussions, a consensus estimate is reached, and, for a more comprehensive view, a range of estimation values is calculated to account for potential variations within the team's assessments.

Differences Between Both Models:
Experience-based models for project estimation leverage insights from past projects, comparing similar subsystems to estimate the effort and cost of a new project. This method relies on the assumption that historical projects share similarities with the current one. However, its effectiveness may be limited when dealing with new projects that diverge significantly from previous experiences. The rapid evolution of software development and the adoption of unfamiliar techniques in a new project can further diminish the applicability of past experiences, making it challenging to accurately estimate effort and cost.

On the other hand, cost-based models employ mathematical functions to estimate project cost, with a common focus on attributes such as code size. This approach provides a structured and systematic framework for estimation, allowing for a more quantitative and objective analysis. The advantage of cost-based models lies in their ability to offer reliable estimates even for projects with unique characteristics or unfamiliar technologies. By focusing on attributes that can be quantified, such as code size, these models provide a more consistent and adaptable method for project estimation, irrespective of the novelty of the project.

Reflections on Case Study/Course Work:

The weekly exercise on agile software methodology cost was an excellent illustration of how project estimation can vary based on different methodologies and other influencing factors. The case study

offered a practical example of how a company's budget could undergo significant changes throughout the development process. This real-time scenario effectively demonstrated the application of learned concepts and provided valuable insights into the dynamic nature of project budgeting, particularly in the context of agile methodologies. The exercise highlighted the importance of adaptability and responsiveness in managing costs, showcasing the nuanced considerations that come into play during software development projects. Overall, it served as a valuable learning experience, connecting theoretical concepts with real-world applications and offering a deeper understanding of the complexities involved in project estimation.

Collaborative Learning:

Engaging in discussions and group activities with my peers proved to be a valuable aspect of my learning experience. The collaborative nature of these interactions not only deepened my understanding of the subjects but also provided insights from diverse perspectives and experiences. The collective exploration of topics allowed for a more thorough examination, and the exchange of ideas among group members significantly enriched the overall learning process.

Further Research/Readings:

I used both my textbook and online resources to really understand the concepts better. When working on the exercises about agile software methods, I checked out several websites to find more information and solutions. This mix of using a regular textbook and exploring online sources helped me get a deeper grasp of the subject and tackle the exercises more effectively.

Adjustments to Goals:

The lecture covered this week provided a great insight about project estimation and cost. The upcoming chapters are based other phases of project management. My goal is to explore the topics learned and compare it with real world examples and get an outline of chapter 4 about Risk Management to gain a better understanding during lecture hours.