

Learning Journal -3

Student Name: Mobbu Mahathi

Course: SOEN 6841 Software Project Management

Journal URL: <https://github.com/mmobbu/SPM>

Dates Range of activities: 6 February 2025 to 20 February 2025

Date of the journal: 22 February 2025

Key Concepts Learned:	Application in Real Projects:	Peer Interactions:	Challenges Faced:	Personal development activities:	Goals for the Next Week:
1. Software project attributes: invisibility, complexity, conformity, and flexibility. 2. Project management phases: initiation, planning, monitoring & control, and closure. 3. Project Division: A method where professionals estimate effort and cost prior to bidding on the project. 4. Function Point Analysis (FPA): calculates software size using function points, which are tied to end-user interactions. 5. Wideband Delphi Technique: Involves a number of expert judgments to improve effort estimates.	1. Project initiation understanding assisted me in establishing well-defined objectives and scope, minimizing project failure risks. 2. Methods like function point analysis and analogy-based estimation helped with budgeting and resource planning.	1. Investigated function point analysis in study groups and compared it to story points in Agile. 2. Contended whether analogy-based estimation is reliable in current software projects. 3. Learned from feedback on improving project scheduling using automated tools. 4. Understood the practical budgeting challenges in real life.	1. Lack of ability to distinguish between different estimation techniques and when to apply each. 2. Grasping the boundaries of experience-based estimation in ever-evolving technology environments. 3. Understanding cost models, in this case, COCOMO, because of its complexity. 4. More actual-world	1. Examined Agile estimation techniques like T-Shirt Sizing and Planning Poker. 2. Attended few online classes on software project management risk assessment. 3. Used estimation tools such as JIRA and Trello to hone project planning.	1. Gain a deeper insight into risk management techniques in software projects. 2. Discover Agile estimation techniques and how they work better than the old ones. 3. Research cloud and AI-based software project cost estimation. 4. Identify research automation tools for estimating effort and cost to reduce human dependency.

<p>6.Algorithmic Cost Modeling: Using mathematical formulas that rely on project attributes.</p> <p>7.COCOMO Model: Empirical cost estimation model with several sub-models based on project phases.</p>			<p>project estimating case studies.</p> <p>5.How to incorporate estimation techniques to enhance accuracy.</p>		
<p>1.Definition of risk and its potential impact on software projects.</p> <p>2.The need for iterative development to minimize risks.</p> <p>3.How risk prioritization is used in software development.</p> <p>4.Importance of traceability and version control in preventing software defects.</p> <p>5.Manage change request strategies and baselining software components.</p> <p>6.The function responsible for authorizing changes by configuration control boards (CCB).</p> <p>7.Components of a software project plan: scheduling, budgeting, resource planning, communication planning, and quality planning.</p>	<p>1.An official risk management plan in agile projects minimized uncertainties.Prioritization of risks in early development phases prevented cost overruns and delays.</p> <p>2.With the use of tools like Git, Jenkins, or CI/CD pipelines, version control and change management were enhanced.</p>	<p>1.Talked about peer risk prioritization methods, including how they learned about actual case scenarios of risks.</p> <p>2.Provided a group discussion of best practices for software configuration management, namely how version control can prevent software deployment issues.</p> <p>3.Obtained feedback on project scheduling techniques, confirming the importance of adaptable planning in agile environments.</p>	<p>1.Quantitative risk assessment models needed to be understood through further research.</p> <p>2.The application of Goldratt's critical chain technique to real-world applications seemed to be cumbersome due to the nature of software projects.</p>	<p>1.Researched more sophisticated risk management models beyond the textbook, such as FAIR (Factor Analysis of Information Risk).</p> <p>2.Practiced creating a Work Breakdown Structure on a hypothetical project to reinforce project planning concepts.</p> <p>3.Performed case studies of unsuccessful software projects to determine risk factors and mitigation strategies.</p>	<p>1.Improve understanding of risk mitigation strategies and their application in software development.</p> <p>2.Investigate industry-standard configuration management tools</p> <p>3. Employ scheduling techniques like Gantt charts and CPM with project management software.</p> <p>4. Link learnings to eventual career objectives by indicating how ideas in project management are utilized in leadership positions in software engineering.</p>