

**Student Name:** Mihir Patel

**Course:** SOEN 6841 Software Project Management

**Journal URL:**

[[https://github.com/Mihir8754/mihir8754/blob/SOEN\\_6841\\_LR's/Learning%20Journal%202.pdf](https://github.com/Mihir8754/mihir8754/blob/SOEN_6841_LR's/Learning%20Journal%202.pdf)]  
]

**Dates Range of activities:** 20/01/25 – 03/02/25

**Date of the journal:** 07/02/25

### Chapter 3

<b>Key Concepts Learned:</b>	Effort Estimation: Experience-based (Analogy, Expert Judgment) and Algorithmic (COCOMO 2). Function Point Analysis (FPA): Measures software size for effort estimation and includes Unadjusted Function Points (UFP) and Value Adjustment Factor (VAF). Delphi Method: Collaborative estimation approach. Cost Estimation: Based on effort and resource parameters. Estimate Uncertainty: Effort estimates require continuous refinement as projects progress.
<b>Application in Real Projects:</b>	FPA is useful in requirement-driven projects, while COCOMO 2 suits large-scale development. Agile projects allow progressive refinement, whereas Waterfall requires precise upfront estimates. Understanding estimation techniques helps prevent project delays and budget overruns.
<b>Peer Interactions:</b>	Discussed challenges in effort estimation with peers, highlighting difficulties in estimating projects involving AI and emerging technologies. Debated whether experience-based or algorithmic estimation is more reliable, with a consensus on using a hybrid approach. Explored real-life case studies on estimation failures and their impact on project success.
<b>Challenges Faced:</b>	COCOMO 2 Complexity: Simplified by using case studies and estimation exercises. Experience vs. Algorithmic Estimation: Balanced through real-world comparisons and hybrid approaches. Adapting Estimation Methods: Recognized the need for flexibility based on project size and complexity.
<b>Personal development activities:</b>	Reviewed additional resources on Function Point Analysis to improve accuracy. Completed practical exercises on effort estimation techniques. Attended an online workshop on software project estimation best practices.
<b>Goals for the Next Week:</b>	Compare different estimation techniques used in Agile and Waterfall methodologies. Deepen understanding of Agile estimation techniques like Story Points and Planning Poker.

## Chapter 4

<b>Key Concepts Learned:</b>	Types of Risks: Estimation, technology, resource, quality, and schedule risks. Risk Management Process: Identification, analysis, prioritization, and mitigation. Response Strategies: Acceptance, Avoidance, Transference, and Mitigation. Quantitative Risk Assessment: Risk exposure formula (Risk Probability × Impact). Risk Control Measures: Strategies for continuous risk monitoring and mitigation. Risk Prioritization Techniques: Risks are ranked based on likelihood and severity to focus mitigation efforts effectively.
<b>Application in Real Projects:</b>	Risk assessment ensures proactive project management. Strategies such as quality gates and knowledge sharing help mitigate risks effectively. Early identification of risks leads to better contingency planning and reduces project failures. Risk buffers are often used in schedules to accommodate unexpected issues and avoid delays. Contract-based risk transference is common in outsourcing agreements, shifting liability to third parties.
<b>Peer Interactions:</b>	Explored risk prioritization techniques with peers, discussing the importance of structured frameworks. Shared insights on risk mitigation strategies in Agile vs. Waterfall projects, noting that iterative models allow early risk identification. Discussed real-world examples of risk management failures and lessons learned.
<b>Challenges Faced:</b>	Risk Prioritization: Overcome by using structured frameworks and scoring models. Balancing Risk Control Costs: Addressed through cost-benefit analysis and trade-off studies. Managing Unknown Risks: Explored methods like risk buffers and continuous monitoring to mitigate unforeseen risks. Resource Availability Risks: Mitigated through knowledge management systems and cross-training team members. Ensuring Quality Despite Tight Deadlines: Implemented quality gates and automated testing to maintain standards.
<b>Personal development activities:</b>	Researched case studies on successful risk management strategies. Practiced using risk assessment frameworks in mock projects. Attended a webinar on risk mitigation in software development projects. Read industry reports on emerging risks in software project management. Analyzed risk prioritization strategies in high-profile software failures. Studied best practices for balancing risk management costs against project constraints.
<b>Goals for the Next Week:</b>	Explore advanced risk assessment models and SDLC-specific risk strategies. Develop a risk assessment matrix for a hypothetical project to reinforce learning. Research real-world software failures and analyse risk management shortcomings.