**Chapters:** 1 to 6

**Student Name:** Thansil Mohamed S

**Course:** Software Project Management

**Journal URL:** https://github.com/ThansilMohamedS/SPM

**Dates Rage of activities:** 13/02/2025 to 20/02/2025

**Date of the journal:** 23/02/2025

|  |
| --- |
| **Key Concepts** |
| This week, I revised every fundamental aspects of Software Project Management for the midterm, focusing on project initiation, estimation techniques, risk management, configuration management, and project planning. The project initiation phase provided insight into how projects begin with a Project Charter, which defines objectives, scope, budget, and schedule. Understanding how effort and cost estimation models such as COCOMO, Function Point Analysis, and Delphi Estimation help predict project timelines and allocate resources effectively was particularly valuable. I also examined algorithmic cost modeling, which utilizes historical data and project attributes to estimate expenses and timelines. These estimation techniques are crucial in ensuring that projects remain within budget and on schedule, minimizing financial risks.  Additionally, I delved into Risk Management and the role of Configuration Management in software development. Risk management covers identifying potential risks, assessing their impact, and planning mitigation strategies such as avoidance, transference, reduction, and acceptance to minimize disruptions. Configuration management, on the other hand, ensures version control, change tracking, and system stability through structured change control processes. Lastly, project planning techniques, including Work Breakdown Structure (WBS), Critical Path Method (CPM), and Gantt Charts, provided a systematic approach to organizing tasks, assigning resources, and maintaining a clear project timeline. Effective planning and scheduling strategies ensure smoother project execution by preventing bottlenecks and optimizing team productivity. |

|  |
| --- |
| **Application in real life** |
| In software projects, effort estimation, risk management, configuration control, and project planning ensure smooth development. Cost estimation optimizes resource allocation, while risk management mitigates delays. Configuration management tracks system versions, preventing update conflicts, and project scheduling techniques like WBS and CPM improve task allocation.  For our Food Expiration Alert System, COCOMO estimation helps determine development time for alerts, databases, and API integrations. Risk management ensures data accuracy to prevent false alerts, while configuration management maintains system stability. Gantt charts and WBS structure tasks like UI design, backend logic, and notification integration, ensuring timely delivery and a scalable, user-friendly system that minimizes food waste. |

|  |
| --- |
| **Peer Interaction** |
| Throughout the week, we engaged in multiple discussions in one of my friend’s place. These interactions were invaluable in refining our project’s approach, particularly in defining effort estimation, risk management, and configuration control. We debated the applicability of COCOMO vs. Function Point Analysis for estimating development effort and discussed potential risks such as notification failures, incorrect expiration alerts, and system integration challenges. By working together, we ensured that our risk assessment strategy was well-structured, addressing potential failures before they could impact system reliability. Additionally, we reviewed our configuration management plan, ensuring a streamlined version control process to track updates and maintain software stability. These collaborative efforts helped solidify our project’s direction and enhanced our ability to manage both technical and operational challenges effectively. |

|  |
| --- |
| Challenges Faced |
| Estimating effort for the Food Expiration Alert System was challenging due to uncertainties in user behavior, notification frequency, and database performance. Unlike predefined datasets, our system relied on real-time tracking and manual inputs, making development effort harder to predict. Ensuring alert accuracy also depended on users entering correct product details.  To address this, we built a minimal viable version to test performance and identify bottlenecks. Automated input suggestions were added to reduce user errors, improving reliability. We also adopted an iterative risk assessment approach, allowing us to refine estimations dynamically and enhance the system’s scalability and accuracy over time. |

|  |
| --- |
| **Personal development activities** |
| To enhance my understanding of effort estimation and risk management, I explored case studies on COCOMO and Function Point Analysis, analyzing how different projects optimized resource allocation. I also researched best practices for real-time notification systems and database optimization, ensuring that expiration alerts are processed efficiently. Additionally, I improved my collaboration and planning skills by actively engaging in team discussions, refining our risk assessment strategies, and contributing to structured decision-making processes. |

|  |
| --- |
| **Goals for next week** |
| Next week, I will focus on our project by defining its scope, applying effort estimation techniques, identifying potential risks, collaborating with peers for task division, and implementing initial components with proper version control. |