**Learning Journal -2**

**Student Name:** Abhigyan Singh

**Course:** SPM SOEN 6841

**Journal URL:** <https://github.com/Abhigyan-singh2001/COMP6841_Learning_Journals>

**Dates Rage of activities:** 29/01/2025 – 9/02/2025

**Chapters covered: 3-6**

**Date of the journal:** 09/02/2025

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Key Concepts Learned:** | **Application in Real Projects:** | **Peer Interactions:** | **Challenges Faced:** | **Personal development activities:** | **Goals for the Next Week:** |
| * Chapter 3 is mainly focused on different effort and cost estimation techniques. * There a re 2 main techniques, **Experience-Based Estimation** (including techniques like **Estimation by analogy**: Compares a new project with similar past projects and adjusts effort estimates accordingly, **Function Point Analysis (FPA**): Measures software size based on user-visible functionality and **Team-Based Delphi Method**) * And 2nd technique is **Algorithmic cost modelling** which uses mathematical formulas to estimate project effort and cost. For Example: **COCOMO (Constructive Cost Model)**, which predicts cost based on lines of code, effort multipliers, and scale factors. * **Cost estimation techniques** involve factors like hourly wages, overhead costs and software/hardware costs. * Scheduling depends on effort estimates and software development lifecycle (SDLC) models like **Waterfall Model** and **Iterative Model**. | * **In Agile Projects we can use** COCOMO II’s Early Design model for sprint planning. * **We can** Combine **Delphi method with risk buffers** to avoid budget overruns. * **In AI/ML Projects we can** Adjust FPA to account for data preprocessing and model training tasks. | Discussed about Delphi estimation techniques and shared thoughts on algorithmic cost modelling technique. | Faced difficulty in understanding the COCOMO II and where it can be applied since there is limited resource. | Researched and read articles about FPA and other estimation techniques. | Goals for the next week is to revise the previous challenging tasks and go through chapters 7 and 8.  Deepen knowledge about project planning and to know about other techniques like SCRUM. |
| Chapter 4 was all about the importance of Risk management in software projects.   * Risks can arise from **resource unavailability, technology obsolescence, wrong tool selection, and service failures**. Risks are categorized into **technical, legal, organizational, economic, schedule, and cost risks**. * Risk management process involves 3 main steps: **risk dentification**, **risk analysis** (Qualitative and Quantitative analysis) and **Risk prioritization** which is done by calculating risk exposure. * % main risk control strategies: Acceptance, Avoidance, Transference Mitigation and Contingency planning. * **Risk Management in Software Development Models. Waterfall Model Risks**: High risk since the final product is only available at the end, making requirement misunderstandings costly. **Iterative Model Benefits**: Lower risk as **small feature sets** are built and reviewed incrementally, allowing early corrections. | **Example**: A startup developing an **AI-based chatbot** relies on an **outdated NLP model** that becomes incompatible with new frameworks.  **Risk Strategy**: **Avoidance** by **regularly updating technology stacks** and **prototyping with new AI models** before full-scale implementation. | Shared thoughts on what potential risks can occur and what strategies can be used to assess those risks for example hacking risks and their impacts. | Understanding how qualitative and quantitative analysis can be balanced simultaneously for risk assessment. |  |  |
| Chapter 5 included Configuration management (CM) concepts which is a systematic process to control and document changes to software systems.   * 4 key functions of CM: Configuration control, configuration status accounting and configuration audits. * Key steps to change control process: Initiation, impact analysis, approval, Implementation and Verification. * Benefits of Configuration management includes Product integrity, traceability, Compliance and Efficiency. | In software development, CM is implemented using tools like Git and GitHub. Developers use Git to track code changes, create branches for new features, and merge updates without conflicts.  Tesla uses CM to manage software updates for its electric vehicles. When deploying a new autopilot feature, engineers **version-control** the code, test it in staged environments, and roll it out incrementally. | Explored and discussed the importance of github in version controlling and versioning. | How **remote teams** handle check-in/check-out protocols, leading to version conflicts or undocumented changes. | Applied CI/CD concept on previous projects using Github. |  |
| Chapter 6 was all about project planning concept.   * Scheduling:   + **Top-Down Planning**   + **Bottom-Up Planning**. * **Work Breakdown Structure (WBS)** * **Resource Allocation** * **Sheduling techniques like Critical path method, Gantt charts and Goldratt’s chain method.** * Also Includes methodologies like Milestones which are key checkpoints and Deliverables. * Contingency planning, Supplier Management, Communication plans and Quality Assurance plans. | Scheduling with Gantt Charts and Critical Path Method  To visualize timelines and dependencies. | How Gantt charts and discussing about the milestones are helpful for the success of a project deliverable. | Understanding the concepts like top down planning and bottom-up planning and how these scheduling techniques help in real world scenarios. |  |  |