Learning Journal

Student Name: Sanjay Dharmendra Upadhyaya
Course: SOEN 6841
Journal URL: https://github.com/sanjay123-321/Learning-Journal/blob/main/40306152_Sanjay%20Upadhyaya_LJ3.pdf
Dates Rage of activities: 02-08-2025 to 03-16-2025
Date of the journal: 03-16-2025

Key Concepts Learned	Application in Real Projects	Peer Interactions	Challenges Faced	Personal Development Activities	Goals for the Next Week
Software Engineering: Understanding the SDLC phases — requirements gathering, design, implementation, testing, deployment, and maintenance. Each phase plays a vital role in ensuring project structure and quality.	Applied SDLC phases to structure a project. Used Agile for iterative delivery, ensuring continuous improvement at each stage.	Discussed with peers the pros and cons of Agile vs. Waterfall in handling changing requirements and project complexity.	Choosing the right SDLC model for different projects and ensuring team alignment was difficult.	Explored case studies of projects using Agile and Waterfall, gaining insights into model selection.	Develop a project roadmap that clearly defines each phase, with milestones and checkpoints.
Development Metrics and Work Products: Learned key metrics like velocity, defect density, code churn, and customer satisfaction to measure project health. Work products include requirement docs, design specs, code, test	Used Jira to track sprint velocity and defect rates, improving visibility into project progress. Created documentation for each stage.	Exchanged ideas on useful metrics for tracking project health and aligning on the best tools to use.	Identifying the right metrics without overwhelming the team was a challenge. Focused on simplifying tracking methods.	Attended webinars on using project management tools for effective metric tracking.	Refine metric-tracking strategies and create automated dashboards to visualize project health.

plans, and reports.					
Customer Requirements: Techniques for requirement elicitation like interviews, surveys, focus groups, and prototyping. Importance of documenting requirements for traceability.	Conducted mock interviews to gather user needs, documented requirements, and validated them with stakeholders.	Shared experiences with peers on techniques for handling ambiguous requirements and refining documentation.	Handling vague requirements and translating them into clear, actionable items was challenging.	Explored tools like Confluence for requirement documentation and collaboration.	Develop a formal requirement gathering template and practice refining requirements in mock projects.
Configuration Management and Quality Assurance: The role of CMS in tracking requirement changes, ensuring consistency, and avoiding version conflicts. QA ensures each stage maintains product quality through inspections and tests.	Integrated Git for version control to manage project artifacts and track changes. Established a branching strategy to avoid conflicts.	Discussed with peers the best practices for managing multiple contributors in a CMS.	Managing merge conflicts and ensuring proper documentation of changes was difficult.	Practiced setting up automated pipelines to enforce quality checks on every code push.	Implement a formal version control policy and set up automated quality checks in the CI/CD pipeline.