Final Learning Journal

Student Name: Vatsal Mukeshkumar Ajmeri
Course: SOEN 6841
Journal URL: https://github.com/vatsal-30/SOEN-6841-learning-journal
Date of the Journal: 30th March 2025

Overall Reflection

1. Overall Course Impact

Studying the provided course content changed my perspective on software development's scope: it's not merely about coding and testing, but involves a systematic progression of defining objectives, analysing feasibility, planning resources, managing risks, and conducting thorough monitoring. The material highlighted how essential processes like configuration management and earned value management are for keeping a project within schedule and budget.

Challenging Component – Reflecting on Prior Knowledge
Before this course, I thought agile sprints alone would suffice to keep projects flexible. However, I realized that structured frameworks—including feasibility studies, formal risk registers, and meticulous scheduling—greatly enhance an agile approach. For instance, in our Automated News Summarizer, my previous mindset would have been "just iterate till we get the model right." But once we delved into the Feasibility & Budget report for GPU costs, data acquisition, and compliance considerations, I saw how vital concrete project controls are. These insights transformed my thinking from a strictly iterative approach to one that blends agile freedom with the discipline of early cost and risk analysis.

2. Application in Professional Life

My professional approach to project management has become more holistic. After examining a variety of estimation techniques—such as Function Point Analysis and COCOMO—I recognized that, although they are classical methods, they can be adapted to machine learning projects. We often had to factor in model complexity, data pre-processing overhead, and cloud resource usage—dimensions not typically captured by simple lines-of-code estimates.

Challenging Component – Potential Long-Term Opportunities
Mastering this fusion of AI-driven project aspects and standard management tools provides a distinct advantage in the job market. I see future opportunities leading teams that balance advanced data science methods with stable project controls. By applying risk assessments and iterative prototypes to large datasets—much like in our project's summarization pipeline—companies can create robust solutions while containing costs. Ultimately, I can envision career paths such as AI Product Management or Technical

Project Lead, roles that demand both domain expertise in machine learning and a systematic approach to scope, quality, and timeline.

Peer Collaboration and Personal Growth

3. Peer Collaboration Insights

A prominent highlight was the active peer collaboration every step of the way. Whether it was brainstorming feature prioritizations or discussing branching strategies in configuration management, interacting with teammates enriched the learning experience. One significant peer insight emerged when a teammate pointed out "model drift" as a risk: we realized the summarizer might degrade if we didn't periodically retrain it on fresh data. Integrating this idea into our plan improved our project's long-term sustainability. Without peer input, I might have overlooked this critical factor.

Another vital lesson came from studying iterative models: we recognized the need for incremental quality checks. A peer suggested automating regression tests for repeated summarizer evaluations. In response, we combined test automation tools with real-time data pipelines, drastically reducing manual testing overhead. In short, peer collaboration fostered an environment where everyone contributed specialized knowledge—from risk logs to AI model tweaks—and we built a more resilient final product as a result.

4. Personal Growth

My personal growth as a developer and budding project manager was significant. Early on, I struggled with maintaining clear, traceable requirements. However, by observing peers who diligently linked requirements to specific commits, I eventually refined our approach: each user story was tracked in a shared repository, with consistent references in code commits. This immediate traceability reduced confusion and boosted accountability.

I also became more comfortable handling feasibility and budget tasks—especially estimating GPU hours or factoring in potential licensing fees for advanced AI libraries. Repeated exercises in cost and schedule forecasting led to a new confidence in bridging technical and financial analyses.

A Notable Example

- ✓ I led a risk workshop focusing on "vendor lock-in," identifying how reliant we had become on a specific cloud platform's GPU service. We then decided to keep containerized builds that could port to other infrastructures if needed. This combined strategic foresight with the day-to-day tasks of coding and testing—an evolution in my critical thinking and proactive risk management.