Student Name: Darshan Kansara
Course: Software Project Management
Journal URL: https://github.com/KansaraDarshan/SOEN6841
Date Range of Activities: 29/10/2024 to 06/11/2024
Date of Journal: 06/11/2024

Key Concepts Learned

This week's focus was on **Project Closure** (Chapter 8) and **Software Lifecycle Management** (Chapter 9), each addressing critical end-to-end aspects of software project management and Project II deliverables.

Chapter 8: Project Closure

- **Project Deliverables**: Ensuring all project deliverables are complete, organized, and ready for hand-off.
- **Source Code Version Control**: Importance of archiving source code and ensuring proper version control for future reference.
- **Data Filtration**: Filtering key metrics for documentation and identifying data to be retained.
- **Lessons Learned**: Systmatic recording of lessons learned to provide insights and reduce repetitive mistakes in future projects.

Chapter 9: Software Lifecycle Management

- **Software Lifecycle Models**: Overview of models, including:
  - **Waterfall Model**: Sequential, ideal for projects with clear, unchanging requirements (e.g., ERP systems).
  - **Iterative Models**: Flexible approaches like Scrum and Extrem Programming that allow rework and adaptation, useful for fast-evolving tech environments.
- **Quality Assurance Gates**: Stratgic checkpoints in the development lifecycle to ensure product quality and adherence to standards.
- **Concurrent Engineering**: Simultaneous development across different stages to accelerate the process while maintaining quality.
- **Work Products in Lifecycle Stages**: Identification and management of documentation and assets produced at each stage of development.

Application in Real Projects

- **Effective Project Closure**:
  - Archiving data and source control history allows new team members to understand project history, reducing onboarding time.
  - Systematic documentation of lessons learned can prevent repeat mistakes and serve as a training resource for similar projects.
- **Choosing Appropriate Lifecycle Models**:
  - **Waterfall Model**: Suitable for large, stable projects like banking software or ERP systems, where changes are minimal and rigorous structure is needed.

    - **Iterative Models**: Ideal for customer-centric software with evolving requirements, such as social media platforms or apps that undergo frequent updates. This allows for quickr, incremental releases and adjustments.
    - **Hybrid Models**: For complex projects with mixed requirements, combining elements from Waterfall and iterative models can help balance flexibility and control, although they may introduce management challenges.
- **Quality Assurance Gates in Practice**:
    - This approach is especially beneficial in regulated industries, like healthcare, where compliance and accuracy are crucial.

Peer Interactions/Collaboration

Peer discussions this week focused on the deliverables:

- **Phase II- project deliverables:** Did Peer collaboration and worked on the project deliverable which includes feasibility study, solution proposal, project plan, risk assessment and budgeting documents.

Challenges Faced

- **Selecting Appropriate Documentation for Archiving**: Determining which project data to archive without creating information overload was challenging. I explored ways to balance thorough documentation with accessibility.
- **Choosing the Right Lifecycle Model**: Understanding when to use Waterfall versus iterative models, especially for hybrid projects with partial predictability, proved challenging. Hybrid models can offer benefits, but they also require careful management to prevent complexity from escalating.

Personal Development Activities

- **Reviewed Case Studies**: I explored projects with strong closure practices, focusing on how they managed version control and documentation.
- **Online SCRUM Tutorial**: I deepened my knowledge of iterative project management, particularly **sprint retrospectives** and how they function as built-in quality gates.
- **Version Control Practice on GitHub**: Practiced version control strategies to enhance my project archiving skills, ensuring smoother future transitions.

Goals for the Next Week

Next week, I will focuus on :

- **Refine Documentation Skills**: I will focus on developing templates for project closure that balance thoroughness and ease of use.
- **Understand Quality Gates Further**: I'll explore quality assurance best practices and their application at each stage of the software lifecycle.
- **Examine Hybrid Models**: I'll undersand additional case studies on hybrid models to identify best practices for managing complexity and maintaining flexibility.