

SOEN 6841 – SOFTWARE PROJECT MANAGEMENT

NAME: Devanshu Kotadiya [40268999]

JOURNAL URL: <https://github.com/devanshu-kotadiya/Software-Project-Management/tree/main/Learning%20Journals>

DATES RANGE OF ACTIVITIES: 22/10/2024 – 8/11/2024

JOURNAL DATE: 09/11/2024

KEY CONCEPTS:

Software Project Deliverables: Software project, User manuals ,User training, project installation, Resource release, Lessons

Source code version management on project

Strategy for project data archiving: Data cleaning, Application area, Industry, Project size, Life-cycle methodology, Programming language

Lessons learned: Better alternative to do things, Better ways to manage projects, Solutions for unique issues, Better negotiation with customer, Better ways to deal with risks, Which techniques work and which do not.

Factors which helped in improving productivity on software projects:

1. Code reuse
2. Maturing software engineering
3. Productivity tools
4. Automatic code generation

Software Life-Cycle Management Processes:

1. Software Requirements
2. Software Design
3. Software Build
4. Software Testing
5. Software Release
6. Software Maintenance

Software development life cycles:

1. Waterfall model
2. Iterative model
3. Incremental model
4. Extreme model
5. Concurrent engineering model

Work products from various software life-cycle phases:

1. **Requirement phase:** Requirement specification document
2. **Design phase:** Product model
3. **Construction phase:** Untested product
4. **UAT testing phase:** UAT tested product
5. **Release phase:** User accepted product+user manuals

Quality assurance mechanism for software projects

Exit criteria after each phase:

Processes completed?

Work product delivery?

Formal review for quality?

Software requirement types: Functional requirements v/s Nonfunctional requirements

Nonfunctional requirements:

1. Security
2. Performance
3. Usability
4. Compatibility

Sources of requirements: Increased revenue potential, Users, Sources of requirements, Standards, Business environment, Maintenance, Customer feedback, Cost saving potential, Business policies changes, Technology changes.

Software requirement development life cycle:

1. Requirement elicitation
2. Requirement documentation
3. Requirement analysis
4. Requirement specification development
5. Requirement verification and validation
6. Verified and validated requirements
7. System modeling

Software requirements change management life cycle:

1. Requirement specification
2. Change request
3. Communication of change
4. Impact analysis
5. Changes in design construction
6. Rework
7. System validation

CASE STUDY REFLECTIONS:

At a SaaS vendor, project and iteration closure are carefully managed to ensure continuity in product development. Once resources finish work on a project, they are quickly reassigned to future projects by the global program manager. The configuration manager then saves all project documents and source code in a dedicated branch on the configuration management system. This branch, integrated with previous software versions, is set as read-only upon project completion, preserving it as the finalized version and enabling new branches for future development.

Knowledge from each project is retained for future learning. In release 6.0, a key lesson emerged: despite detailed planning, unforeseen challenges arose, particularly with the complex appointment scheduling functionality. After extensive problem-solving sessions, the feature was successfully completed, but the original project timeline was strained. Even with a 10% schedule buffer,

additional compromises were necessary—another planned feature was postponed to the next release, and its resources were reallocated to support the appointment scheduling effort.

PEER INTERACTIONS:

Peer interactions were essential in developing our digital skills platform for low-income communities. During the feasibility study, we refined the scope based on team insights. In the solution proposal, each member's input shaped ideas for user engagement and accessibility. Collaborative discussions on task prioritization helped us create a balanced project plan, and risk assessment insights led to comprehensive mitigation strategies. For budgeting, we combined perspectives to ensure financial feasibility. Overall, these interactions strengthened our project and prepared us for effective implementation.

CHALLENGES FACED AND ADDRESSING THOSE CHALLENGES:

A key challenge we faced was navigating the intricate calculations required by the COCOMO model for budgeting. The complexity led to confusion and inconsistencies in our estimates. To address this, we worked together to break down the calculation steps and verified each phase to ensure accuracy. This collaborative approach helped us master the model, align on budget estimates, and improve our confidence in the project's financial feasibility.

PERSONAL DEVELOPMENT ACTIVITIES:

I'm using an ebook on software management to deepen my understanding of project closure, software lifecycle management, and software requirements management—key skills that align with industry practices. For example, insights into project closure are helping me understand finalization steps used in real-world SaaS projects, while lifecycle management techniques are widely applied in agile and iterative development. Practical knowledge in requirements management enhances my ability to ensure alignment with stakeholder expectations, which is critical in software delivery. I plan to review all chapters for exams, using the ebook and other resources to resolve any questions.

GOALS FOR NEXT WEEK:

- Engage in discussions to prepare for demonstrations and presentations of the project report, ensuring clarity and effectiveness in delivery.
- Conduct a thorough review of all relevant chapters, focusing on reinforcing concepts and addressing any gaps identified during the midterm exam.

Long term goal:

My long-term goal is to develop the skills needed for a software development management role in a multinational company. I aim to align my prior experience in software project management with industry expectations, focusing on advanced project lifecycle management, risk assessment, and cross-functional team leadership. To strengthen my culture fit for future roles, I also plan to deepen my understanding of global team dynamics and effective communication strategies, preparing for culture-fit interviews and the demands of software management in diverse, high-performing teams.