Learning Journal

**Student Name:** Priscila Aramouni

**Course:** SOEN 6841 - Software Project Management

**eBook Referenced**: Software Project Management: A Process-Driven Approach

(by Ashfaque Ahmed, CRC Press, 2012)

**Journal URL:** [Link](https://github.com/priscila-aramouni/soen-6841-software-project-management/blob/3dc6e84b6f37cc7e0d00098ee2f70c08506c1f32/learning-journal/learning-journal.docx) to file on GitHub

(Note: To navigate the journal more easily, click View > Navigation Pane.)

# Week 1: Jan 18 – Jan 27

# Chapter(s): 1, 2, 3, 4

## Key Concepts Learned

### **Chapter 1** – Introduction to Software Project Management

#### What is a project?

A project is a set of activities that consumes resources, budget, and time to produce stated goals.

#### What is a software project?

A software project is a specific type of project focused on developing or maintaining software products, applying both project management and software engineering principles to achieve goals efficiently.

#### What processes are involved in a software project?

* Evolving processes beyond a project: a continuous improvement process that runs above projects at the organizational level.
* Project management processes (project initiation, planning, control, monitoring, and closure): these are management processes.
* Software development life cycle (SDLC) processes (requirements, design, build, testing, maintenance, etc.): these are the development processes that build the application.

#### How are people, processes, tools, and technology integrated in a project?

People, processes, tools, and technology are integrated in a project by aligning organization-level processes, managing customer expectations, adapting project manager's style, collaborating with the project team, optimizing supplier management, and selecting technology based on project requirements.

#### What are the characteristics of a good project manager?

A good project manager should possess a blend of general project management skills and domain-specific expertise. They should also be creative and able to understand the project’s environment and how it should be planned and executed.

#### What are the subprocesses in project management processes?

Processes: initiation, planning, monitoring, controlling, and closing.

Subprocesses of initiation: Defining charter, scope, and objective.

#### What management metrics are measured in software projects?

In the case of software development projects, the management metrics are the productivity data for the projects. The software work product quality data are the technical metrics.

### **Chapter 2** – Project Initiation Management

#### How is a project initiated?

Project initiation happens with a kick off meeting involving the project manager and the stakeholders. They define the project charter, project scope, and project objectives. A preliminary effort and cost estimate is chalked out. A sketch for the project schedule is also made so that a tentative duration for the project can be established.

#### What is a project charter?

Project charter is made by the top management of the organization for starting a software project. It includes project objectives, scope, and responsibilities.

#### What is project scope?

A detailed project scope is developed to define boundaries of the project. The scope will include what functionalities are needed in the software product to be developed. It will also define level of quality needed in the software product.

#### What are project objectives?

The project should have a set of objectives that must be met. If clear project objectives are set at the project initiation, it would help the team understand the importance of the project and will help them do their best to achieve the goals.

#### What project activities are performed during project initiation?

* Estimate Initial Project Size
* Estimate Initial Project Effort and Costs
* Estimate Initial Project Schedule
* Create Initial Project Plan
* Feasibility Study
* Quality Planning

### **Chapter 3** – Software Project Effort and Cost Estimation

#### How is an effort estimate for a project made?

Effort estimates involve optimizing high-cost human resources' time and allow service providers to bill customers accurately for the actual effort put into projects. It is used to assemble the project team based on required skills and budget constraints.

What are the different effort estimation techniques.

A white box with black text

Description automatically generated

(Refer to book for formulas and details)

##### How is a cost estimate for a project made?

A cost estimate involves converting the effort estimate into man-months, applying a standard man-month rate, and accounting for additional costs (expenses for hardware, management costs, software tool acquisition costs, training costs…)

Salaries of software professionals are a major cost driver. Salary variations and differences in productivity levels among professionals complicate cost calculations.

##### What are the different cost estimation techniques?

* Cost Factor Analysis
* Activity-Based Cost Estimation

##### How is a schedule estimate for a project made?

The schedule duration is determined as the difference between the project start date and end date.

Effort and schedule are not equal in terms of time, especially in scenarios with parallel processes, which is why effort must be calculated before the schedule.

##### How is a resource estimate for a project made?

A resource estimate is made by first obtaining a list of tasks, identifying required skills and experience for each task, and matching them with available resources.

The amount of work for each task is calculated based on the organization's productivity factor and time duration. The number of resources required is determined by dividing the volume of work by the productivity and time duration for task completion.

### **Chapter 4** – Risk Management

#### What is a risk on a project?

A risk on a project refers to unforeseen or unplanned events that have the potential to badly impact components such as budget, time, resources, quality, and technology.

Risks can be categorized as external or internal, where internal risks arise from aspects dealt with by the project team, while external risks are influenced by environmental factors beyond the project's control.

#### What kinds of risks exist for a project?

* Quality Constraints
* Resource Unavailability
* Disinterest
* Attrition
* Scope Creep
* Cost Constraints
* Bad Negotiation
* Unrealistic Estimate
* Human Error
* Poor Management

The above can be divided into the following categories:

* Budget
* Time/Schedule
* Resource
* Quality
* Technology

#### What kind of impact may risk have on a project?

Risks are dynamic and can occur at any stage, such as schedule delays, budget overruns, rework requirements, or disruptions due to unexpected events like team members falling sick.

The impact of a risk is assessed considering both its impact and likelihood of occurrence. Risks with high probability and high impact are prioritized.

#### What strategy is needed to deal with risks?

Project managers need to formulate mitigation strategies for manageable risks. However, for unmanageable risks, external assistance is relied on, and strategies focus on addressing the consequences.

## Application in Real Projects

In my real-world experience as a full-stack developer within a SCRUM team, the project management concepts learned were regularly applied in our agile workflow. Despite not being a project manager myself, the team collaborated closely to integrate these concepts. We held the following meetings during our iterations:

- Program Increment Planning: Deadlines and objectives are reviewed, and effort is estimated by voting on tickets and checking the team’s capacity to handle them.

- Daily Stand-up: Team members share any roadblocks encountered, what they did, and what they will be doing. This ensured daily synchronization and progress tracking.

- Backlog Refinement: Backlog is reviewed to make sure it contains the appropriate prioritized items, making sure the team is still on track for delivery.

- Sprint Review and Retrospective: “Thorns (negative outcomes) and roses (positive outcomes)” are discussed, risks are identified, and the appropriate action items are created for mitigation.

Integrating these concepts helps organize the workflow and somewhat document the process. Though it is beneficial, it requires time away from development and can sometimes be tedious to maintain.

## Peer Interactions

Teams have been assigned this week. We’ve created a WhatsApp group to get to know each other and familiarize ourselves with the topic of “Community Volunteer Coordination Platform”.

## Challenges Faced

This week’s learning journal and assignment took me about 10 hours in total.

Next week I will attempt to timebox to finish things more quickly.

## Personal Development Activities

A dedicated GitHub repository was established to serve as a central hub for the material related to this course. This repository will serve as a valuable resource, facilitating access to course material and collaborative efforts.

## Goals for the Next Week

* Have a solid understanding of the assigned project topic and understand how the material being learned applies to it.
* Finish the journal and assignments more quickly.
* Ask the professor if there is a more efficient way to be filling out the journal.

# Week 2: Jan 28 – Feb 3

# Chapter(s): 5

## Key Concepts Learned

### **Chapter 5** – Configuration Management

#### What is a configuration management system?

A configuration management system is a tool used to store, organize, and manage artifacts produced during the development lifecycle.

It facilitates version control and supports continuous integration by maintaining a central repository for software builds and revisions.

#### What are the parts of a configuration management system?

The parts of a configuration management system include:

* Software build files
* Work products
* Documents
* Version control mechanisms
* Access control features
* Audit facilities

#### Why is a configuration management system required on a software project?

A configuration management system addresses the challenges of evolving requirements, multiple versions of work products, and the need for secure access and version control. It ensures that team members work on the right versions of documents and artifacts.

#### What strategies can be made to deploy a configuration management system successfully for a

#### project?

Strategies include implementing a centralized system with role-based access control, ensuring continuous integration, providing easy branching mechanisms, and incorporating an audit capability for document verification and version tracking.Top of Form

## Reflections on Case Study/Course Work

Dispersed teams are a very common occurrence in real projects, and Chapter 5’s case study provides insights into the practical application of configuration management principles in real-world software projects.

It demonstrates the importance of a centralized configuration management system accessible to internal, external, and offshore teams across diverse locations and time zones.

Overall, it highlights the necessity of automated smoke testing tools and escalation procedures to address build failures and maintain stability in the development environment.

## Collaborative Learning

We had our first virtual team meeting this week to research and discuss the project description (Community Volunteer Coordination Platform). We shared ideas and discussed how to proceed with the project initiation and market analysis.

## Further Research/Readings

I read several online articles to help me understand the difference between Agile and Traditional methodologies. This helped me better understand why Agile projects require less effort compared to traditional ones.

## Adjustments to Goals

* Make it a habit to go over previous learning journal entries and update them with important notes from the week’s lecture.

# Week 3: Feb 4 – Feb 10

# Chapter(s): 6

## Key Concepts Learned

### **Chapter 6** – Project Planning

#### What is software project plan?

A software project plan is a comprehensive document that outlines the strategy, approach, and activities required to successfully execute a software development project. It serves as a roadmap for project stakeholders, detailing the goals, scope, schedule, resources, risks, and quality standards to be achieved throughout the project lifecycle.

#### What are the parts of a software project plan?

* Task planning: Defining the specific activities and tasks.
* Resource planning: Allocating human and material resources effectively.
* Supplier management planning: Managing relationships with external vendors or suppliers.
* Configuration management planning: Managing changes to project artifacts and ensuring version control.
* Communication planning: Establishing channels and protocols for effective communication among project stakeholders.
* Defect Prevention Strategy (Quality Assurance): Defining quality standards and processes to ensure the delivery of a high-quality product.
* Duration estimation: Predicting the time required for completing tasks, relying on historical data and expert judgment.
* Cost estimation: Determining the financial resources needed.
* Tool planning: Selecting and deploying appropriate tools and technologies.
* Scope planning: Clearly defining the project scope and managing scope changes.
* Effort estimation: Estimating the amount of effort required for each task or activity.
* Risk planning: Identifying and mitigating potential risks.

#### What are the types of software project plans?

* Top-down planning: Typically used in product development scenarios, where project timelines and feature releases are predetermined.
* Bottom-up planning: Commonly employed in custom software development, where project timelines and feature releases are determined based on project requirements and estimation.

#### What inputs go in making a software project plan?

* Project scope and objectives
* Service level agreements (SLAs)
* Project requirements
* Project start date, end date, and duration
* Project budget
* Software engineering considerations, such as technology stack and development methodologies (e.g., waterfall, agile)

#### What techniques are used in making a software project plan?

* Critical Path Method (CPM) or Program Evaluation Review Technique (PERT) identifies the critical path which is the longest sequence of dependent tasks in a project. This helps determine the project duration.
* Goldratt's Critical Chain introduces buffers to manage uncertainty in tasks. Tasks are categorized as fixed or variable, and buffers are allocated to variable tasks to accommodate potential delays or uncertainties.
* Velocity-based iteration planning in Agile methodologies, where team capacity and velocity are used to plan iterative development cycles.

## Reflections on Case Study/Course Work

The case study aligns closely with the core principles of project planning outlined in the chapter. It emphasizes the importance of detailed planning in managing projects effectively. The iterative planning approach adopted by the SaaS vendor reflects the agile methodologies discussed in the chapter, allowing for flexible adjustments based on evolving requirements and market conditions.

Resource allocation and time-boxing are key aspects of the vendor's planning strategy. By assigning resources based on feature priorities and adhering to fixed iteration timelines, the vendor balances adaptability with resource optimization for efficient project execution.

The involvement of key stakeholders, particularly the CTO, in decision-making processes highlights the importance of clear communication in successful projects.

The case study also demonstrates the practical application of planning techniques like feature prioritization, effort estimation, and risk management. By leveraging tools such as Gantt charts, PERT/CPM charts, and earned value management, the vendor enhances project visibility and control.

In summary, the case study provides a tangible example of how project planning principles are applied in real-world scenarios. It stresses the critical role of structured planning processes in navigating project complexities and achieving objectives.

## Collaborative Learning

In our collaborative learning experiences throughout the week, our team came together to brainstorm and develop ideas for our Community Volunteer Coordination Platform. Each member actively participated in identifying the problem statement and defining the scope of the software solution. Through open discussions and sharing of perspectives, we gained a deeper understanding of the challenges faced in coordinating volunteer activities within communities.

As we delved into market analysis, our collaborative efforts enabled us to assess the target audience, potential users, and existing competitors. We identified unique selling points and crafted a value proposition for our platform. Our solution was tailored to meet the specific needs of both volunteers and project organizers by analyzing market trends.

During our collaborative process, we faced the challenge of reaching unanimous agreement among team members. Despite differing opinions and perspectives, our team’s commitment to consensus-building promoted an environment of open communication and mutual respect. This enabled us to sustain momentum towards achieving our goals.

Our collaboration strengthened our project outcomes. By recognizing and valuing each other's contributions, we cultivated an inclusive team culture that encouraged creativity and uniqueness, which are both attributes the professor requested to see in our work.

## Further Research/Readings

Exercise 4.1 of Chapter 4 provided insights into the distinctions between traditional and Agile risk management methodologies. It highlighted how Agile's iterative nature and adaptive planning influence risk handling and control in contrast to the more rigid approach of traditional project management.

Article read: <https://www.projectengineer.net/project-charter-vs-scope-statement/>

## Adjustments to Goals

Based on the feedback provided by the TA regarding the need to detail my answers more in my learning journal entries, here are my adjusted goals:

* I will focus on providing detailed explanations in my learning journal entries instead of simply answering questions.
* I will supplement my explanations with relevant examples, anecdotes, or real-life experiences whenever possible.
* I will consider how the concepts discussed in the course material apply to real-world situations or my personal experiences.