Name Team 9

Members Jairo Galarza, Alex Lundin, Brenden Pace, Daniel Park

Class SE 4381.501 Project Management

Assignment T06 Report

Overview of ISO/IEC/IEEE 12207:

Emphasizing estimating, planning, and tracking progress

Table of Contents

[**Introduction**](#_gjdgxs) **2**

[**Summary of References**](#_30j0zll) **2**

[IEEE 12207 Information technology—Software life cycle processes](#_1fob9te) 2

[IEEE 830 Software Requirements Specifications](#_3znysh7) 4

[IEEE 1058-1998 Standard for Software Project Management Plans](#_tyjcwt) 5

[IEEE 828-1998 Standard for Software Configuration Management Plans](#_3dy6vkm) 6

[IEEE/EIA 12207.1](#_1t3h5sf) 8

[SWEBOK Guide V3.0 Chapter 2](#_4d34og8) 8

[SWEBOK Guide V3.0 Chapter 3](#_2s8eyo1) 9

[**Discussion of Estimating**](#_5y7jjsnc59t9) **10**

[**Discussion of Planning**](#_7lx4hxlogdf) **11**

[**Discussion of Tracking Progress**](#_26in1rg) **12**

[**Conclusion**](#_lnxbz9) **13**

[**References**](#_35nkun2) **14**

# Introduction

The IEEE 12207 document encompasses a set of standards which define a common framework for developing and managing software. This collection of professional practices, first published in 1995, focuses on Software Lifecycle Processes. In the late 1990’s the IEEE revised 12207 three times, modifying it during each change set. In 2017, the IEEE revised and released the most recent version.

# Summary of References

## IEEE 12207 Information technology—Software life cycle processes

From a high level overview, IEEE 12207 splits software lifecycle processes into three main categories: primary lifecycles, supporting lifecycles and organizational life cycles. In addition, there is an Annex designed specifically to support companies who want to tailor 12207 to their current development process, while maintaining compliance to the major headings of 12207. IEEE 12207 outlines the communication and project management practices for compliance to this standard.

Primary lifecycles includes acquisition, supply, development, operation and maintenance. These primary sections mostly pertain to securing resources and using resources to achieve an output. Clause 5.2.4 outlines the planning stage of the supply lifecycle [1]. At this point in the process, the supplier shall review the requirements and select a software life cycle model, if one is not defined by the contract. From here, the planning gets more detailed around the life cycle selected.

Supporting life cycles contain 8 distinct parts, each of them designed to contribute to the overall success and quality of the project. Supporting life cycles cover: documentation process, configuration management process, quality assurance process, verification process, validation process, joint review process, audit process, problem resolution process. These life cycles mostly focus on writing documentation and evaluating software. Clause 6.1.1.1 describes the initial plan for any document generated around this entire software lifecycle [1]. Clause 6.4 contains information on how to evaluate Software progress and efficiency in terms of verification [1]. The supporting life cycles create the most documentation about the software itself.

Organizational life cycles pertain to management, infrastructure and training. These pieces build into the organizations longevity and they are not typically specific to any one project. The real value in the organizations life cycles comes from capturing lessons learned during the course of the project. The management process contains clause 7.1.2 which includes areas for effort estimations and schedules. During effort estimations, the manager prepares all plans to yield an estimation for the execution of the entire process start to finish [1].

IEEE 12207 provides all the necessary requirements that companies need to follow for compliance with this specific IEEE standard. 12207 references other works for in depth descriptions of certain document types used in this package. Instead of covering every granular step in the development process, the major points of 12207 outline what documents are required for companies to achieve compliance to this standard during their development process.

## IEEE 830 Software Requirements Specifications

IEEE 830 covers the Software Requirements Specifications, also known as the SRS. This practice gives recommendations for the specification of requirements, templates and guidelines for how to format this information, and then a compliance section for the IEEE 12207.1[2].

The main two objectives covered for the SRS are the considerations for creating a good SRS and the parts of an SRS. Things to be considered include characteristics, environment of, nature of, and the evolution of the SRS. In conjunction with our project, the important subjects might include knowing the characteristics of good requirements, as they will affect how well we can estimate and track the progress of the project.

The next section of the IEEE 830 gives a good idea of what organizing a good SRS may look like, including giving templates and the individual parts. This includes giving a good definition for the purpose and scope of the project before we start defining the requirements. We can also see how tracking the progress of a project will be easier once we have defined unambiguous requirements.

The last section of the IEEE 830 gives us a direct compliance and comparison to our subject for the project, the IEEE 12207. The IEEE 830 states that the SRS described can be almost directly is transcribed to the SRD of the IEEE 12207, with few additions [2]. Therefore, the SRS gives a good idea of what is contained in the SRD and can describe the processes used in requirements gathering.

## IEEE 1058-1998 Standard for Software Project Management Plans

This document covers the format and content of Software Project Management Plans (SPMP’s). There are two kinds of compliance to this standard: format compliance and content compliance. A plan with format compliance follows the headings in order. A plan with content compliance, meets all the criteria, in a custom order that seems more logical to the writer.

The following list contains all the headings in order: overview, references, definitions, project organization, managerial process plans, technical process plans, supporting, process plans, additional plans, annexes, index [3]. The most important section, for this assignment, is the managerial process plans, which gives insights into estimating, planning and tracking progress.

The overview contains a high-level summary of the project, 5 sub clauses compose it. This breakdown includes: project purpose, assumptions, deliverables, budget summary and evolution. The purpose defines the scope and project objectives [3]. Assumptions draw attention to non-obvious conditions for the project basis and highlight time constraints. The deliverables section describes the work to complete and holds a sub clause about the method for delivering work packages consistently. A budget summary provides a location giving a detailed list of the work packages to complete, such as a work breakdown structure. After the overview section, references and definitions follow. They list any outside sources and provide explanations of terms used throughout the plan.

Plans and organization make up the bulk of SPMP’s. The project organization defines external and internal entities and how they relate to the project. This section becomes important when large corporations subcontract work packages out to other firms. Interactions of the development team also stem from the project organization.

The managerial process plan contains quite a lot of information; clause 4.5.1.1 pertains to the estimation plan [3]. This section defines all tools and techniques for generating an accurate estimate of the project. It also describes the process of re-estimating. The work plan clause 4.5.2, gives a location for recording all tasks which can double as a list to track progress. Schedule allocation, clause 4.5.2.2, outlines time sequencing of tasks for further tracking metrics [3]. This area includes items such as Gantt charts and activity networks.

This document covers the format and content of Software Project Management Plans for the IEEE Standard. Most of the plan itself is more plans. For this assignment, the managerial process plan contains the most pertinent information around estimating, planning and tracking progress.

## IEEE 828-1998 Standard for Software Configuration Management Plans

This document standard covers the minimum requirements for processes for software configuration management (SCM) in systems and software engineering as well as the defined requirements of specific activities for any portion of a software product’s life cycle. This standard revision also defines only the contents of a software configuration management plan.

What is important about this document regarding the team assignment is that SCM deals with the task of tracking and controlling changes (a.k.a. configurations) in software. This is important in project management because crucial project changes can be tracked in terms of what team or team member changed which crucial portion of the project; knowing this information helps address concerns such as schedules, budgets, and software quality.

SCM is the means through which the integrity and traceability of the software system is recorded, communicated, and controlled during both development and maintenance [4]. This is all achieved by this standard providing the structure for identifying and controlling documentation, code interfaces, and databases to support all life cycle phases. In addition, support for a chosen development methodology that fits the requirements, standards, policies, organization, and management philosophy is addressed in the body as well. Finally, the status of baselines, change control, tests, releases, and audits in terms of producing management and product information are also addressed in this document.

Six sub clauses of this standard are addressed in the SCM planning information; they are: introduction (plan’s purpose, scope of application, and references), SCM Management (describes the “who” of the plan), SCM activities (describes the “what” of the plan), SCM schedules (describes the “when” of the plan), SCM resources (describes the “how” of the plan), and SCM plan maintenance. These six sub clauses provide an overview of ‘the plan’ (Software Configuration Management Plan) so that readers of this document know what characteristics each sub clause has and how they contribute to overall SCM.

In a broad sense, the SCM activities (the section that relates most to project tracking) are grouped into four functions: configuration identification, configuration control, status accounting, and configuration audits and reviews. Actions are then taken based on these functions such as change, control change, proper implementation of change made, and the report changes made.

Towards the end of the document, tailoring format of the plan is discussed in terms of upward and downward tailoring. Furthermore, how project managers should conform to the standard is addressed towards the end as well.

This document covers the format and content of Software Configuration Management Plans for the IEEE Standard. Most of the plan itself is more detailed sub plans. For this assignment, as mentioned in the overview, the SCM contains useful information for how tracking progress is performed in projects via controlling how changes are identified, requested, modified, tested, reviewed, merged, and applied.

## IEEE/EIA 12207.1

The IEEE 12207.1 focuses on the life cycle data that is produced during the IEEE 12207.0 processes. This life cycle data is designed to help with action involved in the IEEE 12207 process. The life cycle data is designed to describe actions such as define and control life cycle processes, describe and record information about a software product during its life cycle, and assist the software logistics planning. As many of the clauses in IEEE 12207 require life cycle data, it is essential to the process. This life cycle data helps with tracking progress and planning by showing a record of software product produced and the history of what happened during development [5].

## SWEBOK Guide V3.0 Chapter 2

Chapter 2 of “Guide to the Software Engineering Body of Knowledge” covers the aspects of software design, and what should be planned for when doing so. Software design essentially provides a blueprint for the software engineers to follow. The ISO/IEC/IEEE 12207 process showcases software design consists of two activities that are a part of the process. The first step, software architectural design, is where top-level structure is developed and software is organized to identify various components. The second step is detailed design, which describes the behavior of components. Planning out how the software will be structured and developed is an important part of the process.  
 While software is being developed, there are principles and issues to be aware of. Principles can consist of abstraction, emphasizing the parts of an object relevant at hand. Sufficiency and completeness, which has the software, capture all the characteristics of the abstraction and nothing more. Principles also include primitiveness, which means that the design should be easy to implement. Issues can consist of software’s security, how usable it is, and how reliable it is. These issues must be dealt with in advance, such as using cryptography to ensure security, and allowing the software to handle errors and process them to ensure usability.  
 The way the users interact with the software is also an important concern. Overall, the interface should be easy to use and should not surprise users, and it should be forgiving of any mistakes the user makes. It should subtly guide the user so that they can intuitively figure out how the software works on their own. How the software and user interact, such as selecting options from a menu, or providing commands themselves. User experience is one way to measure and estimate software quality. Other ways to measure or estimate them can involve methods such as function based measurement, which measures the software through functional decomposition, which is based on using a structure chart. Object oriented design can also be used, in which the properties of the internal content of each class can be computed. Overall, there are various methods to plan out and estimate measures of a piece of software due to the ISO/IEC/IEEE 12207 standard [6].

## SWEBOK Guide V3.0 Chapter 3

The Software Engineering Body of Knowledge uses information from several sources, one of these is IEEE 12207, to create a resource of common knowledge relating to software design.

Chapter three covers software construction and how it relates to the software life cycle. Linear models focus on the construction point of view, that is building pieces. Linear models engage construction only after adequate planning has taken place. Iterative models promote planning and construction as concurrent processes that occur at the same time. The life cycle chosen affects what tasks are collected into construction.

Now, for construction to occur, a plan must exist describing how and when to build each component. The construction activity plan outlines the method of construction and order of construction [6]. Both of these choices will impact the software life cycle and the team's ability to reduce complexity. The construction method describes in detail, what events must take place before building the software. The order of construction sequences when to build specific components and when to integrate them into existing systems.

# 

# Discussion of Estimating

Starting from IEEE 12207, the first estimation appears in the management process of organizational life cycle process. The organizational processes comprise the pieces that are not tied to any one specific project, but rather capture the lessons learned from many projects to contribute to the organization as a whole [1]. Conceptually, this makes perfect sense with how estimations operate. The manager will pull data from common projects completed in the past to generate a accurate estimate of the current project. This embodies the main idea that the estimation process is not tied to any one specific project.

During clause 7.1.2 the manger will actually combine the necessary documentation for the planning phase. These pieces include effort estimates, schedules, assigning responsibilities and managing risks. As far as compliance to IEEE 12207, there is a degree of flexibility. The document does not define any set structure for these plans, so the project team has freedom of choice on them. The effort estimates could take form in a work breakdown structure for example. An interesting point of note, the documentation is done by the manager; the development team is not mentioned in any of the clauses for documentation related to the organizational life cycle process.

# 

# Discussion of Planning

Starting from IEEE 12207, the first planning appears after the need for the software is realized in acquisition process. Once the acquirer completes the process of defining the need and submitting a contract, and then the supply process starts. The supplier will review said contract, and determine what specific software life cycle to use, if one is not defined in the contract from the acquirer. The initial planning phases stem from this point on, they originate from the supplier.

A supplier will follow this flow. First, review contract and choose life cycle for software if not defined. Second, establish the requirements for the plans that will guide this project from management through to functionality assurance and finally to a deliverable product. These plans will detail resource needs and acquirer involvement [1]. Third the supplier will evaluate the method for development against the risks involved. At this point decisions are made for completing all work in house or outsourcing. Fourth, the supplier will document the project management plans. This section is the most inclusive set of plans of the entire process. The management plans include effort estimations, quality assurance plans, verification and validation to name a few.

To recap, the entire planning process unfolds in many pieces through the entire project. The starting point is when the acquirer defines a need and submits a contract to the supplier. At this point, the supply process starts and the very first thing the supplier will do, is start planning. These plans evolve and continue until delivery.

# Discussion of Tracking Progress

Tracking progress during the IEEE 12207 process will allow us to give better estimates, and plan accordingly for the rest of the software development process. Two main topics have been considered when tracking progress; the type of data we should track, and how this data is tracked during the IEEE 12207 process.

When it comes to tracking progress, having the proper data types is vital, including gathering requirements. Knowing good characteristics, such as unambiguous requirements will allow us to track the progress easier [2]. The life cycle data that is produced in the IEEE 12207.1 process will lead to us understanding the history of the data that is involved with the project [5]. This will show us how the project has been transformed, allowing us to track the progress with ease.

This information is tracked by the creation of the Software Configuration Managements Plans, or SCM. SCM is the means through which the integrity and traceability of the software system is recorded, communicated, and controlled [4]. This is done by providing a standard that deals with the structure for identifying and controlling documentation, code interfaces, and databases to support all life cycle phases [4]. This makes SCM greatly relate to tracking progress during the IEEE 12207 process and shows the way information is kept up with during the process.

# Conclusion

IEEE 12207 describes, from a 10,000 foot view, how to comply with an International Standard for Software Life Cycle processes. Since this information covers a huge scope, IEEE 12207 refers to quite a few supporting documents for exact descriptions of some specific items required to meet this Standard. The main points of this document cover the management and documentation practices for compliance with this International Standard.

# References

[1] 12207 IEEE Information technology Software life cycle processes. IEEE / Institute of Electrical

And Electronics Engineers Incorporated, 1995.

[2] 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE / Institute of Electrical

And Electronics Engineers Incorporated, 1998.

[3]1058-1098 IEEE Standard for Software Project Management Plans. IEEE / Institute of Electrical And Electronics

Engineers Incorporated, 1998.

[4] 828-1998 IEEE Standard for Software Configuration Management Plans. IEEE / Institute of Electrical And

Electronics Engineers Incorporated, 1998.

[5] 12207.1 IEEE/EIA Software life cycle processes—Life cycle data. IEEE / Institute of Electrical And

Electronics Engineers Incorporated, 1998.

[6] P. Bourque and R. E. Fairley, Guide to the software engineering body of knowledge: SWEBOK. Los Alamitos,

CA: IEEE Computer Society, 2014.