Special Report
CMU/SEI-93-SR-07

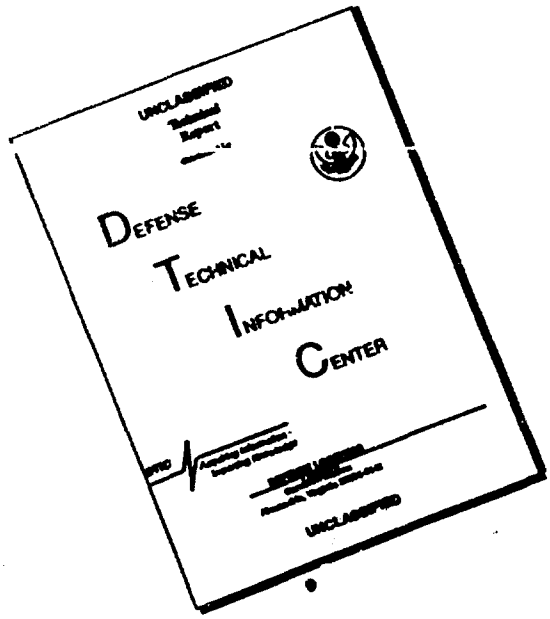
Carnegie-Mellon University
Software Engineering Institute

A Software Process Framework for the SEI Capability Maturity Model: Repeatable Level

Timothy G. Olson
Linda Parker Gates
Julia L. Mullaney
James W. Over
Neal R. Reizer
Marc I. Kellner
Richard W. Phillips
Salvatore J. DiGennaro

June 1993

DISCLAIMER NOTICE



THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

Special Report
CMU/SEI-93-SR-07
June 1993

A Software Process Framework for the SEI Capability Maturity Model: Repeatable Level

Timothy G. Olson
Linda Parker Gates
Julia L. Mullaney
James W. Over
Neal R. Reizer
Marc I. Kellner
Richard W. Phillips

Software Process Definition Project

Salvatore J. DiGennaro

Naval Air Warfare Center

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

This technical report was prepared for the

SEI Joint Program Office
ESC/ENS
Hanscom AFB, MA 01731-2116

The ideas and findings in this report should not be construed as an official DoD position. It is published in the interest of scientific and technical information exchange.

Review and Approval

This report has been reviewed and is approved for publication.

FOR THE COMMANDER

Thomas R. Miller, Lt Col, USAF
SEI Joint Program Office

The Software Engineering Institute is sponsored by the U.S. Department of Defense.
This report was funded by the U.S. Department of Defense.

Carnegie Mellon University
Software Engineering Institute

July 2, 1993

PERMISSION TO REPRODUCE THIS DOCUMENTATION

Permission to reproduce, in whole or in part, the volume of materials released by the Software Engineering Institute under the title *A Software Process Framework for the SEI Capability Maturity Model: Repeatable Level* by Timothy G. Olson, et al is granted under the following conditions:

1. This letter must be reproduced with each copy.

2. All copies must include the copyright notice.

3. The materials are not to be used for commercial gain.

4. The materials are not to be distributed beyond your organization. Refer such requests to the SEI.

5. The materials are to be used in a manner consistent with the framework and methodology advanced by the SEI.

6. Carnegie Mellon University and the Software Engineering Institute are not to be construed as responsible for the results of analyses conducted as a result of this permission. In other words, neither CMU nor the SEI is to be held liable for your use of this material.

Sincerely,

Purvis M. Jackson

Purvis M. Jackson
Sr. Editor and Director
SEI Information Management

Carnegie Mellon University
Pittsburgh, Pennsylvania 15213-3890
(412) 268-7700

Table of Contents

Acknowledgments

The authors would like to thank the customers who sponsored this work and gave us the many ideas for developing this product in the first place. Without customers, this document would not exist. A special thanks to the software engineering process group (SEPG) at Standard Systems Center (SSC), Maxwell Air Force Base, Gunter Annex. The SSC SEPG consisted of Ron Obranovich, Captain Ken Anthonis, Captain Al Braaten, Captain Mark Minkler, and Captain Gloria Trabue. We would also like to thank the primary SSC sponsor, Colonel John Barnhart.

The authors would like to thank the members of the SEI technical staff who contributed their time as technical reviewers of this document. A special thanks to Jim Hart, Michael Konrad, Don McAndrews, Gerald "Mac" McCarty, and Mark Paulk for their review comments which improved the quality of this report.

The authors would also like to thank editor Suzanne Couturiaux for editing this document as well as her helpful review comments.

To The Reader

Document Purpose

This document describes a Software Process Framework (SPF) based on the Software Engineering Institute's (SEI) Capability Maturity Model (CMM). The purpose of the SPF is to support software process improvement by providing guidance for designing, analyzing, and reviewing software processes so that they are consistent with the CMM.

Document Audience

The primary audiences of this document are assumed to be experienced in software process improvement, experienced using the SEI CMM, and familiar with software process assessments and software capability evaluations. The primary audiences of this report are:

- *Software engineering process groups (SEPGs) or process engineers:* Organizational units or personnel responsible for facilitating software process improvement, such as SEPGs and process engineers.
- *Software process improvement groups*: Temporary or permanent organizational teams responsible for improving software processes such as process action teams (PATs), quality improvement teams (QITs) or working groups (e.g., a PAT addressing a software process assessment finding).
- *Software quality assurance groups or process assurance:* Organizational units or personnel responsible for auditing, reviewing, verifying, or validating software processes.

Secondary Audiences of this report include:

- *Managers:* Individuals or groups responsible for planning, controlling, and improving software acquisition, development, or maintenance processes, and are interested in using the CMM for software process improvement.
- *Software process participants:* Individuals or groups responsible for implementing some portion of a software acquisition, development, or maintenance process.
- *Sponsors:* Personnel responsible for funding, authorizing, and providing the needed resources for software process improvement efforts that are based on the CMM.

Document Scope

The scope of this document is the Repeatable Level (level 2) of the SEI Capability Maturity Model. Future versions of the SPF will include the other levels of the CMM, with the Defined Level (level 3) being the next highest priority. The scope of this document includes CMM policies, standards, processes, procedures, training, and tools. Within each key process area of the CMM, the scope of the SPF includes roles, entry and exit criteria, inputs and outputs, reviews and audits, measurements, and work products managed and controlled.

The scope of using the SPF is within an organizational software process improvement effort. Typically, an organization has already conducted a software process assessment, adopted the CMM as a strategy for improvement, and is well under way either planning or addressing the findings of the assessment. The SPF is intended to be used to help process designers and process reviewers to answer the critical question, "Is my software process consistent with the CMM?"

The SPF is not intended to replace software process assessments or software capability evaluations. The SPF is a tool to help organizations design, analyze, and review software processes so that they are consistent with the CMM with the goal of software process improvement.

Report Distribution Guidelines

This document is approved for public release. Requests for this document shall be referred to SEI Information Management.

Preface

Software Process Program Mission

The mission of the SEI Software Process Program is to improve the quality of software development and maintenance processes, and to accelerate the maturity of software engineering as a practice. A fundamental premise of the Software Process Program is that improving the quality of a software process will improve the quality of the software product(s) produced by that process. This premise is based on the principles of process management and statistical process control as successfully applied by Dr. W. E. Deming and Dr. Joseph Juran to Japanese industry after World War II. This relationship between process quality and product quality is a basic assumption that influences the strategy and direction of the Software Process Program and each of its projects, one of which is the Software Process Definition Project.

Software Process Definition Project Mission

The Software Process Definition (SPD) Project supports the Software Process Program mission by advancing the capabilities required to develop and use defined software processes, which are a fundamental prerequisite to improving the quality of software processes within an organization. The mission of the SPD Project is to make the use of defined processes a standard software engineering practice. By "defined process," the SPD Project means that a process is documented, supported (e.g., by training), and practiced. The SPD working definition of "defined process" requires that the actual day-to-day practice, training, and process documentation be equivalent within an organization.

The SPD Project's strategy is to work in close partnership with client organizations to apply process management principles to the practice of software engineering. This strategy allows the SPD Project to gain first-hand knowledge and experience of the real issues and needs facing many software organizations. The SPD Project also collects needs from organizations at strategic events such as the annual Software Engineering Process Group Workshop. From this collection of needs and first-hand experience, the SPD Project can:

- identify and prioritize customer needs,
- build products based on improvement principles that meet customer needs,
- evaluate and refine those products that lead to sustained improvement, and
- transition those products into general use.

The SPD Project has conducted a needs analysis of its customers which are composed of SEPGs, process engineers, process action teams (PATs), management steering committees (MSCs), etc., and has produced the following summary of high-priority needs:

- Software process definition training for SEPGs and PATs: Many SEPGs and PATs are asking for training so they can acquire the needed knowledge and skills for defining software processes.
- Process specification standards and guidelines: Organizations want to know how to design and define software processes, and how to know when a software process is defined well enough to be performed (e.g., minimum information requirements).

Organizations need guidelines for defining, tailoring, planning, performing, and improving software processes. Organizations are also asking for standards and operational definitions for defining software processes.

- Set of process definition frameworks that support CMM key process areas: Organizations are asking the question, "How do I know if my software processes are consistent with the CMM?" Organizations using the CMM have also asked for a format that is more usable for defining software processes.
- A collection of typical process assets as examples to accelerate software process improvement: Many organizations want examples of "good software processes" (e.g., process models and process guides).

This list of prioritized needs guides the SPD product development strategy. The Software Process Framework addresses the third need for a set of process definition frameworks that support a CMM-based software process improvement effort.

There is a misconception that defining software processes only occurs at level 3 of the CMM (the Defined Level). But just as measurement occurs at every level of the CMM, so does process definition. In order for an organization to achieve maturity level 2 of the CMM, the six key process areas (e.g., Software Project Planning) must be defined at some level of granularity. While using the suggested process definition granularity of the Software Process Framework (SPF) is not a requirement to reach maturity level 2, the SPD Project believes the SPF will help organizations not only reach that goal more effectively and efficiently, but reduce the overall time it takes to reach the higher levels of maturity in the CMM.

Chapter 1

Introduction to the Software Process Framework

Overview

Background

Many organizations have started down the path of software process improvement, conducted a software process assessment, and produced action plans to address the assessment findings, only to hit the implementation barrier of "what should we do next?" The Software Process Framework (SPF) helps to address part of this implementation barrier.

Key Questions the SPF Answers

There are many questions that can come in the form of "what should an organization do next in its improvement effort?" These questions typically come from SEPGs, process engineers, PATs, and management steering committees (MSCs). The SPF addresses a few of these key questions:

- How does an organization use the CMM in the context of software process definition and improvement?
- How does an organization know if its software policies, standards, processes, procedures, training, and tools are consistent with the CMM?
- How does an organization know if its software process has been defined with the needed process elements (e.g., *inputs, entry criteria, roles,* etc.) so that the process can be performed?

This chapter addresses these questions.

Continued on next page

Chapter Overview

The table below provides the page number of each section, and describes the key questions that are addressed in each section.

Section Title	Key Question Answered	Page
The Purpose of Software Process Framework	What is the customer need for and purpose of the Software Process Framework?	Intro-3
The SPF is Based Upon the CMM	How does an organization use the CMM in the context of software process definition and improvement?	Intro-4
The SPF is Based Upon an Operational Framework	How does an organization know if its software policies, standards, processes, procedures, training, and tools are consistent with the CMM?	Intro-7
The SPF is Based Upon Process Definition Criteria	How does an organization know if its software process has been defined with the needed process elements (e.g., inputs, entry criteria, roles, etc.) so that the process can be performed?	Intro-11

The Purpose of the Software Process Framework

The Need for a SPF

Many SEPGs have asked the SEI specifically for CMM-based frameworks for defining software processes. One of the top four needs identified by the SPD Project was "a set of process definition frameworks that support (defining and implementing) CMM Key Process Areas."

The SPF is a Companion Document to the CMM

Just as a thesaurus is a useful companion document to a dictionary, the SPF is intended to be a companion document to the CMM for defining software processes. A dictionary and a thesaurus both have the same words in them, but they are both needed because they address a different purpose and are used differently. This is also true of the CMM and the SPF. The SPF is needed because using the CMM to define processes is awkward (just like using a dictionary to find synonyms is awkward).

The Purpose of the SPF

The purpose of the SPF is to provide guidance for designing analyzing, and reviewing software processes so that they are consistent with the CMM. To support that purpose, the SPF:

- is based on the CMM, and principles of quality and process management.
- helps organizations to install defined processes into software engineering practice that are consistent with the CMM.
- describes good process content or "what to define" (i.e., the goal state of a specific maturity level, but not "how to define").
- describes the minimum information criteria for defining software processes that are consistent with the CMM (i.e., what is the minimum I need to define for a process to be able to be performed repeatably).
- helps to make information in the CMM more usable for software process definition and improvement.
- helps to identify policies, standards, processes, procedures, training, and tools required by the CMM at maturity level 2.
- provides checklists for designing, analyzing, and reviewing software policies, standards, processes, procedures, training, and tools so that they can be consistent with the CMM.

The SPF Is Based Upon the CMM

Section Purpose

This section addresses the following question, "How does an organization use the CMM in the context of software process definition and improvement?"

Using the CMM for Process Improvement

The CMM can be applied to software process improvement in several ways:

- as a macro-measure of an organization's institutionalization of continuous software process improvement,
- as a normative model of organizational practices at different maturity levels,
- as the basis for software process assessments and software capability evaluations,
- as a strategy for improving organizational software processes, and
- as a process requirements document (a tactical approach) for defining specific software processes, and installing and institutionalizing key process areas of the CMM into an organization.

Graphically Representing the CMM

Figure 1 describes the five levels of the CMM, the characteristics at each level, the key process areas for each level (i.e., improvement focus areas), and the relationship of increased process maturity and increased productivity and quality to expect in the results (i.e., the software products).

Level	Characteristic	Improvement Focus	Result
5 Optimizing	Continuous improvement	Still human intensive process Maintain organization at optimizing level	Productivity & Quality
4 Managed	Measured process (quantitative basis for improvement)	Defect prevention Technology innovation Process change management	
3 Defined	Process defined and institutionalized (qualitative basis for improvement)	Process measurement Process analysis Quantitative quality plans	
2 Repeatable	Process still dependent on individuals (intuitive)	Organization process focus Organization process defn. Peer reviews Training program Intergroup coordination Software product engineering Integrated software mgt.	
1 Initial	Crisis-driven (ad hoc / chaotic)	Software project planning Software project tracking Software subcontract mgt. Software quality assurance Software configuration mgt. Requirements mgt.	Risk & Waste

Figure 1: Capability Maturity Model for Software

Continued on next page

The SPF is Based Upon the CMM, Continued

The SPF is Based on the CMM

The SPF is founded on the principles and concepts of software process management [Humphrey89a], and based on the SEI's CMM [Paulk93a] for the software process (see Figure 1), which evolved from the 1988 software process maturity framework [Humphrey88]. The reader is assumed to be familiar with the CMM.

Using the CMM as a Strategy

The strategy in the CMM involves maturing organizational processes by building layers of process capability as described by the five level (layer) model. The CMM prescribes an orderly, focused path for organizational process improvement, and strategically recommends a manageable number of key process areas that guide an organization to the next maturity level. Using the Pareto principle [Juran88b], the CMM prescribes the "vital few" key process areas to focus on depending on an organization's process maturity level. By maturing its software processes, an organization can improve quality and productivity, and reduce risk and waste.

The SPF Uses the CMM as a Strategy

The SPF also uses the CMM as a strategy. According to SEI state of the software engineering practice reports, approximately 80% of the software community is at the lowest level of maturity of the CMM (Level 1) [Humphrey89b] [Kitson92]. By being CMM based, the SPF strategically addresses the needs of the software community by focusing on the Level 2 key process areas.

Using the CMM Tactically to Define Software Processes

Assuming that an organization has based its software process improvement strategy on the CMM, the next step is to start focusing tactically on the key process areas. A tactical approach focuses on addressing a specific process area, such as a key process area (KPA) in the CMM. Examples of tactical questions include "how is my organization currently performing software project planning?" and "how is that different from what the CMM suggests or requires?"

Example of Tactically Using the CMM to Define Processes

The following is an example of using the CMM tactically for defining a software process. An organization was assessed to be at maturity level 1, and is addressing the six key process areas (KPAs in level 2). The organization decides to start with a sure win to build success early in the improvement effort and begins with one of its strengths- software configuration management (SCM). The organization forms a process action team (PAT) to define the organization's SCM process. After much struggling, the PAT defines its current SCM process. Someone on the PAT asks the question, "how do we know if our SCM process is a Level 2 SCM process?" The PAT team then pulls out the CMM to check its SCM process against the SCM KPA in the CMM. The PAT compares its SCM process inputs and outputs against the CMM SCM inputs and outputs (as well as entry and entry criteria, roles, activities, etc). The PAT is now using the CMM tactically to define and improve its SCM process.

Continued on next page

The SPF is Based Upon the CMM, Continued

Example of Tactically Using the CMM to Define Policies

An organization had an independent audit of its software policies to find out that 80% of the time the policies required waivers. In other words, in practice few projects were following the policies. A PAT was formed to find out what the problem was, and it was discovered that the policies were over specified or over constraining. The projects could not follow the policies and still be profitable! The PAT decided to rewrite the policies so that waivers would not be needed (i.e., create an unwaiverable policy). Another process requirement was that the new policy meet "Level 2" criteria from the CMM. The PAT then used the CMM to locate all the required policies specified by the CMM. The PAT is using the CMM tactically to define software policies.

Answering the Key Question in this Section

How does an organization use the CMM in the context of software process definition and improvement? An organization can use the CMM strategically to focus on key process areas relevant to its software process maturity, and tactically to define specific software policies, standards, processes, procedures, training, and tools. By using the SPF, an organization can use the information in the CMM strategically and tactically in the context of software process definition and improvement.

The SPF is Based Upon an Operational Framework

Section Purpose

This section addresses the following question, "How does an organization know if its software policies, standards, processes, procedures, training, and tools are consistent with the CMM?" This section also introduces a new framework the SPF is based upon called the Operational Framework.

The Operational Framework

In order to improve operations, an organization should define, document, and communicate its policies, standards, processes, procedures, training, and tools to those that use them. Collectively, these operational elements and their relationships can be referred to as an "Operational Framework" that prescribes, governs, or guides product development (see Figure 2). The Operational Framework may appear to be obvious, but it is a very powerful concept. All too often, policy, standards, process, procedures, and training information are not only all in the same document, they are mixed together. These documents with mixed types of information tend to be very large, confusing, and unfit for use. Using an operational framework to carefully identify and design information will pay large dividends for usable documentation in the long run.

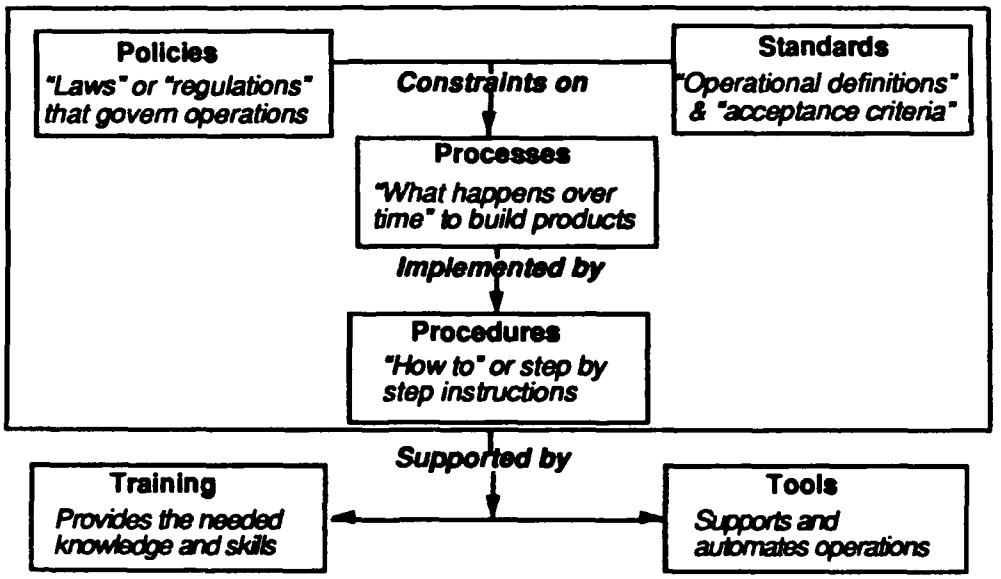


Figure 2. Operational Framework

Continued on next page

The SPF is Based Upon an Operational Framework, Continued

Definitions of the Operational Framework Parts

The parts that make up the Operational Framework and their definitions are listed below:

- **policy** - provides the "law" or "regulations" that govern, guide, or constrain operations.
- **standard** - provides the "operational definitions" or "acceptance criteria" for final or interim products or processes.
- **process** - describes "what happens" over time within the organization to build products that meet the standards in accordance with the organizational policies.
- **procedure** - describes "how-to" or "step-by-step" instructions that implement a process.
- **training** - provides people with necessary knowledge and skills including training on organizational policies, standards, processes, procedures, and tools.
- **tool** - provides the needed support for organizational policies, standards, processes, procedures, and training in order to build software products.

The Relationship of the Operational Framework Parts

The parts of the Operational Framework defined above are related. Policies govern, guide, and constrain processes. For example, a policy may require a process to complete within a certain period of time. Standards also guide and constrain processes. A product standard will constrain the output of a process by providing the structure information or acceptance criteria for that output. A process standard may state operationally what entry or exit criteria are required for a process. Furthermore, processes are supported by procedures. While processes typically involve one or more roles (usually more than one), procedures are written to be performed by one person. Procedures are step by step, or "how-to" information, that implements part of a process. Policies, standards, processes, and procedures are supported by training and tools. For example, policies could be put under a configuration management tool to track changes and maintain control. Training provides the needed knowledge and skills to perform processes and related procedures. The parts of the Operational Framework work together as an operational system so that software operations may work more effectively and efficiently.

Continued on next page

The SPF is Based Upon an Operational Framework, Continued

The Benefits of the Operational Framework

The are many benefits of using the Operational Framework. The Operational Framework helps to:

- *Separate information into usable parts:* The Operational Framework separates information into usable parts used for different purposes. For example, if a senior manager wants to see an organizational policy, then looking at a large confusing document or detailed training mixed in with the policy is irrelevant information (meaning either wasted time or the policy won't be used).
- *Indentify and use only the relevant information for each part:* Only relevant information needs to be in each part of the Operational Framework. Training information should only be in training documents. Policies should contain information that does not change frequently. Process information should be "what happens over time," not a step by step procedure. People will then learn where to look for relevant information.
- *Manage the operational parts to work together as a system:* Once the information has been identified and partitioned correctly into the parts of the Operational Framework, the information can be designed as a system. The relationships between the parts can be optimized so that they work together to communicate to people performing the work. Process documents should help people do their work, not be "red tape."
- *Manage changes and improvements:* Changes and improvements to the operational parts will be easier to manage because the information is well defined. For example, once defined, policies should not frequently change. Processes probably do not need to change if a step by step procedure changes. Training changes can be isolated to training documents. Only the necessary and important relationships between the operational parts need to be managed.
- *Mange and improve communication:* Communication improves because people know where to look for certain types of information, and they know the relationships between the information. Since the changes are more isolated to the operational parts, less communication is needed and only the relevant changes need to be managed and communicated.

Continued on next page

The SPF is Based Upon an Operational Framework, Continued

Process Fidelity and the Operational Framework

Process fidelity is the concept of following or adhering to a defined process. When a defined process is not used or followed, process fidelity is said to be low. The concept of an Operational Framework is very powerful when process fidelity is high, and there is a feedback loop for continuous process improvement. When everyone uses, adheres to, and improves the defined software process consisting of policies, standards, processes, procedures, training, and tools, then the organization can rapidly improve and optimize the operational system.

The SPF is Organized Around the Operational Framework

The SPF is based on and organized around the Operational Framework. In the SPF, there are separate chapters for designing, analyzing, and reviewing policies, standards, processes, and procedures. Training and tools can be found in two places:

- in the chapter "Process Checklists" for each CMM KPA
- in their entirety in the chapter "Cross-Reference for CMM Level 2"

Answering the Key Question in this Section

By using the SPF checklists, an organization can know if its software policies, standards, processes, procedures, training, and tools are consistent with the CMM.

The SPF is Based Upon Process Definition Criteria

Section Purpose

This section answers the question, "How does an organization know if its software process has been defined with the needed process elements (e.g., inputs, entry criteria, etc.) ?"

Key Process Questions

In order to perform a process or activity, a process definition or document must answer key process questions. Questions such as what are the work products produced (artifacts)?, who is involved?, and when does the activity start and end?, must be answered in order to perform a process. What are the minimum number of key process questions that need to be answered in order to perform a process or activity? The table below (Figure 3) is an example of the minimum key process questions that need to be answered in order to define a process or build a process model capable of being performed. The answers to the questions or guidelines are "process elements" that must be documented or defined (hence, the name of "process documentation element").

Process Information Guideline	Process Documentation Element
What artifacts are produced?	list of artifacts
What activities are performed?	list of activities
What agents are involved?	list of agents
How are activities implemented?	sub-activity, procedure, method
When do activities begin and end?	activity entry and exit criteria
What artifacts are produced (or consumed) by which activities?	activity inputs and outputs
What agents are responsible for performing each activity?	activity performed by
Why is an activity performed?	activity purpose
What activity is next?	activity flow

Figure 3. Key Process Questions

Continued on next page

The SPF is Based Upon Process Definition Criteria, Continued

Definitions of the Process Elements

The definitions of the process terminology used in Figure 3 and in the SPF are listed below:

activity: the action taken to create or achieve a work product, service, or result.

entry criteria: describe the conditions under which an activity can be started. Entry criteria take the form of a simple or compound predicate about the state of a work product, agent, or activity.

exit criteria: describe the conditions under which an activity can be declared complete. Exit criteria take the form of a simple or compound predicate about the state of an work product, agent, or activity.

input: the relationship or link between an activity and a work product. Inputs are the results produced by a prior activity and used by the current activity and may be qualified by the state of a work product.

output: the relationship or link between an activity and a work product. Outputs are the results produced by the current activity and used by a subsequent activity and may be qualified by the state of a work product.

role: "... a unit of defined responsibilities that may be assumed by one or more individuals." [Paulk93b]. The accomplisher or performer that carries out the action to achieve or create the work product, service, or result. A role can consist of automation, or even be totally automated.

work product: any product, service, or result of a process or activity.

Continued on next page

The SPF Is Based Upon Process Definition Criteria, Continued

The SPF is a Process Template

The SPF provides a template for defining and modeling software processes. For each key process area in the CMM, the SPF provides checklists for each of the following:

- roles
- entry criteria
- inputs
- activities
- outputs
- exit criteria
- reviews and audits
- measurements
- work products managed and controlled
- documented procedures
- training
- tools

Answering the Key Question in this Section

How does an organization know if its software process has been defined with the needed process elements (e.g., inputs, entry criteria, etc.) ? This question is addressed once an organization defines its software process so that it can answer the key process questions in Figure 3. In addition, each process checklist in Chapter 5 provides the relevant information from the CMM to answer most of the key process questions in Figure 3.

Chapter 2

How to Use the Software Process Framework

Overview

Chapter Purpose

The purpose of this chapter is to provide guidance on how to use the Software Process Framework for designing, analyzing, or reviewing software processes so that they are consistent with the CMM. This guidance is not "how to design" or "how to analyze" software processes. Rather, the guidance is focused on "how to use the Software Process Framework," assuming the reader can already design or analyze software processes. The audience using this chapter is expected to be experienced in software process definition and improvement and familiar with the CMM (e.g., software engineering process groups, process engineers, process action teams, software quality assurance groups, etc).

Two Primary Usages for the SPF

There are two primary uses for the Software Process Framework:

- analyzing and reviewing software processes to check or ensure they are consistent with the CMM (in the context of process assurance or process verification).
- designing software processes to be consistent with the CMM (in the context of software process improvement).

In This Chapter

This chapter is composed of the following sections:

Overview of How to Use the SPF

Section Purpose The purpose of this section is to provide an overview of how to use each chapter and appendix in the SPF.

Using the SPF The following table describes how to use each chapter and appendix in the SPF:

#	Chapter	Use
	To The Reader	Use this section for an overview of the purpose, audience, and scope of the SPF.
1	Introduction	Use this chapter for an overview of the fundamental concepts and principles upon which the SPF is based.
2	How to Use the Software Process Framework	Use this chapter for guidance of how to use the SPF for designing, analyzing, and reviewing software processes to be consistent with the CMM.
3	Policy Checklists	Use this chapter to help design, analyze, and review software policies so that they are consistent with the CMM.
4	Standard Checklists	Use this chapter to help design, analyze, or review software standards so that they are consistent with the CMM.
5	Process Checklists	Use this chapter to help design, analyze, or review software processes so that they are consistent with the CMM. This chapter contains process definition information such as roles, inputs and outputs, entry and exit criteria, etc.
6	Procedure Checklists	Use this chapter to help design, analyze, or review software procedures so that they are consistent with the CMM.
7	Level 2 Cross-Reference Checklists	Use this chapter to view Level 2 of the CMM as a whole from a particular perspective. For example, one checklist contains all the Level 2 KPA purposes on one page. Another checklist contains of all the required procedures at Level 2.

Continued on next page

Overview of How to Use the SPF, Continued

Using the SPF, Continued

The following table describes how to use each appendix in the Software Process Framework, continued from the previous page.

#	Appendix	Use
	References	Use this appendix to identify references in the public domain that the SPF is based upon.
A	List of Acronyms	Use this appendix to understand acronyms used in the SPF.
B	Glossary of Terms	Use this appendix to locate key definitions used in the Software Process Framework.
C	Translation Tables	Use this appendix to translate CMM terminology into your organization's terminology.
D	Process Templates	Use the process templates when you are defining and designing a software process. Also included in this appendix are annotated process templates that describe how to fill out the templates. These process templates are representation independent, and will capture the needed information for almost all process representations.
E	CMM Roles	Use this appendix to locate the definitions of organizational roles from the CMM Version 1.1 (provided for your convenience when using the translation tables - see Page Usage-15).
F	CMM Glossary	Use this appendix to locate definitions from the CMM Version 1.1 (provided for your convenience).

The Software Process Framework is Not...

Section Purpose

The purpose of this section is to describe what the Software Process Framework is not, or should not be used for, and why.

The SPF is Not...

The Software Process Framework is:

- **not "how."** The SPF doesn't tell you "how" to get to maturity Level 2, but rather what the "goal state" or Level 2 looks like from a process definition and improvement perspective.
- **not process definition training.** The SPF does not provide all the needed knowledge and skills for defining a software process. The SPF also does not provide a method or process for defining a software process.
- **not a replacement for the CMM.** As a thesaurus is a companion document to a dictionary, the SPF is a companion document to the CMM. Although a thesaurus and a dictionary have similar information (e.g., they use the same words), they are used for different purposes. The CMM contains information about process maturity, the SPF contains similar information but for the purpose of designing, analyzing, and reviewing software processes so that they are consistent with the CMM.
- **not a process model or process guide.** The SPF is not a process model or a process guide. However, the SPF has been used as a template by SEPGs to produce process guides and build process models in practice.

Using the General Features of the SPF

Using the Features of the SPF

The table below provides an overview of the general features of the Software Process Framework. Each feature is then described in detail in later sections of this chapter (except for the process templates which are in Appendix D).

SPF Feature	How to Use	See Page
CMM References	Every place where the SPF uses information from the CMM, the exact location in the CMM Version 1.1 is referenced for traceability. This feature helps to make the SPF a companion document to the CMM for software process definition and improvement.	Usage-8
Checklists and Checkboxes ❑	Checklists have been designed into the SPF. The SPF checklists allow process designers, analyzers, or reviewers to check whether their software processes are consistent with the CMM. Checkboxes are used within checklists so that every CMM requirement, no matter how small, can be checked off individually.	Usage-14
Space for User References	The checklists have been designed to provide user space for referencing organizational documents. For example, if a user is reviewing an organizational document against the SPF and wants traceability (from the SPF to the organizational document), the user simply puts a reference to the document in the "*References*" box (*e.g., chapter, section, page, paragraph*).	Usage-13
Translation Tables	Example translation tables are provided in this section and templates are provided in Appendix C to help translate CMM terminology into your organization's terminology. For example, the CMM term "software development plan" may be called "software project plan," or something different in your organization.	Usage-11 Appendix C
Process Templates	Annotated process templates and blank process templates for defining and designing software processes are provided in Appendix D. These templates include lists for work products, roles, activities, and detailed templates for process activities.	Appendix D

Using the SPF to Analyze and Review Software Processes

Using the SPF to Analyze and Review Software Processes

This table assumes you have chosen a software policy, standard, process, or procedure to analyze and review. The steps in the following table will help you use the SPF for analysis and review. (The table does not tell you "how to analyze or review.")

Step	Action
1	If not already done, use the translation table of CMM Roles to translate CMM Roles into organizational roles. (See "Using the CMM Translation Tables" in this chapter and Appendix C for templates.)
2	If not already done, use the translation table of CMM General Terms to translate the "CMM Work Products" into organizational work products. (See "Using the CMM Translation Tables" in this chapter and Appendix C for templates.)
3	Translate the CMM inputs and outputs of the Process Checklists into organizational inputs and outputs. (See "Using the CMM Translation Tables" in this chapter and the input and output sections in "Chapter 5: Process Checklists.")
4	Use the checklists in the SPF to analyze and review your policies, standards, processes, or procedures. (See "Using the Checklists and Checkboxes" in this chapter.) Put checks next to the satisfied criteria, and document the reference to the organizational document or model (See "Using the 'User References' in the Checklists" in this chapter.) The references provide traceability between the SPF and the document being analyzed and reviewed.
5	Don't put checks next to the unsatisfied criteria. Document the reference of where in the process document or model the criteria probably should be satisfied, if possible. (This is necessary for improving the document or process being evaluated.) Also mark N/A next to CMM criteria that do not apply to your organization (e.g., if your organization doesn't subcontract out software, then the Software Subcontract Management KPA probably does not apply to you).
6	When the analysis or review is completed, you will have completed checklists of the CMM criteria that are satisfied and unsatisfied. You also have a list of CMM criteria that do not apply to your organization, and you can add rationale of why they do not.
7	If all CMM criteria are satisfied, then the policy, standard, process, or procedure is consistent with the CMM. Otherwise, go to the process design guidance on the next page to start designing improvements to address the unsatisfied criteria.

Using the SPF for Designing Software Processes

Using the SPF for Designing Software Processes

This table assumes you have selected a software policy, standard, process, or procedure to design. The steps in the following table will help you use the SPF for design, but the table does not tell you "how to design."

Step	Action
1	Use the translation table of CMM Roles to translate the "CMM roles" into your organizational roles. (See "Using the CMM Translation Tables" in this chapter and Appendix C for the templates.)
2	Use the translation table of CMM Organization Terms to translate the "CMM organizational terms" into your organizational terms. (See "Using the CMM Translation Tables" in this chapter and Appendix C for templates.)
3	If you're not designing a software process (e.g., if you are designing a software policy), skip to step 6. Use the first 3 process templates to document your process activities, roles, and work products. (See Appendix D "Process Templates.") Use the Process Checklists in Chapter 5 (Roles, Activities, Inputs/Outputs) to make sure you completed process templates in order to satisfy CMM criteria.
4	For each process activity you wish to model or define, fill out the two process activity templates. (See Appendix D "Process Templates.") This 2-page template contains the essential process elements: inputs/outputs, entry/exit criteria, sub-activities, etc. Use the Process Checklists in Chapter 5 to make sure your process activities meet CMM criteria.
5	You will need to select a process model or process guide representation(s) if you haven't already done so.
6	Use the appropriate SPF checklists while designing your software policy, standard, process, or procedure to be consistent with the CMM. For example, an inputs checklist from a process KPA can be used to make sure your software process represents the required inputs.
7	When you are done with your design, use the SPF to help you perform an analysis or review (see previous page).

Using the References to the CMM

Section Purpose

The purpose of this section is to describe the notation used in the Software Process Framework to reference the CMM.

Definition

CMM Reference: identifies the exact location in the CMM Version 1.1 where the source material in the Software Process Framework is derived from.

Syntax

"CMM Reference" is defined as:

([CMM-Page], [Key Practice], [Subpractice].[Subpractice Sub-Bullet])

Each component of the "CMM Reference" is described below.

Description of the CMM-Page Component

The [CMM-Page] component is the page in the CMM where the referenced text is located. For example, the first CMM page number in the Repeatable Level is L2-1, which is referenced as (L2-1) in the SPF. Text which spans page boundaries of the CMM is referenced by the *first* page only.

Description of the Key Practice Component

The [Key Practice] component is an abbreviation based on the CMM common features. (Common features classify the key practices as outlined in the table below.) The number assigned to the CMM key practice is also included with the abbreviation. For example, Activity 3 of Requirements Management is found on page L2-7, so the "CMM Reference" would be (L2-7, A3). Please see the page after the next for a graphical example. For CMM references to descriptions, definitions, or purposes, the [Key Practice] component is omitted.

[Key Practice] := <Abbreviation><Number of Key Practice from CMM>

CMM Common Features (key practice type)	Abbreviation...
Goal (not a common feature, but added as an SPF abbreviation)	G
Commitment to Perform	C
Ability to Perform	Ab
Activity Performed	A
Measurement and Analysis	M
Verifying Implementation	V

Continued on next page

Using the References to the CMM, Continued

Description of Subpractice Component

Many of the key practices found in the CMM contain subpractices. The third component, or [Subpractice] component, is the sentence number of the subpractice in the CMM. For example, Activity 3 (key practice) of Requirements Management has subpractices. The "CMM Reference" to the first subpractice would be (L2-7, A3, 1). Please see the next page for a graphical example.

Description of Subpractice Sub-Bullet Component

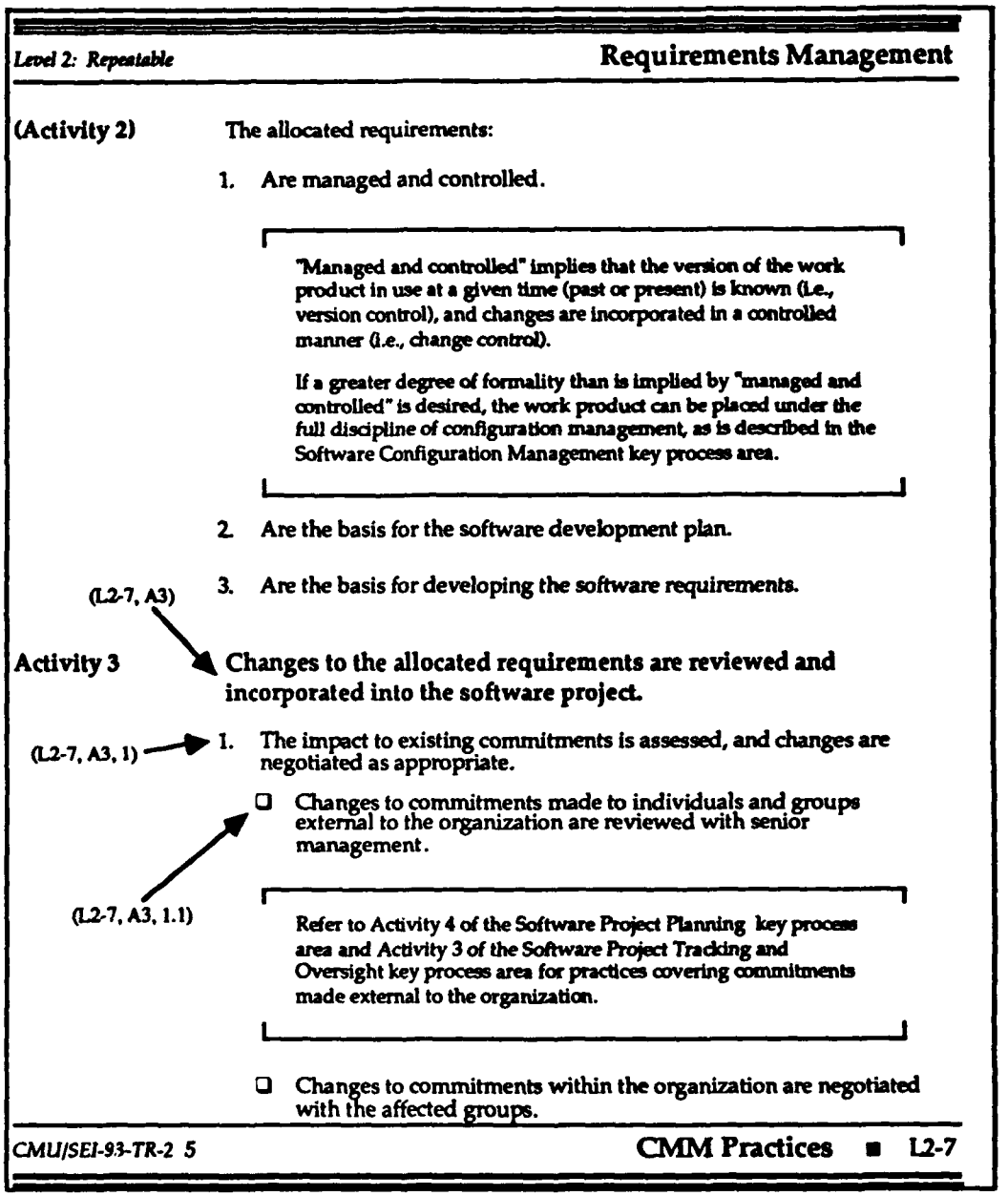
Many subpractices contain one or more sub-bullets (e.g., checkboxed sentences). When one of these sub-bullet sentences is referenced, the number of the sub-bullet becomes appended to the [Subpractice] component as the [Subpractice Sub-Bullet] component after a ".". For example, the "CMM Reference" to the first sub-bullet in subpractice (L2-7, A3, 1) would be (L2-7, A3, 1.1). Please see the next page for a graphical example.

Continued on next page

Using the References to the CMM, Continued

Examples of CMM References

The following diagram illustrates how various passages in the CMM would be referenced.



Using the CMM Translation Tables

Section Purpose

The purpose of this section is to provide examples of translating CMM terminology into your organizational terminology. Although this may seem unnecessary, making the translation assumptions explicit has proven to be beneficial in practice. Experience has shown that there is rarely a one-to-one mapping of terminology, and there are usually "gray" areas that should be documented. Also note that sometimes there can also be different terminologies within the same organization (e.g., projects or divisions within the same organization may use different terminology).

Example Translation Table for CMM Roles/Groups

The following table provides an example of a translation table from CMM roles into a fictitious organization's roles. Appendix D (CMM Roles) and Appendix E (CMM Glossary) provide the definitions from the CMM.

CMM Roles/Groups	Your Organization's Roles/Groups
Affected groups or other affected groups (Affected groups change according to the context of a situation. Make a complete list here of affected groups, and use this list to help build the list of roles in the process checklists for each situation.)	SQA SCM Marketing Sales Technical staff Database group Testing Department Documentation Department Etc.
Customer	Customers
End user or end user representatives	Users
Engineering group	Technical staff
First-line software manager	Project manager
Group responsible for analyzing and allocating system requirements	Technical staff
Manager	N/A
Prime contractor	N/A
Project manager	Project manager
Project software manager	Project manager
Senior management	Senior management steering committee
Software engineering process group	Project
Senior manager	CEO
Software engineering staff	Technical staff

Continued on next page

Example Translation Table for General CMM Terms

The following table provides an example of a translation table from general CMM terms into a fictitious organization's terms. The blank entries are for additional CMM terms that you may wish to translate. Appendix D (CMM Roles) and Appendix E (CMM Glossary) provide the definitions from the CMM.

CMM General Terms	Your Organization's Term
Product	Product Lines
Project	Project
Software product	Product Lines
Software project	Project
System	Product Lines

Translation in Other Areas of the SPF

After using the general translation tables in Appendix C, other areas of the SPF will also need to be translated into your organization's terminology. Roles such as "affected groups" will need to be translated on a case by case basis since they change according to context. (Each context of affected groups is listed in the "Roles" section of the "Process Checklists" in Chapter 5.) The CMM terminology of work products such as inputs and outputs will also need to be translated into your organization's terminology. The input and output translation tables are built into the input and output sections of the "Process Checklists" in Chapter 5.

Example of Using the Translation Tables in Inputs/Outputs

Use the "Org. Input" and "Org. Output" columns from the input and output tables to translate CMM inputs and outputs to your organization's terminology. (These input and output tables can be found in the "Process Checklists" in Chapter 5.) The table below is an example of an input table from Requirements Management translating CMM inputs into a fictitious organization's inputs:

√	Input	Org. Input	References
√	Statement of Work. (L2-14, Ab1) [Refer to SPF Standards for additional information regarding a Statement of Work.]	*SOW*	
√	Allocated requirements. (L2-18, A6, 1.4) [Refer to SPF Standards for additional information regarding allocated requirements.]	*SRS* *IRS*	

Using the "User References" Space in the Checklists

Section Purpose

The purpose of this section is to provide an example of how to use the "User References" space in the SPF checklists.

Example of Using the References

The purpose of the "References" column in the checklists is to provide users space for writing references to their organizational documents. These user references are very important for traceability between the SPF and the document being analyzed and reviewed. For example, if a user is reviewing an organizational document against the SPF and wants traceability (from the SPF to the organizational document), the user simply puts a reference to the organizational document in the "References" box (e.g., chapter, section, page, paragraph). Abbreviations are recommended since blank space is limited, and there are numerous reference boxes to fill in. In the example below, "V1" is an abbreviation for "Volume 1"; "V2" is an abbreviation for "Volume 2"; and "S" is an abbreviation for "Section". In this example, the numbers allow traceability to the exact page and sub-section of the hypothetical organizational document. This is only a simple example, so use abbreviations that work best for you.

Exit Criteria

The table below describes the conditions that must be satisfied in order to exit the software configuration management process.

√	Condition	References
√	A SCM plan is prepared for each software project according to a documented procedure. (L2-76, A1)	*V1 S3.5.2*
	A documented and approved SCM plan is used as the basis for performing the SCM activities. (L2-77, A2)	*V1 S3.5.2*
√	A configuration management library system is established as a repository for the software baselines. (L2-77, A3)	*V1 S7*
√	Change requests and problem reports for all configuration items/units are initiated, recorded, reviewed, approved, and tracked according to a documented procedure. (L2-79, A5)	*VII S4-6*
√	Changes to baselines are controlled according to a documented procedure. (L2-80, A6)	*VII S7*

Some Guidance for User References

Notice that there can be references to items in the checklist that are not checked "√". These references can be used as pointers to areas of the organizational document that can be improved. For example, in the checklist above the second item is not satisfied. The reference points to the exact location in the organizational document where this improvement should be added to satisfy the second item in the checklist in the future.

Using the Checklists and Checkboxes

Section Purpose

The purpose of this section is to describe how the checklists and checkboxes can be used in the SPF.

Example of Using the Checklists

The following example illustrates how the SPF checklists are used to show consistency to the CMM, and illustrate satisfaction and unsatisfaction of specific CMM criteria. Notice that in the example below, only the project manager role has been completely satisfied.

The checkboxes "❑" are used when there are checklists within checklists (nested checklists). When there are nested checklists, the left hand column becomes the parent checklist. Only check the parent checklist if all the checkboxes in that parent are satisfied. In the example below, the SCCB role is not satisfied (the parent checklist is not checked) because all of the checkboxes within that parent have not been checked.

Roles

The table below describes the activities that are performed by the CMM roles in the software configuration management process.

√	Role	Activity	Reference
√	Project Manager	The SCM activities are reviewed with the project manager on both a periodic and event-driven basis. (L2-83, V2)	*V1 S5.2*
	SCCB	The SCCB: (L2-73, Ab1)	
		❑ Authorizes the establishment of software baselines and the identification of configuration items/units.	*V1 S4.6.2*
		☑ Represents the interests of the project manager and all groups who may be affected by changes to the software baselines.	*V1 S4.6*
		☑ Reviews and authorizes changes to the software baselines.	*V1 S4.6.2a*
		☑ Authorizes the creation of products from the software baseline library.	*V1 S4.6.2.b*

Other Possible Uses of the SPF

Section Purpose

The purpose of this section is to provide some ideas of how the SPF could be used other than for software process design, analysis, and review.

List of Other SPF Usages

The following list is preliminary, and should be treated as such. The SPF could be used to:

- provide criteria for a "good process definition."
- help identify who should be involved in a software process improvement effort.
- help communicate Level 2 to upper management (or to whomever).
- guide charter and plan development for PATs.
- guide PATs in pilot selection criteria.
- help measure success criteria for pilot projects, and for software process installation and institutionalization.
- help develop process guides and process models (e.g., SEPG uses the SPF as a template to build a process guide).
- help tailor software processes (e.g., structure and content).
- provide a checklist for SQA reviews and audits.
- help design organizational roles, responsibilities, and scope (e.g., SCM).

Chapter 3

Policy Checklists

Overview

Chapter Purpose

The purpose of the policy checklists is to provide:

- guidance in identifying which policies are required by the CMM.
- criteria that an organization can use to evaluate its software policies to determine if they are consistent with the CMM.
- information that can be used to develop software policies so that they are consistent with the CMM.

Chapter Definitions

policy: A guiding principle, typically established by senior management, that is adopted by an organization to influence and determine decisions. Policy provides the "law" or "regulations" that govern or constrain operations.

Policy Checklist Description

The table below lists the contents associated with a policy checklist and contains a description of each subsection:

Name of Subsection	Description
Definitions	This subsection defines the terms that may cause confusion. Example: The term "software quality assurance" often has different meanings in different organizations.
Policy Checklist	This subsection contains criteria that the organizational policy can be evaluated against. It describes criteria that must be addressed by organizational policy to be consistent with the CMM. Example: The CMM requires that the software quality assurance (SQA) policy ensures the SQA function is in place on all software projects.
Policy Goals	This subsection is a reminder to policy designers and evaluators to keep in mind the KPA goals. The goals can be thought of as the results of implementing an effective policy.

Continued on next page

In This Chapter

This chapter covers the following topics:

Requirements Management (RM) Policy

Definitions

allocated requirements (system requirements allocated to software): The subset of the system requirements that are to be implemented in the software components of the system. The allocated requirements are a primary input to the software development plan. Software requirements analysis elaborates and refines the allocated requirements and results in software requirements which are documented.

RM Policy Checklist

The project follows a written organizational policy for managing the system requirements allocated to software (L2-2, C1). This policy typically specifies that:

√	Description	References
	The allocated requirements are documented. (L2-3, C1, 1)	
	The allocated requirements are reviewed by: (L2-3, C1, 2) ❑ the software managers, and ❑ other affected groups.	
	The software plans, work products, and activities are changed to be consistent with changes to the allocated requirements. (L2-3, C1, 3)	

RM Policy Goals

Implementation of an effective requirements management policy results in:

√	Result of Effective Implementation of RM	References
	System requirements allocated to software are controlled to establish a baseline for software engineering and management use. (L2-2, G1)	
	Software plans, products, and activities are kept consistent with the system requirements allocated to software. (L2-2, G2)	

Software Project Planning (SPP) Policy

Definitions

software development plan: The collection of plans that describe the activities to be performed for the software project. It governs the management of the activities performed by the software engineering group for a software project. It is not limited to the scope of any particular planning standard, such as DOD-STD-2167A and IEEE-STD-1058, which may use similar terminology.

SPP Policy Checklist

The project follows a written organizational policy for planning a software project (L2-12, C2). This policy typically specifies that:

√	Description	References
	The system requirements allocated to software are used as the basis for planning the software project. (L2-12, C2, 1)	
	The software project's commitments are negotiated between: (L2-12, C2, 2) ❑ the project manager, ❑ the project software manager, and ❑ the other software managers.	
	Involvement of other engineering groups in the software activities is negotiated with these groups and is documented. (L2-13, C2, 3)	
	Affected groups review the project's: (L2-13, C2, 4) ❑ software size estimates, ❑ effort and cost estimates, ❑ schedules, and ❑ other commitments.	
	Senior management reviews all software project commitments made to individuals and groups external to the organization. (L2-13, C2, 5)	
	The project's software development plan is managed and controlled. (L2-13, C2, 6)	

SPP Policy Goals

Implementation of an effective software project planning policy results in:

√	Result of Effective Implementation of SPP	References
	Software estimates are documented for use in planning and tracking the software project. (L2-12, G1)	
	Software project activities and commitments are planned and documented. (L2-12, G2)	
	Affected groups and individuals agree to their commitments related to the software project. (L2-12, G3)	

Software Project Tracking and Oversight (SPTO) Policy

Definitions

software development plan: The collection of plans that describe the activities to be performed for the software project. It governs the management of the activities performed by the software engineering group for a software project. It is not limited to the scope of any particular planning standard, such as DOD-STD-2167A and IEEE-STD-1058, which may use similar terminology.

SPTO Policy Checklist

The project follows a written organizational policy for managing the software project (L2-30, C2). This policy typically specifies that:

√	Description	References
	A documented software development plan is used and maintained as the basis for tracking the software project. (L2-30, C2, 1)	
	The project manager is kept informed of the software project's status and issues. (L2-30, C2, 2)	
	Corrective actions are taken when the software plan is not being achieved, either by adjusting performance or by adjusting the plans. (L2-30, C2, 3)	
	Changes to the software commitments are made with the involvement and agreement of the affected groups. (L2-30, C2, 4)	
	Senior management reviews all commitment changes and new software project commitments made to individuals and groups external to the organization. (L2-31, C2, 5)	

SPTO Policy Goals

Implementation of an effective software project tracking and oversight policy results in:

√	Result of Effective Implementation of SPTO	References
	Actual results and performances are tracked against the software plans. (L2-30, G1)	
	Corrective actions are taken and managed to closure when actual results and performance deviate significantly from the software plans. (L2-30, G2)	
	Changes to software commitments are agreed to by the affected groups and individuals. (L2-30, G3)	

Software Subcontract Management (SSM) Policy

Definitions

prime contractor: An individual, partnership, corporation, or association that administers a subcontract to design, develop, and/or manufacture one or more products.

subcontractor: An individual, partnership, corporation, or association that contracts with an organization (i.e., the prime contractor) to design, develop, and/or manufacture one or more products.

SSM Policy Checklist

The project follows a written organizational policy for managing the software subcontract (L2-44, C1). This policy typically specifies that:

√	Description	References
	Documented standards and procedures are used in selecting software subcontractors and managing the software subcontracts. (L2-45, C1, 1)	
	The contractual agreements form the basis for managing the subcontract. (L2-45, C1, 2)	
	Changes to the subcontract are made with the involvement and agreement of both the prime contractor and the subcontractor. (L2-45, C1, 3)	

SSM Policy Goals

Implementation of an effective software subcontract management policy results in:

√	Result of Effective Implementation of SSM	References
	The prime contractor selects qualified software subcontractors. (L2-44, G1)	
	The prime contractor and the software subcontractor agree to their commitments to each other. (L2-44, G2)	
	The prime contractor and the software subcontractor maintain ongoing communications. (L2-44, G3)	
	The prime contractor tracks the software subcontractor's actual results and performance against its commitments. (L2-44, G4)	

Software Quality Assurance (SQA) Policy

Definitions

software quality assurance: (1) A planned and systematic pattern of all actions necessary to provide adequate confidence that a software work product conforms to established technical requirements. (2) A set of activities designed to evaluate the process by which software work products are developed and/or maintained. [Derived from IEEE-STD-610]

SQA Policy Checklist

The project follows a written organizational policy for implementing software quality assurance (L2-60, C1). This policy typically specifies that:

√	Description	References
	The SQA function is in place on all software projects. (L2-60, C1, 1)	
	The SQA group has a reporting channel to senior management that is independent of: (L2-61, C1, 2) ❑ the project manager, ❑ the project's software engineering group, and ❑ other software-related groups.	
	Senior management periodically reviews the SQA activities and results. (L2-61, C1, 3)	

SQA Policy Goals

Implementation of an effective software quality assurance policy results in:

√	Result of Effective Implementation of SQA	References
	Software quality assurance activities are planned. (L2-60, G1)	
	Adherence of software products and activities to applicable standards, procedures, and requirements is verified objectively. (L2-60, G2)	
	Affected groups and individuals are informed of software quality assurance activities and results. (L2-60, G3)	
	Noncompliance issues that cannot be resolved within the software project are addressed by senior management. (L2-60, G4)	

Software Configuration Management (SCM) Policy

Definitions

software baseline audit: An examination of the structure, contents, and facilities of the software baseline library to verify that baselines conform to the documentation that describes the baselines.

software baseline library: The contents of a repository for storing configuration items and the associated records.

SCM Policy Checklist

The project follows a written organizational policy for implementing software configuration management (L2-72, C1). This policy typically specifies that:

√	Description	References
	Responsibility for SCM for each project is explicitly assigned. (L2-72, C1, 1)	
	SCM is implemented throughout the project's life cycle. (L2-72, C1, 2)	
	SCM is implemented for externally deliverable software products, designated internal software work products, and designated support tools used inside the project (e.g., compilers). (L2-72, C1, 3)	
	The projects establish or have access to a repository for storing configuration items/units and the associated SCM records. (L2-72, C1, 4)	
	The software baselines and SCM activities are audited on a periodic basis. (L2-73, C1, 5)	

SCM Policy Goals

Implementation of an effective software configuration management policy results in:

√	Result of Effective Implementation of SCM	References
	Software configuration management activities are planned. (L2-72, G1)	
	Selected software work products are identified, controlled, and available. (L2-72, G2)	
	Changes to identified software work products are controlled. (L2-72, G3)	
	Affected groups and individuals are informed of the status and content of software baselines. (L2-72, G4)	

Chapter 4

Standard Checklists

Overview

Chapter Purpose

The purpose of the standard checklists is to provide:

- guidance in identifying the contents of standard work products that are required by the CMM.
- criteria that an organization can use to evaluate its software standards to determine if they are consistent with the CMM.
- information that can be used to develop software standards that are consistent with the CMM.

Chapter Definitions

standard: A documented, approved, and available set of criteria and operational definitions used to review or audit a work product or process to determine adequacy or compliance. Standards provide the required and recommended contents of a work product.

Standard Checklist Description

The table below lists the contents associated with a standard checklist and contains a description of each subsection.

Name of Subsection	Description
Standard Title	The standard title describes the name of the work product from the CMM.
Definitions	This subsection should define all the terms that may cause confusion. Example: The term "software quality assurance" often has different meanings in different organizations.
Standard Checklist	This subsection contains a checklist for the contents of the work product (standard) required by the CMM. Example: The CMM requires that certain contents be part of a project's software development plan (e.g., estimates of the software project's effort and costs).

Continued on next page

Overview, Continued

What the Standards Checklists are Not

The standards checklists contain only what is required by the CMM, and are not complete standards in themselves. For example, the standard for the software development plan (SDP) contains only CMM requirements, and other important sources for the contents of SDP standards should also be considered such as ANSI/IEEE Std 1058.1-1987, DOD-STD-2167, DI-MCCR-80030, etc.

In This Chapter

This chapter covers the following topics:

Allocated Requirements Standard

Definitions

allocated requirements (system requirements allocated to software): The subset of the system requirements that are to be implemented in the software components of the system. The allocated requirements are a primary input to the software development plan. Software requirements analysis elaborates and refines the allocated requirements and results in software requirements which are documented.

Allocated Requirements Standard Checklist

The following table contains what the CMM describes as the required content of allocated requirements:

√	Required Allocated Requirements Content
	The nontechnical requirements (i.e., the agreements, conditions, and/or contractual terms) that affect and determine the activities of the software project. (L2-4, Ab2, 1)
	The technical requirements for the software. (L2-4, Ab2, 2)
	The acceptance criteria that will be used to validate that the software products satisfy the allocated requirements. (L2-4, Ab2, 3)

Statement of Work Standard

Definitions

statement of work (SOW) - A description of all the work required to complete a project, which is provided by the customer.

SOW Standard Checklist

The following table contains what the CMM describes as the required content of the statement of work:

√	Required SOW Content
	scope of the work (L2-14, Ab1, 1.1)
	technical goals and objectives (L2-14, Ab1, 1.2)
	identification of customers and end users (L2-14, Ab1, 1.3)
	imposed standards (L2-14, Ab1, 1.4)
	assigned responsibilities (L2-14, Ab1, 1.5)
	cost and schedule constraints and goals (L2-15, Ab1, 1.6)
	dependencies between the software project and other organizations (L2-15, Ab1, 1.7)
	resource constraints and goals (L2-15, Ab1, 1.8)
	other constraints and goals for development and/or maintenance (L2-15, Ab1, 1.9)

Software Development Plan Standard

Definitions

software development plan (SDP) - The collection of plans that describe the activities to be performed for the software project. It governs the management of the activities performed by the software engineering group for a software project. It is not limited to the scope of any particular planning standard, such as DOD-STD-2167A and IEEE-STD-1058, which may use similar terminology.

SDP Standard Checklist

The following table contains what the CMM describes as the required content of the Software Development Plan:

√	Required SDP Content
	Software project's purpose, scope, goals, and objectives. (L2-19, A7, 1)
	Selection of a software life cycle. (L2-19, A7, 2)
	Identification of the selected procedures, methods, and standards for developing and/or maintaining the software. (L2-20, A7, 3)
	Identification of software work products to be developed. (L2-20, A7, 4)
	Size estimates of the software work products and any changes to the software work products. (L2-20, A7, 5)
	Estimates of the software project's effort and costs. (L2-20, A7, 6)
	Estimated use of critical computer resources. (L2-20, A7, 7)
	Software project schedules, including identification of milestones and reviews. (L2-20, A7, 8)
	Identification and assessment of the project's software risks. (L2-20, A7, 9)
	Plans for the project's software engineering facilities and support tools. (L2-20, A7, 10)

Contractual Agreement Standard

Definitions

contract terms and conditions - The stated legal, financial, and administrative aspects of a contract.

Contractual Agreement Standard Checklist

The following table contains what the CMM describes as the required content of the Contractual Agreement between the prime contractor and the software subcontractor:

√	Required Contractual Agreement Content
	The terms and conditions. (L2-50, A3, 1)
	The statement of work. (L2-50, A3, 2)
	The requirements for the products to be developed. (L2-50, A3, 3)
	The list of dependencies between the subcontractor and the prime contractor. (L2-50, A3, 4)
	The subcontracted products to be delivered to the prime contractor. (L2-50, A3, 5)
	The conditions under which revisions to products are to be submitted. (L2-50, A3, 6)
	The acceptance procedures and acceptance criteria to be used in evaluating the subcontracted products before they are accepted by the prime contractor. (L2-50, A3, 7)
	The procedures and evaluation criteria to be used by the prime contractor to monitor and evaluate the subcontractor's performance. (L2-51, A3, 8)

Software Quality Assurance Plan Standard

Definitions

software quality assurance (SQA) - (1) A planned and systematic pattern of all actions necessary to provide adequate confidence that a software work product conforms to established technical requirements. (2) A set of activities designed to evaluate the process by which software work products are developed and/or maintained.

SQA Plan Standard Checklist

The following table contains what the CMM describes as the required content of the Software Quality Assurance Plan:

√	Required SQA Plan Content
	Responsibilities and authority of the SQA group. (L2-64, A2, 1)
	Resource requirements for the SQA group (including staff, tools, and facilities). (L2-64, A2, 2)
	Schedule and funding of the project's SQA group activities. (L2-64, A2, 3)
	The SQA group's participation in establishing the software development plan, standards, and procedures for the project. (L2-65, A2, 4)
	Evaluations to be performed by the SQA group. (L2-65, A2, 5)
	Audits and reviews to be conducted by the SQA group. (L2-65, A2, 6)
	Project standards and procedures used as the basis for the SQA group's reviews and audits. (L2-65, A2, 7)
	Procedures for documenting and tracking noncompliance issues to closure. (L2-65, A2, 8)
	Documentation that the SQA group is required to produce. (L2-65, A2, 9)
	Method and frequency of providing feedback to the software engineering group and other software-related groups on SQA activities. (L2-65, A2, 10)

Software Configuration Management Plan Standard

Definitions

configuration management or software configuration management (SCM) - A discipline applying technical and administrative direction and surveillance to identify and document the functional and physical characteristics of a configuration item, control changes to those characteristics, record and report change processing and implementation status, and verify compliance with specified requirements. [IEEE-STD-610]

SCM Plan Standard Checklist

The following table contains what the CMM describes as the required content of the Software Configuration Management Plan:

√	Required SCM Plan Content
	The SCM activities to be performed, the schedule of activities, the assigned responsibilities, and the resources required (including staff, tools, and computer facilities). (L2-77, A2, 1)
	The SCM requirements and activities to be performed by the software engineering group and other software-related groups. (L2-77, A2, 2)

Chapter 5

Process Checklists

Overview

Chapter Purpose

The purpose of the process checklists is to provide:

- guidance in identifying which processes are required by the CMM.
- criteria that an organization can use to evaluate its software processes to determine if they are consistent with the CMM.
- information that can be used to develop software processes that are consistent with the CMM.

Chapter Definitions

process: A process is a series of events or phases that takes place over time and has an identifiable purpose or result. A process has entry criteria, inputs, activities, exit criteria, outputs, roles, etc.

In This Chapter

Requirements Management (RM) Process

RM Process - Overview

RM Process Purpose

The purpose of Requirements Management is to establish a common understanding between the customer and the software project of the customer's requirements that will be addressed by the software project. (L2-1)

RM Process Description

Requirements Management involves establishing and maintaining an agreement with the customer on the requirements for the software project. This agreement is referred to as the "system requirements allocated to the software." The "customer" may be interpreted as the system engineering group, the marketing group, another internal organization, or an external customer. The agreement covers both the technical and nontechnical (e.g., delivery dates) requirements. The agreement forms the basis for estimating, planning, performing, and tracking the software project's activities throughout the software life cycle.

The allocation of the system requirements to software, hardware, and other system components (e.g., humans) may be performed by a group external to the software engineering group (e.g., the system engineering group), and the software engineering group may have no direct control of this allocation. Within the constraints of the project, the software engineering group takes appropriate steps to ensure that the system requirements allocated to software, which they are responsible for addressing, are documented and controlled.

To achieve this control, the software engineering group reviews the initial and revised system requirements allocated to software to resolve issues before they are incorporated into the software project. Whenever the system requirements allocated to software are changed, the affected software plans, work products, and activities are adjusted to remain consistent with the updated requirements. (L2-1)

Continued on next page

Chapter Overview

The table below contains a description and the location of each section in this chapter.

RM Process - Roles

Roles

The table below lists the roles, and the activities in which they participate in the requirements management process.

√	Role	Activities Participated in...	Reference
	Affected Groups	❑ The allocated requirements are reviewed by: (L2-3, C1, 2) ❑ the software managers, and ❑ other affected groups. ❑ Commitments resulting from the allocated requirements are negotiated with the affected groups. (L2-6, A1, 4) ❑ Changes to commitments within the organization are negotiated with the affected groups. (L2-7, A3, 1.2) ❑ Changes that need to be made to the software plans, work products, and activities resulting from changes to the allocated requirements are: (L2-8, A3, 2) ❑ identified, ❑ evaluated, ❑ assessed for risk, ❑ documented, ❑ planned, ❑ communicated to the affected groups and individuals, and ❑ tracked to completion. ❑ Changes to commitments resulting from changes to the allocated requirements are negotiated with the affected groups. (L2-10, V3, 3)	

Continued on next page

RM Process - Roles, Continued

Roles, continued The table below lists the roles, and the activities in which they participate in the requirements management process, *continued from the previous page.*

√	Role	Activities Participated in...	Reference
	(Affected) Individuals	❑ **Individuals who have experience and expertise in the application domain and in software engineering are assigned to manage the allocated requirements.** (L2-5, Ab3, 1) ❑ Changes that need to be made to the software plans, work products, and activities resulting from changes to the allocated requirements are: (L2-8, A3, 2) ❑ identified, ❑ evaluated, ❑ assessed for risk, ❑ documented, ❑ planned, ❑ communicated to the affected groups and individuals, and ❑ tracked to completion.	
	Group Responsible for Analyzing and Allocating System Requirements	Any allocated requirements identified as having potential problems are reviewed with the group responsible for analyzing and allocating system requirements, and necessary changes are made. (L2-6, A1, 3)	
	Individuals and Groups External to the Organization	Changes to commitments made to individuals and groups external to the organization are reviewed by senior management. (L2-7, A3, 1.1)	
	Project Manager	The activities for managing the allocated requirements are reviewed with the project manager on both a periodic and event-driven basis. (L2-9, V2)	

Continued on next page

RM Process - Roles, Continued

Roles, continued The table below lists the roles, and the activities in which they participate in the requirements management process, continued from the previous page.

√	Role	Activities Participated in...	Reference
	Senior Management	❑ Changes to commitments made to individuals and groups external to the organization are reviewed by senior management. (L2-7, A3, 1.1) ❑ The activities for managing the allocated requirements are reviewed with senior management on a periodic basis. (L2-9, V1)	
	Software Engineering Group	❑ Members of the software engineering group and other software-related groups are trained to perform their requirements management activities. (L2-5, Ab4) ❑ The software engineering group reviews the allocated requirements before they are incorporated into the software project. (L2-5, A1) ❑ The software engineering group uses the allocated requirements as the basis for software plans, work products, and activities. (L2-6, A2) ❑ The allocated requirements are reviewed, and problems are resolved before the software engineering group commits to them. (L2-10, V3, 1)	
	Software Manager	The allocated requirements are reviewed by: (L2-3, C1, 2) ❑ the software managers, and ❑ other affected groups.	
	Software-related Groups	Members of the software engineering group and other software-related groups are trained to perform their requirements management activities. (L2-5, Ab4)	
	SQA Group	The software quality assurance group reviews and/or audits the activities and work products for managing the allocated requirements and reports results. (L2-9, V3)	

RM Process - Entry Criteria

Entry Criteria

The table below describes the conditions that must be satisfied in order to begin the requirements management process.

√	Condition	References
	The project follows a written organizational policy for managing the system requirements allocated to software. (L2-2, C1) [Refer to SPF Policies for additional information regarding RM policy.]	
	Allocated requirements are documented. (L2-3, C1, 1)	
	For each project, responsibility is established for analyzing the system requirements and allocating them to hardware, software, and other system components. (L2-3, Ab1) This responsibility covers: ❑ Managing and documenting the system requirements and their allocation throughout the project's life. ❑ Effecting changes to the system requirements and their allocation.	
	Adequate resources and funding are provided for managing the allocated requirements. (L2-5, Ab3)	
	Individuals who have experience and expertise in the application domain and in software engineering are assigned to manage the allocated requirements. (L2-5, Ab3, 1)	
	Tools to support the activities for managing requirements are made available. (L2-5, Ab3, 2)	
	Members of the software engineering group and other software-related groups are trained to perform their requirements management activities. (L2-5, Ab4)	

RM Process - Inputs

Inputs

The table below lists the inputs to the requirements management process.

√	Input	Org. Input	References
	Allocated requirements. (L2-2, C1) [Refer to SPF Standards for additional information regarding allocated requirements.]		
	Changes to the allocated requirements. (L2-4, Ab1, 2)		
	Existing commitments. (L2-7, A3, 1)		
	Software activities. (L2-3, C1, 3)		
	Software plans. (L2-3, C1, 3)		
	Software work products. (L2-3, C1, 3)		
	System requirements. (L2-3, Ab1)		

RM Process - Activities

Activities

The table below lists the required activities for the requirements management process.

√	Activities	References
	The software engineering group reviews allocated requirements before they are incorporated into the software project. (L2-5, A1) [Refer to Reviews and Audits for additional information.]	
	Commitments resulting from the allocated requirements are negotiated with the affected groups. (L2-6, A1,4)	
	The software engineering group uses the allocated requirements as the basis for software plans, work products, and activities. (L2-6, A2)	
	Changes to the allocated requirements are reviewed and incorporated into the software project. (L2-7, A3) ❑ The impact to existing commitments is assessed and changes are negotiated as appropriate. ❑ Changes to commitments made to individuals and groups external to the organization are reviewed with senior management. ❑ Changes to commitments within the organization are negotiated with the affected groups. ❑ Changes that need to be made to the software plans, work products, and activities resulting from changes to the allocated requirements are: ❑ identified, ❑ evaluated, ❑ assessed for risk, ❑ documented, ❑ planned, ❑ communicated to the affected groups and individuals, and ❑ tracked to completion.	
	Measurements are made and used to determine the status of the activities for managing the allocated requirements. (L2-8, M1)	
	The activities for managing requirements are reviewed with senior management on a periodic basis. (L2-9, V1)	

Continued on next page

RM Process - Activities, Continued

Activities, continued

The table below lists the required activities for the requirements management process, continued from the previous page.

√	Activities	References
	The activities for managing the allocated requirements are reviewed with the project manager on both a periodic and event-driven basis. (L2-9, V2)	
	The software quality assurance group reviews and/or audits the activities and work products for managing the allocated requirements and reports the results. (L2-9, V3) [Refer to Reviews and Audits for additional information.]	

RM Process - Outputs

Outputs

The table below lists the outputs produced by the requirements management process.

√	Output	Org. Output	References
	Allocated requirements. (L2-5, A1)		
	Changes that need to be made to the software plans, work products, and activities resulting from changes to the allocated requirements. (L2-8, A3, 2)		
	Changes to allocated requirements. (L2-7, A3)		
	Changes to commitments made to individuals and groups external to the organization. (L2-7, A3, 1.1)		
	Changes to commitments within the organization. (L2-7, A3, 1.2)		
	Commitments resulting from the allocated requirements. (L2-6, A1, 4)		
	Measurements. (L2-8, M1)		
	Software activities. (L2-10, V3, 2)		
	Software plans. (L2-10, V3, 2)		
	Software requirements. (L2-7, A2, 3)		
	Software work products. (L2-10, V3, 2)		

RM Process - Exit Criteria

Exit Criteria

The table below describes the conditions that must be satisfied in order to exit the requirements management process.

√	Condition	References
	The allocated requirements are documented. (L2-3, C1, 1)	
	Allocated requirements are reviewed by: (L2-3, C1, 2) ❑ the software managers, and ❑ other affected groups.	
	The software plans, work products, and activities are changed to be consistent with changes to the allocated requirements. (L2-3, C1, 3)	
	The software engineering group reviews the allocated requirements before they are incorporated into the software project. (L2-5, A1)	
	Incomplete and missing allocated requirements are identified. (L2-5, A1, 1)	
	The allocated requirements are reviewed to determine whether they are: (L2-6, A1, 2) ❑ feasible and appropriate to implement in software, ❑ clearly and properly stated, ❑ consistent with each other, and ❑ testable.	
	Any allocated requirements identified as having potential problems are reviewed with the group responsible for analyzing and allocating system requirements, and necessary changes are made. (L2-6, A1, 3)	
	Commitments resulting from the allocated requirements are negotiated with the affected groups. (L2-6, A1, 4)	
	The allocated requirements are the basis for the software development plan. (L2-7, A2, 2)	
	The allocated requirements are the basis for the software requirements. (L2-7, A2, 3)	
	Changes to allocated requirements are reviewed and incorporated into the software project. (L2-7, A3)	
	The impact of existing commitments is assessed, and changes are negotiated as appropriate. (L2-7, A3, 1) ❑ Changes to commitments made to individuals and groups external to the organization are reviewed with senior management. ❑ Changes to commitments within the organization are negotiated with the affected groups.	

Continued on next page

RM Process - Exit Criteria, Continued

Exit Criteria, continued

The table below describes the conditions that must be satisfied in order to exit the requirements management process, continued from the previous page.

√	Condition	References
	Changes that need to be made to the software plans, work products, and activities resulting from changes to the allocated requirements are: (L2-8, A3, 2) ❑ identified, ❑ evaluated, ❑ assessed for risk, ❑ documented, ❑ planned, ❑ communicated to the affected groups and individuals, and ❑ tracked to completion.	
	Measurements are made and used to determine the status of the activities for managing the allocated requirements. (L2-8, M1)	
	The activities for managing the allocated requirements are reviewed with senior management on a periodic basis. (L2-9, V1)	
	The activities for managing the allocated requirements are reviewed with the project manager on both a periodic and event-driven basis. (L2-9, V2)	
	The software quality assurance group reviews and/or audits the project's activities and work products for managing the allocated requirements and reports the results. (L2-9, V3)	
	The allocated requirements are reviewed, and problems are resolved before the software engineering group commits to them. (L2-10, V3, 1)	

RM Process - Reviews and Audits

Reviews and Audits

The table below lists the required reviews and audits for the requirements management process.

√	Review or Audit	Review Participants	References
	Allocated requirements are reviewed by: (L2-3, C1, 2) ❑ the software managers, and ❑ other affected groups.	Software Managers Affected Groups	
	The software engineering group reviews allocated requirements before they are incorporated into the software project. (L2-5, A1) ❑ Incomplete and missing allocated requirements are identified. ❑ The allocated requirements are reviewed to determine whether they are: ❑ feasible and appropriate to implement in software, ❑ clearly and properly stated, ❑ *consistent* with each other, and ❑ testable. ❑ Any allocated requirements identified as having potential problems are reviewed with the group responsible for analyzing and allocating system requirements, and necessary changes are made.	Software Engineering Group Group Responsible for Analyzing and Allocating System Requirements	
	Changes to the allocated requirements are reviewed and incorporated into the software project. (L2-7, A3)	Not specified in CMM	
	Changes to commitments made to individuals and groups external to the organization are reviewed with senior management. (L2-7, A3, 1.1)	Senior Management	

Continued on next page

RM Process - Reviews and Audits, Continued

Reviews and Audits, continued

The table below lists the required reviews and audits for the requirements management process, continued from the previous page.

√	Review or Audit	Review Participants	References
	The activities for managing requirements are reviewed with senior management on a periodic basis. (L2-9, V1)	Senior Management	
	The activities for managing the allocated requirements are reviewed with the project manager on both a periodic and event-driven basis. (L2-9, V2)	Project Manager	
	The software quality assurance group reviews and/or audits the activities and work products for managing the allocated requirements and reports the results. (L2-9, V3) At a minimum, these reviews and/or audits verify that: ❑ The allocated requirements are reviewed, and problems are resolved before the software engineering group commits to them. ❑ The software plans, work products, and activities are appropriately revised when the allocated requirements change. ❑ Changes to commitments resulting from changes to the allocated requirements are negotiated with the affected groups.	SQA Group Software Engineering Group Affected Groups	

RM Process - Work Products Managed and Controlled

Work Products Managed and Controlled

The table below lists the work products that are required to be managed and controlled during the requirements management process.

√	Work Products Managed and Controlled	References
	Allocated requirements. (L2-7, A2, 1)	

RM Process - Measurements

Measurements

The table below describes the measurements required for the requirements management process.

√	Measurements	References
	Measurements are made and used to determine the status of the activities for managing the allocated requirements. (L2-8, M1)	

RM Process - Documented Procedures

Documented Procedures	There are no activities required to be performed according to a documented procedure for the requirements management process.

RM Process - Training

Training

The table below lists the training required for the requirements management process.

√	Training	References
	Members of the software engineering group and other software-related groups are trained to perform their requirements management activities. (L2-5, Ab4)	

RM Process - Tools

Tools

The table below lists the tools required for the requirements management process.

√	Tools	References
	Tools to support the activities for managing requirements. (L2-5, Ab3, 2)	

Software Project Planning (SPP) Process

SPP Process - Overview

SPP Process Purpose

The purpose of Software Project Planning is to establish reasonable plans for performing the software engineering and for managing the software project. (L2-11)

SPP Process Description

Software Project Planning involves developing estimates for the work to be performed, establishing the necessary commitments, and defining the plan to perform the work.

The software planning begins with a statement of the work to be performed and other constraints and goals that define and bound the software project (those established by the practices of the Requirements Management key process area). The software planning process includes steps to estimate the size of the software work products and the resources needed, produce a schedule, identify and assess software risks, and negotiate commitments. Iterating through these steps may be necessary to establish the plan for the software project (i.e., the software development plan).

This plan provides the basis for performing and managing the software project's activities and addresses the commitments to the software project's customer according to the resources, constraints, and capabilities of the software project. (L2-11)

Continued on next page

Chapter Overview

The table below contains the description and location of each section in this chapter.

Section	Description	Page
Roles	List of roles participating in process activities.	SPP-3
Entry Criteria	Describes when the process can start.	SPP-12
Inputs	A description of the work products consumed by the process.	SPP-13
Activities	Describes the activities of the process.	SPP-14
Outputs	A description of the work products produced by the process.	SPP-16
Exit Criteria	Describes when the process is complete.	SPP-18
Reviews and Audits	List of required reviews and audits.	SPP-20
Work Products Managed and Controlled	Lists work products required to be managed and controlled.	SPP-23
Measurements	Describes required process measurements.	SPP-24
Documented Procedures	Lists which activities must be completed according to a documented procedure.	SPP-25
Training	List of required training.	SPP-26
Tools	List of required tools.	SPP-27

SPP Process - Roles

Roles

The table below lists the roles, and the activities in which they participate in the software project planning process.

√	Role	Activities Participated in...	Reference
	Affected Groups	❑ Affected groups review the software project's: (L2-13, C2, 4) ❑ software size estimates, ❑ effort and cost estimates, ❑ schedules, and ❑ other commitments. ❑ The statement of work is reviewed by: (L2-15, Ab1, 2) ❑ the project manager, ❑ the project software manager, ❑ the other software managers, and ❑ other affected groups. ❑ The software engineering group participates with other affected groups in the overall project planning throughout the project's life. (L2-17, A3) ❑ The software development plan is reviewed by: (L2-19, A6, 4) ❑ the project manager, ❑ the project software manager, ❑ the other software managers, and ❑ other affected groups. ❑ The plans for the project's software engineering facilities and support tools are reviewed by all affected groups. (L2-25, A14, 3) ❑ A summary report from each review with senior management is prepared and distributed to the affected groups and individuals. (L2-26, V1, 5)	
	Role continues on the next page		

Continued on next page

Roles, continued

The table below lists the roles, and the activities in which they participate in the software project planning process, continued from the previous page.

√	Role	Activities Participated in...	Reference
	Affected Groups, continued	❑ The activities for software project planning are reviewed with the project manager on both a periodic and event-driven basis. (L2-26, V2) ❑ **Affected groups are represented.** ❑ A summary report from each review with the project manager is prepared and distributed to the **affected groups** and individuals. (L2-27, V2, 7)	
	(Affected) Individuals	❑ Where feasible, experienced **individuals**, who have expertise in the application domain of the software project being planned, are available to develop the software development plan. (L2-16, Ab3, 1) ❑ The software managers, software engineers, and other **individuals** involved in the software project planning are trained in the software estimating and planning procedures applicable to their areas of responsibility. (L2-16, Ab4,) ❑ A summary report from each review with senior management is prepared and distributed to the affected groups and **individuals**. (L2-26, V1, 5) ❑ A summary report from each review with the project manager is prepared and distributed to the affected groups and **individuals**. (L2-27, V2, 7)	

Continued on next page

SPP Process - Roles, Continued

Roles, continued

The table below lists the roles, and the activities in which they participate in the software project planning process, continued from the previous page.

√	Role	Activities Participated in...	Reference
	Engineering Groups	❑ Involvement of other engineering groups in the software activities is negotiated with these groups and is documented. (L2-13, C2, 3) ❑ Plans for software-related groups and other engineering groups involved in the activities of the software engineering group are negotiated with those groups, the support efforts are budgeted, and the agreements are documented. (L2-18, A6, 2) ❑ Plans for involvement of the software engineering group in the activities of other software-related groups and other engineering groups are negotiated with those groups, the support efforts are budgeted, and the agreements are documented. (L2-19, A6, 3)	
	Individuals and Groups External to the Organization	❑ Senior management reviews all software project commitments made to individuals and groups external to the organization. (L2-13, C2, 5) ❑ Software project commitments made to individuals and groups external to the organization are reviewed with senior management according to a documented procedure. (L2-17, A5)	

Continued on next page

Roles, continued

The table below lists the roles, and the activities in which they participate in the software project planning process, continued from the previous page.

√	Role	Activities Participated in...	Reference
	Project Manager	❑ The software project's commitments are negotiated between: (L2-12, C2, 2) ❑ the project manager, ❑ the project software manager, and ❑ the other software managers. ❑ The statement of work is reviewed by: (L2-15, Ab1, 2) ❑ the project manager, ❑ the project software manager, ❑ the other software managers, and ❑ other affected groups. ❑ The software development plan is reviewed by: (L2-19, A6, 4) ❑ the project manager, ❑ the project software manager, ❑ the other software managers, and ❑ other affected groups. ❑ The activities for software project planning are reviewed with the project manager on both a periodic and event-driven basis. (L2-26, V2)	

Continued on next page

SPP Process - Roles, Continued

Roles, continued

The table below lists the roles, and the activities in which they participate in the software project planning process, continued from the previous page.

√	Role	Activities Participated in...	Reference
	Project Software Manager	❑ A **project software manager** is designated to be responsible for negotiating commitments and developing the project's software development plan. (L2-12, C1) ❑ The software project's commitments are negotiated between: (L2-12, C2, 2) ❑ the project manager, ❑ the **project software manager**, and ❑ the other software managers. ❑ The statement of work is reviewed by: (L2-15, Ab1, 2) ❑ the project manager, ❑ the **project software manager**, ❑ the other software managers, and ❑ other affected groups. ❑ The **project software manager**, directly or by delegation, coordinates the project's software planning. (L2-15, Ab2, 1) ❑ The software development plan is reviewed by: (L2-19, A6, 4) ❑ the project manager, ❑ the **project software manager**, ❑ the other software managers, and ❑ other affected groups.	

Continued on next page

SPP Process - Roles, Continued

Roles, continued

The table below lists the roles, and the activities in which they participate in the software project planning process, continued from the previous page.

√	Role	Activities Participated in...	Reference
	Senior Management	❑ Senior management reviews all software project commitments made to individuals and groups external to the organization. (L2-13, C2, 5) ❑ Software project commitments made to individuals and groups external to the organization are reviewed with senior management according to a documented procedure. (L2-17, A4) ❑ The activities for software project planning are reviewed with senior management on a periodic basis. (L2-26, V1)	

	Software Engineering Group	❑ The software engineering group participates on the project proposal team. (L2-16, A1)
		❑ The software engineering group is involved in: (L2-17, A1, 1)
		  ❑ proposal preparation and submission,
		  ❑ clarification discussions and submissions, and
		  ❑ negotiations of changes to commitments that affect the software project.
		❑ The software engineering group reviews the project's proposed commitments. (L2-17, A1, 2)	

	Role continues on the next page		

Continued on next page

Roles, continued

The table below lists the roles, and the activities in which they participate in the software project planning process, continued from the previous page.

√	Role	Activities Participated in...	Reference
	Software Engineering Group, continued	❑ The software engineering group participates with other affected groups in the overall project planning throughout the project's life. (L2-17, A3) ❑ The software engineering group reviews the project-level plans. (L2-17, A3, 1) ❑ Plans for software-related groups and other engineering groups involved in the activities of the software engineering group are negotiated with those groups, the support efforts are budgeted, and the agreements are documented. (L2-18, A6, 2) ❑ Plans for involvement of the software engineering group in the activities of other software-related groups and other engineering groups are negotiated with those groups, the support efforts are budgeted, and the agreements are documented. (L2-19, A6, 3)	
	Software Engineers	The software managers, software engineers, and other individuals involved in the software project planning are trained in software estimating and planning procedures applicable to their areas of responsibility. (L2-16, Ab4)	

Continued on next page

SPP Process - Roles, Continued

Roles, continued

The table below lists the roles, and the activities in which they participate in the software project planning process, continued from the previous page.

√	Role	Activities Participated in...	Reference
	Software Manager	❑ The software project's commitments are negotiated between: (L2-12, C2, 2) ❑ the project manager, ❑ the project software manager, and ❑ the other software managers. ❑ The statement of work is reviewed by: (L2-15, Ab1, 2) ❑ the project manager, ❑ the project software manager, ❑ the other software managers, and ❑ other affected groups. ❑ Responsibilities for the software work products and activities are partitioned and assigned to software managers in a traceable, accountable manner. (L2-15, Ab2, 2) ❑ The software managers, software engineers, and other individuals involved in the software project planning are trained in software estimating and planning procedures applicable to their areas of responsibility. (L2-16, Ab4) ❑ The software development plan is reviewed by: (L2-19, A6, 4) ❑ the project manager, ❑ the project software manager, ❑ the other software managers, and ❑ other affected groups.	

Continued on next page

SPP Process - Roles, Continued

Roles, continued

The table below lists the roles, and the activities in which they participate in the *software project planning process*, continued from the previous page.

√	Role	Activities Participated in...	Reference
	Software-related Groups	❑ Plans for software-related groups and other engineering groups involved in the activities of the software engineering group are negotiated with those groups, the support efforts are budgeted, and the agreements are documented. (L2-18, A6, 2) ❑ Plans for involvement of the software engineering group in the activities of other software-related groups and other engineering groups are negotiated with those groups, the support efforts are budgeted, and the agreements are documented. (L2-19, A6, 3)	
	SQA Group	The software quality assurance group reviews and/or audits the activities and work products for software project planning and reports the results. (L2-27, V3)	

SPP Process - Entry Criteria

Entry Criteria The table below describes the conditions that must be satisfied in order to begin the software project planning process.

√	Condition	References
	A project software manager is designated to be responsible for negotiating commitments and developing the project's software development plan. (L2-12, C1)	
	The project follows a written organizational policy for planning a software project. (L2-12, C2) [Refer to SPF Policies for additional information regarding SPP policy.]	
	A documented and approved statement of work exists for the software project. (L2-14, Ab1)	
	The statement of work is reviewed by: (L2-15, Ab1, 2) ❑ the project manager, ❑ the project software manager, ❑ the other software managers, and ❑ other affected groups.	
	The statement of work is managed and controlled. (L2-15, Ab1, 3)	
	Responsibilities for developing the software development plan are assigned. (L2-15, Ab2)	
	Responsibilities for the software work products and activities are partitioned and assigned to software managers in a traceable, accountable manner. (L2-15, Ab2, 2)	
	Adequate resources and funding are provided for planning the software project. (L2-16, Ab3)	
	Where feasible, experienced individuals who have expertise in the application domain of the software project being planned are available to develop the software development plan. (L2-16, Ab3, 1)	
	Tools to support the software project planning activities are made available. (L2-16, Ab3, 2)	
	The software managers, software engineers, and other individuals involved in the software project planning are trained in the software estimating and planning procedures applicable to their areas of responsibility. (L2-16, Ab4)	
	Software project planning in initiated in the early stages of, and in parallel with, the overall project planning. (L2-17, A2)	

SPP Process - Inputs

Inputs

The table below lists the inputs to the software project planning process.

√	Input	Org. Input	References
	Allocated requirements. (L2-18, A6, 1.4) [Refer to SPF Standards for additional information regarding allocated requirements.]		
	Cost data. (L2-22, A10, 2.1)		
	Customer's standards. (L2-18, A6, 1.1)		
	Historical data (where available). (L2-21, A9, 3)		
	Productivity data (historical and/or current). (L2-22, A10, 2)		
	Project proposal. (L2-16, A1)		
	Project's standards. (L2-18, A6, 1.2)		
	Proposed commitments. (L2-17, A1, 2)		
	Statement of Work. (L2-14, Ab1) [Refer to SPF Standards for additional information regarding a Statement of Work.]		

SPP Process - Activities

Activities

The table below lists the required activities for the software project planning process.

√	Activities	References
	The software engineering group participates on the project proposal team. (L2-16, A1) ❑ The software engineering group is involved in: (L2-17, A1, 1) ❑ proposal preparation and submission, ❑ clarification discussions and submissions, and ❑ negotiations of changes to commitments that affect the software project. ❑ The software engineering group reviews the project's proposed commitments. (L2-17, A1, 2)	
	The software engineering group reviews the project-level plans. (L2-17, A3, 1)	
	Software project commitments made to individuals and groups external to the organization are reviewed with senior management according to a documented procedure. (L2-17, A4)	
	A software life cycle with predefined stages of manageable size is identified or defined. (L2-17, A5)	
	The project's software development plan is developed according to a documented procedure. (L2-18, A6) [Refer to Documented Procedures for additional information.]	
	The plan for the software project is documented. (L2-19, A7) [Refer to SPF Standards for additional information regarding a software development plan.]	
	Software work products that are needed to establish and maintain control of the software project are identified. (L2-20, A8)	
	Estimates for the size of the software work products (or changes to the size of software work products) are derived according to a documented procedure. (L2-21, A9) [Refer to Documented Procedures for additional information.]	
	Estimates for the software project's effort and costs are derived according to a documented procedure. (L2-22, A10) [Refer to Documented Procedures for additional information.]	

Continued on next page

Activities, continued

The table below describes the activities associated with the software project planning process, continued from the previous page.

√	Activities	References
	Estimates for critical computer resources are derived according to a documented procedure. (L2-23, A11) [Refer to Documented Procedures for additional information.]	
	The project's software schedule is derived according to a documented procedure. (L2-23, A12) [Refer to Documented Procedures for additional information.]	
	The software risks associated with the cost, resource, schedule and technical aspects of the project are identified, assessed, and documented. (L2-24, A13) ❑ The risks are analyzed and prioritized based on their potential impact to the project. ❑ Contingencies for the risks are identified.	
	Plans for the project's software engineering facilities and support tools are prepared. (L2-25, A14) ❑ Responsibilities are assigned and commitments are negotiated to procure or develop these facilities and support tools. (L2-25, A14, 2) ❑ The plans are reviewed by all affected groups. (L2-25, A14, 3)	
	Software planning data are recorded. (L2-25, A15) ❑ Information recorded includes the estimates and the associated information needed to reconstruct the estimates and assess their reasonableness. (L2-25, A15, 1)	
	Measurements are made and used to determine the status of the software planning activities. (L2-25, M1)	
	The activities for software project planning are reviewed with senior management on a periodic basis. (L2-26, V1) [Refer to Reviews and Audits for additional information.]	
	The activities for software project planning are reviewed with the project manager on both a periodic and event-driven basis. (L2-26, V2) [Refer to Reviews and Audits for additional information.]	
	The software quality assurance group reviews and/or audits the activities and work products for software planning and reports the results. (L2-27, V3) [Refer to Reviews and Audits for additional information.]	

SPP Process - Outputs

Outputs

The table below lists the outputs produced by the software project planning process.

√	Output	Org. Outputs	References
	Action items resulting from reviews with senior management. (L2-26, V1, 4)		
	Action items resulting from reviews with the project manager. (L2-27, V2, 6)		
	Assumptions made in deriving the estimates for the software project's effort and costs. (L2-23, A10, 4)		
	Assumptions made in deriving the project's software schedule. (L2-24, A12, 5)		
	Contingencies for the risks associated with the cost, resource, schedule, and technical aspects of the project. (L2-24, A13, 2)		
	Distributions of effort, staffing, and cost estimates over the software life cycle. (L2-23, A10, 3.3)		
	Estimates for the project's critical computer resources. (L2-23, A11)		
	Estimates for the size of the software work products (or changes to the size of software work products). (L2-21, A9)		
	Estimates for the software project's effort and costs. (L2-22, A10)		
	Estimates of capacity requirements for the project's software engineering facilities and support tools. (L2-25, A14, 1)		
	Estimates of the critical computer resources for the project. (L2-23, A11, 3)		
	Measurements to determine the status of the software planning activities. (L2-25, M1)		
	Plans for involvement of the software engineering group in the activities of other software-related groups and other engineering groups. (L2-19, A6, 3)		
	Plans for software-related groups and other engineering groups involved in the activities of the software engineering group. (L2-18, A6, 2)		

Continued on next page

SPP Process - Outputs, Continued

Outputs, continued

The table below lists the outputs produced by the software project planning process, continued from the previous page.

√	Output	Org. Outputs	References
	Plans for the project's software engineering facilities and support tools. (L2-25, A14)		
	Project's Software Development Plan (SDP). (L2-15, Ab2) [Refer to SPF Standards for additional information regarding the software development Plan.]		
	Project's software schedule. (L2-23, A12)		
	Size estimating assumptions. (L2-21, A9, 4)		
	Software life cycle. (L2-17, A5)		
	Software planning data. (L2-25, A15)		
	Software project commitments. (L2-17, A4)		
	Software risks associated with the cost, resource, schedule, and technical aspects of the project. (L2-24, A13)		
	Software work products that are needed to establish and maintain control of the software project. (L2-20, A8)		
	Sources and rationale for productivity data used during estimating the software project's effort and costs. (L2-22, A10, 2)		
	Summary report from each review with senior management. (L2-26, V1, 5)		
	Summary report from each review with the project manager. (L2-27, V2, 7)		
	Time phasing of activities. (L2-22, A10, 3.2)		

SPP Process - Exit Criteria

Exit Criteria

The table below describes the conditions that must be satisfied in order to exit the software project planning process.

√	Condition	References
	The system requirements allocated to software are used as the basis for planning the software project. (L2-12, C2, 1)	
	The software project's commitments are negotiated between: (L2-12, C2, 2) ❑ the project manager, ❑ the project software manager, and ❑ the other software managers.	
	Involvement of other engineering groups in the software activities is negotiated with these groups and is documented. (L2-13, C2, 3)	
	Affected groups review the project's: (L2-13, C2, 4) ❑ software size estimates, ❑ effort and cost estimates, ❑ schedules, and ❑ other commitments.	
	Senior management reviews all software project commitments made to individuals and groups external to the organization. (L2-13, C2, 5)	
	The project's software development plan is managed and controlled. (L2-13, C2, 6)	
	The software engineering group participates on the project proposal team. (L2-16, A1)	
	The software engineering group reviews the project-level plans. (L2-17, A3, 1)	
	A software life-cycle with predefined stages of manageable size is identified or defined. (L2-17, A5)	
	The project's software development plan is developed according to a documented procedure. (L2-18, A6)	
	The plan for the software project is documented. (L2-19, A7)	
	Software work products that are needed to establish and maintain control of the software project are identified. (L2-20, A8)	
	Estimates for the size of the software work products (or changes to the size of software work products) are derived according to a documented procedure. (L2-21, A9)	

Continued on next page

SPP Process - Exit Criteria, Continued

Exit Criteria, continued

The table below describes the conditions that must be satisfied in order to exit the software project planning process, continued from the previous page.

√	Condition	References
	Estimates for the software project's effort and costs are derived according to a documented procedure. (L2-22, A10)	
	Estimates for the project's critical computer resources are derived according to a documented procedure. (L2-23, A11)	
	Project's software schedule is derived according to a documented procedure. (L2-23, A12)	
	The software risks associated with the cost, resource, schedule, and technical aspects of the project are identified, assessed, and documented. (L2-24, A13) ❑ The risks are analyzed and prioritized based on their potential impact to the project. ❑ Contingencies for the risks are identified.	
	Plans for the project's software engineering facilities, and support tools are prepared. (L2-25, A14) ❑ Estimates of capacity requirements for these facilities and support tools are based on the size estimates of the software work products and other characteristics. ❑ Responsibilities are assigned and commitments are negotiated to procure or develop the project's facilities and tools. ❑ The plans are reviewed by all affected groups.	
	Software planning data are recorded. (L2-25, A15) ❑ Information recorded includes the estimates and the associated information needed to reconstruct the estimates and assess their reasonableness. (L2-25, A15, 1)	
	Measurements are made and used to determine the status of the software planning activities. (L2-25, M1)	
	The activities for software project planning are reviewed with senior management on a periodic basis. (L2-26, V1)	
	The activities for software project planning are reviewed with the project manager on both a periodic and event-driven basis. (L2-26, V2)	
	The software quality assurance group reviews and/or audits the activities and work products for software project planning, and reports the results. (L2-27, V3)	

SPP Process - Reviews and Audits

Reviews and Audits

The table below lists the required reviews and audits for the software project planning process.

√	Review or Audit	Review Participants	References
	Affected groups review the software project's: (L2-13, C2, 4) ❑ software size estimates, ❑ effort and cost estimates, ❑ schedules, and ❑ other commitments.	**Affected Groups**	
	Senior management reviews all software project commitments made to **individuals and groups external to the organization.** (L2-13, C2, 5)	**Senior Management**	
	The statement of work is reviewed by: (L2-15, Ab1, 2) ❑ the project manager, ❑ the project software manager, ❑ the other software managers, and ❑ other affected groups.	**Project Manager** **Project Software Manager** **Software Managers** **Affected Groups**	
	The software engineering group reviews the project's proposed commitments. (L2-17, A1, 2)	**Software Engineering Group**	
	The software engineering group reviews the project-level plans. (L2-17, A3, 1)	**Software Engineering Group**	
	Software project commitments made to **individuals and groups external to the organization** are reviewed with **senior management.** (L2-17, A4)	**Senior Management**	
	The software development plan is reviewed by: (L2-19, A6, 4) ❑ the project manager, ❑ the project software manager, ❑ the other software managers, and ❑ other affected groups.	**Project Manager** **Project Software Manager** **Software Managers** **Affected Groups**	

Continued on next page

Reviews and Audits, continued

The table below lists the required reviews and audits for the software project planning process, continued from the previous page.

√	Review or Audit	Review Participants	References
	Size estimates are documented, reviewed, and agreed to. (L2-21, A9, 5)	Not specified in the CMM	
	Estimates and the assumptions made in deriving the estimates are documented, reviewed, and agreed to. (L2-23, A10, 4)	Not specified in the CMM	
	Estimates of the critical computer resources are documented, reviewed, and agreed to. (L2-23, A11, 3)		
	The software schedule is documented, reviewed, and agreed to. (L2-24, A12, 6)	Not specified in the CMM	
	The plans for the project's software engineering facilities and support tools are reviewed by all affected groups. (L2-25, A14, 3)	Affected Groups	
	The activities for software project planning are reviewed with senior management on a periodic basis. (L2-26, V1) ❑ The technical, cost, staffing, and schedule performance is reviewed. ❑ Conflicts and issues not resolvable at lower levels are addressed. ❑ Software project risks are addressed. ❑ Action items are assigned, reviewed, and tracked to closure. ❑ A summary report from each meeting is prepared and distributed to the affected groups and individuals.	Senior Management Affected Groups	

Continued on next page

Reviews and Audits, continued

The table below lists the required reviews and audits for the software project planning process, continued from the previous page.

√	Review or Audit	Review Participants	References
	The activities for software project planning are reviewed with the project manager on both a periodic and event-driven basis. (L2-26, V2) ❑ Affected groups are represented. ❑ Status and current results of the software project planning activities are reviewed against the software project's statement of work and allocated requirements. ❑ Dependencies between groups are addressed. ❑ Conflicts and issues not resolvable at lower levels are addressed. ❑ Software project risks are reviewed. ❑ Action items are assigned, reviewed, and tracked to closure. ❑ A summary report from each meeting is prepared and distributed to the affected groups and individuals.	Project Manager Affected Groups	
	The software quality assurance group reviews and/or audits the activities and work products for software project planning and reports the results. (L2-27, V3) At a minimum, the reviews and/or audits verify: ❑ The activities for software estimating and planning. ❑ The activities for reviewing and making project commitments. ❑ The activities for preparing the software development plan. ❑ The standards used for preparing the software development plan. ❑ The content of the software development plan.	SQA group	

SPP Process - Work Products Managed and Controlled

Work Products Managed and Controlled

The table below lists the work products required to be managed and controlled during the software project planning process.

√	Work Products Managed and Controlled	References
	Project's software development plan (L2-13, C2, 6)	
	Statement of work (L2-15, Ab1, 3)	
	Software planning data (L2-25, A15, 2)	

SPP Process - Measurements

Measurements

The table below lists the measurements required for the software project planning process.

√	Measurements	References
	Software planning data. (L2-25, A15) ❑ Information recorded includes the estimates and the associated information needed to reconstruct the estimates and assess their reasonableness. (L2-25, A15, 1)	
	Estimates and the associated information needed to reconstruct the estimates and assess their reasonableness. (L2-25, A15, 1)	
	Measurements are made and used to determine the status of the software planning activities. (L2-25, M1)	

SPP Process - Documented Procedures

Documented Procedures

The table below lists the software project planning process activities required to be performed according to a documented procedure.

√	Documented procedures	References
	Software project commitments made to individuals and groups external to the organization are reviewed with senior management according to a documented procedure. (L2-17, A4)	
	The project's software development plan is developed according to a documented procedure. (L2-18, A6)	
	Estimates for the size of the software work products (or changes to the size of software work products) are derived according to a documented procedure. (L2-21, A9)	
	Estimates for the software project's effort and costs are derived according to a documented procedure. (L2-22, A10)	
	Estimates for the project's critical computer resources are derived according to a documented procedure. (L2-23, A11)	
	The project's software schedule is derived according to a documented procedure. (L2-23, A12)	

SPP Process - Training

Training

The table below lists the training required for the software project planning process.

√	Training	References
	The software managers, software engineers, and other individuals involved in the software project planning are trained in the software estimating and planning procedures applicable to their areas of responsibility. (L2-16, Ab4)	

SPP Process - Tools

Tools

The table below lists the tools required for the software project planning process.

√	Tools	References
	Tools to support software project planning activities. (L2-16, Ab3, 2)	

Software Project Tracking and Oversight (SPTO) Process

SPTO Process - Overview

SPTO Process Purpose

The purpose of Software Project Tracking and Oversight is to provide adequate visibility into actual progress so that management can take effective actions when the software project's performance deviates significantly from the software plans. (L2-29)

SPTO Process Description

Software Project Tracking and Oversight involves tracking and reviewing the software accomplishments and results against documented estimates, commitments, and plans, and adjusting these plans based on the actual accomplishments and results.

A documented plan for the software project (i.e., the software development plan, as described in the Software Project Planning key process area) is used as the basis for tracking the software activities, communicating status, and revising plans. Software activities are monitored by the management. Progress is primarily determined by comparing the actual software size, effort, cost, and schedule to the plan when selected software work products are completed and at selected milestones. When it is determined that the software project's plans are not being met, corrective actions are taken. These actions may include revising the software development plan to reflect the actual accomplishments and replanning the remaining work or taking actions to improve the performance. (L2-29)

Continued on next page

Chapter Overview

The table below contains the description and location of each section in this chapter.

Section	Description	Page
Roles	List of roles participating in process activities.	SPTO-3
Entry Criteria	Describes when the process can start.	SPTO-10
Inputs	A description of the work products consumed by the process.	SPTO-11
Activities	Describes the activities of the process.	SPTO-12
Outputs	A description of the work products produced by the process.	SPTO-16
Exit Criteria	Describes when the process is complete.	SPTO-20
Reviews and Audits	List of required reviews and audits.	SPTO-25
Work Products Managed and Controlled	Lists work products required to be managed and controlled.	SPTO-29
Measurements	Describes required process measurements.	SPTO-30
Documented Procedures	Lists which activities must be completed according to a documented procedure.	SPTO-31
Training	List of required training.	SPTO-32
Tools	List of required tools.	SPTO-33

SPTO Process - Roles

Roles

The table below lists the roles, and the activities in which they participate in the software project tracking and oversight process.

√	Role	Activities Participated in...	Reference
	Affected Groups	❑ Changes to the software commitments are made with the involvement and agreement of the affected groups. (L2-30, C2, 4) ❑ Changes in size estimates of the software work products that affect software commitments are negotiated with the affected groups and are documented. (L2-36, A5, 5) ❑ Changes in staffing and other software costs that affect software commitments are negotiated with the affected groups and are documented. (L2-36, A6, 4) ❑ Changes in estimates of critical computer resources that affect software commitments are negotiated with the affected groups and are documented. (L2-37, A7, 2) ❑ Software schedule revisions that affect software commitments are negotiated with the affected groups and are documented. (L2-37, A8, 3) ❑ Formal reviews to address the accomplishments and results of the software project are conducted at selected project milestones according to a documented procedure. (L2-39, A13) These reviews: ❑ Are conducted with the customer, end user, and affected groups within the organization, as appropriate. (L2-39, A13, 2) ❑ A summary status report from each review (meeting) with senior management is prepared and distributed to the affected groups. (L2-40, V1, 5)	
		Role/Function continues on the next page	

Continued on next page

Roles, continued

The table below lists the roles, and the activities in which they participate in the software project tracking and oversight process, continued from the previous page.

√	Role	Activities Participated in...	Reference
	Affected Groups, continued	❑ The activities for software project tracking and oversight are reviewed with the project manager on both a periodic and event-driven basis. (L2-41, V2) ❑ **Affected groups** are represented. (L2-41, V2, 1) ❑ A summary status report from each review (meeting) with the project manager is prepared and distributed to the **affected groups**. (L2-40, V2, 8)	
	Customer	❑ Formal reviews to address the accomplishments and results of the software project are conducted at selected project milestones according to a documented procedure. (L2-39, A13) These reviews: ❑ Are conducted with the **customer**, end user, and affected groups within the organization, as appropriate. (L2-39, A13, 2)	
	End User	❑ Formal reviews to address the accomplishments and results of the software project are conducted at selected project milestones according to a documented procedure. (L2-39, A13) These reviews: ❑ Are conducted with the customer, **end user**, and affected groups within the organization, as appropriate. (L2-39, A13, 2)	

Continued on next page

SPTO Process - Roles, Continued

Roles, continued

The table below lists the roles, and the activities in which they participate in the software project tracking and oversight process, continued from the previous page.

√	Role	Activities Participated in...	Reference
	First-line Software Managers	❑ **First-line software managers** receive orientation in the technical aspects of the software project. (L2-32, Ab5) ❑ Members of the software engineering group report their technical status to their **first-line manager** on a regular basis. (L2-37, A9, 1) ❑ The software engineering group conducts periodic internal reviews to track technical progress, plans, performance, and issues against the software development plan.(L2-38, A12) These reviews are conducted between: ❑ The **first-line software managers** and their software task leaders. ❑ The project software manager, **first-line software managers**, and other software managers, as appropriate.	
	Individuals and Groups External to the Organization	❑ Senior management reviews all commitment changes and new software project commitments made to **individuals and groups external to the organization.** (L2-31, C2, 5) ❑ Software project commitments and changes to commitments made to **individuals and groups external to the organization** are reviewed with senior management according to a documented procedure. (L2-35, A3)	

Continued on next page

SPTO Process - Roles Continued

Roles, continued

The table below lists the roles, and the activities in which they participate in the software project tracking and oversight process, continued from the previous page.

√	Role	Activities Participated in...	Reference
	Project Manager	❑ The **project manager** is kept informed of the software project's status and issues. (L2-30, C2, 2) ❑ High-risk areas associated with cost, resource, schedule, and technical aspects of the project are reviewed with the **project manager** on a regular basis. (L2-38, A10, 2) ❑ The activities for software project tracking and oversight are reviewed with the **project manager** on both a periodic and event-driven basis. (L2-41, V2)	
	Project Software Manager	❑ A **project software manager** is designated to be responsible for the project's software activities and results. (L2-30, C1) ❑ The **project software manager** explicitly assigns responsibility for software work products and activities. (L2-31, Ab2) ❑ The software engineering group conducts periodic internal reviews to track technical progress, plans, performance, and issues against the software development plan. (L2-38, A12) These reviews are conducted between: ❑ The first-line software managers and their software task leaders. ❑ The **project software manager**, first-line software managers, and other software managers, as appropriate.	

Continued on next page

SPTO Process - Roles Continued

Roles, continued

The table below lists the roles, and the activities in which they participate in the software project tracking and oversight process, continued from the previous page.

√	Role	Activities Participated in...	Reference
	Senior Management	❑ Senior management reviews all commitment changes and new software project commitments made to individuals and groups external to the organization. (L2-31, C2, 5) ❑ Software project commitments and changes to commitments made to individuals and groups external to the organization are reviewed with senior management according to a documented procedure. (L2-35, A3) ❑ The activities for software project tracking and oversight are reviewed with senior management on a periodic basis. (L2-40, V1)	
	Software Engineering Group	❑ Approved changes to commitments that affect the software project are communicated to the members of the software engineering group and other software-related groups. (L2-35, A4) ❑ Members of the software engineering group report their technical status to their first-line manager on a regular basis. (L2-37, A9, 1) ❑ The software engineering group conducts periodic internal reviews to track technical progress, plans, performance, and issues against the software development plan. (L2-38, A12)	

Continued on next page

Roles, continued

The table below lists the roles, and the activities in which they participate in the software project tracking and oversight process, continued from the previous page.

√	Role	Activities Participated in...	Reference
	Software Manager	❑ The software managers and the software task leaders are assigned specific responsibilities for tracking the software project. (L2-32, Ab3, 1) ❑ The software managers are trained in managing the technical and personnel aspects of the software project. (L2-32, Ab4) ❑ The software engineering group conducts periodic internal reviews to track technical progress, plans, performance, and issues against the software development plan. (L2-38, A12) *These reviews are conducted between:* ❑ The first-line software managers and their software task leaders. ❑ The project software manager, first-line software managers, and other software managers, as appropriate. ❑ Formal review to address the accomplishments and results of the software project are conducted at selected project milestones according to a documented procedure. (L2-39, A13) These reviews: ❑ Use materials that are reviewed and approved by the responsible software managers. (L2-39, A13, 3)	

Continued on next page

SPTO Process - Roles, Continued

Roles, continued

The table below lists the roles, and the activities in which they participate in the software project tracking and oversight process, continued from the previous page.

√	Role	Activities Participated in...	Reference
	Software-related Groups	Approved changes to commitments that affect the software project are communicated to the members of the software engineering group and other **software-related groups.** (L2-35, A4)	
	SQA Group	The **software quality assurance group** reviews and/or audits the activities and work products for software project tracking and oversight and reports the results. (L2-41, V3)	
	Software Task Leaders	❑ The software managers and the **software task leaders** are assigned specific responsibilities for tracking the software project. (L2-32, Ab3, 1) ❑ The software engineering group conducts periodic internal reviews to track technical progress, plans, performance, and issues against the software development plan. (L2-38, A12) These reviews are conducted between: ❑ The first-line software managers and their **software task leaders.** ❑ The project software manager, first-line software managers, and other software managers, as appropriate.	

SPTO Process - Entry Criteria

Entry Criteria The table below describes the conditions that must be satisfied in order to begin the software project tracking and oversight process.

√	Condition	References
	A project software manager is designated to be responsible for the project's software activities and results. (L2-30, C1)	
	The project follows a written organizational policy for managing the software project. (L2-30, C2) [Refer to SPF Policies for additional information regarding SPTO policy.]	
	A software development plan for the software project is documented and approved. (L2-31, Ab1)	
	The project software manager explicitly assigns responsibility for software work products and activities. (L2-31, Ab2) The assigned responsibilities cover: ❑ The software work products to be developed or services to be provided. ❑ The effort and cost for these software activities. ❑ The schedule for these software activities. ❑ The budget for these software activities.	
	Adequate resources and funding are provided for tracking the software project. (L2-32, Ab3)	
	The software managers and the software task leaders are assigned specific responsibilities for tracking the software project. (L2-32, Ab3, 1)	
	Tools to support software tracking are made available. (L2-32, Ab3, 2)	
	The software managers are trained in managing the technical and personnel aspects of the software project. (L2-32, Ab4)	
	First-line software managers receive orientation in the technical aspects of the software project. (L2-32, Ab5)	

SPTO Process - Inputs

Inputs

The table below lists the inputs to the software process tracking and oversight process.

√	Input	Org. Input	References
	Changes to commitments. (L2-34, A2, 2)		
	New software project commitments. (L2-34, A2, 2)		
	Software commitments. (L2-36, A5, 5)		
	Software development plan changes. (L2-34, A2, 1)		
	Software development plan changes. (L2-34, A2, 1)		
	Software development plan refinements. (L2-34, A2, 1)		
	Software development plan refinements. (L2-34, A2, 1)		
	Software development plan. (L2-30, C1) [Refer to SPF Standards for additional information regarding a software development plan.]		
	Software planning data. (L2-38, A11, 3)		

Activities

The table below lists the required activities for the software process tracking and oversight process.

√	Activities	References
	A documented software development plan is used for tracking the software activities and communicating status. (L2-33, A1) This software development plan is: ❑ Updated as the work progresses to reflect accomplishments, particularly when milestones are completed. (L2-33, A1, 1)	
	The project's software development plan is revised according to a documented procedure. (L2-33, A2) [Refer to Documented Procedures for additional information.]	
	Software project commitments and changes to commitments made to individuals and groups external to the organization are reviewed with senior management according to a documented procedure. (L2-35, A3)	
	Approved changes to commitments that affect the software project are communicated to the members of the software engineering group and other software-related groups. (L2-35, A4)	
	The size of the software work products (or size of the changes to the software work products) are tracked, and corrective actions are taken as necessary. (L2-35, A5) ❑ Sizes for all major software work products (or the size of the changes) are tracked. ❑ Actual size of code (generated, fully tested, and delivered) is compared to the estimates documented in the software development plan. ❑ Actual units of delivered documentation are compared to the estimates documented in the software development plan. ❑ Overall projected size of the software work products (estimates combined with actuals) is refined, monitored, and adjusted on a regular basis. ❑ Changes in size estimates of the software work products that affect software commitments are negotiated with the affected groups and are documented.	

Continued on next page

Activities, continued

The table below lists the required activities for the software process tracking and oversight process, continued from the previous page.

√	Activities	References
	The project's software effort and costs are tracked, and corrective actions are taken as necessary. (L2-36, A6) ❑ Actual expenditures of effort and costs over time and against work completed are compared to the estimates documented in the software development plan to identify potential overruns and underruns. ❑ Software costs are tracked and compared to the estimates documented in the software development plan. ❑ Effort and staffing are compared to the estimates documented in the software development plan. ❑ Changes in staffing and other software costs that affect software commitments are negotiated with the affected groups and are documented.	
	The project's critical computer resources are tracked, and corrective actions are taken as necessary. (L2-36, A7) ❑ The actual and projected use of the project's critical computer resources are tracked and compared to the estimates for each major software component as documented in the software development plan. ❑ Changes in estimates of critical computer resources that affect software commitments are negotiated with the affected groups and are documented.	
	The project's software schedule is tracked, and corrective actions are taken as necessary. (L2-37, A8) ❑ Actual completion of software activities, milestones, and other commitments is compared against the software development plan. ❑ Effects of late and early completion of software activities, milestones, and other commitments are evaluated for impacts on future activities and milestones. ❑ Software schedule revisions that affect software commitments are negotiated with the affected groups and are documented.	

Continued on next page

Activities, continued

The table below lists the required activities for the software process tracking and oversight process, continued from the previous page.

√	Activities	References
	Software engineering technical activities are tracked, and corrective actions are taken as necessary. (L2-37, A9) ❑ Members of the software engineering group report their technical status to their first-line manager on a regular basis. ❑ Software release contents for successive builds are compared to the plans documented in the software development plan. ❑ Problems identified in any of the software work products are reported and documented. ❑ Problem reports are tracked to closure.	
	The software risks associated with cost, resource, schedule, and technical aspects of the project are tracked. (L2-37, A10) ❑ The priorities of the risks and the contingencies for the risks are adjusted as additional information becomes available. ❑ High-risk areas are reviewed with the project manager on a regular basis.	
	Actual measurement data and replanning data for the software project are recorded. (L2-38, A11) ❑ Information recorded includes the estimates and associated information needed to reconstruct the estimates and verify their reasonableness. ❑ The software replanning data are managed and controlled. ❑ The software planning data, replanning data, and the actual measurement data are archived for use by ongoing and future projects.	
	The software engineering group conducts periodic internal reviews to track technical progress, plans, performance, and issues against the software development plan. (L2-38, A12) [Refer to Reviews and Audits for additional information.]	
	Formal reviews to address the accomplishments and results of the software project are conducted at selected project milestones according to a documented procedure. (L2-39, A13) [Refer to Reviews and Audits for additional information.]	

Continued on next page

Activities, continued

The table below lists the required activities for the software process tracking and oversight process, continued from the previous page.

√	Activities	References
	Measurements are made and used to determine the status of the software tracking and oversight activities. (L2-39, M1)	
	The activities for software project tracking and oversight are reviewed with senior management on a periodic basis. (L2-40, V1) [Refer to Reviews and Audits for additional information.]	
	The activities for software project tracking and oversight are reviewed with the project manager on both a periodic and event-driven basis. (L2-41, V2) [Refer to Reviews and Audits for additional information.]	
	The software quality assurance group reviews and/or audits the activities and work products for software project tracking and oversight and reports the results. (L2-41, V3) [Refer to Reviews and Audits for additional information.]	

SPTO Process - Outputs

Outputs

The table below lists the outputs produced by the software project tracking and oversight process.

√	Output	Org. Output	References
	Accomplishments and results of the software project. (L2-39, A13)		
	Action items resulting from activities reviews with the project manager. (L2-41, V2, 7)		
	Action items resulting from periodic activities reviews with senior management. (L2-40, V1, 4)		
	Actual measurement data and replanning data for the software project. (L2-38, A11) ❑ Estimates and associated information needed to reconstruct the estimates and verify their reasonableness.		
	Actual measurement data. (L2-38, A11, 3)		
	Approved changes to commitments that affect the software project. (L2-35, A4)		
	Changes to commitments made to individuals and groups external to the organization. (L2-35, A3)		
	Changes to the software commitments. (L2-31, C2, 5)		
	Conflicts and issues. (L2-40, V1, 2)		
	Current estimates and actual use of critical computer resources. (L2-41, V2, 3)		
	Measurements. (L2-39, M1)		
	New software project commitments made to individuals and groups external to the organization. (L2-31, C2, 5)		

Continued on next page

SPTO Process - Outputs, Continued

Outputs, continued

The table below lists the outputs produced by the software project tracking and oversight process, continued from the previous page.

√	Output	Org. Output	References
	Project's critical computer resources. (L2-36, A7) ❑ The actual and projected use of the project's critical computer resources. ❑ Changes in estimates of critical computer resources that affect software commitments.		
	Project's software effort and costs. (L2-36, A6) ❑ Actual expenditures of effort and costs over time and against work completed. ❑ Potential overruns and underruns. ❑ Software costs. ❑ Effort and staffing. ❑ Changes in staffing and other software costs that affect software commitments.		
	Project's software schedule. (L2-37, A8) ❑ Actual completion of software activities, milestones, and other commitments. ❑ Effects of late and early completion of software activities, milestones, and other commitments. ❑ Software schedule revisions that affect software commitments.		
	Results of SQA group reviews and/or audits. (L2-41, V3)		
	Significant issues, action items, and decisions resulting from formal reviews. (L2-39, A13, 5)		

Continued on next page

SPTO Process - Outputs, Continued

Outputs, continued

The table below lists the outputs produced by the software project tracking and oversight process, continued from the previous page.

√	Output	Org. Output	References
	Size of the software work products (or size of the changes to the software work products). (L2-35, A5) ❑ Sizes for all major software work products (or size of the changes). ❑ Actual size of code (generated, fully tested, and delivered). ❑ Actual units of delivered documentation. ❑ Overall projected size of the software work products (estimates combined with actuals). ❑ Changes in size estimates of the software work products that affect software commitments.		
	Software development plan. (L2-33, A2)		
	Software engineering technical activities. (L2-37, A9) ❑ Technical status. ❑ Software release contents for successive builds. ❑ Problems identified in any of the software work products. ❑ Problem reports.		
	Software project commitments made to individuals and groups external to the organization. (L2-35, A3)		
	Software project's status and issues. (L2-30, C2, 2)		
	Software replanning data. (L2-38, A11, 3)		
	Software risks associated with cost, resource, schedule, and technical aspects of the project. (L2-37, A10) ❑ Priorities of the risks and the contingencies for the risks. ❑ High-risk areas.		
	Status of software activities. (L2-33, A1)		

Continued on next page

SPTO Process - Outputs, Continued

Outputs, continued

The table below lists the outputs produced by the software project tracking and oversight process, continued from the previous page.

√	Output	Org. Output	References
	Summary report from each activities review with the project manager. (L2-41, V2, 8)		
	Summary status report from each periodic activities review with senior management. (L2-40, V1, 5)		
	Technical progress, plans, performance, and issues. (L2-38, A12)		
	Technical, cost, staffing, and schedule performance. (L2-40, V1, 1)		

SPTO Process - Exit Criteria

Exit Criteria

The table below describes the conditions that must be satisfied in order to exit the software project tracking and oversight process.

√	Condition	References
	A documented software development plan is used for tracking the software activities and communicating status. (L2-33, A1) This software development plan is: ❑ Updated as the work progresses to reflect accomplishments, particularly when milestones are completed. ❑ Readily available to: ❑ the software engineering group (including all subgroups, such as software design), ❑ the software managers, ❑ the project manager, ❑ senior management, and ❑ other affected groups.	
	The project's software development plan is revised according to a documented procedure. (L2-33, A2)	
	Software project commitments and changes to commitments made to individuals and groups external to the organization are reviewed with senior management according to a documented procedure. (L2-35, A3)	
	Approved changes to commitments that affect the software project are communicated to the members of the software engineering group and other software-related groups. (L2-35, A4)	

Continued on next page

Exit Criteria, continued

The table below describes the conditions that must be satisfied in order to exit the software project tracking and oversight process, continued from the previous page.

√	Condition	References
	The size of the software work products (or size of the changes to the software work products) are tracked, and corrective actions are taken as necessary. (L2-35, A5) ❑ Sizes for all major software work products (or the size of the changes) are tracked. ❑ Actual size of code (generated, fully tested, and delivered) is compared to the estimates documented in the software development plan. ❑ Actual units of delivered documentation are compared to the estimates documented in the software development plan. ❑ Overall projected size of the software work products (estimates combined with actuals) is refined, monitored, and adjusted on a regular basis. ❑ Changes in size estimates of the software work products that affect software commitments are negotiated with the affected groups and are documented.	
	The project's software effort and costs are tracked, and corrective actions are taken as necessary. (L2-36, A6) ❑ Actual expenditures of effort and costs over time and against work completed are compared to the estimates documented in the software development plan to identify potential overruns and underruns. ❑ Software costs are tracked and compared to the estimates documented in the software development plan. ❑ Effort and staffing are compared to the estimates documented in the software development plan. ❑ Changes in staffing and other software costs that affect software commitments are negotiated with the affected groups and are documented.	
	The project's critical computer resources are tracked, and corrective actions are taken as necessary. (L2-36, A7) ❑ The actual and projected use of the project's critical computer resources are tracked and compared to the estimates for each major software component as documented in the software development plan. ❑ Changes in estimates of critical computer resources that affect software commitments are negotiated with the affected groups and are documented.	

Continued on next page

Exit Criteria, continued

The table below describes the conditions that must be satisfied in order to exit the software project tracking and oversight process, continued from the previous page.

√	Condition	References
	The project's software schedule is tracked, and corrective actions are taken as necessary. (L2-37, A8) ❑ Actual completion of software activities, milestones, and other commitments is compared against the software development plan. ❑ Effects of late and early completion of software activities, milestones, and other commitments are evaluated for impacts on future activities and milestones. ❑ Software schedule revisions that affect software commitments are negotiated with the affected groups and are documented.	
	Software engineering technical activities are tracked, and corrective actions are taken as necessary. (L2-37, A9) ❑ Members of the software engineering group report their technical status to their first-line manager on a regular basis. ❑ Software release contents for successive builds are compared to the plans documented in the software development plan. ❑ Problems identified in any of the software work products are reported and documented. ❑ Problem reports are tracked to closure.	
	The software risks associated with cost, resource, schedule, and technical aspects of the project are tracked. (L2-37, A10) ❑ The priorities of the risks and the contingencies for the risks are adjusted as additional information becomes available. ❑ High-risk areas are reviewed with the project manager on a regular basis.	

Continued on next page

SPTO Process - Exit Criteria, Continued

Exit Criteria, continued

The table below describes the conditions that must be satisfied in order to exit the software project tracking and oversight process, continued from the previous page.

√	Condition	References
	Actual measurement data and replanning data for the software project are recorded. (L2-38, A11) ❑ Information recorded includes the estimates and associated information needed to reconstruct the estimates and verify their reasonableness. ❑ The software replanning data are managed and controlled. ❑ The software planning data, replanning data, and the actual measurement data are archived for use by ongoing and future projects.	
	The **software engineering group** conducts periodic internal reviews to track technical progress, plans, performance, and issues against the software development plan. (L2-38, A12)	
	Formal reviews to address the accomplishments and results of the software project are conducted at selected project milestones according to a documented procedure. (L2-39, A13) These reviews: ❑ Are planned to occur at meaningful points in the software project's schedule, such as the beginning or completion of selected stages. ❑ Are conducted with the **customer, end user,** and affected groups within the organization, as appropriate. ❑ Use materials that are reviewed and approved by the responsible **software managers.** ❑ Address the commitments, plans, and status of the software activities. ❑ Result in the identification and documentation of significant issues, action items, and decisions. ❑ Address the software project risks. ❑ Result in the refinement of the software development plan as necessary.	
	Measurements are made and used to determine the status of the software tracking and oversight activities. (L2-39, M1)	
	The activities for software project tracking and oversight are reviewed with **senior management** on a periodic basis. (L2-40, V1)	

Continued on next page

SPTO Process - Exit Criteria, Continued

Exit Criteria, continued

The table below describes the conditions that must be satisfied in order to exit the software project tracking and oversight process, continued from the previous page.

√	Condition	References
	The activities for software project tracking and oversight are reviewed with the **project manager** on both a periodic and event-driven basis. (L2-41, V2)	
	The **software quality assurance group** reviews and/or audits the activities and work products for software project tracking and oversight and reports the results. (L2-41, V3)	

SPTO Process - Reviews and Audits

Reviews and Audits

The table below lists the required reviews and audits for the software project tracking and oversight process.

√	Review or Audit	Review Participants	References
	Senior management reviews all commitment changes and new software project commitments made to individuals and groups external to the organization. (L2-31, C2, 5)	Senior Management	
	The software development plan is reviewed at each revision. (L2-34, A2, 3)	Not Specified in CMM	
	Software project commitments and changes to commitments made to individuals and groups external to the organization are reviewed with senior management according to a documented procedure. (L2-35, A3)	Senior Management	
	High-risk areas are reviewed with the project manager on a regular basis. (L2-38, A10, 2)	Project Manager	
	The software engineering group conducts periodic internal reviews to track technical progress, plans, performance, and issues against the software development plan. (L2-38, A12) These reviews are conducted between: ❑ The first-line software managers and their software task leaders. ❑ The project software manager, first-line software managers, and other software managers, as appropriate.	Software Engineering Group First-line Software Managers Project Software Manager Software Managers Software Task Leaders	

Continued on next page

Reviews and Audits, continued

The table below lists the required reviews and audits for the software project tracking and oversight process, continued from the previous page.

√	Review or Audit	Review Participants	References
	Formal reviews to address the accomplishments and results of the software project are conducted at selected project milestones according to a documented procedure. (L2-39, A13) These reviews: ❑ Are planned to occur at meaningful points in the software project's schedule, such as the beginning or completion of selected stages. ❑ Are conducted with the customer, end user, and affected groups within the organization, as appropriate. ❑ Use materials that are reviewed and approved by the responsible software managers. ❑ Address the commitments, plans, and status of the software activities. ❑ Result in the identification and documentation of significant issues, action items, and decisions. ❑ Address the software project risks. ❑ Result in the refinement of the software development plan as necessary.	Customer End User Software Managers	
	The activities for software project tracking and oversight are reviewed with senior management on a periodic basis. (L2-40, V1) ❑ The technical, cost, staffing, and schedule performance are reviewed. ❑ Conflicts and issues not resolvable at lower levels are addressed. ❑ Software project risks are addressed. ❑ Action items are assigned, reviewed, and tracked to closure. ❑ A summary status report from each meeting is prepared and distributed to the affected groups.	Senior Management Affected Groups	

Continued on next page

SPTO Process - Reviews and Audits, Continued

Reviews and Audits, continued

The table below lists the required reviews and audits for the software project tracking and oversight process, continued from the previous page.

√	Review or Audit	Review Participants	References
	The activities for software project tracking and oversight are reviewed with the project manager on both a periodic and event-driven basis. (L2-41, V2) ❑ Affected groups are represented. ❑ The technical, cost, staffing, and schedule performance is reviewed against the software development plan. ❑ Use of critical computer resources is reviewed; current estimates and actual use of these critical computer resources are reported against the original estimates. ❑ Dependencies between groups are addressed. ❑ Conflicts and issues not resolvable at lower levels are addressed. ❑ Software project risks are addressed. ❑ Action items are assigned, reviewed, and tracked to closure. ❑ A summary report from each meeting is prepared and distributed to the affected groups.	Project Manager Affected Groups	

Continued on next page

SPTO Process - Reviews and Audits, Continued

Reviews and Audits, continued

The table below lists the required reviews and audits for the software project tracking and oversight process, continued from the previous page.

√	Review or Audit	Review Participants	References
	The software quality assurance group reviews and/or audits the activities and work products for software project tracking and oversight and reports the results. (L2-41, V3) At a minimum, the reviews and/or audits verify: ❑ The activities for reviewing and revising commitments. ❑ The activities for revising the software development plan. ❑ The content of the revised software development plan. ❑ The activities for tracking the software project's cost, schedule, risks, technical and design constraints, and functionality and performance. ❑ The activities for conducting the planned technical and management reviews.	**SQA Group**	

SPTO Process - Work Products Managed and Controlled

Work Products Managed and Controlled

The table below lists the work products required to be managed and controlled during the software process tracking and oversight process.

√	Work Products Managed and Controlled	References
	Software development plan. (L2-34, A2, 4)	
	Software replanning data. (L2-38, A11, 2)	

SPTO Process - Measurements

Measurements

The table below lists the measurements required for the software project tracking and oversight process.

√	Measurements	References
	Actual measurement data and replanning data for the software project are recorded. (L2-38, A11)	
	Measurements are made and used to determine the status of the software tracking and oversight activities. (L2-39, M1)	

SPTO Process - Documented Procedures

Documented Procedures

The table below lists the software process tracking and oversight process activities required to be performed according to a documented procedure.

√	Documented procedures	References
	The project's software development plan is revised according to a documented procedure. (L2-33, A2)	
	Software project commitments and changes to commitments made to individuals and groups external to the organization are reviewed with senior management according to a documented procedure. (L2-35, A3)	
	Formal reviews to address the accomplishments and results of the software project are conducted at selected project milestones according to a documented procedure. (L2-39, A13)	

SPTO Process - Training

Training

The table below lists the training required for the software project tracking and oversight process.

√	Training	References
	The software managers are trained in managing the technical and personnel aspects of the software project. (L2-32, Ab4)	
	First-line software managers receive orientation in the technical aspects of the software project. (L2-32, Ab5)	

SPTO Process - Tools

Tools

The table below lists the tools required for the software project tracking and oversight process.

√	Tools	References
	Tools to support software tracking. (L2-32, Ab3, 2)	

Software Subcontract Management (SSM) Process

SSM Process - Overview

SSM Process Purpose

The purpose of Software Subcontract Management is to select qualified software subcontractors and manage them effectively. (L2-43)

SSM Process Description

Software Subcontract Management involves selecting a software subcontractor, establishing commitments with the subcontractor, and tracking and reviewing the subcontractor's performance and results. These practices cover the management of a software (only) subcontract, as well as the management of the software component of a subcontract that includes software, hardware, and possibly other system components.

The subcontractor is selected based on its ability to perform the work. Many factors contribute to the decision to subcontract a portion of the prime contractor's work. Subcontractors may be selected based on strategic business alliances, as well as technical considerations. The practices of this key process area address the traditional acquisition process associated with subcontracting a defined portion of the work to another organization.

When subcontracting, a documented agreement covering the technical and nontechnical (e.g., delivery dates) requirements is established and is used as the basis for managing the subcontract. The work to be done by the subcontractor and the plans for the work are documented. The standards that are to be followed by the subcontractor are compatible with the prime contractor's standards.

The software planning, tracking, and oversight activities for the subcontracted work are performed by the subcontractor. The prime contractor ensures that these planning, tracking, and oversight activities are performed appropriately and that the software products delivered by the subcontractor satisfy their acceptance criteria. The prime contractor works with the subcontractor to manage their product and process interfaces. (L2-43)

Continued on next page

Chapter Overview

The table below contains the description and location of each section in this chapter.

Section	Description	Page
Roles	List of roles participating in process activities.	SSM-3
Entry Criteria	Describes when the process can start.	SSM-10
Inputs	A description of the work products consumed by the process.	SSM-11
Activities	Describes the activities of the process.	SSM-13
Outputs	A description of the work products produced by the process.	SSM-15
Exit Criteria	Describes when the process is complete.	SSM-17
Reviews and Audits	List of required reviews and audits.	SSM-19
Work Products Managed and Controlled	Lists work products required to be managed and controlled.	SSM-24
Measurements	Describes required process measurements.	SSM-25
Documented Procedures	Lists which activities must be completed according to a documented procedure.	SSM-26
Training	List of required training.	SSM-27
Tools	List of required tools.	SSM-28

SSM Process - Roles

Roles

The table below lists the roles, and the activities in which they participate in the software subcontract management process.

√	Role	Activities Participated in...	Reference
	Affected Groups (Parties)	❑ Changes to the software subcontractor's statement of work, subcontract terms and conditions, and other commitments are resolved according to a documented procedure. (L2-51, A6) ❑ This procedure typically specifies that all affected groups of both the prime contractor and the subcontractor are involved. ❑ The subcontract manager is responsible for coordinating the technical scope of work to be subcontracted and the terms and conditions of the subcontract with the affected parties. (L2-45, C2, 2)	
	Customers and End Users	The subcontractor is provided with visibility of the needs and desires of the product's customers and end users, as appropriate. (L2-52, A7, 1)	
	Individuals	❑ The subcontract manager is knowledgeable and experienced in software engineering or has individuals assigned who have that knowledge and experience. (L2-45, C2, 1) ❑ Software managers and other individuals are assigned specific responsibilities for managing the subcontract. (L2-46, Ab1, 1) ❑ Software managers and other individuals who are involved in establishing and managing the software subcontract are trained to perform these activities. (L2-46, Ab2) ❑ Software managers and other individuals who are involved in managing the software subcontract receive orientation in the technical aspects of the subcontract. (L2-46, Ab3)	

Continued on next page

SSM Process - Roles, Continued

Roles, continued The table below lists the roles, and the activities in which they participate in the software subcontract management process, continued from the previous page.

√	Role	Activities Participated in...	Reference
	Prime Contractor	❑ Changes to the subcontract are made with the involvement and agreement of both the **prime contractor** and the subcontractor. (L2-45, C1, 3) ❑ The contractual agreement between the **prime contractor** and the software subcontractor is used as the basis for managing the subcontract. (L2-50,A3) ❑ A documented subcontractor's software development plan is reviewed and approved by the **prime contractor**. (L2-51, A4) ❑ Critical dependencies and commitments between the **prime contractor** and the subcontractor are addressed. (L2-52, A7, 5) ❑ Subcontractor commitments to the **prime contractor** and prime contractor commitments to the subcontractor are *both reviewed.* ❑ The **prime contractor** and the subcontractor coordinate their activities on matters relating to software configuration management to ensure that the subcontractor's products can be readily integrated or incorporated into the project environment of the **prime contractor**. (L2-54, A11, 2) ❑ The **prime contractor** conducts acceptance testing as part of the delivery of the subcontractor's software products according to a documented procedure. (L2-55, A12) ❑ The acceptance procedures and acceptance criteria for each product are defined, reviewed, and approved by both the **prime contractor** and the subcontractor prior to the test. (L2-55, A12, 1)	

Continued on next page

SSM Process - Roles, Continued

Roles, continued The table below lists the roles, and the activities in which they participate in the software subcontract management process, continued from the previous page.

√	Role	Activities Participated in...	Reference
	Prime Contractor's SCM Group	The **prime contractor's software configuration management group** monitors the subcontractor's activities for software configuration management according to a documented procedure. (L2-54, A11)	
	Prime Contractor's SQA Group *or* **SQA Group**	❑ The **prime contractor's software quality assurance group** monitors the subcontractor's software quality assurance activities according to a documented procedure. (L2-53, A10) ❑ The **prime contractor's software quality assurance group** spot checks the subcontractor's software engineering activities and products. (L2-54, A10, 2.1) ❑ The **prime contractor's software quality assurance group** audits the subcontractor's software quality assurance records, as appropriate. (L2-54, A10, 2.2) ❑ The **software quality assurance group** reviews and/or audits the activities and work products for managing the software subcontract and reports the results. (L2-57, V3)	
	Prime Contractor's Management	The **prime contractor's management** conducts periodic status/coordination reviews with the software subcontractor's management. (L2-51, A7)	
	Project Manager	The activities for managing the software subcontract are reviewed with the **project manager** on both a periodic and event-driven basis. (L2-56, V2)	

Continued on next page

SSM Process - Roles, Continued

Roles, continued The table below lists the roles, and the activities in which they participate in the software subcontract management process, continued from the previous page.

√	Role	Activities Participated in...	Reference
	Senior Management	The activities for managing the software subcontract are reviewed with senior management on a periodic basis. (L2-56, V1)	
	Software Managers	❑ Software managers and other individuals are assigned specific responsibilities for managing the subcontract. (L2-46, Ab1, 1) ❑ Software managers and other individuals who are involved in establishing and managing the software subcontract are trained to perform these activities. (L2-46, Ab2) ❑ Software managers and other individuals who are involved in managing the software subcontract receive orientation in the technical aspects of the subcontract. (L2-46, Ab3)	
	Software Subcontractor's Management	The prime contractor's management conducts periodic status/coordination reviews with the software subcontractor's management. (L2-51, A7)	
	Subcontract Bidder	The software subcontractor is selected, based on an evaluation of the subcontract bidders' ability to perform the work, according to a documented procedure. (L2-49, A2)	

Continued on next page

SSM Process - Roles, Continued

Roles, continued The table below lists the roles, and the activities in which they participate in the software subcontract management process, continued from the previous page.

√	Role	Activities Participated in...	Reference
	Subcontract Manager	❑ A subcontract manager is designated to be responsible for establishing and managing the software subcontract. (L2-45, C2) ❑ The subcontract manager is knowledgeable and experienced in software engineering or has individuals assigned who have that knowledge and experience. (L2-45, C2, 1) ❑ The subcontract manager is responsible for coordinating the technical scope of work to be subcontracted and the terms and conditions of the subcontract with the affected parties. (L2-45, C2, 2) ❑ The subcontract manager is responsible for: (L2-45, C2, 3) ❑ selecting the software subcontractor, ❑ managing the software subcontract, and ❑ arranging for the post-subcontract support of the subcontracted products.	

Continued on next page

Roles, continued The table below lists the roles, and the activities in which they participate in the software subcontract management process, continued from the previous page.

√	Role	Activities Participated in...	Reference
	Software Subcontractor	❑ Changes to the subcontract are made with the involvement and agreement of both the prime contractor and the subcontractor. (L2-45, C1, 3) ❑ The software subcontractor is selected, based on an evaluation of the subcontract bidders' ability to perform the work, according to a documented procedure. (L2-49, A2) ❑ The contractual agreement between the prime contractor and the software subcontractor is used as the basis for managing the subcontract. (L2-50,A3) ❑ The subcontractor is provided with visibility of the needs and desires of the product's customers and end users, as appropriate. (L2-52, A7, 1) ❑ Critical dependencies and commitments between the prime contractor and the subcontractor are addressed. (L2-52, A7, 5) ❑ Subcontractor commitments to the prime contractor and prime contractor commitments to the subcontractor are both reviewed. ❑ Periodic technical reviews and interchanges are held with the software subcontractor. (L2-52, A8) ❑ The prime contractor and the subcontractor coordinate their activities on matters relating to software configuration management to ensure that the subcontractor's products can be readily integrated or incorporated into the project environment of the prime contractor. (L2-54, A11, 2)	
	Role continues on the next page		

Continued on next page

SSM Process - Roles, Continued

Roles, continued The table below lists the roles, and the activities in which they participate in the software subcontract management process, continued from the previous page.

√	Role	Activities Participated in...	Reference
	Software Subcontractor, continued	❑ The acceptance procedures and acceptance criteria for each product are defined, reviewed, and approved by both the prime contractor and the subcontractor prior to the test. (L2-55, A12, 1) ❑ The software subcontractor's performance is evaluated on a periodic basis, and the evaluation is reviewed with the subcontractor. (L2-55, A13)	
	Software Subcontractor Groups	Critical dependencies and commitments between the subcontractor's software engineering group and other subcontractor groups are addressed. (L2-52, A7, 4)	
	Software Subcontractor's Software Engineering Group	Critical dependencies and commitments between the subcontractor's software engineering group and other subcontractor groups are addressed. (L2-52, A7, 4)	

SSM Process - Entry Criteria

Entry Criteria The table below describes the conditions that must be satisfied in order to begin the software subcontract management process.

√	Condition	References
	The project follows a written organizational policy for managing the software subcontract. (L2-44, C1) [Refer to SPF Policies for additional information regarding SSM policy.]	
	A subcontract manager is designated to be responsible for establishing and managing the software subcontract. (L2-45, C2)	
	The subcontract manager is knowledgeable and experienced in software engineering or has individuals assigned who have that knowledge and experience. (L2-45, C2, 1)	
	Adequate resources and funding are provided for selecting the software subcontractor and managing the subcontract. (L2-46, Ab1)	
	Software managers and other individuals are assigned specific responsibilities for managing the subcontract. (L2-46, Ab1, 1)	
	Tools to support managing the subcontract are made available. (L2-46, Ab1, 2)	
	Software managers and other individuals who are involved in establishing and managing the software subcontract are trained to perform these activities. (L2-46, Ab2)	
	Software managers and other individuals who are involved in managing the software subcontract receive orientation in the technical aspects of the subcontract. (L2-46, Ab3)	

SSM Process - Inputs

Inputs

The table below lists the inputs to the software subcontract management process.

√	Input	Org. Input	References
	Acceptance procedures and acceptance criteria for each product (L2-55, A12, 1)		
	Computer resources designated as critical for the project. (L2-52, A7, 3)		
	Conflicts and issues not resolvable by the subcontractor. (L2-52, A7, 8)		
	Critical dependencies and commitments between the prime contractor and the subcontractor. (L2-52, A7, 5)		
	Critical dependencies and commitments between the subcontractor's software engineering group and other subcontractor groups. (L2-52, A7, 4)		
	Needs and desires of the product's customers and end users. (L2-52, A7, 1)		
	Prior performance records on similar work, if available. (L2-49, A2, 2)		
	Project risks involving the subcontractor's work. (L2-52, A7, 7)		
	Project's software development plan. (L2-47, A1, 2.4)		
	Project's software requirements. (L2-47, A1, 2.3)		
	Project's standards and procedures. (L2-47, A1, 2.5)		
	Project's statement of work. (L2-47, A1, 2.1) [Refer to SPF Standards for additional information regarding a Statement of Work.]		
	Project's system requirements allocated to software. (L2-47, A1, 2.2) [Refer to SPF Standards for additional information regarding allocated requirements.]		
	Proposals submitted for the planned subcontract. (L2-49, A2, 1)		
	Subcontracted products. (L2-45, C2, 3.3)		

Continued on next page

SSM Process - Inputs, Continued

Inputs, continued The table below lists the inputs to the software subcontract management process, continued from the previous page.

√	Input	Org. Input	References
	Subcontractor's cost performance. (L2-52, A7, 2)		
	Subcontractor's plans. (L2-54, A10, 1)		
	Subcontractor's procedures. (L2-54, A10, 1)		
	Subcontractor's products. (L2-54, A10, 2.1)		
	Subcontractor's resources. (L2-54, A10, 1)		
	Subcontractor's schedule performance. (L2-52, A7, 2)		
	Subcontractor's software baseline library. (L2-54, A11, 3)		
	Subcontractor's software engineering accomplishments and results (L2-53, A9)		
	Subcontractor's SQA records. (L2-54, A10, 2.2)		
	Subcontractor's staffing performance. (L2-52, A7, 2)		
	Subcontractor's standards. (L2-54, A10, 1)		
	Subcontractor's technical performance. (L2-52, A7, 2)		
	Technical and nontechnical characteristics of the project. (L2-47, A1, 1)		

SSM Process - Activities

Activities

The table below lists the required activities for the software project planning process.

√	Activities	References
	The work to be subcontracted is defined and planned according to a documented procedure. (L2-47, A1) [Refer to Documented Procedures for additional information.]	
	The software subcontractor is selected based on an evaluation of the subcontract bidders' ability to perform the work, according to a documented procedure. (L2-49, A2) [Refer to Documented Procedures for additional information.]	
	The contractual agreement between the prime contractor and the software subcontractor is used as the basis for managing the subcontract. (L2-50, A3) [Refer to SPF Standards for additional information regarding the contractual agreement.]	
	A documented subcontractor's software development plan is reviewed and approved by the prime contractor. (L2-51, A4) [Refer to Reviews and Audits for additional information.]	
	A documented and approved subcontractor's software development plan is used for tracking the software activities and communicating status. (L2-51, A5)	
	Changes to the software subcontractor's statement of work, subcontract terms and conditions, and other commitments are resolved according to a documented procedure. (L2-51, A6) [Refer to Documented Procedures for additional information.]	
	The prime contractor's management conducts periodic status/coordination reviews with the software subcontractor's management. (L2-51, A7) [Refer to Reviews and Audits for additional information.]	
	Periodic technical reviews and interchanges are held with the software subcontractor. (L2-52, A8) [Refer to Reviews and Audits for additional information.]	

Continued - next page

Activities, continued

The table below lists the required activities for the software subcontract process, continued from the previous page.

√	Activities	References
	Formal reviews to address the subcontractor's software engineering accomplishments and results are conducted at selected milestones according to a documented procedure. (L2-53, A9) [Refer to Documented Procedures for additional information.]	
	The prime contractor's software quality assurance group monitors the subcontractor's software quality assurance activities according to a documented procedure. (L2-53, A10) [Refer to Documented Procedures for additional information.]	
	The prime contractor's software configuration management group monitors the subcontractor's activities for software configuration management according to a documented procedure. (L2-54, A11) [Refer to Documented Procedures for additional information.]	
	The prime contractor conducts acceptance testing as part of the delivery of the subcontractor's software products according to a documented procedure. (L2-55, A12) [Refer to Documented Procedures for additional information.]	
	The software subcontractor's performance is evaluated on a periodic basis, and the evaluation is reviewed with the subcontractor. (L2-55, A13)	
	Measurements are made and used to determine the status of the activities for managing the software subcontract. (L2-55, M1)	
	The activities for managing the software subcontract are reviewed with senior management on a periodic basis. (L2-56, V1)	
	The activities for managing the software subcontract are reviewed with the project manager on both a periodic and event-driven basis. (L2-56, V2)	
	The software quality assurance group reviews and/or audits the activities and work products for managing the software subcontract and reports the results. (L2-57, V3) [Refer to Reviews and Audits for additional information.]	

SSM Process - Outputs

Outputs

The table below lists the outputs produced by the software subcontract management process.

√	Output	Org. Output	References
	Acceptance procedures and acceptance criteria for each product. (L2-55, A12, 1)		
	Action items resulting from periodic status/coordination reviews with the software subcontractor's management. (L2-52, A7, 9)		
	Action plan for any software product that does not pass its acceptance test. (L2-55, A12, 3)		
	Changes to the software subcontractor's SOW. (L2-51, A6)		
	Changes to the subcontract terms and conditions. (L2-51, A6)		
	Changes to the subcontract. (L2-45, C1, 3)		
	Contractual agreements. (L2-45, C1, 2)		
	Evaluation of the subcontract bidders' ability to perform the work. (L2-49, A2)		
	Functions or subsystems to be subcontracted. (L2-47, A1, 1.1)		
	Nonconformance to the subcontract. (L2-52, A7, 6)		
	Plan for selecting a subcontractor. (L2-48, A1, 4)		
	Results of the acceptance tests. (L2-55, A12, 2)		
	Significant issues, action items, and decisions resulting from formal reviews of the subcontractor's software engineering accomplishments. (L2-53, A9, 3)		
	Software products and activities to be subcontracted. (L2-47, A1, 1)		
	Specification of the software products and activities to be subcontracted. (L2-47, A1, 1.2)		
	Specification of work to be subcontracted. (L2-47, A1, 2)		

Continued on next page

SSM Process - Outputs, Continued

Outputs, continued

The table below lists the outputs produced by the software subcontract management process, continued from the previous page.

√	Output	Org. Output	References
	Subcontract statement of work. (L2-47, A1, 3)		
	Subcontractor software development plan. (L2-51, A4)		
	Work to be subcontracted. (L2-47, A1)		

SSM Process - Exit Criteria

Exit Criteria

The table below describes the conditions that must be satisfied in order to exit the software subcontract management process.

√	Condition	References
	Documented standards and procedures are used in selecting software subcontractors and managing the software subcontracts. (L2-45, C1, 1)	
	The contractual agreements form the basis for managing the subcontract. (L2-45, C1, 2)	
	Changes to the subcontract are made with the involvement and agreement of both the prime contractor and the subcontractor. (L2-45, C1, 3)	
	The work to be subcontracted is defined and planned according to a documented procedure. (L2-47, A1)	
	The software subcontractor is selected based on an evaluation of the subcontract bidders' ability to perform the work, according to a documented procedure. (L2-49, A2)	
	The contractual agreement between the prime contractor and the software subcontractor is used as the basis for managing the subcontract. (L2-50, A3)	
	A documented subcontractor's software development plan is reviewed and approved by the prime contractor. (L2-51, A4) ❑ This software development plan covers (directly or by reference) the appropriate items from the prime contractor's software development plan. (L2-51, A4, 1)	
	A documented and approved subcontractor's software development plan is used for tracking the software activities and communicating status. (L2-51, A5)	
	Changes to the software subcontractor's statement of work, subcontract terms and conditions, and other commitments are resolved according to a documented procedure. (L2-51, A6)	
	The prime contractor's management conducts periodic status/coordination reviews with the software subcontractor's management. (L2-51, A7)	

Continued on next page

SSM Process - Exit Criteria, Continued

Exit Criteria, continued

The table below describes the conditions that must be satisfied in order to exit the software subcontract management process, continued from the previous page.

√	Condition	References
	Periodic technical reviews and interchanges are held with the software subcontractor. (L2-52, A8)	
	Formal reviews to address the subcontractor's software engineering accomplishments and results are conducted at selected milestones according to a documented procedure. (L2-53, A9)	
	The prime contractor's software quality assurance group monitors the subcontractor's software quality assurance activities according to a documented procedure. (L2-53, A10)	
	The prime contractor's software configuration management group monitors the subcontractor's activities for software configuration management according to a documented procedure. (L2-54, A11)	
	The prime contractor conducts acceptance testing as part of the delivery of the subcontractor's software products according to a documented procedure. (L2-55, A12)	
	Measurements are made and used to determine the status of the activities for managing the software subcontract. (L2-55, M1)	
	The activities for managing the software subcontract are reviewed with senior management on a periodic basis. (L2-56, V1)	
	The activities for managing the software subcontract are reviewed with the project manager on both a periodic and event-driven basis. (L2-56, V2)	
	The software quality assurance group reviews and/or audits the activities and work products for managing the software subcontract and reports the results. (L2-57, V3)	

SSM Process - Reviews and Audits

Reviews and Audits

The table below lists the required reviews and audits for the software subcontract management process.

√	Review or Audit	Review Participants	References
	A documented subcontractor's software development plan is reviewed and approved by the prime contractor. (L2-51, A4) ❑ This software development plan covers (directly or by reference) the appropriate items from the prime contractor's software development plan.	Prime Contractor	
	The prime contractor's management conducts periodic status/coordination reviews with the software subcontractor's management. (L2-51, A7) ❑ The subcontractor is provided with visibility of the needs and desires of the product's customers and end users, as appropriate. ❑ The subcontractor's technical, cost, staffing, and schedule performance is reviewed against the subcontractor's software development plan. ❑ Computer resources designated as critical for the project are reviewed; the subcontractor's contribution to the current estimates are tracked and compared to the estimates for each software component as documented in the subcontractor's software development plan.	Prime Contractor's Management Subcontractor Customers End users	
	Review description continues on the next page		

Continued on next page

Reviews and Audits, continued

The table below lists the required reviews and audits for the software subcontract management process, continued from the previous page.

√	Review or Audit	Review Participants	References
	Review description continued from previous page		
	❑ Critical dependencies and commitments between the subcontractor's software engineering group and other subcontractor groups are addressed. ❑ Critical dependencies and commitments between the prime contractor and the subcontractor are addressed. ❑ Subcontractor commitments to the prime contractor and prime contractor commitments to the subcontractor are both reviewed. ❑ Nonconformance to the subcontract is addressed. ❑ Project risks involving the subcontractor's work are addressed. ❑ Conflicts and issues not resolvable internally by the subcontractor are addressed. ❑ Action items are assigned, reviewed, and tracked to closure.	Sub-contractor's Software Engineering Group Prime Contractor Subcontractor	

Continued on next page

Reviews and Audits, continued

The table below lists the required reviews and audits for the software subcontract process, continued from the previous page.

√	Review or Audit	Review Participants	References
	Periodic technical reviews and interchanges are held with the software subcontractor. (L2-52, A8) These reviews: ❑ Provide the subcontractor with visibility of the customer's and end users' needs and desires, as appropriate. ❑ Monitor the subcontractor's technical activities. ❑ Verify that the subcontractor's interpretation and implementation of the technical requirements conform to the prime contractor's requirements. ❑ Verify that commitments are being met. ❑ Verify that technical issues are resolved in a timely manner.	Software Subcontractor Subcontractor Customer End User	
	Formal reviews to address the subcontractor's software engineering accomplishments and results are conducted at selected milestones according to a documented procedure. (L53, A9) ❑ Reviews are preplanned and documented in the statement of work. (L53, A9, 1) ❑ Reviews address the subcontractor's commitments for, plans for, and status of the software activities. (L2-53, A9, 2)	Not specified in CMM	
	The subcontractor's plans, resources, procedures, and standards for software quality assurance are periodically reviewed to ensure they are adequate to monitor the subcontractor's performance. (L2-54, A10, 1)	Prime Contractor's SQA Group	

Continued on next page

Reviews and Audits, continued

The table below lists the required reviews and audits for the software subcontract process, continued from the previous page.

√	Review or Audit	Review Participants	References
	Regular reviews of the subcontractor are conducted to ensure the approved procedures and standards are being followed. (L2-54, A10, 2) ❑ The prime contractor's software quality assurance group spot checks the subcontractor's software engineering activities and products. ❑ The prime contractor's software quality assurance group audits the subcontractor's software quality assurance records, as appropriate.	Subcontractor Prime Contractor's SQA Group	
	The subcontractor's records of its software quality assurance activities are periodically audited to assess how well the software quality assurance plans, standards, and procedures are being followed. (L2-54, A10, 3)	Prime Contractor's SQA Group	
	The subcontractor's plans, resources, procedures, and standards for software configuration management are reviewed to ensure they are adequate. (L2-54, A11, 1)	Prime Contractor's SCM Group	
	The subcontractor's software baseline library is periodically audited to assess how well the standards and procedures for software configuration management are being followed and how effective they are in managing the software baseline. (L2-54, A11, 3)	Prime Contractor's SCM Group	
	The acceptance procedures and acceptance criteria for each product are defined, reviewed and approved by both the prime contractor and the subcontractor prior to the test. (L2-55, A12, 1)	Prime Contractor Subcontractor	
	The software subcontractor's performance is evaluated on a periodic basis, and the evaluation is reviewed with the subcontractor. (L2-55, A13)	Prime Contractor Subcontractor	

Continued on next page

SSM Process - Reviews and Audits, Continued

Reviews and Audits, continued

The table below lists the required reviews and audits for the software subcontract process, continued from the previous page.

√	Review or Audit	Review Participants	References
	The activities for managing the software subcontract are reviewed with senior management on a periodic basis. (L2-56, V1)	Senior Management	
	The activities for managing the software subcontract are reviewed with the project manager on both a periodic and event-driven basis. (L2-56, V2)	Project Manager	
	The software quality assurance group reviews and/or audits the activities and work products for managing the software subcontract and reports the results. (L2-57, V3) At a minimum, the reviews and/or audits verify: ❑ The activities for selecting the subcontractor. ❑ The activities for managing the software subcontract. ❑ The activities for coordinating configuration management activities of the prime contractor and subcontractor. ❑ The conduct of planned reviews with the subcontractor. ❑ The conduct of reviews that establish completion of key project milestones or stages for the subcontract. ❑ The acceptance process for the subcontractor's software products.	SQA Group Subcontractor Prime Contractor	

SSM Process - Work Products Managed and Controlled

Work Products Managed and Controlled

The table below lists the work products required to be managed and controlled during the software subcontract management process.

√	Work Products Managed and Controlled	References
	Subcontract statement of work. (L2-48, A1, 3.5)	

SSM Process - Measurements

Measurements

The table below lists the measurements required for the software subcontract management process.

√	Measurements	References
	Measurements are made and used to determine the status of the activities for managing the software subcontract. (L2-55, M1)	

SSM Process - Documented Procedures

Documented Procedures

The table below lists the software subcontract management activities which are required to be performed according to a documented procedure.

√	Documented Procedure(s)	References
	The work to be subcontracted is defined and planned according to a documented procedure. (L2-47, A1)	
	The software subcontractor is selected based on an evaluation of the subcontract bidders' ability to perform the work, according to a documented procedure. (L2-49, A2)	
	Changes to the software subcontractor's statement of work, subcontract terms and conditions, and other commitments are resolved according to a documented procedure. (L2-51, A6)	
	Formal reviews to address the subcontractor's software engineering accomplishments and results are conducted at selected milestones according to a documented procedure. (L2-53, A9)	
	The prime contractor's software quality assurance group monitors the subcontractor's software quality assurance activities according to a documented procedure. (L2-53, A10)	
	The prime contractor's software configuration management group monitors the subcontractor's activities for software configuration management according to a documented procedure. (L2-54, A11)	
	The prime contractor conducts acceptance testing as part of the delivery of the subcontractor's software products according to a documented procedure. (L2-55, A12)	

SSM Process - Training

Training

The table below lists the training required for the software subcontract management process.

√	Training	References
	Software managers and other individuals who are involved in establishing and managing the software subcontract are trained to perform these activities. (L2-46, Ab2)	
	Software managers and other individuals who are involved in managing the software subcontract receive orientation in the technical aspects of the subcontract. (L2-46, Ab3)	

SSM Process - Tools

Tools

The table below lists the tools required for the software subcontract management process.

√	Tools	References
	Tools to support managing the subcontract. (L2-46, Ab1, 2)	

Software Quality Assurance (SQA) Process

SQA Process - Overview

SQA Process Purpose

The purpose of Software Quality Assurance is to provide management with appropriate visibility into the process being used by the software project and of the products being built. (L2-59)

SQA Process Description

Software Quality Assurance involves reviewing and auditing the software products and activities to verify that they comply with the applicable procedures and standards and providing the software project and other appropriate managers with the results of these reviews and audits.

The software quality assurance group works with the software project during its early stages to establish plans, standards, and procedures that will add value to the software project and satisfy the constraints of the project and the organization's policies. By participating in establishing the plans, standards, and procedures, the software quality assurance group helps ensure they fit the project's needs and verifies that they will be usable for performing reviews and audits throughout the software life cycle. The software quality assurance group reviews project activities and audits software work products throughout the life cycle and provides management with visibility as to whether the software project is adhering to its established plans, standards, and procedures.

Compliance issues are first addressed within the software project and resolved there if possible. For issues not resolvable within the software project, the software quality assurance group escalates the issue to an appropriate level of management for resolution.

This key process area covers the practices for the group performing the software quality assurance function. The practices identifying the specific activities and work products that the software quality assurance group reviews and/or audits are generally contained in the Verifying Implementation common feature of the other key process areas. (L2-59)

Continued on next page

Chapter Overview

The table below contains the description and the location of each section in this chapter.

Section	Description	Page
Roles	List of roles participating in process activities.	SQA-3
Entry Criteria	Describes when the process can start.	SQA-8
Inputs	A description of the work products consumed by the process.	SQA-9
Activities	Describes the activities of the process.	SQA-10
Outputs	A description of the work products produced by the process.	SQA-12
Exit Criteria	Describes when the process is complete.	SQA-13
Reviews and Audits	List of required reviews and audits.	SQA-15
Work Products Managed and Controlled	Lists work products required to be managed and controlled.	SQA-17
Measurements	Describes required process measurements.	SQA-18
Documented Procedures	Lists which activities must be completed according to a documented procedure.	SQA-19
Training	List of required training.	SQA-20
Tools	List of required tools.	SQA-21

SQA Process - Roles

Roles

The table below lists the roles, and the activities in which they participate in the software quality assurance process.

√	Role	Activities Participated in...	Reference
	Affected Groups	The SQA plan is reviewed by the affected groups and individuals. (L2-63, A1, 2)	
	Customer	The deliverable software products are evaluated before they are delivered to the customer. (L2-66, A5, 1)	
	Customer SQA Personnel	The SQA group conducts periodic reviews of its activities and findings with the customer's SQA personnel, as appropriate. (L2-67, A8)	
	Experts Independent of the SQA Group	Experts independent of the SQA group periodically review the activities and software work products of the project's SQA group. (L2-69, V3)	
	Individuals	The SQA plan is reviewed by the affected groups and individuals. (L2-63, A1, 2)	
	Manager	❑ A manager is assigned specific responsibilities for the project's SQA activities. (L2-62, Ab2, 1) ❑ All managers in the SQA reporting chain to the senior manager are knowledgeable in the SQA role, responsibilities, and authority. (L2-62, Ab2, 2.1)	

Continued on next page

SQA Process - Roles, Continued

Roles, continued

The table below lists the roles, and the activities in which they participate in the software quality assurance process, continued from the previous page.

√	Role	Activities Participated in...	Reference
	Project Manager	❑ Deviations from the software development plan and the designated project standards and procedures are documented and resolved with the appropriate software task leaders, software managers, or project manager, where possible. (L2-67, A7, 1) ❑ Deviations from the software development plan and the designated project standards and procedures not resolvable with the software task leaders, software managers, or project manager are documented and presented to the senior manager designated to receive noncompliance items. (L2-67, A7, 2) ❑ The SQA activities are reviewed with the project manager on both a periodic and event-driven basis. (L2-69, V2)	
	Senior Management	❑ Senior management periodically reviews the SQA activities and results. (L2-61, C1, 3) ❑ The SQA activities are reviewed with senior management on a periodic basis. (L2-68, V1)	

Continued on next page

SQA Process - Roles, Continued

Roles, continued

The table below lists the roles, and the activities in which they participate in the software quality assurance process, continued from the previous page.

√	Role	Activities Participated in...	Reference
	Senior Manager	❑ A senior manager, who is knowledgeable in the SQA role and has the authority to take appropriate oversight actions, is designated to receive and act on software noncompliance items. (L2-62, Ab2, 2) ❑ Deviations from the software development plan and the designated project standards and procedures not resolvable with the software task leaders, software managers, or project manager are documented and presented to the senior manager designated to receive noncompliance items. (L2-67, A7, 2) ❑ Noncompliance items presented to the senior manager are periodically reviewed until they are resolved. (L2-67, A7, 3)	
	Software Engineering Group	The SQA group periodically reports the results of its activities to the software engineering group. (L2-67, A6)	
	Software Manager	❑ Deviations from the software development plan and the designated project standards and procedures are documented and resolved with the appropriate software task leaders, software managers, or project manager, where possible. (L2-67, A7, 1) ❑ Deviations from the software development plan and the designated project standards and procedures not resolvable with the software task leaders, software managers, or project manager are documented and presented to the senior manager designated to receive noncompliance items. (L2-67, A7, 2)	

Continued on next page

SQA Process - Roles, Continued

Roles, continued

The table below lists the roles, and the activities in which they participate in the software quality assurance process, continued from the previous page.

√	Role	Activities Participated in...	Reference
	Software Task Leader	❑ Deviations from the software development plan and the designated project standards and procedures are documented and resolved with the appropriate software task leaders, software managers, or project manager, where possible. (L2-67, A7, 1) ❑ Deviations from the software development plan and the designated project standards and procedures not resolvable with the software task leaders, software managers, or project manager are documented and presented to the senior manager designated to receive noncompliance items. (L2-67, A7, 2)	
	SQA Group	❑ Members of the SQA group are trained to perform their SQA activities. (L2-62, Ab3) ❑ The SQA group's activities are performed in accordance with the SQA plan. (L2-64, A2) ❑ The SQA group participates in the preparation and review of the project's software development plan, standards, and procedures. (L2-65, A3)	
		Role continues on next page	

Continued on next page

SQA Process - Roles, Continued

Roles, continued

The table below lists the roles, and the activities in which they participate in the software quality assurance process, continued from the previous page.

√	Role	Activities Participated in...	Reference
	SQA Group, continued	❑ The SQA group provides consultation and review of the plans, standards, and procedures with regard to: (L2-66, A3, 1) ❑ compliance to organizational policy, ❑ compliance to externally imposed standards and requirements (e.g., standards imposed by the statement of work), ❑ standards that are appropriate for use by the project, ❑ topics that should be addressed in the software development plan, and ❑ other areas assigned by the project. ❑ The SQA group verifies that plans, standards, and procedures are in place and can be used to review and audit the software project. (L2-66, A3, 2) ❑ The SQA group reviews the software engineering activities to verify compliance. (L2-66, A4) ❑ The SQA group audits designated software work products to verify compliance. (L2-66, A5) ❑ The SQA group periodically reports the results of its activities to the software engineering group. (L2-67, A6) ❑ The SQA group conducts periodic reviews of its activities and findings with the customer's SQA personnel, as appropriate. (L2-67, A8)	

SQA Process - Entry Criteria

Entry Criteria

The table below describes the conditions that must be satisfied in order to begin the software quality assurance process.

√	Condition	References
	The project follows a written organizational policy for implementing software quality assurance (SQA). (L2-60, C1) [Refer to SPF Policies for additional information regarding SQA policy.]	
	A group that is responsible for coordinating and implementing SQA for the project (i.e., the SQA group) exists. (L2-61, Ab1)	
	Adequate resources and funding are provided for performing the SQA activities. (L2-62, Ab2)	
	A manager is assigned specific responsibilities for the project's SQA activities. (L2-62, Ab2, 1)	
	A senior manager, who is knowledgeable in the SQA role and has the authority to take appropriate oversight actions, is designated to receive and act on software noncompliance items. (L2-62, Ab2, 2)	
	All managers in the SQA reporting chain to the senior manager are knowledgeable in the SQA role, responsibilities, and authority. (L2-62, Ab2, 2.1)	
	Tools to support the SQA activities are made available. (L2-62, Ab2, 3)	
	Members of the SQA group are trained to perform their SQA activities. (L2-62, Ab3)	
	The members of the software project receive orientation on the role, responsibilities, authority, and value of the SQA group. (L2-63, Ab4)	

SQA Process - Inputs

Inputs

The table below lists the inputs to the software quality assurance process.

√	Input	Org. Input	References
	Deliverable software products. (L2-66, A5, 1)		
	Designated contractual requirements. (L2-66, A5, 2)		
	Designated procedures. (L2-66, A4, 1)		
	Designated software standards. (L2-66, A4, 1)		
	Designated software work products. (L2-66, A5)		
	Externally imposed requirements. (L2-66, A3, 1.2)		
	Externally imposed standards. (L2-66, A3, 1.2)		
	Project's procedures. (L2-65, A3)		
	Project's software development plan. (L2-65, A3)		
	Project's standards. (L2-65, A3)		
	Software work products. (L2-66, A5, 2)		
	SQA plan. (L2-64, A2)		

SQA Process - Activities

Activities

The table below lists the required activities for the software quality assurance process.

√	Activities	References
	A SQA plan is prepared for the software project according to a documented procedure. (L2-63, A1) [Refer to Documented Procedures for additional information.]	
	The SQA group participates in the preparation and review of the project's software development plan, standards, and procedures. (L2-65, A3) ❑ The SQA group provides consultation and review of the plans, standards, and procedures with regard to: ❑ compliance to organizational policy, ❑ compliance to externally imposed standards and requirements (e.g., standards required by the statement of work), ❑ standards that are appropriate for use by the project, ❑ topics that should be addressed in the software development plan, and ❑ other areas as assigned by the project. ❑ The SQA group verifies that plans, standards, and procedures are in place and can be used to review and audit the software project.	
	The SQA group reviews the software engineering activities to verify compliance. (L2-66, A4) ❑ The activities are evaluated against the software development plan and the designated software standards and procedures. ❑ Deviations are identified, documented, and tracked to closure. ❑ Corrections are verified.	
	The SQA group audits designated software work products to verify compliance. (L2-66, A5) ❑ The deliverable software products are evaluated before they are delivered to the customer. ❑ The software work products are evaluated against the designated software standards, procedures, and contractual requirements. ❑ Deviations are identified, documented, and tracked to closure. ❑ Corrections are verified.	

Continued on next page

SQA Process - Activities, Continued

Activities, continued

The table below lists the required activities for the software quality assurance process, continued from the previous page.

√	Activities	References
	The SQA group periodically reports the results of its activities to the software engineering group. (L2-67, A6)	
	Deviations identified in the software activities and software work products are documented and handled according to a documented procedure. (L2-67, A7) [Refer to Documented Procedures for additional information.]	
	The SQA group conducts periodic reviews of its activities and findings with the customer's SQA personnel, as appropriate. (L2-67, A8)	
	Measurements are made and used to determine the cost and schedule status of the SQA activities. (L2-68, M1)	
	The SQA activities are reviewed with senior management on a periodic basis. (L2-68, V1)	
	The SQA activities are reviewed with the project manager on a periodic and event-driven basis. (L2-69, V2)	
	Experts independent of the SQA group periodically review the activities and software work products of the project's SQA group. (L2-69, V3)	

SQA Process - Outputs

Outputs

The table below lists the outputs produced by the software quality assurance process.

√	Output	Org. Output	References
	Corrections to deviations. (L2-66, A4, 3)		
	Deviations between the contractual requirements and the designated software work products. (L2-67, A5, 3)		
	Deviations between the designated software procedures and the designated software work products. (L2-67, A5, 3)		
	Deviations between the designated software procedures and the software engineering activities. (L2-66 A4, 2)		
	Deviations between the designated software standards and the designated software work products. (L2-67, A5, 3)		
	Deviations between the designated software standards and the software engineering activities. (L2-66 A4, 2)		
	Deviations between the software development plan and the software engineering activities. (L2-66 A4, 2)		
	Deviations from the software development plan and the designated project standards and procedures. (L2-67, A7, 1)		
	Deviations identified in the software activities. (L2-67, A7)		
	Deviations identified in the software work products. (L2-67, A7)		
	Documentation of noncompliance items. (L2-67, A7, 4)		
	Measurements. (L2-68, M1)		
	Noncompliance items. (L2-62, Ab2, 2)		
	Software work products of the SQA group. (L2-69, V3)		
	SQA findings. (L2-67, A8)		
	SQA plan. (L2-63, A1) [Refer to SPF Standards for additional information regarding a SQA plan.]		
	SQA results. (L2-61, C1, 3)		

SQA Process - Exit Criteria

Exit Criteria

The table below describes the conditions that must be satisfied in order to exit the software quality assurance process.

√	Condition	References
	A SQA Plan is prepared for the software project according to a documented procedure. (L2-63, A1)	
	The SQA group's activities are performed in accordance with the SQA plan. (L2-64, A2)	
	The SQA group participates in the preparation and review of the project's software development plan, standards, and procedures. (L2-65, A3) ❑ The SQA group provides consultation and review of the plans, standards, and procedures with regard to: ❑ compliance to organizational policy, ❑ compliance to externally imposed standards and requirements (e.g., standards required by the statement of work), ❑ standards that are appropriate for use by the project, ❑ topics that should be addressed in the software development plan, and ❑ other areas as assigned by the project. ❑ The SQA group verifies that plans, standards, and procedures are in place and can be used to review and audit the software project.	
	The SQA group reviews the software engineering activities to verify compliance. (L2-66, A4) ❑ The activities are evaluated against the software development plan and the designated software standards and procedures. ❑ Deviations are identified, documented, and tracked to closure. ❑ Corrections are verified.	
	The SQA group audits designated software work products to verify compliance. (L2-66, A5) ❑ The deliverable software products are evaluated before they are delivered to the customer. ❑ The software work products are evaluated against the designated software standards, procedures, and contractual requirements. ❑ Deviations are identified, documented, and tracked to closure. ❑ Corrections are verified.	

Continued on next page

SQA Process - Exit Criteria, Continued

Exit Criteria, continued

The table below describes the conditions that must be satisfied in order to exit the software quality assurance process, continued from the previous page.

√	Condition	References
	Results of SQA group activities are reported to the software engineering group. (L2-67, A6)	
	Deviations identified in the software activities and software work products are documented and handled according to a documented procedure. (L2-67, A7)	
	SQA findings are periodically reviewed with the customer's SQA personnel, as appropriate. (L2-67, A8)	
	Measurements of SQA activities are made and used to determine cost and schedule status. (L2-68, M1)	
	The SQA activities are reviewed with senior management on a periodic basis. (L2-68, V1)	
	The SQA activities are reviewed with the project manager on a periodic and event-driven basis. (L2-69, V2)	
	Experts independent of the SQA group periodically review the activities and software work products of the project's SQA group. (L2-69, V3)	

SQA Process - Reviews and Audits

Reviews and Audits

The table below lists the required reviews and audits for the software quality assurance process.

√	Review or Audit	Review Participants	References
	Senior management periodically reviews the SQA activities and results. (L2-61, C1, 3)	Senior Management	
	The SQA plan is reviewed by the affected groups and individuals. (L2-63, A1, 2)	Affected Groups Affected Individuals.	
	The SQA group participates in the preparation and review of the project's software development plan, standards, and procedures. (L2-65, A3)	SQA Group	
	The SQA group provides consultation and review of the plans, standards, and procedures with regard to: (L2-66, A3, 1) ❑ compliance to organizational policy, ❑ compliance to externally imposed standards and requirements (e.g., standards required by the statement of work), ❑ standards that are appropriate for use by the project, ❑ topics that should be addressed in the software development plan, and ❑ other areas as assigned by the project.	SQA Group	
	The SQA group reviews the software engineering activities to verify compliance. (L2-66, A4)	SQA Group	
	The SQA group audits designated software work products to verify compliance. (L2-66, A5)	SQA Group	
	The SQA group conducts periodic reviews of its activities and findings with the customer's SQA personnel, as appropriate. (L2-67, A8)	SQA Group Customer's SQA Personnel	
	The SQA activities are reviewed with senior management on a periodic basis. (L2-68, V1)	Senior Management	

Continued on next page

SQA Process - Reviews and Audits, Continued

Reviews and Audits, continued

The table below lists the required reviews and audits for the software quality assurance process, continued from the previous page.

√	Review or Audit	Review Participants	References
	The SQA activities are reviewed with the project manager on both a periodic and event-driven basis. (L2-69, V2)	Project Manager	
	Experts independent of the SQA group periodically review the activities and software work products of the project's SQA group. (L2-69, V3)	Experts Independent of the SQA Group SQA Group	

SQA Process - Work Products Managed and Controlled

Work Products Managed and Controlled

The table below lists the work products required to be managed and controlled during the software quality assurance process.

√	Work Products Managed and Controlled	References
	SQA plan. (L2-64, A1, 3)	
	Documentation of noncompliance items. (L2-67, A7, 4)	

SQA Process - Measurements

Measurements

The table below lists the measurements required for the software quality assurance process.

√	Measurements	References
	Measurements are made and used to determine the cost and schedule status of SQA activities. (L2-68, M1)	

SQA Process - Documented Procedures

Documented Procedures

The table below lists the software quality assurance process activities required to be performed according to a documented procedure.

√	Documented procedures	References
	A SQA plan is prepared for the software project according to a documented procedure. (L2-63, A1)	
	Deviations identified in the software activities and software work products are documented and handled according to a documented procedure. (L2-67, A7)	

SQA Process - Training

Training

The table below lists the training required for the software quality assurance process.

√	Training	References
	Members of the SQA group are trained to perform their SQA activities. (L2-62, Ab3)	
	The members of the software project receive orientation on the role, responsibilities, authority, and value of the SQA group. (L2-63, Ab4)	

SQA Process - Tools

Tools

The table below lists the tools required for the software quality assurance process.

√	Tools	References
	Tools to support the SQA activities. (L2-62, Ab2, 3)	

Software Configuration Management (SCM) Process

SCM Process - Overview

SCM Process Purpose

The purpose of Software Configuration Management is to establish and maintain the integrity of the products of the software project throughout the project's software life cycle. (L2-71)

SCM Process Description

Software Configuration Management involves identifying the configuration of the software (i.e., selected software work products and their descriptions) at given points in time, systematically controlling changes to the configuration, and maintaining the integrity and traceability of the configuration throughout the software life cycle. The work products placed under software configuration management include the software products that are delivered to the customer (e.g., the software requirements document and the code) and the items that are identified with or required to create these software products (e.g., the compiler).

A software baseline library is established containing the software baselines as they are developed. Changes to baselines and the release of software products built from the software baseline library are systematically controlled via the change control and configuration auditing functions of software configuration management.

This key process area covers the practices for performing the software configuration management function. The practices identifying specific configuration items/units are contained in the key process areas that describe the development and maintenance of each configuration item/unit. (L2-71)

Continued on next page

Chapter Overview

The below table contains the description and location of each section in this chapter.

Section	Description	Page
Roles	List of roles participating in process activities.	SCM-3
Entry Criteria	Describes when the process can start.	SCM-7
Inputs	A description of the work products consumed by the process.	SCM-8
Activities	Describes the activities of the process.	SCM-9
Outputs	A description of the work products produced by the process.	SCM-11
Exit Criteria	Describes when the process is complete.	SCM-12
Reviews and Audits	List of required reviews and audits.	SCM-14
Work Products Managed and Controlled	Lists work products required to be managed and controlled.	SCM-16
Measurements	Describes required process measurements.	SCM-17
Documented Procedures	Lists which activities must be completed according to a documented procedure.	SCM-18
Training	List of required training.	SCM-19
Tools	List of required tools.	SCM-20

SCM Process - Roles

Roles

The table below lists the roles, and the activities in which they participate in the software configuration management process.

√	Role	Activities Participated in...	Reference
	Affected Groups	❑ The SCM plan is reviewed by the **affected groups.** ❑ The configuration management library system provides for the sharing and transfer of configuration items/units between the **affected groups** and between control levels within the library. (L2-78, A3, 3) ❑ Standard reports documenting the SCM activities and the contents of the software baseline are developed and made available to **affected groups** and individuals. (L2-81, A9)	
	Individuals	Standard reports documenting the SCM activities and the contents of the software baseline are developed and made available to affected groups and **individuals.** (L2-81, A9)	
	Manager	A **manager** is assigned specific responsibilities for SCM. (L2-75, Ab3, 1)	
	Person Responsible foe Each Configuration Unit/Item	The **person responsible for each configuration item/unit** (i.e., the owner, from a configuration management point of view) is identified. (L2-79, A4, 6)	
	Project Manager	The SCM activities are reviewed with the **project manager** on both a periodic and event-driven basis. (L2-83, V2)	
	Project Software Manager	The results of software baseline audits are reported to the **project software manager.** (L2-82, A10, 6)	
	Senior Management	The SCM activities are reviewed with **senior management** on a periodic basis. (L2-82, V1)	

Continued on next page

SCM Process - Roles, Continued

Roles, continued

The table below lists the roles, and the activities in which they participate in the software configuration management process, continued from the previous page.

√	Role	Activities Participated in...	Reference
	SCCB	❑ The SCCB: (L2-73, Ab1) ❑ Authorizes the establishment of software baselines and the identification of configuration items/units. ❑ Represents the interests of the project manager and all groups *who may be affected* by changes to the software baselines. ❑ Reviews and authorizes changes to the software baselines. ❑ Authorizes the creation of products from the software baseline library. ❑ Only configuration items/units that are approved by the SCCB are entered into the software baseline library. (L2-80, A6, 2)	

Continued on next page

SCM Process - Roles, Continued

Roles, continued

The table below lists the roles, and the activities in which they participate in the software configuration management process, continued from the previous page.

√	Role	Activities Participated in...	Reference
	SCM Group	❑ A group that is responsible for coordinating and implementing SCM for the project (i.e., the SCM group) exists. (L2-74, Ab2) The SCM group coordinates or implements: ❑ Creation and management of the project's software baseline library. ❑ Development, maintenance, and distribution of the SCM plans, standards, and procedures. ❑ The identification of the set of work products to be placed under SCM. ❑ Management of the access to the software baseline library. ❑ Updates of the software baselines. ❑ Creation of products from the software baseline library. ❑ Recording of SCM actions. ❑ Production and distribution of SCM reports. ❑ Members of the SCM group are trained in the objectives, procedures, and methods for performing their SCM activities. (L2-76, Ab4) ❑ The SCM group periodically audits software baselines to verify that they conform to the documentation that defines them. (L2-83, V3)	
	Software Engineering Group	Members of the software engineering group and other software-related groups are trained to perform their SCM activities. (L2-76, Ab5)	

Continued on next page

SCM Process - Roles, Continued

Roles, continued

The table below lists the roles, and the activities in which they participate in the software configuration management process, continued from the previous page.

√	Role	Activities Participated in...	Reference
	Software-related Groups	❑ Members of the software engineering group and other software-related groups are trained to perform their SCM activities. (L2-76, Ab5) ❑ The SCM plan covers the SCM requirements and activities to be performed by the software engineering group and other software-related groups. (L2-77, A2, 2)	
	SQA Group	The software quality assurance group reviews and/or audits the activities and work products for SCM and reports the results. (L2-83, V4)	

SCM Process - Entry Criteria

Entry Criteria

The table below describes the conditions that must be satisfied in order to begin the software configuration management process.

√	Condition	References
	The project follows a written organizational policy for implementing software configuration management (SCM). (L2-72, C1) [Refer to SPF Policies for additional information regarding SCM policy.]	
	A board having the authority for managing the project's software baselines (i.e., a software configuration control board - SCCB) exists or is established. (L2-73, Ab1)	
	A group that is responsible for coordinating and implementing SCM for the project (i.e., the SCM group) exists. (L2-74, Ab2)	
	Adequate resources and funding are provided for performing the SCM activities. (L2-75, Ab3)	
	A manager is assigned specific responsibilities for SCM. (L2-75, Ab3, 1)	
	Tools to support the SCM activities are made available. (L2-75, Ab3, 2)	
	Members of the SCM group are trained in the objectives, procedures, and methods for performing their SCM activities. (L2-76, Ab4)	
	Members of the software engineering group and other software-related groups are trained to perform their SCM activities. (L2-76, Ab5)	

SCM Process - Inputs

Inputs

The table below lists the inputs to the software configuration management process.

√	Input	Org. Input	References
	Change requests for configuration items/units. (L2-79, A5)		
	Changes to baselines. (L2-80, A6)		
	Configuration items/units. (L2-78, A3, 2)		
	Designated internal software work products. (L2-72, C1, 3)		
	Designated support tools used inside the project. (L2-72, C1, 3)		
	Externally deliverable software products. (L2-72, C1, 3)		
	Problem reports for configuration items/units. (L2-79, A5)		
	SCM plan. (L2-77, A2) [Refer to SPF Standards for additional information regarding a SCM plan.]		
	Software baselines. (L2-73, C1, 5)		
	Software work products. (L2-78, A4)		
	Updates of the software baselines. (L2-75, Ab2, 5)		
	Work products (to be placed under SCM). (L2-75, Ab2, 3)		

SCM Process - Activities

Activities

The table below lists the required activities for the software configuration management process.

√	Activities	References
	A SCM plan is prepared for each software project according to a documented procedure. (L2-76, A1) [Refer to Documented Procedures for additional information.]	
	A documented and approved SCM plan is used as the basis for performing the SCM activities. (L2-77, A2)	
	A configuration management library system is established as a repository for the software baselines. (L2-77, A3) [Refer to Tools for additional information regarding a configuration management library system.]	
	The software work products to be placed under configuration management are identified. (L2-78, A4) ❑ The configuration items/units are selected based on documented criteria. ❑ The configuration items/units are assigned unique identifiers. ❑ The characteristics of each configuration item/unit are specified. ❑ The software baselines to which each configuration item/unit belongs are specified. ❑ The point in its development that each configuration item/unit is placed under configuration management is specified. ❑ The person responsible for each configuration item/unit (i.e., the owner, from a configuration management point of view) is identified.	
	Change requests and problem reports for all configuration items/units are initiated, recorded, reviewed, approved, and tracked according to a documented procedure. (L2-79, A5)	
	Changes to baselines are controlled according to a documented procedure. (L2-80, A6) [Refer to Documented Procedures for additional information.]	

Continued on next page

SCM Process - Activities, Continued

Activities, continued

The table below lists the required activities for the software configuration management process, continued from the previous page.

√	Activities	References
	Products from the software baseline library are created and their release is controlled according to a documented procedure. (L2-80, A7,) [Refer to Documented Procedures for additional information.]	
	The status of configuration items/units is recorded according to a documented procedure. (L2-80, A8) [Refer to Documented Procedures for additional information.]	
	Standard reports documenting the SCM activities and the contents of the software baseline are developed and made available to affected groups and individuals. (L2-81, A9)	
	Software baseline audits are conducted according to a documented procedure. (L2-81, A10) [Refer to Documented Procedures for additional information.]	
	Measurements are made and used to determine the status of the SCM activities. (L2-82, M1)	
	The SCM activities are reviewed with senior management on a periodic basis. (L2-82, V1)	
	The SCM activities are reviewed with the project manager on both a periodic and event-driven basis. (L2-83, V2)	
	The SCM group periodically audits software baselines to verify that they conform to the documentation that defines them. (L2-83, V3)	
	The software quality assurance group reviews and/or audits the activities and work products for SCM and reports the results. (L2-83, V4) [Refer to Reviews and Audits for additional information.]	

SCM Process - Outputs

Outputs

The table below lists the outputs produced by the software configuration management process.

√	Output	Org. Output	References
	Action items from the software baseline audit. (L2-82, A10, 7)		
	Archive versions of configuration items/units. (L2-78, A3, 5)		
	Changes to the software baselines. (L2-74, Ab1, 3)		
	Configuration items/units. (L2-73, Ab1, 1)		
	Current status and history (i.e., changes and other actions) of each configuration item/unit. (L2-81, A8, 2)		
	Measurements. (L2-82, M1)		
	Products from the software baseline library. (L2-74, Ab1, 4)		
	Project's software baseline library (or repository). (L2-75, Ab2, 1)		
	Results of the software baseline audit. (L2-82, A10, 6)		
	SCM actions. (L2-75, Ab2, 7)		
	SCM plan. (L2-75, Ab2, 2)		
	SCM procedures. (L2-75, Ab2, 2)		
	SCM records. (L2-72, C1, 4)		
	SCM reports. (L2-75, Ab2, 8)		
	SCM standards. (L2-75, Ab2, 2)		
	Software baselines. (L2-73, Ab1)		
	Standard reports documenting the SCM activities and the contents of the software baseline. (L2-81, A9)		
	Work products (to be placed under SCM). (L2-75, Ab2, 3)		

SCM Process - Exit Criteria

Exit Criteria

The table below describes the conditions that must be satisfied in order to exit the software configuration management process.

√	Condition	References
	The SCCB: (L2-73, Ab1) ❑ Authorizes the establishment of software baselines and the identification of configuration items/units. ❑ Represents the interests of the project manager and all groups who may be affected by changes to the software baselines. ❑ Reviews and authorizes changes to the software baselines. ❑ Authorizes the creation of products from the software baseline library.	
	A SCM plan is prepared for each software project according to a documented procedure. (L2-76, A1)	
	A documented and approved SCM plan is used as the basis for performing the SCM activities. (L2-77, A2)	
	A configuration management library system is established as a repository for the software baselines. (L2-77, A3)	
	The software work products to be placed under configuration management are identified. (L2-78, A4) ❑ The configuration items/units are selected based on documented criteria. ❑ The configuration items/units are assigned unique identifiers. ❑ The characteristics of each configuration item/unit are specified. ❑ The software baselines to which each configuration item/unit belongs are specified. ❑ The point in its development that each configuration item/unit is placed under configuration management is specified. ❑ The person responsible for each configuration item/unit (i.e., the owner, from a configuration management point of view) is identified.	
	Change requests and problem reports for all configuration items/units are initiated, recorded, reviewed, approved, and tracked according to a documented procedure. (L2-79, A5)	
	Changes to baselines are controlled according to a documented procedure. (L2-80, A6)	

Continued on next page

SCM Process - Exit Criteria, Continued

Exit Criteria, continued

The table below describes the conditions that must be satisfied in order to exit the software configuration management process, continued from the previous page.

√	Condition	References
	Products from the software baseline library are created and their release is controlled according to a documented procedure. (L2-80, A7)	
	The status of configuration items/units is recorded according to a documented procedure. (L2-80, A8)	
	Standard reports documenting the SCM activities and the contents of the software baseline are developed and made available to affected groups and individuals. (L2-81, A9)	
	Software baseline audits are conducted according to a documented procedure. (L2-81, A10)	
	Measurements are made and used to determine the status of the SCM activities. (L2-82, M1)	
	The SCM activities are reviewed with senior management on a periodic basis. (L2-82, V1)	
	The SCM activities are reviewed with the project manager on both a periodic and event-driven basis. (L2-83, V2)	
	The SCM group periodically audits software baselines to verify that they conform to the documentation that defines them. (L2-83, V3)	
	The software quality assurance group reviews and/or audits the activities and work products for SCM and reports the results. (L2-83, V4)	

SCM Process - Reviews and Audits

Reviews and Audits

The table below lists the required reviews and audits for the software configuration management process.

√	Review or Audit	Review Participants	References
	The software baselines and SCM activities are audited on a periodic basis. (L2-73, C1, 5)	Not Specified in CMM	
	The SCCB reviews and authorizes changes to the software baselines. (L2-74, Ab1, 3)	SCCB	
	The SCM plan is reviewed by the affected groups. (L2-77, A1, 2)	Affected Groups	
	Change requests and problem reports for all configuration items/units are initiated, recorded, reviewed, approved, and tracked according to a documented procedure. (L2-79, A5)	Not Specified in CMM	
	Reviews and/or regression tests are performed to ensure that changes have not caused unintended effects on the baseline. (L2-80, A6, 1)	Not Specified in CMM	
	Software baseline audits are conducted according to a documented procedure. (L2-81, A10)	Not Specified in CMM	
	The SCM activities are reviewed with senior management on a periodic basis. (L2-82, V1)	Senior Manager	
	The SCM activities are reviewed with the project manager on both a periodic and event-driven basis. (L2-83, V2)	Project Manager	
	The SCM group periodically audits software baselines to verify that they conform to the documentation that defines them. (L2-83, V3)	SCM Group	

Continued on next page

SCM Process - Reviews and Audits, continued

Reviews and Audits, continued

The table below lists the required reviews and audits for the software configuration management process, continued from the previous page.

√	Review or Audit	Review Participants	References
	The software quality assurance group reviews and/or audits the activities and work products for SCM and reports the results. (L2-83, V4) At a minimum, the reviews and/or audits verify: ❑ Compliance with the SCM standards and procedures by: ❑ the SCM group, ❑ the SCCB, ❑ the software engineering group, and ❑ other software-related groups. ❑ Occurrence of periodic baseline audits.	SQA Group SCM Group SCCB Software Engineering Group Softwa related roups	

SCM Process - Work Products Managed and Controlled

Work Products Managed and Controlled

The table below lists the work products required to be managed and controlled during the configuration management process.

√	Work Products Managed and Controlled	References
	The SCM plan. (L2-77, A1, 3)	

SCM Process - Measurements

Measurements

The table below lists the measurements required for the software configuration management process.

√	Measurements	References
	Measurements are made and used to determine the status of the SCM activities. (L2-82, M1)	

SCM Process - Documented Procedures

Documented Procedures

The table below lists the software configuration management process activities required to be performed according to a documented procedure.

√	Documented procedures	References
	A SCM plan is prepared for each software project according to a documented procedure. (L2-76, A1)	
	Change requests and problem reports for all configuration items/units are initiated, recorded, reviewed, approved, and tracked according to a documented procedure. (L2-79, A5)	
	Changes to baselines are controlled according to a documented procedure. (L2-80, A6)	
	Products from the software baseline library are created and their release is controlled according to a documented procedure. (L2-80, A7)	
	The status of configuration items/units is recorded according to a documented procedure. (L2-80, A8)	
	Software baseline audits are conducted according to a documented procedure. (L2-81, A10)	

SCM Process - Training

Training

The table below lists the training required for the software configuration management process.

√	Training	References
	Members of the SCM group are trained in the objectives, procedures, and methods for performing their SCM activities. (L2-76, Ab4)	
	Members of the software engineering group and other software-related groups are trained to perform their SCM activities. (L2-76, Ab5)	

SCM Process - Tools

Tools

The table below lists the tools required for the software configuration management process.

√	Tools	References
	Tools to support the SCM activities. (L2-75, Ab3, 2)	
	A configuration management library system is established as a repository for the software baselines. (L2-77, A3) *This library system:* ❑ Supports multiple control levels of SCM. ❑ Provides for the storage and retrieval of configuration items/units. ❑ Provides for the sharing and transfer of configuration items/units between the affected groups and between control levels within the library. ❑ Helps in the use of product standards for configuration items/units. ❑ Provides for the storage and recovery of archive versions of configuration items/units. ❑ Helps to ensure correct creation of products from the software baseline library. ❑ Provides for the storage, update, and retrieval of SCM records. ❑ Supports production of SCM reports. ❑ Provides for the maintenance of the library structure and contents.	

Chapter 6

Procedure Checklists

Overview

Chapter Purpose

The purpose of the procedure checklists is to provide:

- provide guidance in identifying which procedures are required by the CMM.
- provide criteria that an organization can use to evaluate its software procedures to determine if those procedures are consistent with the CMM.
- provide information that can be used to develop software procedures that are consistent with the CMM.

Chapter Definitions

procedure: describes "how-to" or "step-by-step" instructions that implement a process in a repeatable way.

"A documented procedure is usually needed so that the individuals responsible for a task or activity are able to perform it in a repeatable way and so that others with general knowledge of the area will be able to learn and perform the task or activity in the same way. This is one aspect of institutionalizing a process."

"The formality and level of detail of a documented procedure can vary significantly, from a hand-written individual desk procedure to a formal organizational standard operating procedure. The formality and level of detail depends on who will perform the task or activity (e.g., individual or team), how often it is performed, the importance and intended use of the results, and the intended recipients of the results." [CMM, Overview, O-42]

In This Chapter

This chapter covers the following topics:

Requirements Management (RM) Procedures

Documented Procedures

There are no required documented procedures for requirements management.

Software Project Planning (SPP) Procedures

Documented Procedures

The table below lists the required documented procedures for software project planning.

√	Documented Procedures	References
	Software project commitments made to individuals and groups external to the organization are reviewed with senior management according to a documented procedure. (L2-17, A4)	
	The project's software development plan is developed according to a documented procedure. (L2-18, A6) This procedure typically specifies that: ❑ The software development plan is based on and conforms to: ❑ the customer's standards, as appropriate; ❑ the project's standards; ❑ the approved statement of work; and ❑ the allocated requirements. ❑ Plans for software-related groups and other engineering groups involved in the activities of the software engineering group are negotiated with those groups, the support efforts are budgeted, and the agreements are documented. ❑ Plans for involvement of the software engineering group in the activities of other software-related groups and other engineering groups are negotiated with those groups, the support efforts are budgeted, and the agreements are documented. ❑ The software development plan is reviewed by: ❑ the project manager, ❑ the project software manager, ❑ the other software managers, and ❑ other affected groups. ❑ The software development plan is managed and controlled.	

Continued on next page

Documented Procedures, Continued

The table below lists the required documented procedures for software project planning, continued from the previous page.

√	Documented Procedures	References
	Estimates for the size of the software work products (or changes to the size of software work products) are derived according to a documented procedure. (L2-21, A9) This procedure typically specifies that: ❑ Size estimates are made for all major software work products and activities. ❑ Software work products are decomposed to the granularity needed to meet the estimating objectives. ❑ Historical data are used where available. ❑ Size estimating assumptions are documented. ❑ Size estimates are documented, reviewed, and agreed to.	
	Estimates for the software project's effort and costs are derived according to a documented procedure. (L2-22, A10) This procedure typically specifies that: ❑ Estimates for the software project's effort and costs are related to the size estimates of the software work products (or the size of the changes). ❑ Productivity data (historical and/or current) are used for the estimates when available; sources and rationale for these data are documented. ❑ The productivity and cost data are from the organization's projects when possible. ❑ The productivity and cost data take into account the effort and significant costs that go into making the software work products. ❑ Effort, staffing, and cost estimates are based on past experience. ❑ Similar projects should be used when possible. ❑ Time phasing of activities is derived. ❑ Distributions of the effort, staffing, and cost estimates over the software life cycle are prepared. ❑ Estimates and the assumptions made in deriving the estimates are documented, reviewed, and agreed to.	

Continued on next page

Documented Procedures, Continued

The table below lists the required documented procedures for software project planning, continued from the previous page.

√	Documented Procedures	References
	Estimates for the project's critical computer resources are derived according to a documented procedure. (L2-23, A11) This procedure typically specifies that: ❑ Critical computer resources for the project are identified. ❑ Estimates for the critical computer resources are related to the estimates of: ❑ the size of the software work products, ❑ the operational processing load, and ❑ the communications traffic. ❑ Estimates of the critical computer resources are documented, reviewed, and agreed to.	
	The project's software schedule is derived according to a documented procedure. (L2-23, A12) This procedure typically specifies that: ❑ The software schedule is related to: ❑ the size estimate of the software work products (or the size of changes), and ❑ the software effort and costs. ❑ The software schedule is based on past experience. ❑ Similar projects are used when possible. ❑ The software schedule accommodates the imposed milestone dates, critical dependency dates, and other constraints. ❑ The software schedule activities are of appropriate duration and the milestones are of appropriate time separation to support accuracy in progress measurement. ❑ Assumptions made in deriving the schedule are documented. ❑ The software schedule is documented, reviewed, and agreed to.	

Software Project Tracking & Oversight (SPTO) Procedures

Documented Procedures

The table below lists the required documented procedures for software process tracking and oversight.

√	Documented Procedures	References
	The project's software development plan is revised according to a documented procedure. (L2-33, A2) This procedure typically specifies that: ❑ The software development plan is revised, as appropriate, to incorporate plan refinements and incorporate plan changes, particularly when plans change significantly. ❑ The software development plan is updated to incorporate all new software project commitments and changes to commitments. ❑ The software development plan is reviewed at each revision. ❑ The software development plan is managed and controlled.	
	Software project commitments and changes to commitments made to individuals and groups external to the organization are reviewed with senior management according to a documented procedure. (L2-35. A3)	
	Formal reviews to address the accomplishments and results of the software project are conducted at selected project milestones according to a documented procedure. (L2-39. A13)	

Software Subcontract Management (SSM) Procedures

Documented Procedures

The table below lists the required documented procedures for software subcontract management.

√	Documented Procedures
	The work to be subcontracted is defined and planned according to a documented procedure. (L2-47, A1) This procedure typically specifies that: ❑ The software products and activities to be subcontracted are selected based on a balanced assessment of both technical and nontechnical characteristics of the project. (L2-47, A1, 1) ❑ The functions or subsystems to be subcontracted are selected to match the skills and capabilities of potential subcontractors. ❑ The specification of the software products and activities to be subcontracted is determined based on a systematic analysis and appropriate partitioning of the system and software requirements. ❑ The specification of the work to be subcontracted and the standards and procedures to be followed are derived from the project's: (L2-47, A1, 2) ❑ statement of work, ❑ system requirements allocated to software, ❑ software requirements, ❑ software development plan, and ❑ software standards and procedures. ❑ A subcontract statement of work is: (L2-47, A1, 3) ❑ prepared, ❑ reviewed, ❑ agreed to, ❑ revised when necessary, and ❑ managed and controlled. ❑ A plan for selecting a subcontractor is prepared concurrent with the subcontract statement of work and is reviewed, as appropriate. (L2-47, A1, 4)

Continued on next page

Documented Procedures, Continued

The table below lists the required documented procedures for software subcontract management, continued from the previous page.

√	Documented Procedures
	The software subcontractor is selected based on an evaluation of the subcontract bidders' ability to perform the work, according to a documented procedure. (L2-49, A2) This procedure covers the evaluation of: ❑ Proposals submitted for the planned subcontract. (L2-49, A2, 1) ❑ Prior performance records on similar work, if available. (L2-49, A2, 2) ❑ The geographic locations of the subcontract bidders' organizations relative to the prime contractor. (L2-49, A2, 3) ❑ Software engineering and software management capabilities. (L2-49, A2, 4) ❑ Staff available to perform the work. (L2-49, A2, 5) ❑ Prior experience in similar applications, including software expertise on the subcontractor's software management team. (L2-49, A2, 6) ❑ Available resources. (L2-49, A2, 7)
	Changes to the software subcontractor's statement of work, subcontract terms and conditions, and other commitments are resolved according to a documented procedure. (L2-51, A6) This procedure typically specifies that: ❑ All affected groups of both the prime contractor and the subcontractor are involved. (L2-51, A6, 1)

Continued on next page

Documented Procedures, Continued

The table below lists the required documented procedures for software subcontract management, continued from the previous page.

√	Documented Procedures
	Formal reviews to address the subcontractor's software engineering accomplishments and results are conducted at selected milestones according to a documented procedure. (L2-53, A9) This procedure typically specifies that: ❑ Reviews are preplanned and documented in the statement of work. (L2-53, A9, 1) ❑ Reviews address the subcontractor's commitments for, plans for, and status of the software activities. (L2-53, A9, 2) ❑ Significant issues, action items, and decisions are identified and documented. (L2-53, A9, 3) ❑ Software risks are addressed. (L2-53, A9, 4) ❑ The subcontractor's software development plan is refined, as appropriate. (L2-53, A9, 5)
	The prime contractor's software quality assurance group monitors the subcontractor's software quality assurance activities according to a documented procedure. (L2-53, A10) This procedure typically specifies that: ❑ The subcontractor's plans, resources, procedures, and standards for software quality assurance are periodically reviewed to ensure they are adequate to monitor the subcontractor's performance. (L2-53, A10, 1) ❑ Regular reviews of the subcontractor are conducted to ensure the approved procedures and standards are being followed. (L2-53, A10, 2) ❑ The prime contractor's software quality assurance group spot checks the subcontractor's software engineering activities and products. ❑ The prime contractor's software quality assurance group audits the subcontractor's software quality assurance records, as appropriate. ❑ The subcontractor's records of its software quality assurance activities are periodically audited to assess how well the software quality assurance plans, standards, and procedures are being followed. (L2-53, A10, 3)

Continued on next page

Documented Procedures, Continued

The table below lists the required documented procedures for software subcontract management, continued from the previous page.

√	Documented Procedures
	The prime contractor's software configuration management group monitors the subcontractor's activities for software configuration management according to a documented procedure. (L2-54, A11) This procedure typically specifies that: ❑ The subcontractor's plans, resources, procedures, and standards for software configuration management are reviewed to ensure they are adequate. (L2-54, A11, A1) ❑ The prime contractor and the subcontractor coordinate their activities on matters relating to software configuration management to ensure that the subcontractor's products can be readily integrated or incorporated into the project environment of the prime contractor. (L2-54, A11, A2) ❑ The subcontractor's software baseline library is periodically audited to assess how well the standards and procedures for software configuration management are being followed and how effective they are in managing the software baseline. (L2-54, A11, A3)
	The prime contractor conducts acceptance testing as part of the delivery of the subcontractor's software products according to a documented procedure. (L2-55, A12) This procedure typically specifies that: ❑ The acceptance procedures and acceptance criteria for each product are defined, reviewed, and approved by both the prime contractor and the subcontractor prior to the test. (L2-55, A12, 1) ❑ The results of the acceptance tests are documented. (L2-55, A12, 2) ❑ An action plan is established for any software product that does not pass its acceptance test. (L2-55, A12, 3)

Software Quality Assurance (SQA) Procedures

Documented Procedures

The table below lists the required documented procedures for software quality assurance.

√	Documented Procedures	References
	A SQA plan is prepared for the software project according to a documented procedure. (L2-63, A1) This procedure typically specifies that: ❑ The SQA plan is developed in the early stages of, and in parallel with, the overall project planning. ❑ The SQA plan is reviewed by the affected groups and individuals. ❑ The SQA plan is managed and controlled.	
	Deviations identified in the software activities and software work products are documented and handled according to a documented procedure. (L2-67, A7) This procedure typically specifies that: ❑ Deviations from the software development plan and the designated project standards and procedures are documented and resolved with the appropriate software task leaders, software managers, or project manager, where possible. ❑ Deviations from the software development plan and the designated project standards and procedures not resolvable with the software task leaders, software managers, or project manager are documented and presented to the senior manager designated to receive noncompliance items. ❑ Noncompliance items presented to the senior manager are periodically reviewed until they are resolved. ❑ The documentation of noncompliance items is managed and controlled.	

Software Configuration Management (SCM) Procedures

Documented Procedures

The table below lists the required documented procedures for software configuration management.

√	Documented Procedures	References
	A SCM plan is prepared for each software project according to a documented procedure. (L2-76, A1) This procedure typically specifies that: ❑ The SCM plan is developed in the early stages of, and in parallel with, the overall project planning. ❑ The SCM plan is reviewed by the affected groups. ❑ The SCM plan is managed and controlled.	
	Change requests and problem reports for all configuration items/units are initiated, recorded, reviewed, approved, and tracked according to a documented procedure. (L2-79, A5)	
	Changes to baselines are controlled according to a documented procedure. (L2-80, A6) This procedure typically specifies that: ❑ Reviews and/or regression tests are performed to ensure that changes have not caused unintended effects on the baseline. ❑ Only configuration items/units that are approved by the SCCB are entered into the software baseline library. ❑ Configuration items/units are checked in and out in a manner that maintains the correctness and integrity of the software baseline library.	
	Products from the software baseline library are created and their release is controlled according to a documented procedure. (L2-80, A7) This procedure typically specifies that: ❑ The SCCB authorizes the creation of products from the software baseline library. ❑ Products from the software baseline library, for both internal and external use, are built only from configuration items/units in the software baseline library.	

Continued on next page

Documented Procedures, Continued

The table below lists the required documented procedures for software configuration management, continued from the previous page.

√	Documented Procedures	References
	The status of configuration items/units is recorded according to a documented procedure. (L2-80, A8) This procedure typically specifies that: ❑ The configuration management actions are recorded in sufficient detail so that the content and status of each configuration item/unit are known and previous versions can be recovered. ❑ The current status and history (i.e., changes and other actions) of each configuration item/unit are maintained.	
	Software baseline audits are conducted according to a documented procedure. (L2-81, A10). This procedure typically specifies that: ❑ There is adequate preparation for the audit. ❑ The integrity of software baselines is assessed. ❑ The structure and facilities of the configuration management library system are reviewed. ❑ The completeness and correctness of the software baseline library contents are verified. ❑ Compliance with applicable SCM standards and procedures is verified. ❑ The results of the audit are reported to the **project software manager**. ❑ Action items from the audit are tracked to closure.	

Chapter 7

Cross-References for Level 2 of the CMM

Overview

Chapter Purpose

The purpose of this chapter is to provide checklists that have a view across the entire Repeatable Level (Level 2). Chapters 3-6 provided views of the Operational Framework from the perspective of specific key process areas (KPAs). This chapter provides a high level view of the Operational Framework across all of Level 2. Other viewpoints across Level 2 are provided such as KPA purposes, KPA goals, reviews and audits, work products managed and controlled, and measurements.

Chapter Overview

The table below contains a description and the location of each section in this chapter.

Section	Description	Page
KPA Purposes	Checklist of KPA purposes for Level 2.	XRef - 2
KPA Goals	Checklist of KPA goals for Level 2.	XRef - 3
Policies	Checklist of required policies.	XRef - 5
Standards	Checklist of required content of work products (standards)	XRef - 8
Process Descriptions	List of process descriptions at Level 2.	XRef - 9
Procedures	Checklist of required documented procedures.	XRef - 12
Training	Checklist of required training.	XRef - 15
Tools	Checklist of required tools.	XRef - 16
Reviews and Audits	Checklist of required reviews and audits.	XRef - 17
Work Products Managed and Controlled	Checklist or work products required to be managed and controlled.	XRef - 22
Measurements	Checklist of CMM required measurements.	XRef - 23

The Purposes of the CMM Level 2 KPAs

KPA Purposes in the CMM at Level 2

The following table describes the purposes of the KPAs in the CMM at Level 2.

√	KPA	Purpose of KPAs at Level 2
	RM	The purpose of Requirements Management is to establish a common understanding between the customer and the software project of the customer's requirements that will be addressed by the software project. (L2-1, P)
	SPP	The purpose of Software Project Planning is to establish reasonable plans for performing the software engineering and for managing the software project. (L2-11, P)
	SPTO	The purpose of Software Project Tracking and Oversight is to provide adequate visibility into actual progress so that management can take effective actions when the software project's performance deviates significantly from the software plans. (L2-29, P)
	SSM	The purpose of Software Subcontract Management is to select qualified software subcontractors and manage them effectively. (L2-43, P)
	SQA	The purpose of Software Quality Assurance is to provide management with appropriate visibility into the process being used by the software project and of the products being built. (L2-59, P)
	SCM	The purpose of Software Configuration Management is to establish and maintain the integrity of the products of the software project throughout the project's software life cycle. (L2-71, P)

The Goals of the CMM Level 2 KPAs

KPA Goals in the CMM at Level 2

The following table lists the goals that are described in the CMM at Level 2 for each key process area.

√	KPA	CMM Goals at Level 2	References
	RM	System requirements allocated to software are controlled to establish a baseline for software engineering and management use. (L2-2, G1)	
	RM	Software plans, products, and activities are kept consistent with the system requirements allocated to software. (L2-2, G2)	
	SPP	Software estimates are documented for use in planning and tracking the software project. (L2-12, G1)	
	SPP	Software project activities and commitments are planned and documented. (L2-12, G2)	
	SPP	Affected groups and individuals agree to their commitments related to the software project. (L2-12, G3)	
	SPTO	Actual results and performances are tracked against the software plans. (L2-30, G1)	
	SPTO	Corrective actions are taken and managed to closure when actual results and performance deviate significantly from the software plans. (L2-30, G2)	
	SPTO	Changes to software commitments are agreed to by the affected groups and individuals. (L2-30, G3)	
	SSM	The prime contractor selects qualified software subcontractors. (L2-44, G1)	
	SSM	The prime contractor and the software subcontractor agree to their commitments to each other. (L2-44, G2)	
	SSM	The prime contractor and the software subcontractor maintain ongoing communications. (L2-44, G3)	
	SSM	The prime contractor tracks the software subcontractor's actual results and performance against its commitments. (L2-44, G4)	

Continued on next page

The Goals of CMM Maturity Level 2, Continued

KPA Goals in the CMM at Level 2, Continued

The following table lists the goals that are described in the CMM at Level 2 for each key process area, continued from the previous page.

√	KPA	CMM Goals at Level 2	References
	SQA	Software quality assurance activities are planned. (L2-60, G1)	
	SQA	Adherence of software products and activities to applicable standards, procedures, and requirements is verified objectively. (L2-60, G2)	
	SQA	Affected groups and individuals are informed of software quality assurance activities and results. (L2-60, G3)	
	SQA	Noncompliance issues that cannot be resolved within the software project are addressed by senior management. (L2-60, G4)	
	SCM	Software configuration management activities are planned. (L2-72, G1)	
	SCM	Selected software work products are identified, controlled, and available. (L2-72, G2)	
	SCM	Changes to identified software work products are controlled. (L2-72, G3)	
	SCM	Affected groups and individuals are informed of the status and content of software baselines. (L2-72, G4)	

CMM Level 2 - Policies

CMM Level 2 Policies

The following table lists the required policies in the CMM at Level 2.

√	KPA	Description	References
	RM	The allocated requirements are documented. (L2-3, C1, 1)	
	RM	The allocated requirements are reviewed by: (L2-3, C1, 2) ❑ the software managers, and ❑ other affected groups.	
	RM	The software plans, work products, and activities are changed to be consistent with changes to the allocated requirements. (L2-3, C1, 3)	
	SPP	The system requirements allocated to software are used as the basis for planning the software project. (L2-12, C2, 1)	
	SPP	The software engineering groups' commitments are negotiated between: (L2-12, C2, 2) ❑ the project manager, ❑ the project software manager, and ❑ the other software managers.	
	SPP	Involvement of other engineering groups in the software activities is negotiated with these groups and is documented. (L2-13, C2, 3)	
	SPP	Affected groups review the project's: (L2-13, C2, 4) ❑ software size estimates, ❑ effort and cost estimates, ❑ schedules, and ❑ other commitments.	
	SPP	Senior management reviews all software project commitments made to individuals and groups external to the organization. (L2-13, C2, 5)	
	SPP	The project's software development plan is managed and controlled. (L2-13, C2, 6)	

Continued on next page

CMM Level 2 - Policies, Continued

CMM Level 2 Policies, Continued

The following table lists the required policies in the CMM at Level 2, continued from the previous page.

√	KPA	Description	References
	SPTO	A documented software development plan is used and maintained as the basis for tracking the software project. (L2-30, C2, 1)	
	SPTO	The project manager is kept informed of the software project's status and issues. (L2-30, C2, 2)	
	SPTO	Corrective actions are taken when the software plan is not being achieved, either by adjusting performance or by adjusting the plans. (L2-30, C2, 3)	
	SPTO	Changes to the software commitments are made with the involvement and agreement of the affected groups. (L2-30, C2, 4)	
	SPTO	Senior management reviews all commitment changes and new software project commitments made to individuals and groups external to the organization. (L2-31, C2, 5)	
	SSM	Documented standards and procedures are used in selecting software subcontractors and managing the software subcontracts. (L2-45, C1, 1)	
	SSM	The contractual agreements form the basis for managing the subcontract. (L2-45, C1, 2)	
	SSM	Changes to the subcontract are made with the involvement and agreement of both the prime contractor and the subcontractor. (L2-45, C1, 3)	
	SQA	The SQA function is in place on all software projects. (L2-60, C1, 1)	
	SQA	The SQA group has a reporting channel to senior management that is independent of: (L2-61, C1, 2) ❑ the project manager, ❑ the project's software engineering group, ❑ other software-related groups.	
	SQA	Senior management periodically reviews the SQA activities and results. (L2-61, C1, 3)	

Continued on next page

CMM Level 2 - Policies, Continued

CMM Level 2 Policies, Continued

The following table lists the required policies in the CMM at Level 2, continued from the previous page.

√	KPA	Description	References
	SCM	Responsibility for SCM for each project is explicitly assigned. (L2-72, C1, 1)	
	SCM	SCM is implemented throughout the project's life cycle. (L2-72, C1, 2)	
	SCM	SCM is implemented for externally deliverable software products, designated internal software work products, and designated support tools used inside the project (e.g., compilers). (L2-72, C1, 3)	
	SCM	The projects establish or have access to a repository for storing configuration items/units and the associated SCM records. (L2-72, C1, 4)	
	SCM	The software baselines and SCM activities are audited on a periodic basis. (L2-73, C1, 5)	

CMM Level 2 - Standards

Standards for CMM Level 2

The CMM prescribes the contents of the following work products at Level 2:

√	KPA	Standards at Level 2	References
	RM	Allocated Requirements (L2-4, Ab2, 1-3)	
	SPP	Statement of Work (L2-14, Ab1, 1)	
	SPP	Software Development Plan (L2-19, A7, 1-10)	
	SSM	Contractual Agreement (L2-50, A3, 1-8)	
	SQA	Software Quality Assurance Plan (L2-64, A2, 1-10)	
	SCM	Software Configuration Management Plan (L2-77, A2, 1-2)	

CMM Level 2 - Process Descriptions

RM Process Description

Requirements Management involves establishing and maintaining an agreement with the customer on the requirements for the software project. This agreement is referred to as the "system requirements allocated to the software." The "customer" may be interpreted as the system engineering group, the marketing group, another internal organization, or an external customer. The agreement covers both the technical and nontechnical (e.g., delivery dates) requirements. The agreement forms the basis for estimating, planning, performing, and tracking the software project's activities throughout the software life cycle.

The allocation of the system requirements to software, hardware, and other system components (e.g., humans) may be performed by a group external to the software engineering group (e.g., the system engineering group), and the software engineering group may have no direct control of this allocation. Within the constraints of the project, the software engineering group takes appropriate steps to ensure that the system requirements allocated to software, which they are responsible for addressing, are documented and controlled.

To achieve this control, the software engineering group reviews the initial and revised system requirements allocated to software to resolve issues before they are incorporated into the software project. Whenever the system requirements allocated to software are changed, the affected software plans, work products, and activities are adjusted to remain consistent with the updated requirements. (L2-1)

SPP Process Description

Software Project Planning involves developing estimates for the work to be performed, establishing the necessary commitments, and defining the plan to perform the work.

The software planning begins with a statement of the work to be performed and other constraints and goals that define and bound the software project (those established by the practices of the Requirements Management key process area). The software planning process includes steps to estimate the size of the software work products and the resources needed, produce a schedule, identify and assess software risks, and negotiate commitments. Iterating through these steps may be necessary to establish the plan for the software project (i.e., the software development plan).

This plan provides the basis for performing and managing the software project's activities and addresses the commitments to the software project's customer according to the resources, constraints, and capabilities of the software project. (L2-11)

Continued on next page

SPTO Process Description

Software Project Tracking and Oversight involves tracking and reviewing the software accomplishments and results against documented estimates, commitments, and plans, and adjusting these plans based on the actual accomplishments and results.

A documented plan for the software project (i.e., the software development plan, as described in the Software Project Planning key process area) is used as the basis for tracking the software activities, communicating status, and revising plans. Software activities are monitored by the management. Progress is primarily determined by comparing the actual software size, effort, cost, and schedule to the plan when selected software work products are completed and at selected milestones. When it is determined that the software project's plans are not being met, corrective actions are taken. These actions may include revising the software development plan to reflect the actual accomplishments and replanning the remaining work or taking actions to improve the performance. (L2-29)

SSM Process Description

Software Subcontract Management involves selecting a software subcontractor, establishing commitments with the subcontractor, and tracking and reviewing the subcontractor's performance and results. These practices cover the management of a software (only) subcontract, as well as the management of the software component of a subcontract that includes software, hardware, and possibly other system components.

The subcontractor is selected based on its ability to perform the work. Many factors contribute to the decision to subcontract a portion of the prime contractor's work. Subcontractors may be selected based on strategic business alliances, as well as technical considerations. The practices of this key process area address the traditional acquisition process associated with subcontracting a defined portion of the work to another organization.

When subcontracting, a documented agreement covering the technical and nontechnical (e.g., delivery dates) requirements is established and is used as the basis for managing the subcontract. The work to be done by the subcontractor and the plans for the work are documented. The standards that are to be followed by the subcontractor are compatible with the prime contractor's standards.

The software planning, tracking, and oversight activities for the subcontracted work are performed by the subcontractor. The prime contractor ensures that these planning, tracking, and oversight activities are performed appropriately and that the software products delivered by the subcontractor satisfy their acceptance criteria. The prime contractor works with the subcontractor to manage their product and process interfaces. (L2-43)

Continued on next page

CMM Level 2 - Processes, Continued

SQA Process Description

Software Quality Assurance involves reviewing and auditing the software products and activities to verify that they comply with the applicable procedures and standards and providing the software project and other appropriate managers with the results of these reviews and audits.

The software quality assurance group works with the software project during its early stages to establish plans, standards, and procedures that will add value to the software project and satisfy the constraints of the project and the organization's policies. By participating in establishing the plans, standards, and procedures, the software quality assurance group helps ensure they fit the project's needs and verifies that they will be usable for performing reviews and audits throughout the software life cycle. The software quality assurance group reviews project activities and audits software work products throughout the life cycle and provides management with visibility as to whether the software project is adhering to its established plans, standards, and procedures.

Compliance issues are first addressed within the software project and resolved there if possible. For issues not resolvable within the software project, the software quality assurance group escalates the issue to an appropriate level of management for resolution.

This key process area covers the practices for the group performing the software quality assurance function. The practices identifying the specific activities and work products that the software quality assurance group reviews and/or audits are generally contained in the Verifying Implementation common feature of the other key process areas. (L2-59)

SCM Process Description

Software Configuration Management involves identifying the configuration of the software (i.e., selected software work products and their descriptions) at given points in time, systematically controlling changes to the configuration, and maintaining the integrity and traceability of the configuration throughout the software life cycle. The work products placed under software configuration management include the software products that are delivered to the customer (e.g., the software requirements document and the code) and the items that are identified with or required to create these software products (e.g., the compiler).

A software baseline library is established containing the software baselines as they are developed. Changes to baselines and the release of software products built from the software baseline library are systematically controlled via the change control and configuration auditing functions of software configuration management.

This key process area covers the practices for performing the software configuration management function. The practices identifying specific configuration items/units are contained in the key process areas that describe the development and maintenance of each configuration item/unit. (L2-71)

CMM Level 2 - Procedures

Procedure Checklist

The table below lists the activities required to be performed according to a documented procedure in the CMM at Level 2.

√	KPA	Documented Procedures	References
√	RM	There are no required procedures for the RM process.	
	SPP	Software project commitments made to individuals and groups external to the organization are reviewed with senior management according to a documented procedure. (L2-17, A4)	
	SPP	The project's software development plan is developed according to a documented procedure. (L2-18, A6)	
	SPP	Estimates for the size of the software work products (or changes to the size of software work products) are derived according to a documented procedure. (L2-21, A9)	
	SPP	Estimates for the software project's effort and costs are derived according to a documented procedure. (L2-22, A10)	
	SPP	Estimates for the project's critical computer resources are derived according to a documented procedure. (L2-23, A11)	
	SPP	The project's software schedule is derived according to a documented procedure. (L2-23, A12)	
	SPTO	The project's software development plan is revised according to a documented procedure. (L2-33, A2)	
	SPTO	Software project commitments and changes to commitments made to individuals and groups external to the organization are reviewed with senior management according to a documented procedure. (L2-35, A3)	
	SPTO	Formal reviews to address the accomplishments and results of the software project are conducted at selected project milestones according to a documented procedure. (L2-39, A13)	

Continued on next page

CMM Level 2 - Procedures, Continued

Procedure Checklist, Continued

The table below lists the activities required to be performed according to a documented procedure in the CMM at Level 2, continued from the previous page.

√	KPA	Documented Procedures	References
	SSM	The work to be subcontracted is defined and planned according to a documented procedure. (L2-47, A1)	
	SSM	The software subcontractor is selected based on an evaluation of the subcontract bidders' ability to perform the work, according to a documented procedure. (L2-49, A2)	
	SSM	Changes to the software subcontractor's statement of work, subcontract terms and conditions, and other commitments are resolved according to a documented procedure. (L2-51, A6)	
	SSM	Formal reviews to address the subcontractor's software engineering accomplishments and results are conducted at selected milestones according to a documented procedure. (L2-53, A9)	
	SSM	The prime contractor's software quality assurance group monitors the subcontractor's software quality assurance activities according to a documented procedure. (L2-53, A10)	
	SSM	The prime contractor's software configuration management group monitors the subcontractor's activities for software configuration management according to a documented procedure. (L2-54, A11)	
	SSM	The prime contractor conducts acceptance testing as part of the delivery of the subcontractor's software products according to a documented procedure. (L2-55, A12)	
	SQA	A SQA plan is prepared for the software project according to a documented procedure. (L2-63, A1)	

Continued on next page

Procedure Checklist, Continued

The table below lists the activities required to be performed according to a documented procedure in the CMM at Level 2, continued from the previous page.

√	KPA	Documented Procedures	References
	SQA	Deviations identified in the software activities and software work products are documented and handled according to a documented procedure. (L2-67, A7)	
	SCM	A SCM plan is prepared for each software project according to a documented procedure. (L2-76, A1)	
	SCM	Change requests and problem reports for all configuration items/units are initiated, recorded, reviewed, approved, and tracked according to a documented procedure. (L2-79, A5)	
	SCM	Changes to baselines are controlled according to a documented procedure. (L2-80, A6)	
	SCM	Products from the software baseline library are created and their release is controlled according to a documented procedure. (L2-80, A7)	
	SCM	The status of configuration items/units is recorded according to a documented procedure (L2-80, A8)	
	SCM	Software baseline audits are conducted according to a documented procedure. (L2-81, A10).	

CMM Level 2 - Training

Training Checklist

The table below lists the training required in the CMM at Level 2.

√	KPA	Training	References
	RM	Members of the software engineering group and other software-related groups are trained to perform their requirements management activities. (L2-5, Ab4)	
	SPP	The software managers, software engineers, and other individuals involved in the software project planning are trained in the software estimating and planning procedures applicable to their areas of responsibility. (L2-16, Ab4)	
	SPTO	The software managers are trained in managing the technical and personnel aspects of the software project. (L2-32, Ab4)	
	SPTO	First-line software managers receive orientation in the technical aspects of the software project. (L2-32, Ab5)	
	SSM	Software managers and other individuals who are involved in establishing and managing the software subcontract are trained to perform these activities. (L2-46, Ab2)	
	SSM	Software managers and other individuals who are involved in managing the software subcontract receive orientation in the technical aspects of the subcontract. (L2-46, Ab3)	
	SQA	Members of the SQA group are trained to perform their SQA activities. (L2-62, Ab3)	
	SQA	The members of the software project receive orientation on the role, responsibilities, authority, and value of the SQA group. (L2-63, Ab4)	
	SCM	Members of the SCM group are trained in the objectives, procedures, and methods for performing their SCM activities. (L2-76, Ab4)	
	SCM	Members of the software engineering group and other software-related groups are trained to perform their SCM activities. (L2-76, Ab5)	

CMM Level 2 - Tools

Tools Checklist The table below lists the tool requirements in the CMM for Level 2.

√	KPA	Tools	References
	RM	Tools to support the activities for managing requirements. (L2-5, Ab3, 2)	
	SPP	Tools to support software project planning activities. (L2-16, Ab3, 2)	
	SPTO	Tools to support software tracking. (L2-32, Ab3, 2)	
	SSM	Tools to support managing the subcontract. (L2-46, Ab1, 2)	
	SQA	Tools to support the SQA activities. (L2-62, Ab2, 3)	
	SCM	Tools to support the SCM activities. (L2-75, Ab3, 2)	
	SCM	A configuration management library system is established as a repository for the software baselines. (L2-77, A3) This library system: ❑ Supports multiple control levels of SCM. ❑ Provides for the storage and retrieval of configuration items/units. ❑ Provides for the sharing and transfer of configuration items/units between the affected groups and between control levels within the library. ❑ Helps in the use of product standards for configuration items/units. ❑ Provides for the storage and recovery of archive versions of configuration items/units. ❑ Helps to ensure correct creation of products from the software baseline library. ❑ Provides for the storage, update, and retrieval of SCM records. ❑ Supports production of SCM reports. ❑ Provides for the maintenance of the library structure and contents.	

CMM Level 2 - Reviews and Audits

Reviews and Audits

The table below lists the major reviews and audits in the CMM at Level 2.

√	KPA	Review or Audit	Review Participants	References
	RM	The software engineering group reviews allocated requirements before they are incorporated into the software project. (L2-5, A1)	Software Engineering Group	
	RM	Changes to the allocated requirements are reviewed and incorporated into the software project. (L2-7, A3)	Not specified in CMM	
	RM	The activities for managing requirements are reviewed with senior management on a periodic basis. (L2-9, V1)	Senior Management	
	RM	The activities for managing the allocated requirements are reviewed with the project manager on both a periodic and event-driven basis. (L2-9, V2)	Project Manager	
	RM	The software quality assurance group reviews and/or audits the activities and work products for managing the allocated requirements and reports the results. (L2-9, V3)	SQA Group	
	SPP	Software project commitments made to individuals and groups external to the organization are reviewed with senior management. (L2-17, A4)	Senior Management	
	SPP	The activities for software project planning are reviewed with senior management on a periodic basis. (L2-26, V1)	Senior Management	
	SPP	The activities for software project planning are reviewed with the project manager on both a periodic and event-driven basis. (L2-26, V2)	Project Manager	

Continued on next page

CMM Level 2 - Reviews and Audits, Continued

Reviews and Audits, Continued

The table below lists the major reviews and audits in the CMM at Level 2, continued from the previous page.

√	KPA	Review or Audit	Review Participants	References
	SPP	The software quality assurance group reviews and/or audits the activities and work products for software project planning and reports the results (L2-27, V3)	SQA Group	
	SPTO	Software project commitments and changes to commitments made to individuals and groups external to the organization are reviewed with senior management according to a documented procedure. (L2-35, A3)	Senior Management	
	SPTO	The software engineering group conducts periodic internal reviews to track technical progress, plans, performance, and issues against the software development plan. (L2-38, A12)	Software Engineering Group First-line Software Managers Project Software Manager Software Managers Software Task Leaders	
	SPTO	Formal reviews to address the accomplishments and results of the software project are conducted at selected project milestones according to a documented procedure. (L2-39, A13)	Customer End User Software Managers	
	SPTO	The activities for software project tracking and oversight are reviewed with senior management on a periodic basis. (L2-40, V1)	Senior Management	

Continued on next page

CMM Level 2 - Reviews and Audits, Continued

Reviews and Audits, Continued

The table below lists the major reviews and audits in the CMM at Level 2, continued from the previous page.

√	KPA	Review or Audit	Review Participants	References
	SPTO	The activities for software project tracking and oversight are reviewed with the project manager on both a periodic and event-driven basis. (L2-41, V2)	Project Manager	
	SPTO	The software quality assurance group reviews and/or audits the activities and work products for software project tracking and oversight and reports the results. (L2-41, V3)	SQA Group	
	SSM	A documented subcontractor's software development plan is reviewed and approved by the prime contractor. (L2-51, A4)	Prime Contractor	
	SSM	The prime contractor's management conducts periodic status/coordination reviews with the software subcontractor's management. (L2-51, A7)	Prime Contractor's Management Sub-Contractor's Management	
	SSM	Periodic technical reviews and interchanges are held with the software subcontractor. (L2-52, A8)	Software Sub-Contractor	
	SSM	Formal reviews to address the subcontractor's software engineering accomplishments and results are conducted at selected milestones according to a documented procedure. (L2-53, A9)	Prime Contractor	
	SSM	The software subcontractor's performance is evaluated on a periodic basis, and the evaluation is reviewed with the subcontractor. (L2-55, A13)	Prime Contractor Sub-Contractor	

Continued on next page

CMM Level 2 - Reviews and Audits, Continued

Reviews and Audits, Continued

The table below lists the major reviews and audits in the CMM at Level 2, continued from the previous page.

√	KPA	Review or Audit	Review Participants	References
	SSM	The activities for managing the software subcontract are reviewed with senior management on a periodic basis. (L2-56, V1)	Senior Management	
	SSM	The activities for managing the software subcontract are reviewed with the project manager on both a periodic and event-driven basis. (L2-56, V2)	Project Manager	
	SSM	The software quality assurance group reviews and/or audits the activities and work products for managing the software subcontract and reports the results. (L2-57, V3)	SQA Group	
	SQA	The SQA group participates in the preparation and review of the project's software development plan, standards, and procedures. (L2-65, A3)	SQA Group	
	SQA	The SQA group reviews the software engineering activities to verify compliance. (L2-66, A4)	SQA Group	
	SQA	The SQA group audits designated software work products to verify compliance. (L2-66, A5)	SQA Group	
	SQA	The SQA group conducts periodic reviews of its activities and findings with the customer's SQA personnel, as appropriate. (L2-67, A8)	SQA Group Customer's SQA Personnel	
	SQA	The SQA activities are reviewed with senior management on a periodic basis. (L2-68, V1)	Senior Management	

Continued on next page

CMM Level 2 - Reviews and Audits, Continued

Reviews and Audits, Continued

The table below lists the major reviews and audits in the CMM at Level 2, continued from the previous page.

√	KPA	Review or Audit	Review Participants	References
	SQA	The SQA activities are reviewed with the project manager on both a periodic and event-driven basis. (L2-69, V2)	Project Manager	
	SQA	Experts independent of the SQA group periodically review the activities and software work products of the project's SQA group. (L2-69, V3)	Experts Independent of the SQA Group and SQA Group	
	SCM	Change requests and problem reports for all configuration items/units are initiated, recorded, reviewed, approved, and tracked according to a documented procedure. (L2-79, A5)	Not Specified in CMM	
	SCM	Software baseline audits are conducted according to a documented procedure. (L2-81, A10)	Not Specified in CMM	
	SCM	The SCM activities are reviewed with senior management on a periodic basis. (L2-82, V1)	Senior Management	
	SCM	The SCM activities are reviewed with the project manager on both a periodic and event-driven basis. (L2-83, V2)	Project Manager	
	SCM	The SCM group periodically audits software baselines to verify that they conform to the documentation that defines them. (L2-83, V3)	SCM Group	
	SCM	The software quality assurance group reviews and/or audits the activities and work products for SCM and reports the results. (L2-83, V4)	SQA Group	

CMM Level 2 - Work Products Managed and Controlled

Work Products Managed and Controlled

The table below lists the work products that are required to managed and controlled in the CMM at Level 2.

√	KPA	Work Products Managed and Controlled	References
	RM	Allocated requirements. (L2-7, A2, 1)	
	SPP	Project's software development plan. (L2-13, C2, 6)	
	SPP	Statement of work. (L2-15, Ab1, 3)	
	SPP	Software planning data. (L2-25, A15, 2)	
	SPTO	Software development plan. (L2-34, A2, 4) (Same as Project's SDP above in SPP)	
	SPTO	Software replanning data. (L2-38, A11, 2)	
	SSM	Subcontract statement of work. (L2-48, A1, 3.5)	
	SQA	The SQA plan. (L2-64, A1, 3)	
	SQA	The documentation of noncompliance items. (L2-67, A7, 4)	
	SCM	The SCM plan. (L2-77, A1, 3)	

CMM Level 2 - Measurements

Measurements

The table below describes the required measurements in the CMM at Level 2.

√	KPA	Description	References
	RM	Measurements are made and used to determine the status of the activities for managing the allocated requirements. (L2-8, M1)	
	SPP	Software planning data. (L2-25, A15) ❑ Information recorded includes the estimates and the associated information needed to reconstruct the estimates and assess their reasonableness. (L2-25, A15, 1)	
	SPP	Estimates and the associated information needed to reconstruct the estimates and assess their reasonableness. (L2-25, A15, 1)	
	SPP	Measurements are made and used to determine the status of the software planning activities. (L2-25, M1)	
	SPTO	Actual measurement data and replanning data for the software project are recorded. (L2-38, A11)	
	SPTO	Measurements are made and used to determine the status of the software tracking and oversight activities. (L2-39, M1)	
	SSM	Measurements are made and used to determine the status of the activities for managing the software subcontract. (L2-55, M1)	
	SQA	Measurements are made and used to determine the cost and schedule status of the SQA activities. (L2-68, M1)	
	SCM	Measurements are made and used to determine the status of the SCM activities. (L2-82, M1)	

References

[Brooks75] Brooks, Frederick P. *No Silver Bullet.* Reading, MA: Addison-Wesley, 1975.

[Covey89] Covey, Stephen R. *The 7 Habits of Highly Effective People.* Fireside, Simon & Schuster, New York, N.Y., 1989.

[Crosby79] Crosby, P. B. *Quality is Free.* New York, NY: McGraw-Hill, 1979.

[Curtis87] Curtis, Bill, Herb Krasner, Vincent Shen, and Neil Iscoe. "On Building Software Process Models Under the Lamppost," pp. 96-103. *Proceeding of the Ninth International Conference on Software Engineering,* Monterey, CA: IEEE Computer Society, March 30 - April 2, 1987.

[Deming82a] Deming, W. E. *Quality, Productivity, and Competitive Position.* Cambridge, MA: Massachusetts Institute of Technology Center for Advanced Engineering Study, 1982.

[Deming82b] Deming, W. Edward. *Out of the Crisis.* Cambridge, MA: Massachusetts Institute of Technology Center for Advanced Engineering Study, 1982.

[Dion92] Dion, Raymond. "Elements of a Process Improvement Program." *IEEE Software,* July 1992, pp. 83-85.

[Fowler90] Fowler, Priscilla, and Stan Rifkin. *Software Engineering Process Group Guide,* CMU/SEI-90-TR-24, September 1990.

[Humphrey87] Humphrey, W. S., and W. L. Sweet. *A Method for Assessing Software Engineering Capability of Contractors - Preliminary Technical Report* CMU/SEI-87-TR-23 (ADA200542). Pittsburgh, PA: Carnegie Mellon University, 1987.

[Humphrey88] Humphrey, Watts S. "Characterizing the Software Process: A Maturity Framework." *IEEE Software 5,* 2 (March 1988), pp. 73-79.

[Humphrey89a] Humphrey, W. S. *Managing the Software Process.* Reading, MA: Addison-Wesley Publishing Company, 1989.

[Humphrey89b] Humphrey, Watts S., David H. Kitson, and Tim C. Kasse. *The State of Software Engineering Practice: A Preliminary Report.* CMU/SEI-89-TR-1 (ADA206573). Pittsburgh, PA: Carnegie Mellon University, February 1989.

[Humphrey91] Humphrey, Watts S., Terry R. Snyder, and Ronald R. Willis. "Software Process Improvement at Hughes Aircraft." *IEEE Software,* July 1991, pp. 11-23.

[Ishikawa86] Ishikawa, K. *Guide to Quality Control.* Tokyo, Japan: Asian Productivity Organization, 1986.

[Juran89] Juran, Joseph M. *Juran on Leadership for Quality: An Executives Handbook.* New York, NY: The Free Press, 1989.

[Juran88a] Juran, J. M. *Juran on Planning for Quality.* New York, NY: Macmillan, 1988.

[Juran88b] Juran, J. M. *Juran's Quality Control Handbook.* Fourth Edition, New York, NY: McGraw Hill, 1988.

[Kitson92] Kitson, David, H., and Stephen Masters. *An Analysis of SEI Software Process Assessment Results: 1987-1991,* Software Engineering Institutute, CMU/SEI-92-TR-24, July 1992.

[Olson89] Olson, Timothy G., Watts Humphrey, and David Kitson. *Conducting SEI-Assisted Software Process Assessments.* CMU/SEI-89-TR-7 (ADA219065). Pittsburgh, PA: Carnegie Mellon University, 1989.

[Paulk93a] Paulk, Mark C., Bill Curtis, and Mary Beth Chrissis. *Capability Maturity Model for Software, Version 1.1* (CMU/SEI-93-TR-24). Pittsburgh, PA: Carnegie Mellon University, February 1993.

[Paulk93b] Paulk Mark, Charles Weber, Suzanne Garcia, Mary Beth Chrisis, and Marilyn Bush. *Key Practices of the Capability Maturity Model , Version 1.1* CMU/SEI-93-TR-25. Pittsburgh, PA: Carnegie Mellon University, February 1993.

[Peters92] Peters, Tom. "First Word: To Win with Quality and Service, Focus on Culture, not on Rewards.", from *On Achieving Excellence*, Volume 7, Issue #4, April 1992, pp.1-12.

[Radice85] Radice, R. A., J. T. Harding, P. E. Munnis, and R. W. Phillips. "A programming process study." *IBM Systems Journal, 24*, 2 (1985), pp. 91-101.

[Radice85] Radice, R. A., N. K. Roth Jr., A. C. O'Hara, and W. A. Ciargella. "A Programming Process Architecture." *IBM Systems Journal 24*, 2 (1985), pp. 79-90.

[Sirkin90] Sirkin, Harold and George Stalk, Jr. "Fix the Process, Not the Problem." *Harvard Business Review 4*, 90411 (July-August 1990), pp. 3-8.

Appendix A: List of Acronyms

List of Acronyms Used in SPF

The table below lists the acronyms in the Software Process Framework and their meaning.

Acronym	Meaning
CMM	Capability Maturity Model
KPA	Key process area
PAT	Process action team
RM	Requirements management
SCM	Software configuration management
SEI	Software Engineering Institute
SEPG	Software engineering process group
SPF	Software Process Framework (this document)
SPP	Software project planning
SPTO	Software project tracking and oversight
SQA	Software quality assurance
SSM	Software subcontract management

Appendix B: Glossary of Terms

Glossary of Terms Used in the SPF

activity: the action (e.g., process, procedure, etc.) taken to create or achieve a work product, service, or result.

entry criteria: the conditions under which an activity can be started. Entry criteria take the form of a simple or compound predicate about the state of a work product, role, or activity.

exit criteria: the conditions under which an activity can be declared complete. Exit criteria take the form of a simple or compound predicate about the state of an artifact, role, or activity.

input: the relationship or link between an activity and a work product. Inputs are the results produced by a prior activity and used by the current activity and may be qualified by the state of a work product.

output: the relationship or link between an activity and a work product. Outputs are the results produced by the current activity and used by a subsequent activity and may be qualified by the state of a work product.

policy: provides the "law" or "regulations" that guide, govern, or constrain operations.

procedure: describes "how-to" or "step-by-step" instructions that implement a process.

process: describes "what happens" over time within the organization to build products that meet the standards in accordance with the organizational policies.

role (agent): "... a unit of defined responsibilities that may be assumed by one or more individuals." [Paulk93b]. The accomplisher or performer that carries out the action to achieve or create the work product, service, or result. A role can consist of automation, or even be totally automated.

standard: the "operational definitions" or "acceptance criteria" for final or interim products or processes.

tool: provides the needed support for organizational policies, standards, processes, procedures, and training in order to build software products.

training: provides people with necessary knowledge and skills including training on organizational policies, standards, processes, procedures, and tools.

work product: any final or intermediate product, service, or result of a process or activity.

Appendix C: Translation Tables

Translation Table for CMM Roles/Groups

Fill in the equivalent role for your organization in the table below. Sometimes an organizational role will perform more than one CMM role. Appendix E (CMM Glossary) and Appendix F (CMM Roles) provide the definitions from the CMM.

CMM Roles/Groups	Your Organization's Roles/Groups
Affected groups or other affected groups (Affected groups change according to the context of a situation. Make a complete list here of affected groups, and use this list to help build the list of roles in the process checklists for each situation.)	
Customer	
End user or end user representatives	
Engineering group	
First-line software manager	
Group responsible for analyzing and allocating system requirements	
Manager	
Prime contractor	
Project manager	
Project software manager	
Senior management	
Software engineering process group	
Senior manager	
Software engineering staff	
Software manager	
Software-related groups	
Software task leader	
Staff	

Continued on next page

Translation Tables, Continued

Translation Table for CMM Roles/Groups, continued

Fill in the equivalent role for your organization in the table below, continued from the previous page. Blank entries are provide for your use.

CMM Roles/Groups	Your Organization's Roles/Groups
Subcontractor	
Subcontract manager	
System engineering group	
System test group	
Task leader	

Translation Tables, Continued

Translation Table for General CMM Terms

Fill in the equivalent term for your organization in the table below. Blank entries are provided for your use. Appendix E (CMM Glossary) and Appendix F (CMM Roles) provide the definitions from the CMM.

CMM General Terms	Your Organization's Term
Organization	
Product	
Project	
Software product	
Software project	
System	

Appendix D: Process Templates - Annotated Catalogue of Agents, Work Products, and Activities

Agents

Who is involved in this process?

Agents: *organizational units, roles, or automation who perform the process*

Id Code	Name and Description
Enter the ID	*Enter a short name and optional brief description of each organizational unit, role, or automation.*

Work Products

What work products are consumed and produced by this process?

Products: *items transformed or produced by the process*

Id Code	Name and Description
Enter the ID	*Enter a short name and optional description of each work product.*

Activities

What activities are performed?

Activities: *actions taken to transform or produce a work product*

Id Code	Name and Description
Enter the ID	*Enter a short name and optional brief description of each activity.*

Activity: *Enter the ID and name of the Activity*

Purpose

Why is this activity performed?

Describe the purpose or rationale for this activity.

Performed by

Who is responsible for performing this activity?

Name or ID of Agent
List the organizational units, roles, or automation

Entry Criteria

When can this activity begin?

State or Condition	From Activity	[and] [or]
State as a simple or compound rule in terms of the state of an activity, work product, or agent.	*List the source activity that results in this state or condition*	

Inputs

What work products are consumed by this activity?

Work product name or ID	Source activity name or ID
List the ID and work product name of each input to the activity.	*List the source activity for this input.*

Continued on next page

Activity: *ID or Name of Activity, continued*

Parent Activity *Enter the ID or name of the parent activity to describe the activity hierarchy.*

Sub-activity, Procedure, or Method

How is this activity implemented?

Step	Description
	Describe the sub-activities or procedures to be followed for this activity. For activities at the bottom of the hierarchy, enumerate the steps.

Exit Criteria

When is this activity completed? What activity is next?

State or Condition	To Activity	[and] [or]
State as a simple or compound rule in terms of the state of an activity, work product, or agent.	*List the destination activity for this state or condition.*	

Outputs

What work products are produced by this activity?

Work product name or ID	Destination activity name or ID
List the ID and work product name of each output from the activity	*List the destination activity for this output.*

Process Templates

Catalogue of Agents, Work Products, and Activities

Agents — **Who is involved in this process?**

Id Code	Name and Description

Work Products — **What work products are consumed and produced by this process?**

Id Code	Name and Description

Activities — **What activities are performed?**

Id Code	Name and Description

Activity:

Purpose **Why is this activity performed?**

Performed by **Who is responsible for performing this activity?**

Name or ID of Agent

Entry Criteria **When can this activity begin?**

State or Condition	From Activity	[and] [or]

Inputs **What work products are consumed by this activity?**

Work product name or ID	Source activity name or ID

Continued on next page

Activity: , *continued*

Parent Activity

Sub-activity, Procedure, or Method

How is this activity implemented?

Step	Description

Exit Criteria

When is this activity completed? What activity is next?

State or Condition	To Activity	[and] [or]

Outputs

What work products are produced by this activity?

Work product name or ID	Destination activity name or ID

Appendix E: CMM Roles

The CMM Organizational Roles

A copy of the original CMM organizational roles from Version 1.1 [Paulk93b], from the "Overview of the Key Practices," is attached for your convenience.

4.4 Organizational Structure and Roles

Although the CMM attempts to remain independent of specific organizational structures and models, it is necessary to express the practices in the CMM consistently using terminology related to organizational structure and roles, which may differ from that followed by any specific organization. The following sections describe the various concepts related to organizations, projects, and roles that are necessary for interpreting the key practices of the CMM.

4.4.1 Organizational Roles

A role is a unit of defined responsibilities that may be assumed by one or more individuals. The following descriptions of roles are frequently used in the key practices:

Manager — A manager fulfills a role that encompasses providing technical and administrative direction and control to individuals performing tasks or activities within the manager's area of responsibility. The traditional functions of a manager include planning, resourcing, organizing, directing, and controlling work within an area of responsibility.

Senior manager

A senior manager fulfills a management role at a high enough level in an organization that the primary focus is the long-term vitality of the organization, rather than short-term project and contractual concerns and pressures. In general, a senior manager for engineering would have responsibility for multiple projects. A senior manager also provides and protects resources for long-term improvement of the software process (e.g., a software engineering process group).

Senior management, as used in the CMM, can denote any manager who satisfies the above description, up to and including the head of the whole organization. As used in the key practices, the term senior management should be interpreted in the context of the key process area and the projects and organization under consideration. The intent is to include specifically those senior managers who are needed to fulfill the leadership and oversight roles essential to achieving the goals of the key process area.

Project manager

A project manager fulfills the role with total business responsibility for an entire project; the project manager is the individual who directs, controls, administers, and regulates a project building a software or hardware/software system. The project manager is the individual ultimately responsible to the customer.

In a project-oriented organizational structure, most of the people working on a project would report to the project manager, although some disciplines might have a matrixed reporting relationship. In a matrixed organizational structure, it may be only the business staff who reports to the project manager. The engineering groups would then have a matrixed reporting relationship.

Project software manager

A project software manager fulfills the role with total responsibility for all the software activities for a project. The project software manager is the individual the project manager deals with in terms of software commitments and who controls all the software resources for a project.

The software engineering groups on a project would report to the project software manager, although some activities such as tools development might have a matrixed reporting relationship.

In a large project, the project software manager is likely to be a second-, third-, or fourth-line manager. In a small project or department with a single project, the project software manager might be the first-line software manager or might be at a higher level.

First-line software manager

A first-line software manager fulfills the role with direct management responsibility (including providing technical direction and administering the personnel and salary functions) for the staffing and activities of a single organizational unit (e.g., a department or project team) of software engineers and other related staff.

Software task leader

A software task leader fulfills the role of leader of a technical team for a specific task, who has technical responsibility and provides technical direction to the staff working on the task.

The software task leader usually reports to the same first-line software manager as the other people who are working on the task.

Staff, software engineering staff, individuals

Several terms are used in the CMM to denote the individuals who perform the various technical roles described in various key practices of the CMM. The staff are the individuals, including task leaders, who are responsible for accomplishing an assigned function, such as software development or software configuration management, but who are not managers.

The software engineering staff are the software technical people (e.g., analysts, programmers, and engineers), including software task leaders, who perform the software development and maintenance activities for the project, but who are not managers.

The term "individuals" as used in the key practices is qualified and bounded by the context in which the term appears (e.g., "the individual involved in managing the software subcontract").

A similar breakout of roles can be identified for other engineering groups such as system engineering or system test.

In a particular project or organization, there does not need to be a one-to-one correspondence between these roles and individuals. One person could perform in multiple roles, or each role could be performed by separate individuals.

For example, on a small, software-only project, one person might have as many as six roles: the system engineering first-line manager, the project system engineering manager, the software first-line manager, the project software manager, the project manager, and the software configuration management manager.

On a slightly larger project, one person might be the system engineering first-line manager, the project system engineering manager, and the project manager while another person might be both the first-line software manager and the project software manager. These two managers might be in the same second-line organization or in different second-line organizations.

On a large project, many roles, especially those of management, would likely be filled by separate individuals.

4.4.2 Organizational Structure

The fundamental concepts of organization, project, and group must be understood to properly interpret the key practices of the Capability Maturity Model. The following paragraphs define the use of these concepts in the CMM:

Organization

An organization is a unit within a company or other entity (e.g., government agency or branch of service) within which many projects are managed as a whole. All projects within an organization share a common top-level manager and common policies.

Project

A project is an undertaking requiring concerted effort, which is focused on developing and/or maintaining a specific product. The product may include hardware, software, and other components. Typically a project has its own funding, cost accounting, and delivery schedule.

Group

A group is the collection of departments, managers, and individuals who have responsibility for a set of tasks or activities. A group could vary from a single individual assigned part time, to several part-time individuals assigned from different departments, to several individuals dedicated full time.

Groups commonly referred to in the CMM are described below:

Software engineering group

The software engineering group is the collection of individuals (both managers and technical staff) who have responsibility for software development and maintenance activities (i.e., requirements analysis, design, code, and test) for a project.

Groups performing software-related work, such as the software quality assurance group, the software configuration management group, and the software engineering process group, are not included in the software engineering group. These groups are considered to be one of the "other software-related groups."

Software-related groups

A software-related group is the collection of individuals (both managers and technical staff) representing a software engineering discipline that supports, but is not directly responsible for, software development and/or maintenance.

Examples of software engineering disciplines include software quality assurance and software configuration management.

Software engineering process group

The software engineering process group is the group of specialists who facilitate the definition, maintenance, and improvement of the software process used by the organization. In the key practices, this group is generically referred to as "the group responsible for the organization's software process activities."

System engineering group

The system engineering group is the collection of individuals (both managers and technical staff) who have responsibility for specifying the system requirements; allocating the system requirements to the hardware, software, and other components; specifying the interfaces between the hardware, software, and other components; and monitoring the design and development of these components to ensure conformance with their specifications.

System test group

The system test group is the collection of individuals (both managers and technical staff) who have responsibility for planning and performing the independent system testing of the software to determine whether the software product satisfies its requirements.

Software quality assurance group

The software quality assurance group is the collection of individuals (both managers and technical staff) who plan and implement the project's quality assurance activities to ensure the software process steps and standards are followed. Organizational issues concerning software quality assurance are discussed in Section 4.4.3.

Software configuration management group

The software configuration management group is the collection of individuals (both managers and technical staff) who have responsibility for planning, coordinating, and implementing the formal configuration management activities for the software project.

Training group The training group is the collection of individuals (both managers and staff) who are responsible for coordinating and arranging the training activities for an organization. This group typically prepares and conducts most of the training courses and coordinates use of other training vehicles.

4.4.3 Independence and Organizational Structure

The organization must take care that the key practices that call for independence are appropriately interpreted and followed. This is particularly true for small projects and small organizations. The key practices call for independence when technical or organizational biases may affect the quality or risks associated with the project. For example, two practices dealing with independence are:

- The SQA group has a reporting channel to senior management that is independent of the project manager, the project's software engineering group, and the other software-related groups (Commitment 1.2 in Software Quality Assurance).
- The (system and acceptance) test cases and test procedures are planned and prepared by a test group that is independent of the software developers (Activity 7.3 in Software Product Engineering).

The need for independence of the system and acceptance testing is based on technical considerations. This independence ensures that the testers are not inappropriately influenced by the design and implementation decisions made by the software developers or maintainers.

The independence of the SQA group is necessary so its members can perform their jobs without being influenced by project schedule and cost

Appendix F: CMM Glossary

Version 1.1 of the CMM Glossary

A copy of the original CMM glossary from Version 1.1 [Paulk93b], "Appendix B: Glossary of Terms," is attached for your convenience.

Appendix B: Glossary of Terms

ability to perform - (See *common features.*)

acceptance criteria - The criteria that a system or component must satisfy in order to be accepted by a user, customer, or other authorized entity. [IEEE-STD-610]

acceptance testing - Formal testing conducted to determine whether or not a system satisfies its acceptance criteria and to enable the customer to determine whether or not to accept the system. [IEEE-STD-610]

activity - Any step taken or function performed, both mental and physical, toward achieving some objective. Activities include all the work the managers and technical staff do to perform the tasks of the project and organization. (See *task* for contrast.)

activities performed - (See *common features.*)

action item- (1) A unit in a list that has been assigned to an individual or group for disposition. (2) An action proposal that has been accepted.

action proposal- A documented suggestion for change to a process or process-related item that will prevent the future occurrence of defects identified as a result of defect prevention activities. (See also *software process improvement proposal.*)

allocated requirements - (See *system requirements allocated to software.*)

application domain - A bounded set of related systems (i.e., systems that address a particular type of problem). Development and maintenance in an application domain usually requires special skills and/or resources. Examples include payroll and personnel systems, command and control systems, compilers, and expert systems.

assessment - (See *software process assessment.*)

audit - An independent examination of a work product or set of work products to assess compliance with specifications, standards, contractual agreements, or other criteria. [IEEE-STD-610]

baseline - A specification or product that has been formally reviewed and agreed upon, that thereafter serves as the basis for further development, and that can be changed only through formal change control procedures. [IEEE-STD-610]

baseline configuration management- The establishment of baselines that are formally reviewed and agreed on and serve as the basis for further development. Some software work products, e.g., the software design and the code, should have baselines established at predetermined points, and a rigorous change control process should be applied these items. These baselines provide control and stability when interacting with the customer. (See also *baseline management*.)

baseline management- In configuration management, the application of technical and administrative direction to designate the documents and changes to those documents that formally identify and establish baselines at specific times during the life cycle of a configuration item. [IEEE-STD-610]

benchmark - A standard against which measurements or comparisons can be made. [IEEE-STD-610]

bidder - An individual, partnership, corporation, or association that has submitted a proposal and is a candidate to be awarded a contract to design, develop, and/or manufacture one or more products.

capability maturity model - A description of the stages through which software organizations evolve as they define, implement, measure, control, and improve their software processes. This model provides a guide for selecting process improvement strategies by facilitating the determination of current process capabilities and the identification of the issues most critical to software quality and process improvement.

causal analysis - The analysis of defects to determine their underlying root cause.

causal analysis meeting - A meeting, conducted after completing a specific task, to analyze defects uncovered during the performance of that task.

CMM - Acronym for *capability maturity model.*

commitment - A pact that is freely assumed, visible, and expected to be kept by all parties.

commitment to perform - (See *common features.*)

common cause (of a defect) - A cause of a defect that is inherently part of a process or system. Common causes affect every outcome of the process and everyone working in the process. (See *special cause* for contrast.)

common features - The subdivision categories of the CMM key process areas. The common features are attributes that indicate whether the implementation and institutionalization of a key process area is effective, repeatable, and lasting. The CMM common features are the following:

- *commitment to perform* - The actions the organization must take to ensure that the process is established and will endure. Commitment to Perform typically involves establishing organizational policies and senior management sponsorship.
- *ability to perform* - The preconditions that must exist in the project or organization to implement the software process competently. Ability to Perform typically involves resources, organizational structures, and training.
- *activities performed* - A description of the roles and procedures necessary to implement a key process area. Activities Performed typically involve establishing plans and procedures, performing the work, tracking it, and taking corrective actions as necessary.
- *measurement and analysis* - A description of the need to measure the process and analyze the measurements. Measurement and Analysis typically includes examples of the measurements that could be taken to determine the status and effectiveness of the Activities Performed.

- *verifying implementation* - The steps to ensure that the activities are performed in compliance with the process that has been established. Verification typically encompasses reviews and audits by management and software quality assurance.

configuration - In configuration management, the functional and physical characteristics of hardware or software as set forth in technical documentation or achieved in a product. [IEEE-STD-610]

configuration control - An element of configuration management, consisting of the evaluation, coordination, approval or disapproval, and implementation of changes to configuration items after formal establishment of their configuration identification. [IEEE-STD-610]

configuration identification - An element of configuration management, consisting of selecting the configuration items for a system and recording their functional and physical characteristics in technical documentation. [IEEE-STD-610]

configuration item - An aggregation of hardware, software, or both, that is designated for configuration management and treated as a single entity in the configuration management process. [IEEE-STD-610]

configuration management - A discipline applying technical and administrative direction and surveillance to identify and document the functional and physical characteristics of a configuration item, control changes to those characteristics, record and report change processing and implementation status, and verify compliance with specified requirements. [IEEE-STD-610]

configuration management library system - The tools and procedures to access the contents of the software baseline library.

configuration unit - The lowest level entity of a configuration item or component that can be placed into, and retrieved from, a configuration management library system.

consistency - The degree of uniformity, standardization, and freedom from contradiction among the documents or parts of system or component. [IEEE-STD-610]

contingency factor - An adjustment (increase) of a size, cost, or schedule plan to account for likely underestimates of these parameters due to incomplete specification, inexperience in estimating the application domain, etc.

contract terms and conditions - The stated legal, financial, and administrative aspects of a contract.

critical computer resource - The parameters of the computing resources deemed to be a source of risk to the project because the potential need for those resources may exceed the amount that is available. Examples include target computer memory and host computer disk space.

critical path - A series of dependent tasks for a project that must be completed as planned to keep the entire project on schedule.

customer - The individual or organization that is responsible for accepting the product and authorizing payment to the developing organization.

defect - A flaw in a system or system component that causes the system or component to fail to perform its required function. A defect, if encountered during execution, may cause a failure of the system.

defect density - The number of defects identified in a product divided by the size of the product component (expressed in standard measurement terms for that product).

defect prevention - The activities involved in identifying defects or potential defects and preventing them from being introduced into a product.

defect root cause - The underlying reason (e.g., process deficiency) that allowed a defect to be introduced.

defined level - (See *maturity level.*)

defined software process - (See *project's defined software process.*)

dependency item - A product, action, piece of information, etc., that must be provided by one individual or group to a second individual or group so that the second individual or group can perform a planned task.

developmental configuration management - The application of technical and administrative direction to designate and control software and associated technical documentation that define the evolving configuration of a software work product during development. Developmental configuration management is under the direct control of the developer. Items under developmental configuration management are not baselines, although they may be baselined and placed under baseline configuration management at some point in their development.

deviation - A noticeable or marked departure from the appropriate norm, plan, standard, procedure, or variable being reviewed.

documented procedure - (See *procedure.*)

effective process - A process that can be characterized as practiced, documented, enforced, trained, measured, and able to improve. (See also *well-defined process.*)

end user - The individual or group who will use the system for its intended operational use when it is deployed in its environment.

end user representatives - A selected sample of end users who represent the total population of end users.

engineering group - A collection of individuals (both managers and technical staff) representing an engineering discipline. Examples of engineering disciplines include systems engineering, hardware engineering, system test, software engineering, software configuration management, and software quality assurance.

evaluation - (See *software capability evaluation.*)

event-driven review/activity - A review or activity that is performed based on the occurrence of an event within the project (e.g., a formal review or the completion of a life cycle stage). (See *periodic review/activity* for contrast.)

findings - The conclusions of an assessment, evaluation, audit, or review that identify the most important issues, problems, or opportunities within the area of investigation.

first-line software manager - A manager who has direct management responsibility (including providing technical direction and administering the personnel and salary functions) for the staffing and activities of a single organizational unit (e.g., a department or project team) of software engineers and other related staff.

formal review - A formal meeting at which a product is presented to the end user, customer, or other interested parties for comment and approval. It can also be a review of the management and technical activities and of the progress of the project.

function - A set of related actions, undertaken by individuals or tools that are specifically assigned or fitted for their roles, to accomplish a set purpose or end.

goals - A summary of the key practices of a key process area that can be used to determine whether an organization or project has effectively implemented the key process area. The goals signify the scope, boundaries, and intent of each key process area.

group - The collection of departments, managers, and individuals who have responsibility for a set of tasks or activities. A group could vary from a single individual assigned part time, to several part-time individuals assigned from different departments, to several individuals dedicated full time.

host computer - A computer used to develop software. (See *target computer* for contrast.)

initial level - (See *maturity level.*)

institutionalization - The building of infrastructure and corporate culture that support methods, practices, and procedures so that they are the ongoing way of doing business, even after those who originally defined them are gone.

integrated software management - The unification and integration of the software engineering and management activities into a coherent defined software process based on the organization's standard software process and related process assets.

integration - (See *software integration.*)

key practices - The infrastructures and activities that contribute most to the effective implementation and institutionalization of a key process area.

key process area - A cluster of related activities that, when performed collectively, achieve a set of goals considered important for establishing process capability. The key process areas have been defined to reside at a single maturity level. They are the areas identified by the SEI to be the principal building blocks to help determine the software process capability of an organization and understand the improvements needed to advance to higher maturity levels. The Level 2 key process areas in the CMM are Requirements Management, Software Project Planning, Software Project Tracking and Oversight, Software Subcontract Management, Software Quality Assurance, and Software Configuration Management. The Level 3 key process areas in the CMM are Organization Process Focus, Organization Process Definition, Training Program, Integrated Software Management, Software Product Engineering, Intergroup Coordination, and Peer Reviews. The Level 4 key process areas are Quantitative Process Management and Software Quality Management. The Level 5 key process areas are Defect Prevention, Technology Change Management, and Process Change Management.

life cycle - (See *software life cycle.*)

maintenance - The process of modifying a software system or component after delivery to correct faults, improve performance or other attributes, or adapt to a changed environment. [IEEE-STD-610]

managed and controlled - The process of identifying and defining software work products that are not part of a baseline and, therefore, are not placed under configuration management but that must be controlled for the project to proceed in a disciplined manner. "Managed and controlled" implies that the version of the work product in use at a given time (past or present) is known (i.e., version control), and changes are incorporated in a controlled manner (i.e., change control).

managed level - (See *maturity level.*)

manager - A role that encompasses providing technical and administrative direction and control to individuals performing tasks or activities within the manager's area of responsibility. The traditional functions of a manager include planning, resourcing, organizing, directing, and controlling work within an area of responsibility.

maturity level - A well-defined evolutionary plateau toward achieving a mature software process. The five maturity levels in the SEI's Capability Maturity Model are:

- *initial* - The software process is characterized as ad hoc, and occasionally even chaotic. Few processes are defined, and success depends on individual effort.
- *repeatable* - Basic project management processes are established to track cost, schedule, and functionality. The necessary process discipline is in place to repeat earlier successes on projects with similar applications.
- *defined* -The software process for both management and engineering activities is documented, standardized, and integrated into a standard software process for the organization. All projects use an approved, tailored version of the organization's standard software process for developing and maintaining software.

- *managed* - Detailed measures of the software process and product quality are collected. Both the software process and products are quantitatively understood and controlled.
- *optimizing* - Continuous process improvement is enabled by quantitative feedback from the process and from piloting innovative ideas and technologies.

maturity questionnaire - A set of questions about the software process that sample the key practices in each key process area of the CMM. The maturity questionnaire is used as a springboard to appraise the capability of an organization or project to execute a software process reliably.

measure - A unit of measurement (such as source lines of code or document pages of design).

measurement - The dimension, capacity, quantity, or amount of something (e.g., 300 source lines of code or 7 document pages of design).

method - A reasonably complete set of rules and criteria that establish a precise and repeatable way of performing a task and arriving at a desired result.

methodology - A collection of methods, procedures, and standards that defines an integrated synthesis of engineering approaches to the development of a product.

milestone - A scheduled event for which some individual is accountable and that is used to measure progress.

nontechnical requirements - Agreements, conditions and/or contractual terms that affect and determine the management activities of a software project.

operational software - The software that is intended to be used and operated in a system when it is delivered to its customer and deployed in its intended environment.

optimizing level - (See *maturity level.*)

organization - A unit within a company or other entity (e.g., government agency or branch of service) within which many projects are managed as a whole. All projects within an organization share a common top-level manager and common policies.

organization's measurement program - The set of related elements for addressing an organization's measurement needs. It includes the definition of organization-wide measurements, methods and practices for collecting organizational measurement data, methods and practices for analyzing organizational measurement data, and measurement goals for the organization.

organization's software process assets - A collection of entities, maintained by an organization, for use by projects in developing, tailoring, maintaining, and implementing their software processes. These software process assets typically include:

- the organization's standard software process,
- descriptions of the software life cycles approved for use,
- the guidelines and criteria for tailoring the organization's standard software process,
- the organization's software process database, and
- a library of software process-related documentation.

Any entity that the organization considers useful in performing the activities of process definition and maintenance could be included as a process asset.

organization's software process database - A database established to collect and make available data on the software processes and resulting software work products, particularly as they relate to the organization's standard software process. The database contains or references both the actual measurement data and the related information needed to understand the measurement data and assess it for reasonableness and applicability. Examples of process and work product data include estimates of software size, effort, and cost; actual data on software size, effort, and cost; productivity data; peer review coverage and efficiency; and number and severity of defects found in the software code.

organization's standard software process - The operational definition of the basic process that guides the establishment of a common software process across the software projects in an organization. It describes the fundamental software process elements that each software project is expected to incorporate into its defined software process. It also describes the relationships (e.g., ordering and interfaces) between these software process elements.

orientation - An overview or introduction to a topic for those overseeing or interfacing with the individuals responsible for performing in the topic area. (See *train* for contrast.)

Pareto analysis - The analysis of defects by ranking causes from most significant to least significant. Pareto analysis is based on the principle, named after the 19th-century economist Vilfredo Pareto, that most effects come from relatively few causes, i.e., 80% of the effects come from 20% of the possible causes.

peer review - A review of a software work product, following defined procedures, by peers of the producers of the product for the purpose of identifying defects and improvements.

peer review leader - An individual specifically trained and qualified to plan, organize, and lead a peer review.

periodic review/activity - A review or activity that occurs at specified regular time intervals. (See *event-driven review/activity* for contrast.)

policy - A guiding principle, typically established by senior management, which is adopted by an organization or project to influence and determine decisions.

prime contractor - An individual, partnership, corporation, or association that administers a subcontract to design, develop, and/or manufacture one or more products.

procedure - A written description of a course of action to be taken to perform a given task. [IEEE-STD-610]

process - A sequence of steps performed for a given purpose; for example, the software development process. [IEEE-STD-610]

process capability - The range of expected results that can be achieved by following a process. (See *process performance* for contrast.)

process capability baseline - A documented characterization of the range of expected results that would normally be achieved by following a specific process under typical circumstances. A process capability baseline is typically established at an organizational level. (See *process performance baseline* for contrast.)

process database - (See *organization's software process database.*)

process description- The operational definition of the major components of a process. Documentation that specifies, in a complete, precise, verifiable manner, the requirements, design, behavior, or other characteristics of a process. It may also include the procedures for determining whether these provisions have been satisfied. Process descriptions may be found at the task, project, or organizational level.

process development- The act of defining and describing a process. It may include planning, architecture, design, implementation, and validation.

process measurement - The set of definitions, methods, and activities used to take measurements of a process and its resulting products for the purpose of characterizing and understanding the process.

process performance - A measure of the actual results achieved by following a process. (See *process capability* for contrast.)

process performance baseline - A documented characterization of the actual results achieved by following a process, which is used as a benchmark for comparing actual process performance against expected process performance. A process performance baseline is typically established at the project level, although the initial process performance baseline will usually be derived from the process capability baseline. (See *process capability baseline* for contrast.)

process tailoring - The activity of creating a process description by elaborating, adapting, and/or completing the details of process elements or other incomplete specifications of a process. Specific business needs for a project will usually be addressed during process tailoring.

product - (See *software product* and *software work product.*)

profile - A comparison, usually in graphical form, of plans or projections versus actuals, typically over time.

project - An undertaking requiring concerted effort, which is focused on developing and/or maintaining a specific product. The product may include hardware, software, and other components. Typically a project has its own funding, cost accounting, and delivery schedule.

project's defined software process - The operational definition of the software process used by a project. The project's defined software process is a well-characterized and understood software process, described in terms of software standards, procedures, tools, and methods. It is developed by tailoring the organization's standard software process to fit the specific characteristics of the project. (See also *organization's standard software process, effective process,* and *well-defined process.*)

project manager - The role with total business responsibility for an entire project; the individual who directs, controls, administers, and regulates a project building a software or hardware/software system. The project manager is the individual ultimately responsible to the customer.

project software manager - The role with total responsibility for all the software activities for a project. The project software manager is the individual the project manager deals with in terms of software commitments and who controls all the software resources for a project.

quality - (1) The degree to which a system, component, or process meets specified requirements. (2) The degree to which a system, component, or process meets customer or user needs or expectations. [IEEE-STD-610]

quality assurance - (See *software quality assurance.*)

quantitative control - Any quantitative or statistically-based technique appropriate to analyze a software process, identify special causes of variations in the performance of the software process, and bring the performance of the software process within well-defined limits.

repeatable level - (See *maturity level.*)

required training - Training designated by an organization to be required to perform a specific role.

risk - Possibility of suffering loss.

risk management - An approach to problem analysis which weighs risk in a situation by using risk probabilities to give a more accurate understanding of the risks involved. Risk management includes risk identification, analysis, prioritization, and control.

risk management plan - The collection of plans that describe the risk management activities to be performed on a project.

role - A unit of defined responsibilities that may be assumed by one or more individuals.

SCE - Acronym for *software capability evaluation.*

SCM - Acronym for *software configuration management.*

senior manager - A management role at a high enough level in an organization that the primary focus is the long-term vitality of the organization, rather than short-term project and contractual concerns and pressures. In general, a senior manager for engineering would have responsibility for multiple projects.

software architecture - The organizational structure of the software or module. [IEEE-STD-610]

software baseline audit - An examination of the structure, contents, and facilities of the software baseline library to verify that baselines conform to the documentation that describes the baselines.

software baseline library - The contents of a repository for storing configuration items and the associated records.

software build - An operational version of a software system or component that incorporates a specified subset of the capabilities the final software system or component will provide. [IEEE-STD-610]

software capability evaluation - An appraisal by a trained team of professionals to identify contractors who are qualified to perform the software work or to monitor the state of the software process used on an existing software effort.

software configuration control board - A group responsible for evaluating and approving or disapproving proposed changes to configuration items, and for ensuring implementation of approved changes.

software development plan - The collection of plans that describe the activities to be performed for the software project. It governs the management of the activities performed by the software engineering group for a software project. It is not limited to the scope of any particular planning standard, such as DOD-STD-2167A and IEEE-STD-1058, which may use similar terminology.

software engineering group - The collection of individuals (both managers and technical staff) who have responsibility for software development and maintenance activities (i.e., requirements analysis, design, code, and test) for a project. Groups performing software-related work, such as the software quality assurance group, the software configuration management group, and the software engineering process group, are not included in the software engineering group.

software engineering process group - A group of specialists who facilitate the definition, maintenance, and improvement of the software process used by

the organization. In the key practices, this group is generically referred to as "the group responsible for the organization's software process activities."

software engineering staff - The software technical people (e.g., analysts, programmers, and engineers), including software task leaders, who perform the software development and maintenance activities for the project, but who are not managers.

software integration - A process of putting together selected software components to provide the set or specified subset of the capabilities the final software system will provide.

software life cycle - The period of time that begins when a software product is conceived and ends when the software is no longer available for use. The software life cycle typically includes a concept phase, requirements phase, design phase, implementation phase, test phase, installation and checkout phase, operation and maintenance phase, and, sometimes, retirement phase. [IEEE-STD-610]

software manager - Any manager, at a project or organizational level, who has direct responsibility for software development and/or maintenance.

software plans - The collection of plans, both formal and informal, used to express how software development and/or maintenance activities will be performed. Examples of plans that could be included: software development plan, software quality assurance plan, software configuration management plan, software test plan, risk management plan, and process improvement plan.

software process - A set of activities, methods, practices, and transformations that people use to develop and maintain software and the associated products (e.g., project plans, design documents, code, test cases, and user manuals).

software process assessment - An appraisal by a trained team of software professionals to determine the state of an organization's current software process, to determine the high-priority software process-related issues facing

an organization, and to obtain the organizational support for software process improvement.

software process assets - (See *organization's software process assets.*)

software process capability - (See *process capability.*)

software process description - The operational definition of a major software process component identified in the project's defined software process or the organization's standard software process. It documents, in a complete, precise, verifiable manner, the requirements, design, behavior, or other characteristics of a software process. (See also *process description.*)

software process element - A constituent element of a software process description. Each process element covers a well-defined, bounded, closely related set of tasks (e.g., software estimating element, software design element, coding element, and peer review element). The descriptions of the process elements may be templates to be filled in, fragments to be completed, abstractions to be refined, or complete descriptions to be modified or used unmodified.

software process improvement plan - A plan, derived from the recommendations of a software process assessment, that identifies the specific actions that will be taken to improve the software process and outlines the plans for implementing those actions. Sometimes referred to as an action plan.

software process improvement proposal - A documented suggestion for change to a process or process-related item that will improve software process capability and performance. (See also *action proposal.*)

software process maturity - The extent to which a specific process is explicitly defined, managed, measured, controlled, and effective. Maturity implies a potential for growth in capability and indicates both the richness of an organization's software process and the consistency with which it is applied in projects throughout the organization.

software process performance - (See *process performance.*)

software process-related documentation - Example documents and document fragments, which are expected to be of use to future projects when they are tailoring the organization's standard software process. The examples may cover subjects such as a project's defined software process, standards, procedures, software development plans, measurement plans, and process training materials.

software product - The complete set, or any of the individual items of the set, of computer programs, procedures, and associated documentation and data designated for delivery to a customer or end user. [IEEE-STD-610] (See *software work product* for contrast.)

software project - An undertaking requiring concerted effort, which is focused on analyzing, specifying, designing, developing, testing, and/or maintaining the software components and associated documentation of a system. A software project may be part of a project building a hardware/software system.

software quality assurance - (1) A planned and systematic pattern of all actions necessary to provide adequate confidence that a software work product conforms to established technical requirements. (2) A set of activities designed to evaluate the process by which software work products are developed and/or maintained.

software quality goal - Quantitative quality objectives defined for a software work product.

software quality management - The process of defining quality goals for a software product, establishing plans to achieve these goals, and monitoring and adjusting the software plans, software work products, activities, and quality goals to satisfy the needs and desires of the customer and end users.

software-related group - A collection of individuals (both managers and technical staff) representing a software engineering discipline that supports, but is not directly responsible for, software development and/or maintenance. Examples of software engineering disciplines include software quality assurance and software configuration management.

software requirement - A condition or capability that must be met by software needed by a user to solve a problem or achieve an objective. [IEEE-STD-610]

software work product - Any artifact created as part of defining, maintaining, or using a software process, including process descriptions, plans, procedures, computer programs, and associated documentation, which may or may not be intended for delivery to a customer or end user. (See *software product* for contrast.)

SPA - Acronym for *software process assessment.*

special cause (of a defect) - A cause of a defect that is specific to some transient circumstance and not an inherent part of a process. Special causes provide random variation (noise) in process performance. (See *common cause* for contrast.)

SQA - Acronym for *software quality assurance.*

staff - The individuals, including task leaders, who are responsible for accomplishing an assigned function, such as software development or software configuration management, but who are not managers.

stage - A partition of the software effort that is of a manageable size and that represents a meaningful and measurable set of related tasks which are performed by the project. A stage is usually considered a subdivision of a software life cycle and is often ended with a formal review prior to the onset of the following stage.

standard - Mandatory requirements employed and enforced to prescribe a disciplined uniform approach to software development.

standard software process - (See *organization's standard software process.*)

statement of work - A description of all the work required to complete a project, which is provided by the customer.

subcontract manager - A manager in the prime contractor's organization who has direct responsibility for administering and managing one or more subcontracts.

subcontractor - An individual, partnership, corporation, or association that contracts with an organization (i.e., the prime contractor) to design, develop, and/or manufacture one or more products.

system - A collection of components organized to accomplish a specific function or set of functions. [IEEE-STD-610]

system engineering group - The collection of individuals (both managers and technical staff) who have responsibility for specifying the system requirements; allocating the system requirements to the hardware, software, and other components; specifying the interfaces between the hardware, software, and other components; and monitoring the design and development of these components to ensure conformance with their specifications.

system requirement - A condition or capability that must be met or possessed by a system or system component to satisfy a condition or capability needed by a user to solve a problem. [IEEE-STD-610]

system requirements allocated to software - The subset of the system requirements that are to be implemented in the software components of the system. The allocated requirements are a primary input to the software development plan. Software requirements analysis elaborates and refines the allocated requirements and results in software requirements which are documented.

tailor - To modify a process, standard, or procedure to better match process or product requirements.

target computer - The computer on which delivered software is intended to operate. (See *host computer* for contrast.)

task - (1) A sequence of instructions treated as a basic unit of work. [IEEE-STD-610] (2) A well-defined unit of work in the software process that

provides management with a visible checkpoint into the status of the project. Tasks have readiness criteria (preconditions) and completion criteria (postconditions). (See *activity* for contrast.)

task kick-off meeting - A meeting held at the beginning of a task of a project for the purpose of preparing the individuals involved to perform the activities of that task effectively.

task leader - The leader of a technical team for a specific task, who has technical responsibility and provides technical direction to the staff working on the task.

team - A collection of people, often drawn from diverse but related groups, assigned to perform a well-defined function for an organization or a project. Team members may be part-time participants of the team and have other primary responsibilities.

testability - (1) The degree to which a system or component facilitates the establishment of test criteria and the performance of tests to determine whether those criteria have been met. (2) The degree to which a requirement is stated in terms that permit establishment of test criteria and performance of tests to determine whether those criteria have been met. [IEEE-STD-610]

technical requirements - Those requirements that describe what the software must do and its operational constraints. Examples of technical requirements include functional, performance, interface, and quality requirements.

technology - The application of science and/or engineering in accomplishing some particular result.

traceability - The degree to which a relationship can be established between two or more products of the development process, especially products having a predecessor-successor or master-subordinate relationship to one another. [IEEE-STD-610]

train - To make proficient with specialized instruction and practice. (See also *orientation.*)

training group - The collection of individuals (both managers and staff) who are responsible for coordinating and arranging the training activities for an organization. This group typically prepares and conducts most of the training courses and coordinates use of other training vehicles.

training program - The set of related elements that focus on addressing an organization's training needs. It includes an organization's training plan, training materials, development of training, conduct of training, training facilities, evaluation of training, and maintenance of training records.

training waiver - A written approval exempting an individual from training that has been designated as required for a specific role. The exemption is granted because it has been objectively determined that the individual already possesses the needed skills to perform the role.

unit - (1) A separately testable element specified in the design of a computer software component. (2) A logically separable part of a computer program. (3) A software component that is not subdivided into other components. [IEEE-STD-610]

user- (See *end user.)*

validation- The process of evaluating software during or at the end of the development process to determine whether it satisfies specified requirements. [IEEE-STD-610]

verification- The process of evaluating software to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase. [IEEE-STD-610]

verifying implementation - (See *common features.*)

waiver - (See *training waiver).*

well-defined process - A process that includes readiness criteria, inputs, standards and procedures for performing the work, verification mechanisms (such as peer reviews), outputs, and completion criteria. (See also *effective process.*)

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS None
2a. SECURITY CLASSIFICATION AUTHORITY N/A	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for Public Release Distribution Unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) CMU/SEI-93-SR-07	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Software Engineering Institute 6b. OFFICE SYMBOL (if applicable) SEI	7a. NAME OF MONITORING ORGANIZATION SEI Joint Program Office
6c. ADDRESS (city, state, and zip code) Carnegie Mellon University Pittsburgh PA 15213	7b. ADDRESS (city, state, and zip code) HQ ESC/ENS 5 Eglin Street Hanscom AFB, MA 01731-2116
8a. NAME OFFUNDING/SPONSORING ORGANIZATION SEI Joint Program Office 8b. OFFICE SYMBOL (if applicable) ESC/ENS	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F1962890C0003
8c. ADDRESS (city, state, and zip code)) Carnegie Mellon University Pittsburgh PA 15213	10. SOURCE OF FUNDING NOS. PROGRAM ELEMENT NO: 63756E PROJECT NO.: N/A TASK NO: N/A WORK UNIT NO.: N/A
11. TITLE (Include Security Classification) A Software Process Framework for the SEI Capability Maturity Model: Repeatable Level	
12. PERSONAL AUTHOR(S) Timothy G. Olson, L. P. Gates. J. L. Mullaney, J. W. Over, N. R. Reizer, M. I. Kellner, R. W. Phillips, S. J. DiGennaro	
13a. TYPE OF REPORT Final 13b. TIME COVERED FROM TO	14. DATE OF REPORT (year, month, day) June 1993 15. PAGE COUNT 300 pp.
16. SUPPLEMENTARY NOTATION	
17. COSATI CODES FIELD / GROUP / SUB. GR.	18. SUBJECT TERMS (continue on reverse of necessary and identify by block number) capability maturity model; software process; process definition; software engineering process group (SEPG); repeatable process

19. ABSTRACT (continue on reverse if necessary and identify by block number)

This document describes a Software Process Framework (SPF) based on the Software Engineering Institute's (SEI) Capability Maturity Model (CMM) for the Repeatable Level (Level 2). The purpose of the SPF is to provide guidance for designing, analyzing, and reviewing software processes (consisting of standards, processes, procedures, training, and tools) so that they are consistent with the CMM. As a thesaurus is a companion book to a dictionary (both are useful for different purposes) the SPF is a companion guide to the CMM for defining software processes. For each key process area in the CMM, the SPF defines roles, entry and exit criteria, inputs and outputs, activities, reviews and audits, measurements, etc. This information can also be used to help build process models and

(please turn over)

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED ■ SAME AS RPT ☐ DTIC USERS ■	21. ABSTRACT SECURITY CLASSIFICATION Unclassified, Unlimited Distribution
22a. NAME OF RESPONSIBLE INDIVIDUAL Thomas R. Miller, Lt Col, USAF	22b. TELEPHONE NUMBER (include area code) (412) 268-7631 22c. OFFICE SYMBOL ESC/ENS (SEI)

ABSTRACT — continued from page one, block 19

process guides that are consistent with the CMM. The primary audiences of the SPF are software engineering process groups (SEPGs), process engineers, process action teams, and software quality assurance groups. Secondary audiences include anyone interested in software process improvement, or anyone using the CMM.