Carnegie Mellon University
Software Engineering Institute

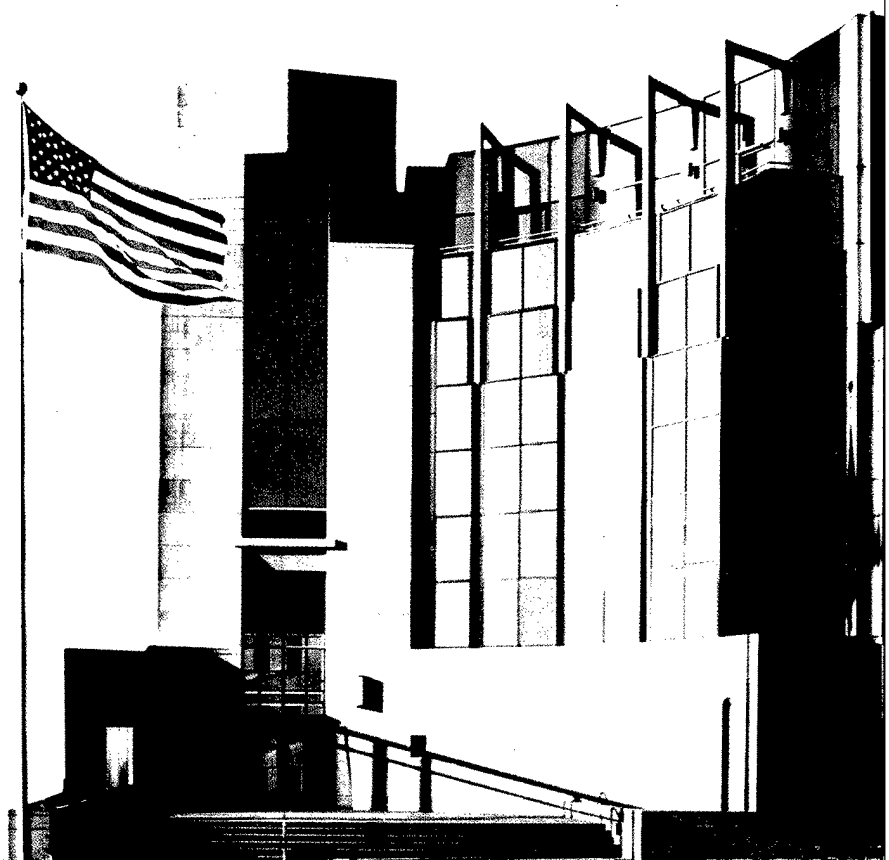
How to Use the Software Process Framework

Linda Parker Gates

October 1997

SR

SPECIAL REPORT
CMU/SEI-97-SR-009



Carnegie Mellon University does not discriminate and Carnegie Mellon University is required not to discriminate in admission, employment, or administration of its programs or activities on the basis of race, color, national origin, sex or handicap in violation of Title VI of the Civil Rights Act of 1964, Title IX of the Educational Amendments of 1972 and Section 504 of the Rehabilitation Act of 1973 or other federal, state, or local laws or executive orders.

In addition, Carnegie Mellon University does not discriminate in admission, employment or administration of its programs on the basis of religion, creed, ancestry, belief, age, veteran status, sexual orientation or in violation of federal, state, or local laws or executive orders. However, in the judgment of the Carnegie Mellon Human Relations Commission, the Department of Defense policy of, "Don't ask, don't tell, don't pursue," excludes openly gay, lesbian and bisexual students from receiving ROTC scholarships or serving in the military. Nevertheless, all ROTC classes at Carnegie Mellon University are available to all students.

Inquiries concerning application of these statements should be directed to the Provost, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, telephone (412) 268-6684 or the Vice President for Enrollment, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, telephone (412) 268-2056.

Obtain general information about Carnegie Mellon University by calling (412) 268-2000.

Special Report
CMU/SEI-97-SR-009
October 1997

How to Use the Software Process Framework

Linda Parker Gates

Transition Enabling Program

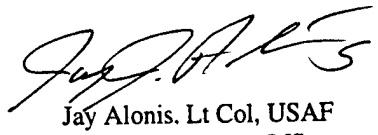
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

This report was prepared for the

SEI Joint Program Office
HQ ESC/AXS
5 Eglin Street
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.

FOR THE COMMANDER



Jay Alonis. Lt Col, USAF
SEI Joint Program Office

This work is sponsored by the U.S. Department of Defense.

This work was created in the performance of Federal Government Contract Number F19628-95-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work. in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 52.227-7013.

This document is available through Asset Source for Software Engineering Technology (ASSET): 1350 Earl L. Core Road: PO Box 3305; Morgantown, West Virginia 26505 / Phone:—(304) 284-9000 / FAX—(304) 284-9001 World Wide Web: http://www.asset.com / e-mail: sei@asset.com

Copies of this document are available through the National Technical Information Service (NTIS). For information on ordering, please contact NTIS directly: National Technical Information Service, U.S. Department of Commerce, Springfield, VA 22161. Phone—(703) 487-4600.

This document is also available through the Defense Technical Information Center (DTIC). DTIC provides access to and transfer of scientific and technical information for DoD personnel, DoD contractors and potential contractors, and other U.S. Government agency personnel and their contractors. To obtain a copy, please contact DTIC directly: Defense Technical Information Center / Attn: BRR / 8725 John J. Kingman Road / Suite 0944 / Ft. Belvoir, VA 22060-6218 / Phone—(703) 767-8274 or toll-free in the U.S.—1-800 225-3842.

Table of Contents

List of Tables

List of Figures

Acknowledgments

I would like to acknowledge Tim Olson, Neal Reizer, and Jim Over, the authors of the *Software Process Framework* (SPF) [Olson 94], for documenting their thoughts on how to use the SPF in the original document, which formed the basis for several sections of this report. I would like to thank Jonathan Addelston and Jim Over for their thoughtful input regarding their experiences using the SPF with government and industry organizations, and Mark Kasunic for his help in diagramming the organization of the SPF checklists.

I would also like to thank Marc Kellner for his guidance and careful review of this report, Paul Arnold for his assistance in gathering user input to this report and for his review of the early drafts, and Mark Paulk for his insightful comments. I would also like to thank Suzanne Couturiaux for her outstanding editorial assistance.

How to Use the Software Process Framework

Abstract: This report is intended to provide guidance on how to use the *Software Process Framework* (SPF) [Olson 94] for reviewing, analyzing, and designing software process documents that are consistent with the *Capability Maturity Model℠ (CMM®) for Software, Version 1.1* [Paulk 93a]. This guidance is not "how to design" or "how to analyze" software process documents in general. Rather, the guidance is focused on *how to use the Software Process Framework* for those purposes. The purpose of this report is to clarify the intended usage of the SPF and describe usage scenarios that have evolved through the use of the SPF in the software development community over several years. This report is intended to be used as a supplement to the SPF and in conjunction with the SPF, not by itself. It is assumed that the reader is familiar with the CMM, is experienced in software process improvement and definition, and has skill in designing or analyzing software process documents.

1. Introduction

1.1 Purpose of This Report

This report is intended to provide guidance on how to use the *Software Process Framework* (SPF) [Olson 94] for reviewing, analyzing, and designing software process documents that are consistent with the *Capability Maturity Model (CMM) for Software, Version 1.1* [Paulk93a]. This guidance is not "how to design" or "how to analyze" software process documents in general. Rather, the guidance is focused on *how to use the Software Process Framework* for those purposes. It is assumed that the reader is familiar with the CMM, is experienced in software process improvement and definition, and has skill in designing or analyzing software process documents.

The purpose of this report is to clarify the intended usage of the SPF and describe usage scenarios that have evolved through the use of the SPF by the software development community over several years. This report is intended to be used as a supplement to the SPF and in conjunction with the SPF, not by itself.

℠ Capability Maturity Model is a service mark of Carnegie Mellon University.

® CMM is registered in the U.S. Patent and Trademark Office.

1.2 What Is the SPF?

Just 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 (they include a lexicon of the same words), they are structured differently to accommodate different uses. The SPF contains the key practices of the CMM but reorganizes them to facilitate process definition activities: process design, process review, and process analysis.

The SPF was *not* designed to do the following:

- *The SPF is not a replacement for the CMM.* The CMM is designed to help software organizations determine current process maturity and identify the issues most critical to software quality and process improvement [Paulk 93a, p. 5]. The SPF is a process definition tool designed for process design, review, and analysis.
- *The SPF is not a "how to" guide.* It does not describe *how* to get to higher maturity levels. Rather, like the CMM, it contains information about *what* can be done in a particular key process area (KPA) to improve process maturity.
- *The SPF is not process definition training.* The SPF does not transfer the necessary knowledge and skills for defining a software process. For process definition training, the SEI offers the public course *Defining Software Processes.*
- *The SPF is not a process definition process.* The SPF does not provide a method or process for defining a software process or developing software process documents. A process for defining software processes is presented in the SEI public course *Defining Software Processes.*

2. Overview of the SPF and CMM

2.1 Purpose of the SPF

The SPF is a process definition tool intended to help users access the process maturity criteria, or key practices, established in the CMM. The SPF

- presents information recommended by the CMM in a format that is convenient for software process definition tasks
- identifies the polices, standards, processes, procedures, training, and tools recommended by the CMM
- provides checklists for ensuring that process documents are consistent with the CMM

2.2 Uses for the SPF

The SPF has two primary uses. The first is as a tool for reviewing and analyzing existing software process documents to ensure that they are consistent with the CMM. The second is as an aid in designing new software process documents that are consistent with the CMM.

Additional uses for the SPF include the following:

- performing process reviews and audits for software quality assurance
- defining organizational roles and responsibilities for key process areas
- evaluating and measuring improvements in process improvement pilot projects

2.3 The Purpose and Structure of the CMM

The information in the CMM forms the basis for the SPF. To use the SPF effectively, it is important to understand the purpose, structure, and content of the CMM. The user can absorb the content of the CMM only through study and experience with the CMM itself. The purpose, structure, and content of the CMM, however, is summarized here.

The CMM is a guide that is designed to help software organizations select process improvement strategies. The primary uses of the CMM are to help organizations determine current process maturity and identify the issues most critical to software quality and process improvement [Paulk 93a, p. 5]. Secondary uses of the CMM are to understand the activities necessary to plan and implement a software process improvement program, and to help define and improve the software process.

The CMM is structured to support organizational process improvement. It is organized by maturity levels: repeatable (Level 2) through optimizing (Level 5). Each maturity level contains two to seven key process areas. The key process areas contain key practices that are organized by common features. The common features are "attributes that indicate whether

the implementation and institutionalization of a key process area is effective, repeatable, and lasting" [Paulk 93a, p. 37]. The common features in the CMM are Commitment to Perform, Ability to Perform, Activities Performed, Measurement and Analysis, and Verifying Implementation. The organization of key practices into the common features emphasizes organizational performance.

The key practices in the CMM describe *what* is to be done in a particular key process area. They "should not be interpreted as mandating 'how' the goals should be achieved" [Paulk 93a, p.41]. The SPF reorganizes the CMM key practices to facilitate analyzing, reviewing, and designing software process documents and carries the same caveat.

2.4 The Operational Framework

Like the CMM, the SPF is organized by maturity levels and key process areas. The SPF, however, reorganizes the key practices of the CMM to facilitate process design, review, and analysis. The key practices are grouped according to an *operational framework*. The content material in the SPF is taken directly from the CMM. No new key practices or process requirements have been added.

The operational framework used in the SPF describes the operational elements that govern organizational software development. The operational framework is shown in Figure 1. Each component of the operational framework is described on the following page.

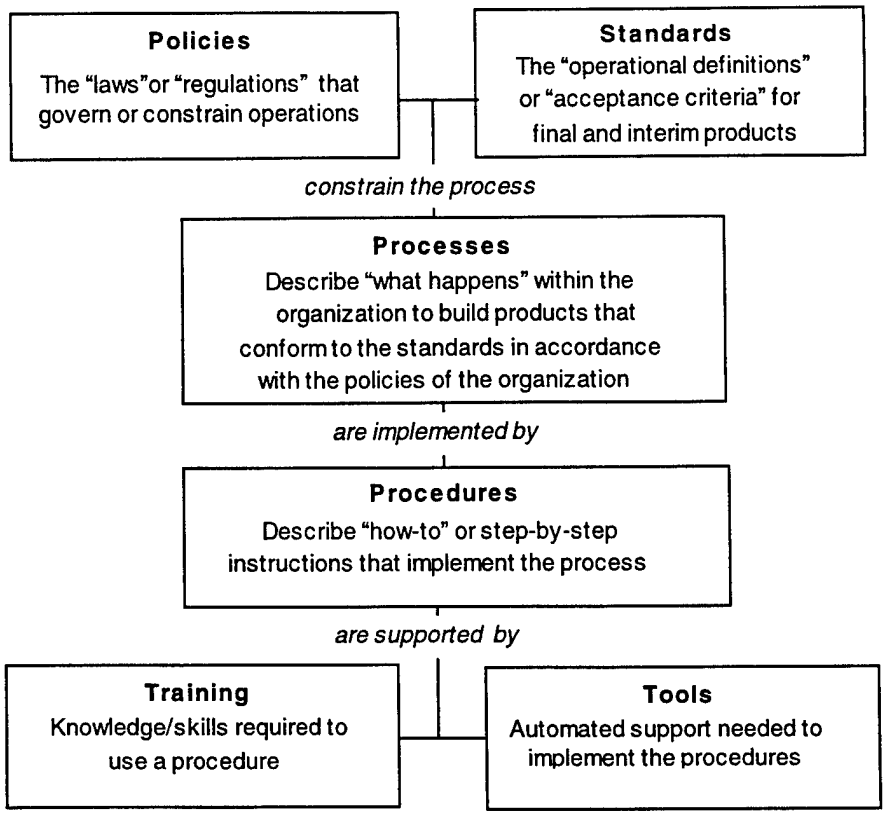


Figure 1. The Operational Framework

Policies

Organizational policies dictate the rules that govern operations. A policy statement usually is used to enforce the use of organizational processes. A policy statement, therefore, constrains organizational processes by identifying required or acceptable processes, or ways of doing work.

Standards

Standards are the operational definitions of final or interim organizational work products. Standards constrain organizational processes by setting acceptance criteria on the output of those processes.

Processes

A process is what happens in the organization to build products. Processes are constrained by organizational policies and standards in that they must specify ways to develop products that conform to organizational standards, in accordance with organizational policies.

Procedures

Procedures are the step-by-step instructions for implementing a process or a portion of a process. Procedural information focuses on how to perform a certain task identified in the process. Processes are, therefore, implemented by specific procedures.

Training

Training addresses the knowledge and skills required to execute a process or use a procedure. Training is used to support the use of processes and procedures.

Tools

Tools are any automated support needed to implement a procedure. Tools, like training, are used to support the use of processes and procedures.

2.5 Checklists

In the SPF, the CMM key practices are placed in checklists based on the operational framework. Within each maturity level, there are four types of checklists:

- Policy checklists contain key practices that convey policy information and information about policy goals.
- Standards checklists describe the content of the fundamental work products for each maturity level.
- Process checklists contain information about common process elements, including tools and training.
- Procedure checklists describe those documented procedures that are specifically called out in the CMM.

2.5.1 Organization of the Checklists

The process checklists are organized according to common elements of enactable software process definitions.[1] These process elements comprise the set of information that makes process descriptions usable by people performing a process. Basic process elements and the information they convey are listed in Table 1. Each of the process elements is represented by a checklist in the process checklists sections of the SPF, with the exception of procedures, which are listed separately.

[1] For more information, see the following SEI special report with limited distribution: James Armitage et al. *Software Process Definition Guide: Content of Enactable Software Process Definitions, Version 1.0* (CMU/SEI-93-SR-018). Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University, 1993.

Process elements	Information conveyed...
Roles	Who (or what) performs the process?
Entry criteria	When (under what circumstances) can a process activity begin?
Inputs	What work products are used to accomplish the goal(s) of the process?
Activities	What is done?
Outputs	What work products are produced?
Exit criteria	When (under what circumstances) can a process activity end?
Reviews and audits	What validation and verification steps are taken?
Work products managed and controlled	Which work products are under project or organizational control?
Measurements	What process measurements are taken?
Procedures	How are activities implemented?
Training	What kind of training is necessary or recommended?
Tools	What tools are necessary?

Table 1. Process Elements and the Information They Convey

Figure 2 illustrates how key practices and subpractices are organized into checklists that are based on the elements of the operational framework and the common elements of enactable process definitions.

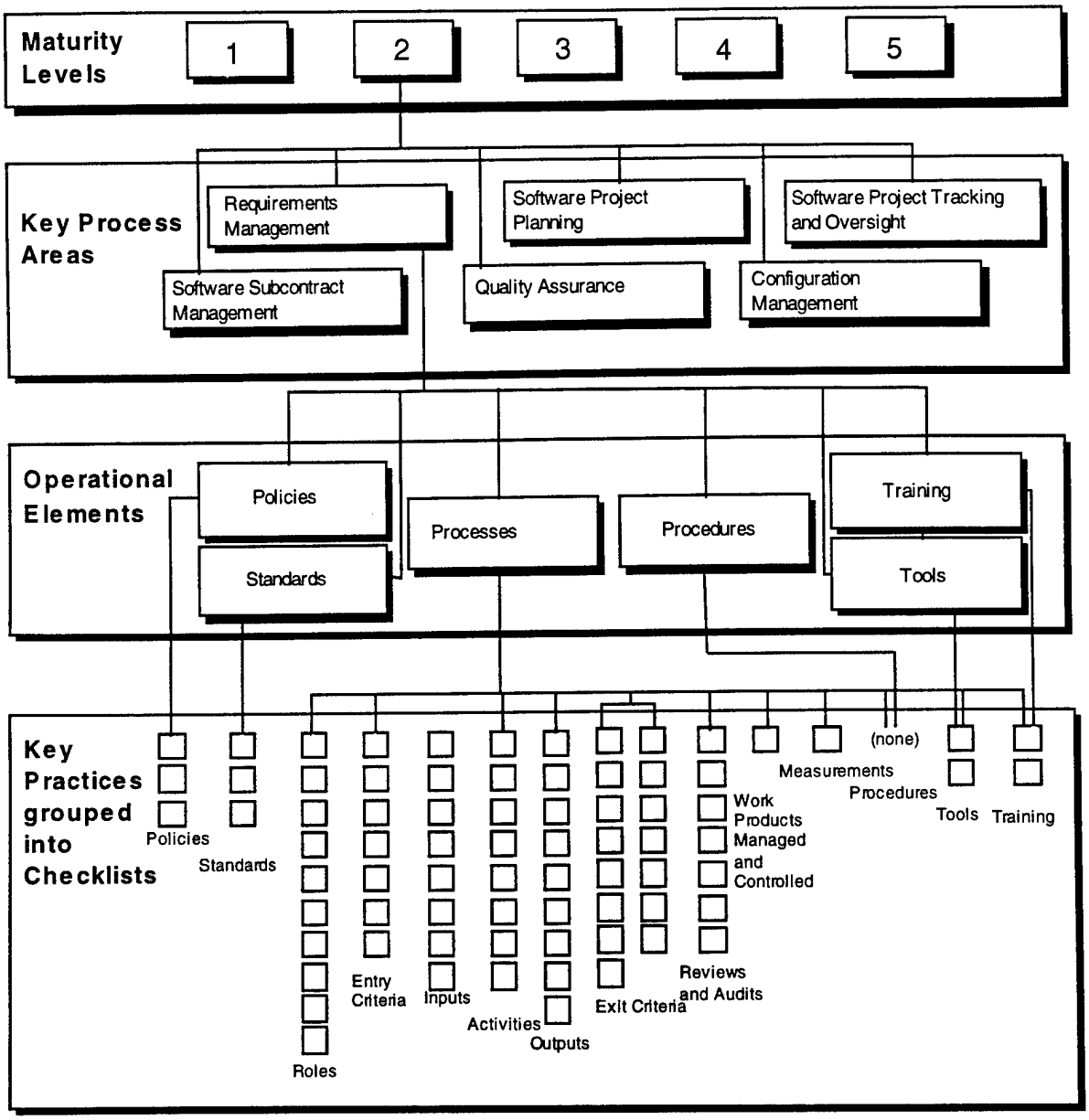


Figure 2. Organization of the SPF Checklists: Example Showing the Requirements Management KPA

2.5.2 Redundancy in the SPF Checklists

The CMM key practices and subpractices are expressed in a way that enables organizations to implement them most practically. For this reason, the key practices and subpractices usually appear in more than one process checklist in the SPF. A typical key practice describing Activities Performed in the CMM may specify a role and a work product (input or

output) in addition to the activity it describes. For example, Activity 9 in the Software Project Tracking and Oversight KPA in the CMM states

"Software engineering technical activities are tracked, and corrective actions are taken as necessary.

1. Members of the software engineering group report their technical status to their first-line manager on a regular basis.
2. Software release contents for successive builds are compared to the plans documented in the software development plan.
3. Problems identified in any of the software work products are reported and documented.
4. Problem reports are tracked to closure" [Paulk 93b, p. L2-37].

This "activity" in the CMM implies that a variety of process elements be in place in an enactable process definition. In the first subpractice alone, the following process elements are identified: two roles (software engineering group and first-line manager), an activity (reporting technical status), and an overall set of process exit criteria (i.e., conditions that must be met in order to exit a software project tracking and oversight process). As a result, the subpractice described in Activity 9.1 of the CMM will appear in the roles, activities, and entry criteria checklists for the Software Project Tracking and Oversight key process area.

This kind of redundancy is included in the SPF because it allows users to examine a key process area from many points of view. When a process is being designed or analyzed, it is often useful to focus on a single process element at a time and to ask questions such as the following: Are all the major activities represented in the process? Are all the entry criteria specified for this process? Are all work products accounted for? The individual process checklists (activities checklists, entry criteria checklists, input and output checklists, etc.) provide a very helpful view of the key practices for answering these kinds of questions. Additionally, a process definer may want to provide information about what an individual's role is with respect to an entire maturity level, for example. That person would rely heavily on the roles checklists within that maturity level. Similarly, process analysts may want to provide a training coordinator or measurement specialist with information pertinent to establishing their programs. In that case, the training or measurement checklists would be central to the analysis.

2.5.3 General Mapping of the SPF Checklists to the CMM Common Features

The organization of the CMM key practices by common features focuses on organizational process improvement. However, the common features can be broadly mapped to the operational elements that organize the SPF, and hence, can be mapped to some of the SPF process checklists.

The SPF *policy checklists* generally contain the Commitment to Perform key practices. The Commitment to Perform key practices generally communicate policy information, or

the commitments that the organization must make to software process maturity via its policy statements. Additionally, the policy checklists contain the key process area goals. The key process area goals are not a common feature of the CMM, but they are included in the SPF because they serve as a good reminder to policy developers that the policies should be designed to achieve the goals of the key process areas.

Standards information can be found in the CMM anywhere that a work product is described in detail. Standards information generally comes from the Activities Performed common feature or, on occasion, the Ability to Perform common feature.

The *process checklists* contain key practices from a wider range of common features. The Ability to Perform key practices tend to cover training and organizational structures. These key practices translate into process entry criteria, and generally make up the entry criteria checklists in the SPF. Activities checklists are composed of key practices from several of the common features, typically Activities Performed, Measurement and Analysis, and Verifying Implementation. The key practices in the inputs, outputs, and roles checklists can come from almost any of the common features. The exit criteria checklists, which describe conditions that must be met in order to declare a process finished, also usually encompass key practices from across many of the common features. Refer to Table 2 for the full set of process elements.

Procedure information is typically found in the Activities Performed common feature of the CMM.

Table 2 summarizes the general mapping of the CMM common features to the SPF checklists.

SPF Process Checklists	Common Features Typically Represented
Policy checklists	Commitment to Perform Goals
Standards checklists	Activities Performed Ability to Perform
Process: Roles checklists	Any
Process: Entry Criteria checklists	Ability to Perform
Process: Inputs checklists	Any
Process: Activities checklists	Activities Performed Measurement and Analysis Verifying Implementation
Process: Outputs checklists	Any
Process: Exit Criteria checklists	Any
Process: Reviews and Audits	Verifying Implementation
Process: Work Products Managed and Controlled	Activities Performed
Process: Measurements	Measurement and Analysis
Process: Training	Ability to Perform
Process: Tools	Any
Procedure checklists	Activities Performed

Table 2. General Mapping of the SPF Checklists to the CMM Common Features

3. Features of the SPF Checklists

3.1 Overview of the Checklist Features

The SPF checklists have several structural characteristics that facilitate their use as a process definition tool. Table 3 provides an overview of the features of the SPF checklists. Each feature is then described in detail in this section. Figure 4 presents an example SPF checklist.

SPF Feature	Usage
CMM References	Where information in the SPF is taken directly from the CMM, the exact source location in the CMM Version 1.1 is referenced for traceability.
Checkboxes	Checkboxes are used within the checklists so that every CMM requirement, no matter how small, can be accounted for individually.
User References	Space is provided for users to reference organizational documents that address CMM key practices or subpractices.
Translation Tables	Translation tables are provided to help users translate CMM terminology into their organization's terminology.

Table 3. Structural Characteristics of the SPF Checklists

3.2 CMM References

The CMM references in the SPF identify the exact location in the CMM Version 1.1 from which the material is derived.

Each CMM reference is defined as follows:

([CMM page], [Key practice], [Subpractice].[Subpractice sub-bullet])

Each component of a CMM reference is described below.

Note that the CMM references in the SPF are based on the notebook (binder) version of the CMM Version 1.1, which has since been published as a hardback book. The page numbers in the book edition of CMM Version 1.1, unfortunately, do not correspond to the references used in the SPF. If you are using a version of the CMM other than the notebook version, make sure to note the KPA you are working with. Then eliminate the CMM page component of

the reference and use the key practice, subpractice, and subpractice sub-bullet components to identify material.

3.2.1 The CMM Page Component

The [CMM page] component refers to the page in the CMM Version 1.1 notebook where the referenced text is located. For example, the first CMM page number in the repeatable level is L2-1, which is referenced simply as (L2-1) in the SPF. Text that spans page boundaries of the CMM is referenced by the page on which it begins.

3.2.2 The Key Practice Component

The [Key practice] component is an abbreviation of the CMM common feature addressed b y a given SPF item, followed by the number assigned to that key practice in the CMM.

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

For example, Activity 3 of Requirements Management is found on page L2-7 of the CMM, so the CMM reference for Activity 3 is (L2-7, A3). The abbreviations for the common features are listed in Table 4. Note that for references to CMM description, definition, or purpose blocks, the [Key practice] component is omitted.

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

Table 4. Abbreviations for CMM Common Features

3.2.3 The Subpractice Component

Many of the key practices found in the CMM contain numbered subpractices. The third component of the CMM reference, or [Subpractice] component, is the number of the subpractice in the CMM. For example, Activity 3 (Key practice) of Requirements Management has several subpractices. The CMM reference to the first subpractice would be (L2-7, A3, 1) as illustrated in Figure 3.

3.2.4 The Subpractice Sub-Bullet Component

Many subpractices contain one or more sub-bullets (e.g., checkboxed sentences in the CMM). When one of these sub-bullet sentences is included in an SPF checklist, the number of the sub-bullet becomes appended to the [Subpractice] component as the [Subpractice sub-bullet] component following a period. For example, the CMM reference to the first sub-bullet in subpractice (L2-7, A3, 1) would be (L2-7, A3, 1.1), as illustrated in Figure 3.

3.2.5 Examples of CMM References

Figure 3 illustrates how the CMM reference components are used to identify various passages in the CMM.

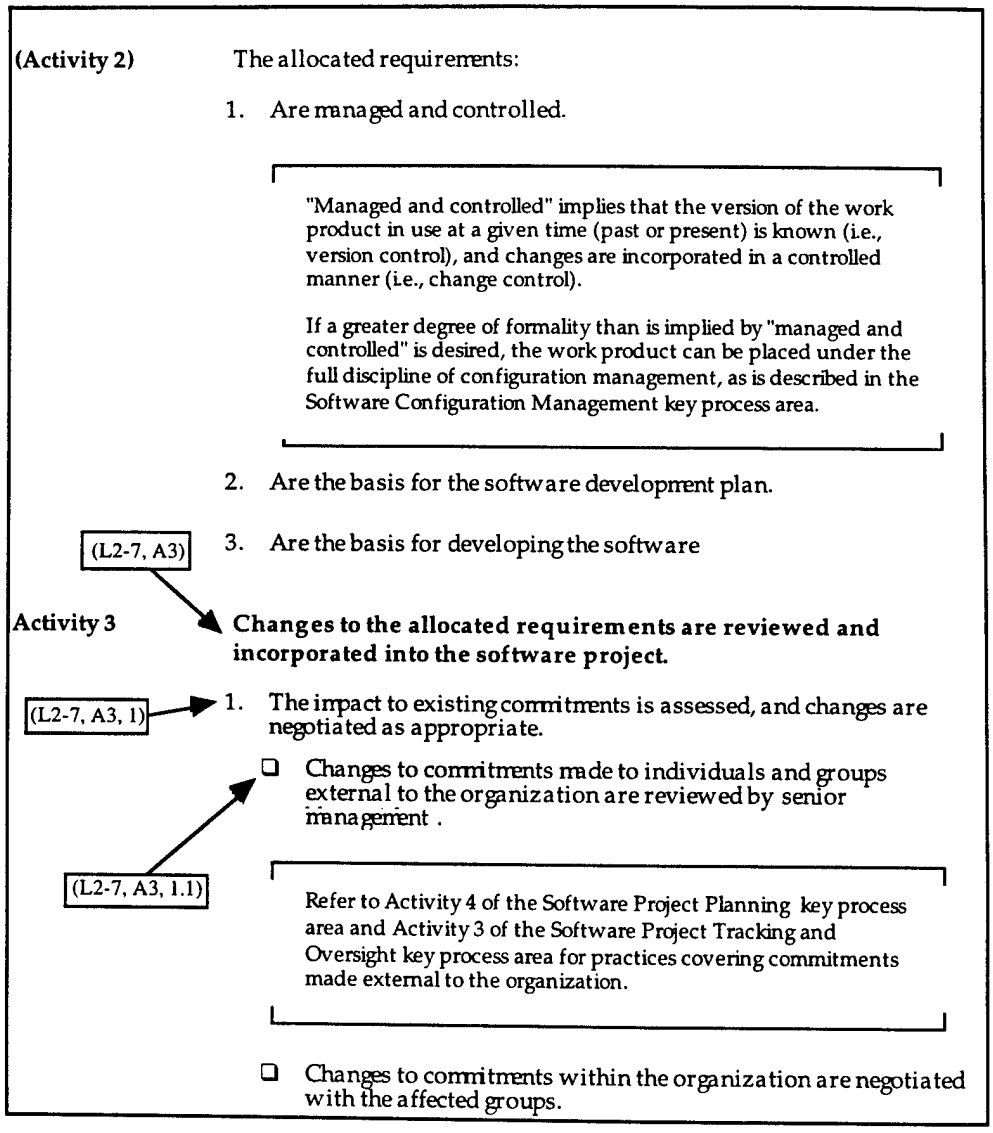


Figure 3. Example of CMM References

3.3 Checkboxes

The SPF checklists are for comparing the content of the CMM to the content of the target process. To use the checklists, examine each item in the checklists and determine whether it is satisfied or not satisfied in the target document. When an item is satisfied, check off the checkbox associated with the item. For nested checklists (checklists within checklists), the higher level checklist item functions as a "parent" item. Check the parent checkbox only if all the "children" items for that parent are satisfied.

The example in Figure 4 illustrates how an SPF roles checklist can be used to determine consistency between a target process and the CMM. Notice that in the example below, the SCCB (software configuration control board) role is not satisfied (the parent checklist is not checked) because not all of the checkboxes associated with the children of that parent have been checked. In this example, only the project manager role has been completely satisfied.

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*

Figure 4. Example of an SPF Checklist

3.4 User References

The purpose of the "References" column in the checklists is to provide users space to reference their organizational documents. These user references allow traceability between the SPF and the organizational document being analyzed and reviewed. As shown in Figure 5, they also allow the user to reference the chapter, section, page, paragraph, etc., of the organizational document that addresses each key practice or subpractice in the SPF.

SCM Process - Exit Criteria , Continued

Output-based Exit Criteria, continued

The table below describes the states that outputs must satisfy to exit the software configuration management process, continued from the previous page.

√	Output	State	References
	SCM plan	☑ development is coordinated or implemented by the **SCM group**. (L2-75, Ab2, 2)	*V1 S2.3.6*
		☑ distribution is coordinated or implemented by the **SCM group**. (L2-75, Ab2, 2)	*V1 S2.3.6*
		☐ maintenance is coordinated or implemented by the **SCM group**. (L2-75, Ab2, 2)	*V1 S2.3.6*
		☑ is prepared for each software project according to a documented procedure. (L2-76, A1)	*V2 S1.1*
		☑ is developed in the early stages of, and in parallel with, overall project planning. (L2-76, A1, 1)	*V2 S1.2*
		☑ is reviewed by the **affected groups**. (L2-77, A1, 2)	*V2 S1.3*
		☑ is managed and controlled. (L2 - 77, A1, 3)	*V2 S1.4*
		☐ is documented. (L2-77, A2)	*V2 S1.5*
		☐ is approved. (L2-77, A2)	*V2 S1.6*
		☐ is used as the basis for performing the SCM activities. (L2-77, A2)	*V2 S1.7*

Figure 5. Example of User References

Use abbreviations since the user reference space is limited, and there are numerous reference boxes to fill in. In the example above, "V1" is an abbreviation for "Volume 1"; "V2" is an abbreviation for "Volume 2"; and "S" is an abbreviation for "Section" in a fictitious user document. This is only an example; use abbreviations that make sense in your situation.

Note that you may want to make references on items in the checklist that are not satisfied in the target document. These references can be used as pointers to areas of the organizational document that can be improved. For example, in the checklist above the third item is not satisfied. A reference is made, however, to the location in the organizational document where this improvement might be added to satisfy that particular CMM requirement. If there are CMM criteria that do not apply to your organization, you may want to write N/A (not applicable) in the reference column.

3.5 Translation Tables

Software development terminology is not yet standardized. In order to be widely applicable, the CMM uses terminology that is quite general. The purpose of the SPF translation tables is to provide an easy way to translate CMM terminology into your organizational terminology. Although this may seem unnecessary because translations will be understood, making translation assumptions explicit has proven to be beneficial in practice.

Note that there is rarely a one-to-one mapping of CMM terminology to organizational terminology, and there are usually "gray" areas that need to be documented. Also note that organizations sometimes have multiple uses for a software term in different projects or divisions. There are three types of translation tables in the SPF: role translation tables, general term translation tables, and work product translation tables.

The role translation tables are accompanied by a role/KPA matrix that indicates which CMM roles occur in each key process area. The role/KPA matrix can be used as a reference when you are focusing on a single maturity level or a limited number of KPAs, helping you to translate only those roles that are relevant to your organization's effort.

3.5.1 Translating CMM Roles—An Example

Table 5 provides an example of a translation of CMM roles into a fictitious organization's roles. The role translation tables and role/KPA matrix are included in Appendix C in the SPF. See Appendix B in the SPF for a glossary of terms.

CMM Role	Your Organization's Role(s)
Affected groups or other affected groups	*SQA* *SCM* *Marketing* *Sales* *Technical staff* *Testing*
Project manager	*Project leader*
Project software manager	*Project leader*
Senior management	*Senior management steering committee*
Software engineering process group	*SEPG*
Senior manager	*CEO*
Software engineering staff	*Technical staff*

Table 5. Example of CMM Roles Translation

3.5.2 Translating General CMM Terms—An Example

Table 6 provides examples of a translation of general CMM terms into a fictitious organization's terms. The general terms translation tables are included in Appendix D in the SPF. See Appendix B in the SPF for a glossary of CMM terms.

CMM General Term	Your Organization's Term
Product	*Deliverable*
Project	*Project*
Software product	*CSCI/CSCU*
Software project	*Software task*
System	*Product*

Table 6. Example of CMM General Terms Translation

3.5.3 Translating Work Products—An Example

Table 7 provides an example of a translation of work products into a fictitious organization's terms. The work product translation tables are embedded in the input and output process checklists in the SPF. Use the "Org. Input" and "Org. Output" columns from the checklists to note your organization's terminology. See Appendix B in the SPF for a glossary of CMM terms.

√	Input	Org. Input	References
√	Statement of Work. (L2-14, Ab1) [Refer to Level 2 Standards for additional information regarding a statement of work.]	*SOW*	*DID 1000.5*
√	Allocated requirements. (L2-18, A6, 1.4) [Refer to Level 2 Standards for additional information regarding allocated requirements.]	*SRS* *IRS*	*DID 1000.6*

Table 7. Example of CMM Work Products Translation

4. Using the SPF to Design Software Process Documents

The purpose of this chapter is to provide some guidance on using the SPF to design organizational software process documents. It is not intended to provide training on software process definition, but to describe how the SPF can be used as a process design tool when creating a software process that is compliant with the CMM. This guidance is based on experience gained through application of the SPF for this use in various organizations.

4.1 General Procedure for Using the SPF to Design Software Processes

A general approach to using the SPF to design a process has the following steps:

1. Review the goals of the key process area that the document is intended to support.

   Once you have selected a software policy, standard, process, or procedure to design, the process designer should review the goals of the key process area that the document is intended to support. The key process area goals are listed in the policy checklists in the tables labeled "<Key Process Area> Policy Goals."

2. Translate the organizational software terminology to CMM terminology.

   The translation tables can be used to translate the organizational software terminology to CMM terminology. Translate CMM roles and general terms to your organizational terminology using the translation tables in Appendices C and D. You can translate work products when you use the input and output process checklists. For guidance on using the translation tables, see Section 3.5 of this document.

   If an electronic version of the SPF is being used, search-and-replace techniques can be used to translate the CMM terms in the actual checklists to organizational terminology. It is recommended that organizational terms be inserted into the checklists [in brackets], rather than allowed to replace the CMM terminology. Such an approach will facilitate making changes to the translations as you work with the checklists. In all cases, you should verify the results of using automated techniques to change the content of any of the checklists.

3. Use the appropriate policy, standard, process, or procedure checklist(s) in the SPF as input to the design of your process document.

   The information in the checklists will help you design CMM compliance into the policy, standard, process, or procedure.

4. Place references to your organizational document in the checklists to provide traceability between the CMM and your document.

   This will make it simpler to review and analyze the document against the CMM. (See "User References" in Section 3.4 of this document.)

Note that the CMM key practices alone will not generally form a complete process or a complete process document, but will help build the skeleton for a CMM-compliant process.

When the process document is complete, you should have a document that is compliant with the CMM. You will also have a detailed record of where the key practices and subpractices are addressed in your organizational document.

4.2 Using the Policies and Standards Checklists to Design Software Process Documents

The policies and standards checklists should be used to ensure that policies and standards are put in place to guide the use of the process. Specific uses of the policies and standards checklists are described below.

- Use the policy checklists as a reference when designing policy documents that are consistent with the CMM.
- Use the policy goals checklists to ensure that your documents address the *goals* of the appropriate key process area.
- Use the standards checklist(s) when designing standards documents that are compliant with the CMM.

4.3 Using the Process Checklists to Design Software Process Documents

Before you begin creating CMM-compliant process documents, it is recommended that you review the goals of the CMM key process area that the document is intended to support. The pertinent question in considering CMM compliance is whether a practice meets the goals of a key process area [Paulk 93b]. A goal-oriented approach to process design encourages process definers to develop processes that best suit the organization, while still satisfying the intent of the CMM. You can thus avoid inserting CMM key practices into an organizational culture that cannot embrace them effectively.

After reviewing the goals of a key process area, use the process checklists to ensure that the key practices of the key process area are designed into the process. The ordering of the process checklists follows a behavioral view of process descriptions: entry criteria -> input -> activity -> output -> exit criteria. This ordering supports process design by encouraging you to design the process from a behavioral point of view. You may find it helpful to use the process checklists to block out your process in terms of the common process elements. For example, the process entry criteria, inputs, activities, outputs, and exit criteria checklists can be used to help you develop a CMM-compliant peer review that is expressed in terms of these common process elements. The additional checklists (roles, reviews and audits, work products managed and controlled, measurements, procedures, training, and tools) can be consulted to ensure that these critical aspects of process design are also addressed.

Note that it may be effective to focus on high-priority activities, work products, and roles before lower priority items. For example, if you are designing a software project planning

process you may want to focus on how the software development plan is created (L2-19, A5, A7, A9, A10, A11, A12, A13, A14), what the software development plan should contain (L2-19, A7), how it is reviewed (L2-19, A6, 4), managed and controlled (L2-13, C2, 6), etc., before specifying the way that summary reports from senior management reviews are generated (L2-26, V1, 5).

5. Using the SPF to Review and Analyze Software Process Documents

The purpose of this chapter is to provide some guidance on using the SPF to review and analyze organizational software process documents. It is not intended to provide training on software process analysis, but to describe how the SPF can be used as a process analysis tool when examining a software process for CMM compliance. This guidance is based on experience gained through application of the SPF for this use in various organizations.

While the SPF checklists allow the process analyst to check for compliance to each key practice and subpractice of the CMM, remember that a process is compliant with the CMM when it satisfies the *goals* of a CMM key process area. Again, the pertinent question in considering CMM compliance is whether a practice meets the goals of a key process area [Paulk 93b].

5.1 General Procedure for Using the SPF to Review and Analyze Software Process Documents

A general approach to using the SPF to review and analyze process documents has the following steps:

1. Determine which key process area(s) the process document is intended to support by examining the goals of the candidate key process area(s).

   Once a process document has been selected for review, the process analyst should determine which key process area(s) it is intended to support by examining the goals of the candidate key process area(s). The key process area goals are listed in the policy checklists in the tables labeled "<Key Process Area> Policy Goals."

2. Use the translation tables to translate the CMM terminology to organizational software terminology.

   The translation tables can be used to translate the CMM terminology to organizational software terminology. Translate CMM roles and general terms to your organizational terminology using the translation tables in Appendices C and D. You can translate work products when you use the input and output process checklists (see below). For guidance on using the translation tables, see Section 3.5 of this document.

   If an electronic version of the SPF is being used, search-and-replace techniques can be used to translate the CMM terms in the actual checklists to organizational terminology. It is recommended that organizational terms be inserted into the checklists [in brackets], rather than allowed to replace the CMM terminology. Such an approach will facilitate making changes to the translations as you work with the checklists. In all cases, you should verify the results of using automated techniques to change the content of any of the checklists.

3. Use the appropriate policy, standard, process, or procedure checklist(s) in the SPF to review and analyze your process document.

   Check off the key practices and subpractices that are satisfied. (See Section 3.3, "Checkboxes.")

4. Add references to an organizational document, section, or page in the "Reference" column, as appropriate, to provide traceability of your work. (See "User References" in Section 3.4 of this document.)

When the review is complete, you will have a detailed record of which key practices and subpractices are addressed in your organizational document and which are not. You can use this information as input to designing improvements to your policies, standards, processes, and procedures that support the key process area.

5.2 Using the Policies and Standards Checklists to Review and Analyze Software Process Documents

The policies and standards checklists should be used to verify that policies and standards are in place to guide the use of the process. Specific uses of the policies and standards checklists are described below.

- Use the policy checklists to verify that policy documents are consistent with the CMM.
- Use the policy goals checklists to ensure that your documents address the *goals* of the appropriate key process area.
- Use the standards checklist(s) to verify the content of the work products that are used or produced in your target process.

5.3 Using the Process Checklists to Review and Analyze Software Process Documents

When reviewing process documents for CMM compliance or analyzing their content against the CMM, it is recommended that the process first be compared against the *goals* of the CMM key process area that it supports. The pertinent question in considering CMM compliance is whether a practice meets the goals of a key process area [Paulk 93b]. A goal-oriented approach to process review and analysis encourages process analysts to recognize practices that are well suited to the organization and still satisfy the intent of the CMM.

After reviewing the goals of a key process area and determining that the target process is intended to achieve those goals, use the process checklists to determine whether the key practices and subpractices of that key process area are addressed in the target process.

The ordering of the process checklists in the SPF follows a behavioral view of process descriptions: entry criteria -> input -> activity -> output -> exit criteria. This ordering is quite logical for process design (see Chapter 4). For process review or analysis, it is most effective to examine the checklists in a more counter-intuitive order. Begin by evaluating the target process against the outputs checklist. If the outputs for a key process area are being produced in the target process, you can review the activities checklist to determine if the outputs are being created via the recommended actions. Next, you can use the roles and

inputs checklists to ensure that the proper personnel and work products are consulted or are involved in the production of work products (outputs).

The benefit of starting at the "behavioral end" of the process for process review is that if you are merely examining a process for CMM compliance, it may be sufficient, once you have determined that an output is *not* being produced, to declare the process non-compliant without examining the remaining, and partially redundant, checklists. For example, if a software development plan (SDP) is not being created as part of an organization's software project planning process (L2-15, Ab2), there may be no need to verify that the process contains no SDP creation activity (L2-18, A6), or that the project manager is not involved in creating the SDP (L2-12, C1).

If, however, you are analyzing the target process in search of specific process improvement opportunities, you may want to examine more of the checklists and provide more complete input to process improvement planning. To build on the example above, for process analysis, it may be useful to communicate that the project manager is not involved in the creation of an SDP, there is no SDP creation activity, and there is, in fact, no SDP being created. Using the checklists in the reverse order (as described above) will speed the analysis, because it will probably be easier to recognize non-compliance items with the knowledge you gain from examining "later" checklists.

Whether using the lists for process review or analysis, it is recommended that you use the process checklists in the following order:

- outputs
- activities
- roles
- inputs
- exit criteria
- entry criteria

The remaining checklists can be used to fill in any gaps. You will probably want to make use of the work product translation column when going through the outputs and inputs checklists. The use of these columns is explained in Section 3.5.3.

Note that it may be quicker when you are reviewing a process solely for CMM compliance to verify the existence of "high-priority" work products (outputs) before looking for lower priority items. For example, if a software project planning process does not include the creation of an SDP, it is not necessary to ensure the existence of action items resulting from reviews with the project manager (L2-27, V2, 6) to determine that the process will not satisfy the goals of CMM Level 2, or specifically, of the Software Project Planning key process area.

6. Summary

This document is based on experience gained from the use of the SPF in the software development community.

The SPF, a companion document to the CMM, contains the key practices of the CMM, reorganized to facilitate process design, process review, and process analysis. The SPF is a process definition tool intended to help users access the process maturity criteria established in the CMM. It is intended to be used as a tool for reviewing and analyzing existing software process documents to ensure that they are consistent with the CMM or as an aid in designing new software process documents that are consistent with the CMM. The SPF has been used successfully in a number of organizations for these and related purposes.

The SPF is *not* intended to be a replacement for the CMM. It is not a "how-to" guide for reaching higher maturity levels. It does not constitute process definition training, and it does not specify a method for defining a process.

References

[Olson 94] Olson, Timothy; Reizer, Neal; and Over, James. *A Software Process Framework for the Capability Maturity Model* (CMU/SEI-94-HB-01, ADA 285595). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1994.Available WWW:<http://www.sei.cmu.edu/pub/-documents/94.reports/pdf/hb01.94.pdf>.

[Paulk 93a] Paulk, Mark C.; Curtis, Bill M.; and Chrissis, Mary Beth. *Capability Maturity Model for Software, Version 1.1* (CMU/SEI-93-TR-024, ADA 2634034). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1993.Available WWW:<http://www.sei.cmu.edu/-pub/documents/93.reports/pdf/tr24.93.pdf>.

[Paulk 93b] Paulk, Mark C; Weber, Charles V.; Garcia, Suzanne M.; Chrissis, Mary Beth; and Bush, Marilyn. *Key Practices of the Capability Maturity Model, Version 1.1* (CMU/SEI-93-TR-025, ADA 263432). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1993. Available WWW:<http://www.sei.cmu.edu/pub/documents/93.reports/-pdf/tr25.93.pdf>.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

AGENCY USE ONLY (LEAVE BLANK)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	October 1997	Final

TITLE AND SUBTITLE	5. FUNDING NUMBERS
w to Use the Software Process Framework	C — F19628-95-C-0003

AUTHOR(S)	
da Parker Gates	

PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213	CMU/SEI-97-SR-009

SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING/MONITORING AGENCY REPORT NUMBER
HQ ESC/AXS 5 Eglin Street Hanscom AFB, MA 01731-2116	

SUPPLEMENTARY NOTES

A DISTRIBUTION/AVAILABILITY STATEMENT	12.B DISTRIBUTION CODE
Unclassified/Unlimited, DTIC, NTIS	

ABSTRACT (MAXIMUM 200 WORDS)

This report is intended to provide guidance on how to use the *Software Process Framework* (SPF) [Olson 94] for reviewing, analyzing, and designing software process documents that are consistent with the *Capability Maturity Model℠ (CMM®) for Software, Version 1.1* [Paulk 93a]. This guidance is not "how to design" or "how to analyze" software process documents in general. Rather, the guidance is focused on *how to use the Software Process Framework* for those purposes. The purpose of this report is to clarify the intended usage of the SPF and describe usage scenarios that have evolved through the use of the SPF in the software development community over several years. This report is intended to be used as a supplement to the SPF and in conjunction with the SPF, not by itself. It is assumed that the reader is familiar with the CMM, is experienced in software process improvement and definition, and has skill in designing or analyzing software process documents.

SUBJECT TERMS	15. NUMBER OF PAGES
ability maturity model (CMM), key practices, key process area, policies, procedures, process	40
ecklists,software process documents, software process framework, software process	16. PRICE CODE
provement	

SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL