Technical Report
CMU/SEI-94-TR-024
ESC-TR-94-024

Carnegie Mellon University
Software Engineering Institute

DTIC
ELECTE
JAN 0 4 1996
S D
F

Process Tailoring and the Software Capability Maturity Modelsm

Mark P. Ginsberg
Lauren H. Quinn

November 1995

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

19960102 044

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.

Technical Report
CMU/SEI-94-TR-024
ESC-TR-94-024
November 1995

Process Tailoring and the Software Capability Maturity Model

Mark P. Ginsberg

SEI Resident Affiliate - Hughes Missile Systems Company

Lauren H. Quinn

SEI Resident Affiliate - Loral Federal Systems

Accesion For	
NTIS CRA&I	☑
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution /	
Availability Codes	
Dist	Avail and / or Special
A-1	

Approved for public release.
Distribution unlimited.

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

This report was prepared for the

SEI Joint Program Office
HQ ESC/ENS
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.

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

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

Copyright © 1995 by Carnegie Mellon University

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 part and in any manner, and to have or permit others to do so, for government purposes.

This material may be reproduced by or for the U.S. Government pursuant to the copyright license under the clause at 52.227-7013.

This document is available through Research Access, Inc., 800 Vinial Street, Pittsburgh, PA 15212: Phone: 1-800-685-6510. FAX: (412) 321-2994.

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: DTIC-OCP, 8725 John J. Kingman Road, Suite 0944, Ft. Belvoir, VA 22060-6218. Phone: (703) 767-8019/8021/8022/8023. Fax: 703-767-8032/DSN-427.

Use of any trademarks in this report is not intended in any way to infringe on the rights of the trademark holder.

Table of Contents

1 Introduction **1**
- 1.1 Formality of Tailoring 2
- 1.2 Purpose 2
- 1.3 Intended Audience 2
- 1.4 Organization of This Report 3
- 1.5 Interpretation vs. Tailoring 3
- 1.6 General Approach 4

2 Tailoring Framework **7**
- 2.1 Contexts for the Capability Maturity Model 7
- 2.2 Considerations for Tailoring with the SW-CMM 11
  - 2.2.1 Organizational Structure 11
  - 2.2.2 Customer and End-User Relationships 12
  - 2.2.3 Tailoring by Degree 12
  - 2.2.4 Business Goals 13
  - 2.2.5 Impact of Maturity Level on Tailoring 13

3 Requirements Analysis / Tailoring with the SW-CMM **15**
- 3.1 Tailoring Approach 16
- 3.2 Output Mapping 17
  - 3.2.1 Extensions/ Alternatives 18
- 3.3 Activity Mapping 18
  - 3.3.1 Extensions/ Alternatives 20
- 3.4 Role Mapping 21
  - 3.4.1 Role Translation 21
  - 3.4.2 Role Checklist 21
  - 3.4.3 Extensions/ Alternatives 24
- 3.5 Tailoring and the Infrastructure Common Features 25
- 3.6 Additional Requirements 27

4 The Organization's Standard Software Process and Tailoring Guidelines **29**
- 4.1 Tailoring Guidelines 29
- 4.2 Process Tailoring 29
  - 4.2.1 Project Structure 30
  - 4.2.2 Customer and End-User Relationships and Requirements 30
  - 4.2.3 Tailoring by Degree 31
  - 4.2.4 Business Goals 31
  - 4.2.5 Impact of Maturity Level on Tailoring 32
- 4.3 Process Elements 32
- 4.4 Tailoring Approach 33
  - 4.4.1 Developing the Tailoring Guidelines 33
  - 4.4.2 Developing the Project's Specific Process 35

4.5 The Software Development Plan 36

5 Data Sources 39

6 Summary 41

References 43

Appendix A. Glossary 45

Appendix B. Acronyms 51

Appendix C. Example Codes for Tailoring Tables 53

List of Figures

Figure 1-1: Volumes of the Software Capability Maturity Model 2

Figure 1-2: Correlating /Adjusting to Alternate Environments 4

Figure 2-1: The IDEAL Model for Process Improvement 9

Figure 2-2: A Tailoring Framework 10

Figure 2-3: Process Tailoring at Levels 1 and 2 14

Figure 4-1: Project Tailoring 36

List of Tables

Table 3-1: Example of SPF Output Checklist (for Requirements Management) 17

Table 3-2: Example of SPF Activities Checklist (for Organization Process Focus) 19

Table 3-3: Description of Disposition Code 20

Table 3-4: Example of SPF Role Translation Table 22

Table 3-5: Example of SPF Role Checklist (for Training Program) 23

Table 3-6: Example of Matrix of SW-CMM Key Practices and Activity Roles 24

Table 3-7: Focus Areas of CMM Common Features 26

Table 4-1: Example of Tailorable Process Elements 34

Process Tailoring and the Software Capability Maturity Model

Abstract The Software Capability Maturity Model[sm] (SW-CMM)[sm] is serving as the foundation for a major portion of the process improvement being undertaken in the software industry. It is composed of two volumes: the *Capability Maturity Model for Software*, and the *Key Practices of the Capability Maturity Model*. The key practices of the SW-CMM are expressed in terms that reflect normal practices of organizations that work on large, government contracts. There is, however, a significant population of software-producing and -acquiring organizations, operating in different environments, for which the key practices require significant interpretation and/or tailoring prior to application. This report presents a tailoring framework that identifies process artifacts, tailoring processes, and their relationships to project artifacts, and explores the nature of various kinds of tailoring used in the definition and development of software process descriptions. Techniques appropriate to each type of tailoring are then discussed. The general approach utilizes and builds upon the *Software Process Framework,* whose purpose is to provide guidance for designing, analyzing, and reviewing software processes for consistency with the SW-CMM.

1 Introduction

The Software Capability Maturity Model (SW-CMM), developed by the SEI, serves as the foundation for a major portion of the process improvement being undertaken in the software industry. The SW-CMM is composed of two volumes: The *Capability Maturity Model for Software* [Paulk 93a] and the *Key Practices of the Capability Maturity Model* [Paulk 93b]. The first volume contains a description of the five-level model, descriptions of each of the five levels, and an operational definition of the model. The second volume contains key practices that correspond to the key process areas (KPAs) at each maturity level of the model.

Since the set of all possible software projects and project environments (project space) is so large, a set of key practices suitable for use by all potential organizations and projects would be either very general or very complex, and would not be easily applied to any one project or environment. Therefore, the key practices were expressed with a specific subset of project space in mind: contractors concerned with the development of large, software-intensive critical systems (see Figure 1-1). However, It should be noted that the SW-CMM has been successfully adapted to other environments and serves as the basis for software process improvement efforts in many different arenas.

[sm] Capability Maturity Model[sm] and CMM[sm] are service marks of Carnegie Mellon University.

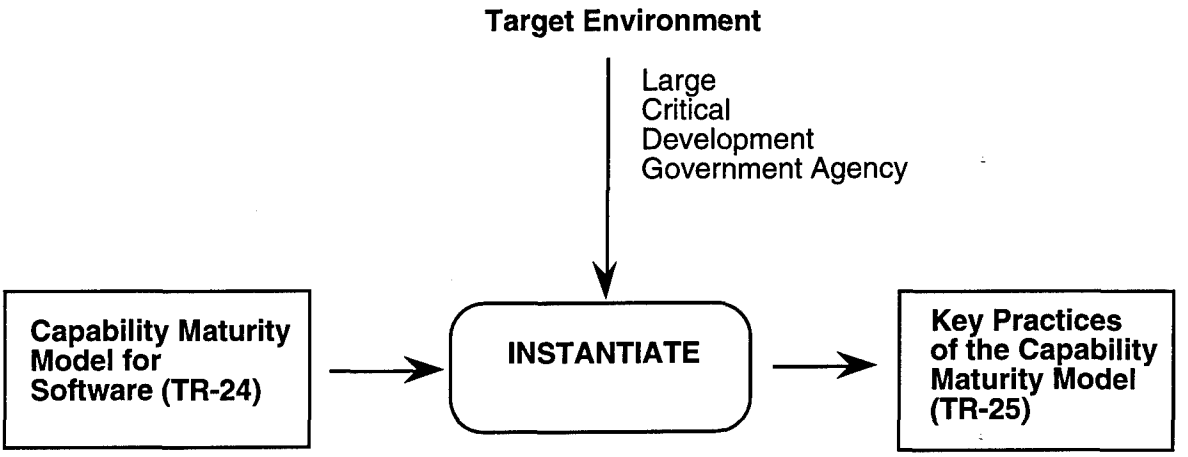

Figure 1-1: Volumes of the Software Capability Maturity Model

1.1 Formality of Tailoring

The tailoring approaches described in this report are somewhat formal. Frequently tailoring (and interpretation) will be more informally performed. The concepts and approaches described here should be useful for tailoring (and interpretation) at various levels of formality.

1.2 Purpose

Although the SW-CMM is recognized as a valuable contribution to the state of software process improvement, there is a large population of software-producing and -acquiring organizations that need to interpret and/ or tailor the key practices before they can be applied. This report explores some of the areas where adjustments to the SW-CMM practices will most likely be required and proposes techniques designed to resolve the conflict between maximizing both process commonality and project efficiency.

This work represents the knowledge and analysis of the authors, to date. In the spirit of the SW-CMM and continuous process improvement, we welcome any feedback, comments, or criticisms.

1.3 Intended Audience

This report is intended for use by members of process groups and others interested in the development of software processes, which are compatible with the SW-CMM, for organizations and projects. It is assumed that the user is familiar with the structure and content of the SW-CMM and is knowledgeable in software engineering and software management principles.

1.4 Organization of This Report

The report is organized into six primary chapters and three appendices. This chapter describes the overall document, defines the terms "tailoring" and "interpretation," and presents an overview of the way tailoring is described in the remainder of the report.

Chapter 2 places tailoring in an organizational framework. It identifies the different contexts in which the word "tailoring" is used and relates tailoring activities to the software management process. It also describes key issues an organization should consider before tailoring the practices in the SW-CMM.

Chapters 3 and 4 address two kinds of tailoring: the tailoring used in the definition and development of the organization's standard software processes (OSSP) and the tailoring used in the definition and development of the project's defined software processes. In the first case, the effort is to tailor the practices in the SW-CMM to generate a set of suitable requirements for the OSSP, and in the second case, the effort is to tailor the OSSP into a suitable process for the project. These chapters also describe techniques appropriate for accomplishing these two types of tailoring. The general approach is to utilize and build upon the Software Process Framework [Olson94].

Chapter 5 briefly describes additional data sources.

Chapter 6 concludes with a brief summary. The three appendices contain a glossary, a list of acronyms, and a list of codes used in the example tailoring tables.

1.5 Interpretation vs. Tailoring

The terms *interpret* and *tailor* are both used to describe a process that involves establishing a relationship (e.g., identification, correlation, and/or derivation) between different levels of process definitions or between process requirements and process definitions. For example, one might interpret the key practices of the SW-CMM in terms of an organization's standard software process (a possible instantiation of the model), or tailor the organization's standard software process to derive a project's defined software process. In this report, the terms are differentiated by the following definitions.

Interpret: The act of analyzing the definitions and/or terms of a general process description with respect to an existing instantiation of the description in order to facilitate the understanding of the relationship between the description and the instantiation, e.g., interpreting the key practices of a key process area of the SW-CMM in terms of the processes present at a potential software contractor.

Tailor: The act of adjusting the definitions and/or particularizing the terms of a general description to derive a description applicable to an alternate (less general) environment, e.g., tailoring the key practices of the software configuration management (SCM) key process area for use on a small project or tailoring the key practices of the SW-CMM to produce process requirements for an organization's standard software process.

Figure 1-2 summarizes these two concepts in graphical form. Interpreting is defined as correlating practice with the model or correlating different instantiations of the practice across environments. Tailoring is defined as adjusting practices for differences across environments.

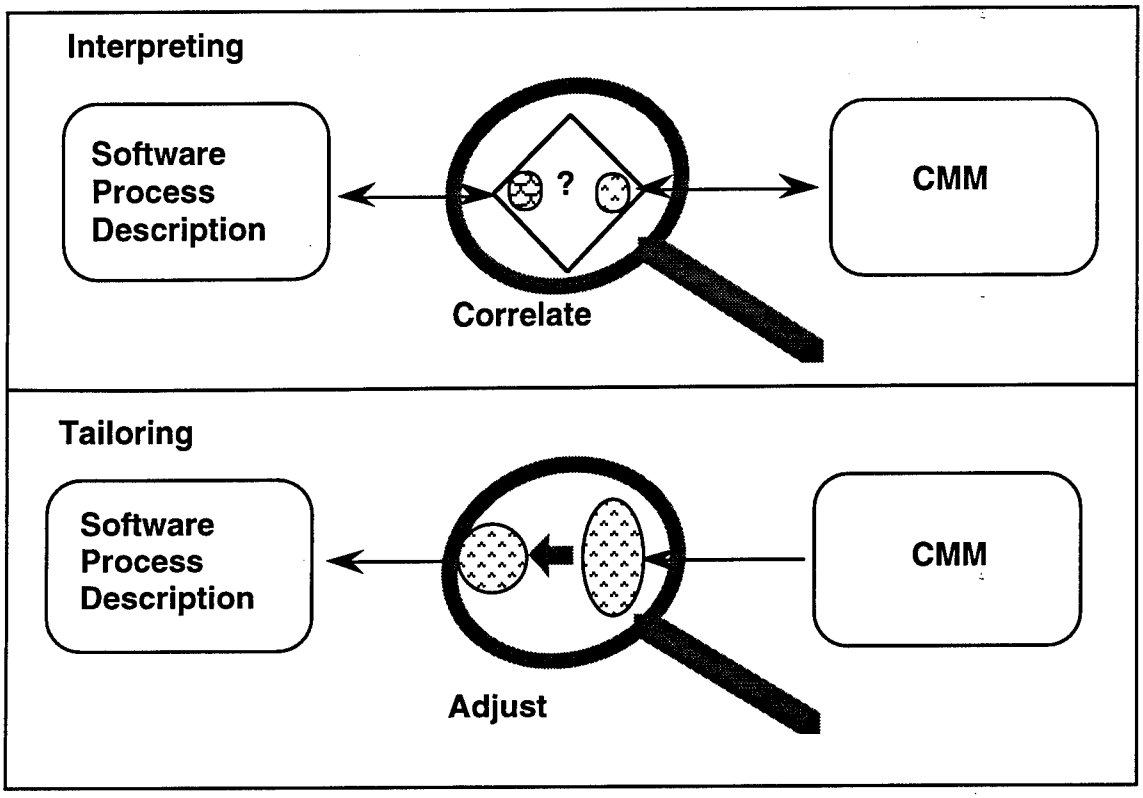

Figure 1-2: Correlating /Adjusting to Alternate Environments

1.6 General Approach

Interpretation and/or tailoring need to be done in a thoughtful, disciplined manner. Interpreting the SW-CMM terminology (for the various groups, documents, process artifacts, etc.) in terms that each organization understands, is not a trivial task.

In exploring the characteristics of organizations and projects that contribute significantly to the need to tailor the practices of the SW-CMM, we identified two very important issues.

First, similar types of projects or similar applications appear to require different interpretations and/or tailoring of the practices. For example, different contractors developing large software systems for the same government agency have varying levels of difficulty in applying the various key process areas in their environments. A significant cause of this appears to be that some organizational structures seem to be more compatible with the key practices than others.

Second, many large organizations (especially large government contractors) contain multiple environments with very different characteristics. There are often small-project

environments within the large organization that have more in common with small-company environments than with the large-project environments in their own organization. This is especially true of the amount of resources available to provide organizational infrastructure.

The main point is that almost every organization or project will have to perform some tailoring or interpretation in order to apply the key practices in their specific environment. Indeed, the SW-CMM itself includes provisions for developing and applying tailoring within organizations (in the Organization Process Definition and Integrated Software Management KPAs, respectively).

This report presents a tailoring framework that identifies process artifacts, tailoring processes, and their relationships to project artifacts, and explores the nature of various kinds of tailoring used in the definition and development of software process descriptions. Techniques appropriate to each type of tailoring are then discussed.

The general approach of these techniques is to utilize and build upon the Software Process Framework (SPF), whose purpose "is to provide guidance for designing, analyzing, and reviewing software processes for consistency with the SW-CMM." [Olson94] The SPF provides checklists to aid in the efficiency of the analysis and to capture the results. In this report, we recommend extending many of the checklists by utilizing disposition codes, in place of the simple checkmarks. We also recommend a specific subset and order of the SPF checklists. In addition to the SPF checklists, alternative approaches have been suggested in some areas.

2 Tailoring Framework

The tailoring framework presented in this chapter assumes that the SW-CMM forms the "requirements" for an organization's standard software process (OSSP), which in turn is the basis for a project's defined software process (DSP), which provides the foundation for a project's software development plan. Before the details of the framework are discussed, it is important to note that all tailoring must be done in the context of the organization's (or project's) overall process improvement program.

2.1 Contexts for the Capability Maturity Model

One of the key contexts in the SW-CMM is expressed in the following quote: "The key practices of the SW-CMM are expressed in terms of what is expected to be the normal practices of organizations that work on large, government contracts." [Paulk 93b] In presenting the key practices in this way, the SW-CMM uses terminology common to certain organizational structures and a certain set of roles (referred to below as an environment). However, the extension to non-government contracts has been quite successful and extensive. Indeed in 1994, 39% of the organizations reporting SW-CMM-based evaluation or assessment results to the SEI, identified themselves as commercial or in-house software developers.

The SW-CMM is primarily used by organizations in two main contexts: process improvement and capability evaluations. The following quote from the SW-CMM describes its intended use for process improvement. "The Capability Maturity Model (CMM) for Software provides software organizations with guidance on how to gain control of their processes for developing and maintaining software. . . The CMM was designed to guide software organizations in selecting process improvement strategies by determining current process maturity and identifying the few issues most critical to software quality and process improvement." [Paulk 93a]

The SW-CMM is also sometimes used as the underlying model for capability evaluations as part of a procurement cycle. In fact, the SW-CMM "effort was initiated in response to a request to provide the federal government with a method for assessing the capability of its software contractors. . . . The document can be used . . . by acquisition organizations or prime contractors wanting to know the risks of having a particular software organization perform the work of a contract." [Paulk 93a]

Both contexts have induced many organizations (especially those involved in government contracting) to use the SW-CMM as a set of process requirements for their OSSP. However, there may be many other requirements driving the OSSP, including business goals, ISO 9000, customer requirements, etc. All of these requirements need to be integrated into a single process improvement program. Many organizations are using the IDEAL[sm] model as the basic steps for their process improvement program. This model is

---

[sm] IDEAL[sm] is a service mark of Carnegie Mellon University.

depicted in Figure 2-1. A procurement evaluation, then, may act as the stimulus for improvement (in the Initiating phase), but the primary use of the SW-CMM is to appraise and characterize the current practice (in the Diagnosing phase). The IDEAL model also shows that process improvement in general (of which tailoring is a part) is an iterative process. Tailoring is not performed once. It is an ongoing, continuously evaluated process that is repeated many, many times.

It is in this capacity that the SW-CMM appears as the starting point for process definition in Figure 2-2, which presents a framework for the tailoring discussion. This diagram and discussion is most relevant to an organization that is operating near or above the defined level (Level 3) of the SW-CMM. However, the techniques and analysis performed can and should be used by organizations at all levels of maturity. This diagram and discussion assume that a project will be managed using a software development plan that is based on the project's defined software process, which is, in turn, derived from the organization's standard software process (OSSP), which is derived from the SW-CMM. Another key assumption is that the organization intends both to implement a set of processes that are compatible with the goals of the key process areas of the SW-CMM and build an infrastructure to ensure that the processes are practiced and institutionalized.

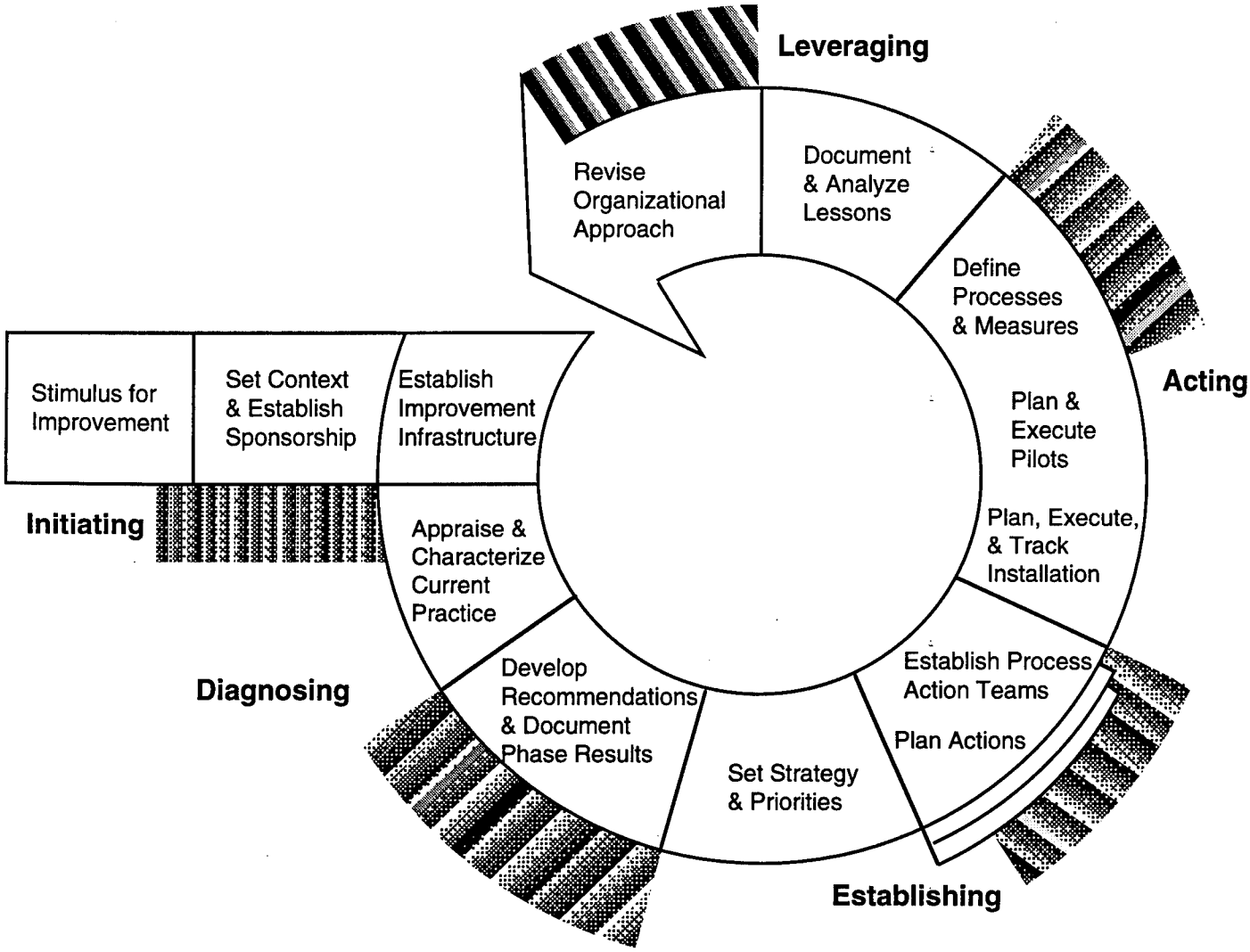

Figure 2-1: The IDEAL Model for Process Improvement

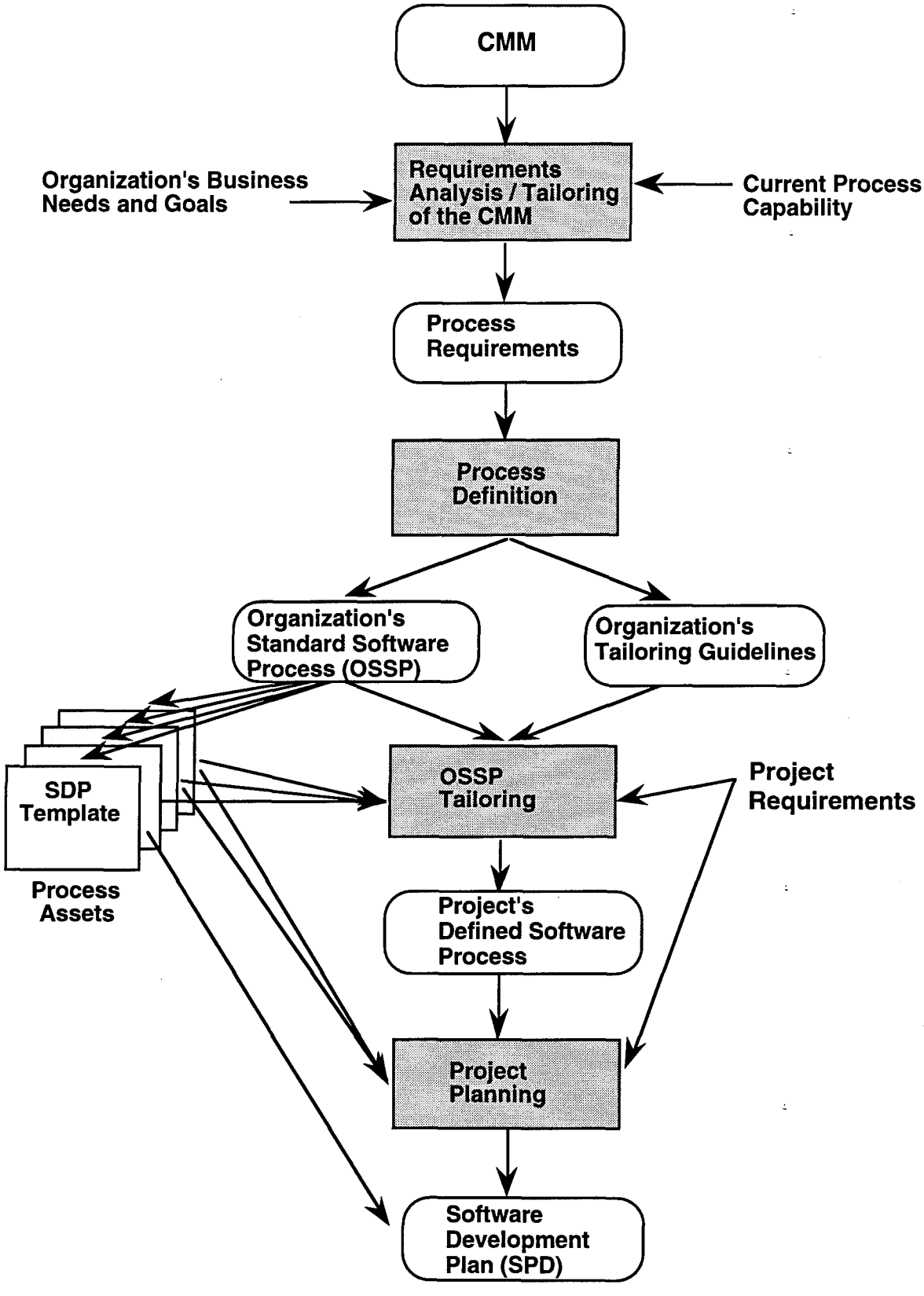

Figure 2-2: A Tailoring Framework

2.2 Considerations for Tailoring with the SW-CMM

As a prerequisite to using the key practices of the SW-CMM as process requirements, an organization must determine the similarities and differences between the environment expressed in the terminology of the SW-CMM and the organization's environment. The results of this analysis are a principle input into the organization's process definition activity. Important areas to address in this analysis include the following:

- similarities and differences between the organizational structure implied by the SW-CMM terminology and the organization's structure
- similarities and differences in the assumed and actual customer relationships
- degree of formality, frequency, granularity, or scope of the OSSP for the organization in general
- the specific business goals and needs to be addressed by implementing a process improvement program
- the current process capability of the organization

The following sections further describe these issues to consider when tailoring with the SW-CMM.

2.2.1 Organizational Structure

The SW-CMM's terminology implies an organizational structure and set of roles that influences the key practices in several ways. The most obvious is size. It can be argued that the SW-CMM assumes a rather large organization with well-defined and separate functional roles developing and maintaining large systems. Examples of these roles are software quality assurance group, software configuration group, and training group. The SW-CMM defines a group as

"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." [Paulk93b]

Even though the SW-CMM goes to great pains to define a group as such, many smaller organizations have difficulty mapping or tailoring the defined SW-CMM roles to their current organizational structure. The key thing to remember is that *someone* is responsible for the task or activity, not that they belong to a group with a specific title, or that the group with the appropriate title is responsible for everything in the SW-CMM.

Since process formality and rigor are most clearly required with large teams (to ensure clarity of communications and role boundaries), the key practices tend to be biased towards formality and rigor. Smaller organizations sometimes assume that this formality or discipline is unduly burdensome. However, the experiences with the Personal Software Process (PSP) indicate that formality and discipline, even at the individual level, produce measurable

increases in product quality, productivity and schedule commitment [Humphrey95]. We recommend that each organization at least pilot these best practices to determine what works best for them in their environment.

In organizations that have a different structure than that implied by the SW-CMM, the key practices will have to be adjusted, mapped, or correlated to the actual structure of the organization.

2.2.2 Customer and End-User Relationships

A specific type of relationship to the customer and/or end user is assumed in the key practices. The contract environment assumes that there is a single known customer who specifies the system requirements. It is further assumed that the customer has the time, money, knowledge, and desire to participate in development reviews. For multicustomer environments, when end users are not known, or when the customer cares only about the product and not the development process, the organization may have to provide surrogate customers and/or users.

In commercial environments, these concepts need to be translated into something meaningful. For example, the single customer contract that specifies the system requirements, might refer to the marketing analysis that specifies the desired product qualities/functions. One needs to read beyond the literal interpretation of the words in the key practice and understand the intent. In doing this analysis, it is very beneficial to refer back to the goals of the key process area.

2.2.3 Tailoring by Degree

Tailoring by degree is perhaps the most common form of tailoring in relation to the SW-CMM. Tailoring by degree means that the intent of the tailored object is met with only minor changes to the detail. In the case of the SW-CMM, the tailored objects include activities, work products, and process artifacts that may need to be altered in some minor way. In order for the tailoring to be consistent across all of the objects, the way in which an object is altered is specified by attributes of each object. In the case of the SW-CMM, various attributes of the activities, work products, and process artifacts can be defined for each organization. Some of the more common attributes are

- formality - The essential aspects of an activity can be performed with varying degree of detail, or attention to formal rules, procedures, or standards. For example, using "managed and controlled" procedures (simple version and change control) is considered informal, whereas full configuration management as described in the configuration management KPA is considered very formal (change control boards, configuration status reports, etc.)
- frequency - There are many activities in the SW-CMM that are performed on a "periodic" or "event-driven" basis. The frequency of each activity needs to be interpreted in light of the organization's and project's needs.

- granularity - The level of detail needed in the process definition may vary. The SW-CMM often states, "This document typically includes..." An organization may want to include more or less detail than suggested, depending on how its process artifacts are structured, on the desired consistency in level of detail with other artifacts, etc.
- scope - It may not make sense to perform certain activities, due to organizational constraints, business environment, etc. The easy example here is subcontract management — if an organization never uses subcontractors, it does not need to consider this KPA. Better examples might be the decision to forgo the independent review of the SQA organization, or the decision not to measure the effort expended in performing the tracking and oversight activities. Wholesale elimination of KPAs or activities should be done with an understanding of the risks involved and appropriate cost/benefit justifications.

2.2.4 Business Goals

The business goals and needs of the organization must be known to tailor the SW-CMM to a specific organization. Since the key practices are intended for a wide audience, the SW-CMM assumes the following business goals: decreased cost, increased quality, better schedule performance, and a continuously improving software process. All of these issues are important, but one or more of these issues may be relatively more important to any particular organization.

2.2.5 Impact of Maturity Level on Tailoring

To tailor the key practices of the SW-CMM to a specific organization, it is necessary to know the current process capability of that organization. Since the SW-CMM is organized by maturity levels of increasing process capability, the extent of organizational involvement in the projects' processes must be based on how the organization is currently operating.

For example, if the organization is currently at or near the repeatable level, the approach of Figure 2-2 may not be appropriate. A more appropriate approach might be that presented in Figure 2-3. In this approach, which is consistent with the repeatable level,[1] the organization indicates the kinds of practices that each project must implement as organizational-level policies, but does not specify the details of the practices. For example, a tracking policy at the organizational level might indicate that the projects must track and report development progress against their software development plan, but might not specify the metrics to be used or the reporting frequency; these might be left for the project to decide.

It should be clear from the above discussion that an organization's tailoring needs and process requirements issues will change as the organization matures. In other words, as the organization moves around on the circle of the IDEAL model, it will continually analyze

---

[1]Figure 2-3 represents a minimum approach to attaining the repeatable level. Some organizations may find it more effective to provide some of the capability present in Figure 2-2 early in the improvement process. The question of how much organizational support to provide to projects at the repeatable level must be based on the needs of the organization.

the new set of process requirements, assess the current process, and institute process improvements. Thus, as an organization moves up the maturity ladder, it can expect to analyze and tailor its process requirements on a regular basis. This reinforces the earlier concept that tailoring is not a one-time event, but a repeated, ongoing analysis that is embedded within the process improvement framework.

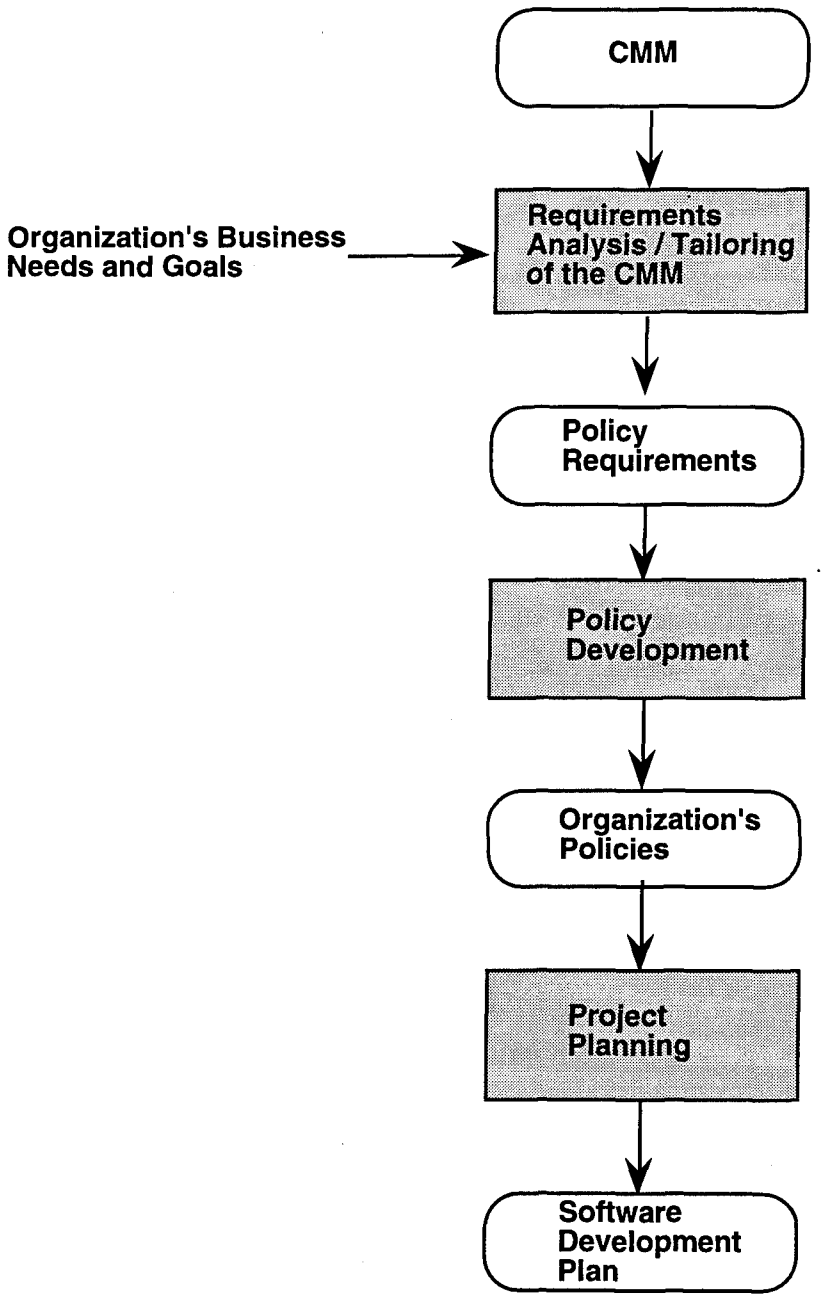

Figure 2-3: Process Tailoring at Levels 1 and 2

3 Requirements Analysis / Tailoring with the SW-CMM

Each organization creates an OSSP to meet a set of requirements. These requirements may or may not be formally documented or understood. In any event, the requirements typically come from a variety of inputs, such as the SW-CMM, ISO-9000, Total Quality Management or other quality initiatives, the business needs of the organization, and the current maturity level of the organization. The purpose of requirements analysis/tailoring with the SW-CMM is to ensure that the requirements used to define the OSSP are compatible with the SW-CMM.[2] As noted previously, as an organization matures, there will be a recurring need to examine the requirements for the OSSP and augment the OSSP and other process assets to incorporate activities that are associated with higher level KPAs.

Compatibility with the SW-CMM is addressed at the key process area level. To ensure compatibility with a given KPA, the organization must ensure that the goals of the KPA are satisfied and that the processes that achieve these goals are institutionalized. The key practices and subpractices associated with each KPA provide examples of typical practices that would meet the intent of the KPA as implemented in a large organization developing and maintaining software-intensive systems. They are not requirements, but do serve the useful purpose of demonstrating a common way of meeting the intent of the KPA.[3] It is important to note that the KPA goals address implementation and are comparatively silent on the subject of infrastructure and institutionalization. It is in the area of infrastructure and institutionalization that the key practices can provide especially useful examples of suitable practice.

The tailoring approach that is presented in this report is to examine systematically *three* process elements in each KPA to determine the action necessary to ensure that the organization generates an appropriate OSSP. The SPF describes a process element as those portions of a process description that satisfy the process definition criteria. Process definition criteria are "the set of information that must be included in a software process description for it to be usable by the people performing the process." [Olson94] Process definition criteria are typically stated as questions (who, what, when, why). Process elements, therefore, include purpose, input, output, role, activity, entry and exit criteria, and procedure to name a few. Each process element is presented in the SPF as a series of checklists. We extend the use of the checklists by suggesting disposition codes, instead of using a simple (yes/no) checkmark. (For some of the more complex KPAs, it may be

---

[2]The activities described in this section correspond to the top shaded box of Figure 2-2. They are appropriate for an organization operating near or above the defined level. At the repeatable level and below, organizational policies may be used to communicate process requirements to individual projects.

[3]There are no "shall" statements in the CMM, and the key practices were not intended to be used as required process statements.

beneficial for the organization to extend the examination to include other process elements or additional detail.)

The organization may use the key practices as written, introduce vocabulary changes to harmonize the SW-CMM with organizational structure and culture, or implement an equivalent set of practices that achieve the same process capability represented by the KPA goals. When replacing one or more practices, it is important to determine the intended value to ensure that the replacement set of practices is capable of providing an equivalent process capability. Since some of the SW-CMM terminology is most common to large, government contracting organizations, some translation and tailoring may be required for other organizations. Additionally, a different environment and set of characteristics may generate requirements for practices that were not necessary in the case assumed by the SW-CMM. This is probably most true in areas that address organizational and project interfaces.

3.1 Tailoring Approach

The SPF defines several process elements. Some of the most important ones are listed below:

- *Roles:* Describe individuals or collections of individuals participating in process activities. They may be suppliers, customers, agents, reviewers, or verifiers of a practice.
- *Entry criteria:* Describe the conditions under which an activity can start. The entry criteria can be a simple or compound predicate about the state of a work product, role, or activity. "Software requirements shall be baselined under formal configuration management control prior to starting software design activities," is an example of an entry criteria.
- *Inputs:* Describe those items or work products produced by a prior activity and used by the current activity. Software requirements are an input to the software design activities.
- *Activities:* Describe what is being done. They may be directly associated with the production of a product, a management function, or a service provided to help those directly involved to operate more effectively or efficiently.
- *Outputs/work products:* Describe those items that are produced by the process, i.e., items that are the direct result of executing a process step. Software modules, tested code, meeting minutes, and SQA reports are all examples of work products. Work products exist only if the process is executed.
- *Exit criteria:* Describe the conditions under which an activity can be declared complete. The exit criteria can be a simple or compound predicate about the state of a work product, role, or activity. "Software requirements have been reviewed by software managers and other affected groups," is an example of an exit criteria.

Tailoring must be managed and executed in an orderly manner. The approach used in this report is to use the three process elements of outputs, activities, and roles, in that order. The SPF provides checklists to aid in the efficiency of the analysis and to capture the

results. Techniques for analyzing and tailoring each of the three process elements are described in the following sections.

3.2 Output Mapping

A good first step is reviewing the outputs of each KPA. The outputs are fundamental in meeting the goals of each KPA — if an organization does not produce the output, it is difficult to show that the goals are met. The SPF output checklists identify the recommended outputs for each KPA. An example SPF output checklist is shown in Table 3-1. The checklists have columns for indicating (with a checkmark [√]) if the output is produced, entering the organization's terminology for an output, and including a reference to the specified output in the OSSP.

Table 3-1: Example of SPF Output Checklist (for Requirements Management)

√	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)		
	Impact to existing commitments. (L2-7, A3, 1)		
	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)		

Since an organization may not have an exact match to the document set referenced in the SW-CMM, it is useful to map document content from the documents referenced in the SW-CMM to those present in the organization. The first step is to ask the question: Does our organization produce this output at all? If so, does the organization's document include all of the content specified or recommended in the SW-CMM? Rather than a simple checkmark

(√), codes can be used in the column to indicate the degree of coverage. For example, the following codes could be used:

C - Complete coverage

S - Shared coverage - All items are covered, but spread across multiple documents. Specify the names of the different documents in the Organizational Output column.

P - Partial coverage - Some items recommended in SW-CMM are not in the current organization document.

Blank entries (indicating that no organizational document exists) or entries marked with a 'P' indicate that the organization should pay special attention to the content of the document identified in the SW-CMM. It may be that the content is not necessary or inappropriate to the organization; however, it may also mean that the organization's document set is missing some useful content.

3.2.1 Extensions/ Alternatives

The key practices of the SW-CMM reference and define a considerable volume of documentation and artifacts, besides outputs. The documents vary in purpose, scope, formality, and subject area. Examples of documents include policies, practices, procedures, reports, plans, specifications, and standards. The presence of these references is not intended to require an organization to use these exact document descriptions. Rather, it indicates the content of a typical document set for an organization that has implemented a set of KPAs. The document mapping may be extended to these other documents by using other SPF checklists. Specifically, the checklists for inputs, policies, procedures, and standards cover most, if not all, documents mentioned in the SW-CMM.

3.3 Activity Mapping

Once the outputs (and potentially other documents) have been identified and mapped, the next logical step is to examine in detail the activities that generate and support those outputs. The SPF activity checklists, shown in Table 3-2, list the recommended activities for each KPA. The checklists have a column for indicating (with a checkmark [√]) if the activity is performed, and a column for including a reference to the specified activity in the OSSP. Rather than using a simple checkmark, however, we recommend the following approach to address each of the activities of a KPA and record an initial disposition for tailoring each activity.

1. Examine each activity in the KPA to decide the relevance of the practice to the organization. This analysis should be performed carefully, keeping in mind that many of the activities are closely related to other activities and, thus, decisions on one activity may affect decisions for other activities. Factors to consider when making this decision include those discussed in Section 2.2 of this report, namely, the organizational structure, customer and end-user relationships, attributes for tailoring by degree, business goals, and current/future maturity level.

2. For each activity, record the disposition code (Accept, Expand, Tailor, Optional, Not Recommended) for your organization. The disposition codes are detailed in Table 3-3. Note: The results of this examination can be used in further tailoring activities. Therefore, it is useful to include the rationale for decisions made during the examination, especially if a practice is found not to be necessary or desirable.

Table 3-2: Example of SPF Activities Checklist (for Organization Process Focus)

√	Activities	References
	The software process is assessed periodically, and action plans are developed to address the assessment findings. (L3-6, A1)	
	The organization develops and maintains a plan for its software process development and improvement activities. (L3-7, A2)	
	The organization's and projects' activities for developing and improving their software processes are coordinated at the organization level. (L3-7, A3) This coordination covers the development and improvement of: ❑ The organization's standard software process. ❑ The projects' defined software processes.	
	The use of the organization's software process database is coordinated at the organizational level. (L3-8, A4)	
	New processes, methods, and tools in limited use in the organization are monitored, evaluated, and, where appropriate, transferred to other parts of the organization. (L3-8, A5)	
	Training for the organization's and projects' software processes is coordinated across the organization. (L3-8, A6) ❑ Plans for training on subjects related to the organization's and projects' software processes are prepared. ❑ Where appropriate, training may be prepared and conducted by the **group responsible for the organization's software process activities** (e.g., software engineering process group) or by the **training group**.	
	The **groups involved in implementing the software processes** are informed of the organization's and projects' activities for software process development and improvement. (L3-9, A7)	

Table 3-3: Description of Disposition Codes

Code	Description
A	Accept - Necessary and desirable practice that is acceptable as written.
E	Expand - Necessary and desirable practice that requires the addition of local definitions for one or more terms to be used in this environment. Different environments may require different definitions. Each unique set of definitions should be identified with its own subscript, and environments using the same definitions should use the same subscript. (An example of expanding would be requiring the use of a local documented procedure for reviewing changes to allocated requirements. This is an expansion, since the SW-CMM (Requirements Management, Activity 3) specifies that the review be performed, but not that a documented procedure be used.)
T	Tailor - Necessary and desirable practice that requires some adjustment to be used in this environment. The adjustment involves more than the inclusion of local definitions. Different environments may require different adjustments. Each unique set of adjustments should be identified with its own subscript. Environments using the same adjustments should use the same subscript. (An example of tailoring would be having an organizational policy that requires the use of local procedures for all phases/aspects of software development, instead of detailed policies for requirements management, project planning, etc. This is tailoring, since the details of each policy are not included, but the intent of setting a policy for the organization is met.)
O	Optional - Practice may be useful for some, but not all, projects in this environment.
NR	Not recommended - Practice is not recommended for this environment.

The analysis should focus on whether or not the activity is performed, not by whom, or where the output is documented. The outputs were mapped previously, and the roles will be mapped in the next section. The analysis should reference where in the OSSP this activity is mentioned.

3.3.1 Extensions/ Alternatives

This analysis can be extended to include other key practices from the SW-CMM, not just the activities. The SPF has checklists for reviews and audits, training, tools, and measurements. These can be used to supplement the activity checklists, if desired, using the disposition codes above.

Alternatively, instead of using the checklists, an organization can simply use the SW-CMM itself. In this approach, an organization painstakingly steps through each and every key practice, including all the other common features — commitment to perform, ability to perform, measurement and analysis, and verifying implementation, not just activities. A disposition for each practice is recorded, similar to that above. This approach is exhaustive and thorough, but may take significant time and resources to complete.

3.4 Role Mapping

As stated earlier, there are many different styles of organizations that are involved in software development and maintenance. Further, within groups using the same organizational principles, there are differences in the way the principles have been applied. Given this multiplicity of organizational structures, it is necessary to provide a method to identify specific attributes of structure that can be used when tailoring structure-sensitive practices. Tabulating the existence, nature, and interactions of software roles appears to be a reasonable approach.

When an organization's structure differs significantly from the typical structure assumed by the SW-CMM, the following techniques are useful in identifying practices that require tailoring.

3.4.1 Role Translation

This technique is intended to identify which organizational role or roles are responsible for performing the roles referenced in each of the key practices of a KPA. The information developed is useful in adjusting the key practice definitions to fit an organization's structure as expressed by its software development roles. It is also useful in identifying any existing "process holes" which can be addressed by future process improvement activities.

The SPF role translation table, shown in Table 3-4, can be used to record which role in the organization is responsible for each SW-CMM role. The table is contained in Appendix C of the SPF, which also contains definitions of frequently used roles to assist in the translation. The role translation table has one column for SW-CMM roles/groups and a column for the organization's roles/groups. The intent is to identify each role in terms that are meaningful to the organization. If a specific role is not filled, it is useful to review the definition of the role (contained in Appendix C of the SPF) and the activities in which each role participates. These activities are contained in the KPA sections of the SPF and referenced in the next section of this report. SW-CMM roles may also be shared across organizational roles.

3.4.2 Role Checklist

Once the SW-CMM roles have been translated to the organization's roles, then you can perform a final check to determine if the SW-CMM activities that each role participates in, are also covered. In the SPF, each KPA has a role checklist (see Table 3-5) that lists the roles and the activities in which they participate. The checklists have a column for indicating with a checkmark (√) if the role performs the specified activity and a reference column for identifying where the OSSP references the activity.

Table 3-4 : Example of SPF Role Translation Table

SW-CMM Roles/Groups	Your Organization's Roles/Groups
Administrative personnel	
Affected groups	
Affected individuals	
Affected managers	
Customer	
Customer SQA personnel	
Documentation specialist	
End user	
Engineering group	
Experienced individuals who have expertise in defining and analyzing software processes	
Experts independent of the SQA group	
First-line software managers	
Group responsible for analyzing and allocating system requirements	
Group responsible for coordinating the organization's software process activities (e.g., SEPG)	
Group responsible for coordinating the quantitative process management activities for the organization	
Group responsible for providing the critical dependency item	
Group responsible for system and acceptance testing	
Group responsible for the organization's technology change management activities	
Group responsible for the organization's software process activities	
Group responsible for the system requirements	
Group that defines and maintains the affected process descriptions	

The intent here is to determine how the SW-CMM roles/activities translate into the organizational structure. The following steps clarify the recommended usage of the role checklist:

- If the activity is performed by the organizational role identified in the role translation table, then mark the activity with a simple checkmark.
- If an activity is performed by another organizational role, or shared across roles, then that can be indicated by writing the name of the organizational role(s) in the Reference column, along with the OSSP reference.
- If the activity is modified, or not performed at all, then an activity disposition code can be used (see Table 3-3) to indicate if the activity is appropriate for the organization. If each of the activities was assigned a disposition code (see section 3.3), this can reference the previous work.

Table 3-5: Example of SPF Role Checklist (for Training Program)

√	Role	Activities Participated in...	Reference
	Affected groups	The organization's training plan is readily available to the **affected groups** and individuals. (L3-31, A2, 6)	
	Affected individuals	❑ The organization's training plan is reviewed by the **affected individuals** when it is initially released and whenever major revisions are made. (L3-31, A2, 4) ❑ The organization's training plan is readily available to the **affected** groups and **individuals**. (L3-31, A2, 6)	
	Manager	A **manager** is designated to be responsible for implementing the organization's training program. (L3-28, Ab2, 1)	
	Senior management	The training program activities are reviewed with **senior management** on a periodic basis. (L3-35, V1)	
	Software manager	**Software managers** receive orientation on the training program. (L3-29, Ab4)	
	Training group	Members of the **training group** have the necessary skills and knowledge to perform their training activities. (L3-28, Ab3)	

For organizations developing software for a multicustomer environment (as opposed to development under contract for a single external customer), some of the activities assumed for the customer and end-user roles may require special attention. While some of these activities will be not applicable, others will be performed by roles outside the software organization, but inside the company (for example, marketing).

3.4.3 Extensions/ Alternatives

An alternative to the above two techniques (role translation and role checklist) is the Activity Role technique. This technique is intended to identify which organizational role or roles are responsible for performing the various "activity roles" referenced in each of the key practices of a KPA. An "activity role" is defined as all the explicit and implied players needed to execute an activity successfully. Typically, the activity roles include

- supplier - person or group responsible for supplying requirements or other necessary inputs for the activity
- agent - person or group responsible for performing the main elements of the activity
- customer - person or group receiving the output of the activity
- verifier - person or group ensuring that the quality of the output is satisfactory and that the process to perform the activity was correctly implemented
- reviewer - person or group responsible for reviewing and/or approving outputs. (Reviews are typically performed against specific standards and criteria.)

It should be noted that the SW-CMM does not explicitly define all of these roles for each activity. Therefore, performing this analysis is complicated because it is not always clear which group the SW-CMM would assign to each of these roles. However, in order for an organization to implement an activity successfully, a clear definition of all these roles is extremely helpful. The information in this section is useful in adjusting the key practice definitions to fit an organization's structure as expressed by its software development roles. It is also useful in identifying any existing "process holes" which can be addressed by future process improvement activities.

A matrix similar to that shown in Table 3-6 can be used to record which role in the organization is responsible for each key practice. The matrix has one column for each activity role plus a column to indicate if the key practice is not applicable to the organization. There is one row for each key practice of the KPA.

Table 3-6: Example Matrix of SW-CMM Key Practices and Activity Roles

SW-CMM	**Activity Role**					
Key Practice	Supplier	Agent	Customer	Verifier	Reviewer	N/A
CO_1						X
CO_2	O	SEPG	SEPG	N/A	SEPG	
.						
AB_1	—	SEPG	SW	SQA	SEPG	
AB_2						

The cell at the intersection of a row and column contains the organizational role responsible for performing the activity role for the key practices. In addition to the organizational roles, the following special cell codes are used:

— An activity role is not referenced in the key practice.

O The activity role is not performed or not defined by the organization.

N/A The activity role is not appropriate for the organization.

X (in the N/A column) The practice does not apply to the organization.

The examples shown in Table 3-6 are interpreted in the following way (the left column refers to the key practice row)

CO_1 This practice does not apply to the organization. (X in N/A column)

CO_2 For this practice, the organization has not defined responsibilities for specified inputs (O in the Supplier column)

The software engineering process group (SEPG) is tasked to do something, acts as the customer, and reviews the work product (SEPG in the Agent, Customer, and Reviewer columns)

SQA is tasked to review process and standard compliance (SQA in Verifier column).

AB_1 For this activity, there are no specified inputs (— in the Supplier column).

The SEPG is tasked to do something, and to review the work product (SEPG in the Agent and Reviewer column).

The software developers are the customer (SW in the customer column).

SQA is tasked to review process and standard compliance (SQA in the Verifier column).

For organizations developing software for a multicustomer environment (as opposed to development under contract for a single external customer), some of the activities assumed for the customer and end-user roles may require special attention. While some of these activities will not apply, others will be performed by roles outside the software organization, but inside the company (for example, marketing).

3.5 Tailoring and the Infrastructure Common Features

Within KPAs, practices are organized into common features. Each common feature focuses on one aspect of achieving and institutionalizing the process capability associated with the KPA. Practices located within the *activities performed* common feature address such issues as establishing plans and procedures, performing the work, tracking work in progress, and

taking corrective action.[4] The goals of the KPA most commonly address these same issues and are relatively silent about infrastructure and institutionalization.

The other common features (commitment to perform, ability to perform, measurement and analysis, and verifying implementation) are more concerned with infrastructure and institutionalization. It is important to understand the purpose of each of these common features and ensure that their purposes are achieved. Implementing the practices that meet the intent of the activities performed common feature on projects is not sufficient to establish an organizational process capability. Sufficient infrastructure must be in place to ensure that, in addition to being practiced, processes are documented, trained, supported and maintained, controlled, verified and validated, measured, and able to improve.

Table 3-7 describes the primary focus areas of each of the common features concerned primarily with infrastructure and institutionalization. KPAs differ in the amount of emphasis placed on these common features, but every KPA requires some effort in each common feature.

Table 3-7: Focus Areas of SW-CMM Common Features

Common Feature	Primary Focus Areas
Commitment to perform	Establish policy Establish leadership
Ability to perform	Establish structure/responsibility Ensure appropriate resources Provide training/orientation Identify prerequisites
Measurement and analysis	Establish process measures
Verifying implementation	Senior management oversight Project management oversight Software quality assurance activity

The structure and formality of the key practices in the common features describing infrastructure are typical of what would be expected in contracting organizations concerned with developing large, critical systems for a government agency. For organizations operating in different environments, alternate approaches to these common features may be appropriate. It is, however, necessary to ensure that these alternative approaches provide the necessary framework to ensure that the processes endure.

---

[4] Subpractices associated with the practices usually address such issues as managing and controlling selected work products, and ensuring that certain work products are reviewed by appropriate groups.

Tailoring a practice in a common feature that describes infrastructure most often takes the form of adjusting its scope, formality, or precision; *not* deleting it. For example, in a small-project environment, one individual might perform multiple roles which could span the practices of several KPAs. In this situation, requirements for measuring the status of each of these part-time activities could be replaced with a single measure of the combined effort. Thus, the status is known, but with less precision than if individual measurements were made for each activity. As another example, multiple formal, periodic, single-purpose meetings (that are of value on large programs) could be met in a small-team environment by having appropriate items on the agenda of periodic team meetings.

It is often the case in tailoring that one or more general requirements in the practices are replaced with a more specific statement directed at a single environment. Thus, a tailored version of a set of practices is usually compatible with, but more prescriptive than, the original practices.

Again, the SPF has several detailed checklists that can aid in this analysis. These include checklists for training, tools, reviews and audits, measurements, policies, and input and exit criteria. An extremely thorough and detailed analysis can be done by using all the checklists in the SPF. A minimal set are the three described — outputs, activities, and roles.

3.6 Additional Requirements

The scope of the requirements that result from applying the above tailoring techniques is very close to the scope of the requirements of the original SW-CMM practices. If the organization includes one or more environments that are significantly different than the typical environment expressed in the SW-CMM, there may be additional process requirements which have not been addressed. For example, if the organization's structure is different than the structure implied by the terminology in the SW-CMM, some of the requirements for role interfaces may be deleted as not applicable. This is appropriate, but it is also necessary to investigate the unique role interfaces in the organization to ensure that any additional process requirements are captured.

When the organization includes significantly different project environments, the requirements of each of the environments must be addressed. Since it is also highly desirable to maintain process commonality across the organization, the process requirements for all of the environments should be analyzed together. Organizations with multiple environments may examine the SW-CMM practices either by considering the organization as a whole, or by treating each environment as an individual organization. The former approach is more complicated, but has a greater potential long-term benefit. For example, in considering the whole organization there are opportunities to stress the process elements that are common and to transfer best practices across the organization. On the other hand, considering each environment as a separate organization is simpler, but it tends to emphasize process differences and it may be more difficult to maintain and implement process improvements.

Since the intent here is to create an OSSP that is compatible with the SW-CMM, the approach used should depend on whether the organization has (or intends to have) a single

OSSP or multiple OSSPs for each environment. Each organization needs to balance a "one-size-fits-all" approach, against maintaining several "custom-fit" OSSPs.

For an organization with a single OSSP, whenever a practice or set of practices requires different tailoring for different environments, a requirement for a set of acceptable alternatives should be documented. These requirements do not have to associate a specific environment with a specific tailoring need. It is sufficient to document the range of capability necessary to cover the needs of the relevant environments. The selection of alternatives for specific environments will be addressed when the OSSP and its associated tailoring guidelines are developed.

For an organization with multiple OSSPs, each environment will have its own set of tables. That is, the analysis performed and documented in the above sections is repeated for each environment. This set of tables can then form the basis of the requirements analysis for the OSSP for each environment. Experience has shown that a single OSSP is the norm, but each organization needs to determine what structure best meets its needs. The analysis for each environment can also be compared to determine the degree of overlap and hence aid in the decision to have one or multiple OSSPs.

4 The Organization's Standard Software Process and Tailoring Guidelines

This chapter deals with tailoring the organization's standard software process to derive the project's defined software process. It focuses on activities associated with the box labeled "OSSP Tailoring" in Figure 2-2. It assumes that the organization has developed an OSSP based on a tailored set of process requirements, derived from the SW-CMM and other business needs.

4.1 Tailoring Guidelines

An organization's standard software process is the means by which the organization expresses the requirements that all projects' software development processes must meet. The OSSP may take many forms and may include alternatives to support multiple life-cycle models. The objective of the OSSP is to establish process commonality across the organization's projects in order to support process measurement, process continuity, and process improvement. The associated tailoring guidelines are the means by which the organization recognizes the project's responsibility to address the impact of project-specific needs in the project's defined software process.

The specific form and content of an OSSP can vary from abstract descriptions to detailed implementations depending on the range of products and life cycles that it must support. In general, one would expect an OSSP to contain elements at several levels of abstraction. At the abstract description level, the OSSP will contain the primary elements of software process that all projects are expected to include and their interrelationships. At the detailed implementation level, descriptive data sufficient to execute the process element may be present. If there are elements of software process that the organization has decided will be executed in the same manner on all projects, the OSSP would probably contain or reference complete implementation descriptions. On the other hand, when the organization does not constrain a process element to the implementation level, the element's description will of necessity be at a higher level of abstraction. It is then up to the individual projects to complete the element's description to the implementation level. For example, the organization might require all projects to perform formal code inspections and include requirements for advance scheduling and package distribution, quality and performance metrics, and meeting minutes, but leave remaining details such as content of checklists and specific meeting roles to the project.

4.2 Process Tailoring

When organizations are operating near or above the defined level, an OSSP and a set of tailoring guidelines are used to develop the software processes used on each project. In the SW-CMM, the process of adjusting the OSSP to a process suitable for the particular business and technical needs of a project is referred to as tailoring the OSSP to form the project's defined software process.

The organization's tailoring guidelines must be developed and applied in a manner that will preserve the benefits of having common practices based on the OSSP. The guidelines must grant projects the flexibility to operate efficiently, while also preserving the maximum amount of commonality possible. At the individual practice level, the goal is to maintain as much of the practice as possible while adjusting practice attributes to achieve an implementation that is compatible with the nature and goals of the project. That is, try to limit the tailoring to changes in degree and not introduce changes in kind. As an example of this kind of a change, a project might vary the source of data to be collected in a practice, but it would not negate the need to collect the data.

Tailoring the OSSP into a project's specific process is remarkably similar to tailoring the SW-CMM into the OSSP. Some of the important areas to address are the same and include

- similarities and differences between the organizational structure implied by the OSSP and the project's structure
- customer relationships and requirements
- degree of formality, frequency, granularity, and scope desired for the project in general
- the specific business goals and needs to be addressed by the project
- the current process capability of the organization and the desired process capability of the project

The following sections further describe these issues to consider when tailoring the OSSP.

4.2.1 Project Structure

The OSSP may assume an organizational structure and set of roles that are not appropriate for every environment or project within an organization. For example, many OSSPs are written assuming a full development life cycle, and many small and maintenance projects may have difficulty in translating all the roles to their specific needs. The organization's tailoring guidelines should provide guidance in mapping the roles to known organizational environments. Certain OSSP activities will undoubtedly rely on assistance or inputs from other organizations. In tailoring the OSSP, one must ask: For this specific project, can that assistance or input be provided?

4.2.2 Customer and End-User Relationships and Requirements

The OSSP may assume a specific type of customer relationship that may not always be valid for all projects in an organization (maintenance projects and software that is internally developed and used come to mind). The OSSP may need to be tailored to reflect a specific customer or end-user relationship. Additionally, certain customers may levy specific requirements that conflict with the OSSP. The project's specific software process needs to be developed with the customer requirements in mind.

4.2.3 Tailoring by Degree

Tailoring by degree is perhaps the most common form of tailoring for the OSSP as well as the SW-CMM. Various attributes of the activities, work products, and process artifacts can be tailored or defined for each project. The attributes are identical to the SW-CMM tailoring discussed in Section 2.2.3 and are repeated here for convenience:

- formality - The essential aspects of an activity can be performed with varying degree of detail, or attention to formal rules, procedures, or standards. For example, using "managed and controlled" procedures (simple version and change control) is considered informal, whereas full configuration management as described in the configuration management KPA is considered very formal (change control boards, configuration status reports, etc.)
- frequency - Many OSSPs will define activities that are performed on a "periodic" or "event-driven" basis. The frequency of each activity needs to be interpreted in light of the organization's and project's needs.
- granularity - The level of detail needed in the process definition may vary. The OSSP may contain detailed process descriptions for some practices and high level descriptions for others. The projects will need to add detail where necessary. A project may want to include more or less detail than suggested, depending on how its process artifacts are structured, the consistency in level of detail with other artifacts, etc.
- scope - It may not make sense to perform certain activities, due to organizational constraints, business environment, etc. The easy example here is subcontract management — if a project never uses subcontractors, it does not need to consider these procedures in the OSSP. Better examples might be the decision to forgo the independent review of the SQA organization, or the decision not to measure the effort expended in performing the tracking and oversight activities. Wholesale elimination of KPAs or activities should be done with an understanding of the risks involved and appropriate cost/benefit justifications.

For a practice that is viewed as generally applicable to the project's environment, but not to the degree specified in the OSSP, tailoring by degree can be applied. This form of tailoring recognizes that, for some project environments, one or more aspects of the practice may require a different degree of execution. For such practices, the organization develops a set of alternative implementations varying with one or more of the attributes described above. Each organization needs to define a set of attributes that is useful and meaningful to the environments and projects that it serves.

4.2.4 Business Goals

The business goals and needs of the organization and the project must be known to tailor the OSSP to a specific project. In addition to the business goals assumed in the creation of the OSSP (goals such as decreased cost, increased quality, better schedule performance, and a continuously improving software process), each individual project will have specific business goals that may have an impact on the project's specific process. For example, has this project been chosen to pilot a new technology, is cost an overriding customer concern,

or is schedule more important? Also, how does this project intend to meet its own process improvement goals and aid the organization in meeting its overall process improvement goals? All of these may influence the implementation of the OSSP for the project.

4.2.5 Impact of Maturity Level on Tailoring

To tailor the OSSP to a specific project, it is necessary to know the current process capability of that organization and the desired process capability for the project. Higher level maturity activities may depend on the existence of supporting infrastructure (e.g., training and tools). The OSSP may be updated before the infrastructure is fully deployed, or the project may be attempting to operate at a maturity level above that defined in the OSSP. It is possible, and perhaps desirable in some circumstances, to have projects within an organization operating at different maturity levels. In such circumstances, the organization must decide whether the OSSP should reflect the leading edge (that which most or all project's should strive for) or the average (the current state for most projects). All of these factors clearly influence how a project will tailor the OSSP into its specific process.

4.3 Process Elements

Again, tailoring the OSSP into a project's specific process is remarkably similar to tailoring the SW-CMM into the OSSP. The process elements (see Section 3.1) are the same, and the same subset is repeated here for convenience:

Roles: Describe individuals or collections of individuals participating in process activities. They may be suppliers, customers, agents, reviewers, or verifiers of a practice.

Entry criteria: Describe the conditions under which an activity can start. The entry criteria can be a simple or compound predicate about the state of a work product, role, or activity. "Software requirements shall be baselined under formal configuration management control prior to starting software design activities," is an example of an entry criteria.

Inputs: Describe those items or work products produced by a prior activity and used by the current activity. Software requirements are an input to the software design activities.

Activities: Describe what is being done. They may be directly associated with the production of a product, a management function, or a service provided to help those directly involved to operate more effectively or efficiently.

Outputs/work products: Describe those items that are produced by the process, i.e., items that are the direct result of executing a process step. Software modules, tested code, meeting minutes, and SQA reports are all examples of work products. Work products exist only if the process is executed.

Exit criteria: Describe the conditions under which an activity can be declared complete. The exit criteria can be a simple or compound predicate about the state of a work product, role, or activity. "Software requirements shall be reviewed by software managers and other affected groups," is an example of an exit criteria.

4.4 Tailoring Approach

As stated earlier, the analysis and considerations for tailoring the OSSP into a project's specific process is remarkably similar to tailoring the SW-CMM into a set of requirements for the OSSP. However, in tailoring from the SW-CMM into requirements for the OSSP, we relied heavily on the SPF and the use of already developed checklists. In tailoring from the OSSP into a project's specific process, it is unlikely that the organization has developed an equivalent SPF for its OSSP, although the OSSP may be organized in a manner, such that an equivalent SPF is not necessary. In any event, the process elements of the OSSP should be analyzed and tailored to fit the project's needs, but an alternative method for capturing the analysis needs to be developed.

The approach we recommend (and recommended in the SW-CMM itself) is to capture the range of possibilities in the organization's tailoring guidelines. This limits a project's choices somewhat, but considerably eases the burden in performing the analysis. A shorter analysis would only cover the same three process elements – outputs, activities and roles – but this depends on the needs of the project and the desired level of detail in the project's defined software process.

4.4.1 Developing the Tailoring Guidelines

To minimize the variation in a practice across similar project environments in an organization and reduce the amount of process development required in tailoring, we recommend a controlled form of tailoring. The tailoring is controlled by publishing a set of tailoring guidelines for the OSSP.

We recommend that the tailoring guidelines be developed by initially creating a table that indicates the process elements, the tailorable attributes for each element, the range for each attribute, and the considerations for selecting a particular range. This approach ties together the process elements, tailoring by degree, and the tailoring considerations — all described in earlier sections of this report. An example of such a table is shown in Table 4-1.

Table 4-1: Example of Tailorable Process Elements

Process Element	Process Element Example	Likely Tailorable Attribute	Alternatives	Considerations
Role	Agent Supplier Customer Reviewer Verifier	Performer	Individual	Size (Small = < 5 people, < 6 months, <$200K or <0.5 KSLOC) Structure (IPT, Matrix, Functional) customer = internal, small external
			Team	Size (Medium = 5-15 people, 0.5-1 year, or < $.5M, or 0.5-5 KSLOC) customer = large external, gov't agency
			Group Department	Size (Large = 10-25 people, <1-3 years, < $1M, or 5-20 KSLOC) Size(Mega = >Large)
Activity	Design Code Test Review	Frequency	Once/week Once/month Once/quarter At major milestones	Highly critical, mega or large
	Communicate	Formality	Meeting w/ minutes Informal meeting Memo Email Phone call	Mega or large Large or medium Large, medium or small Medium or small Small
Inputs/ Outputs/ Work product	Core module	Formality	CCB controlled document CM controlled document Date/Version control	
	Document	Scope	Formal standard Suggested template Engineering notes	
	Report	Precision	Documented reviews Informal reviews Self review	

The steps for developing the tailoring guidelines are as follows:

1. Identify the process elements. The SPF contains a list of process elements, but depending on the structure of the OSSP, different groupings or elements may be desired. For the purpose of this exercise, you may want to concentrate on outputs, activities, and roles to start with, and expand to the other elements as needed.
2. Identify the attributes of each process element that are likely to be tailored. Attributes are descriptive clauses associated with process elements, and some of the more common attributes are described in the section on tailoring by degree (Section 4.2.3). Each organization needs to develop a set of attributes that are meaningful to their environment and business goals. Attributes serve to make more explicit the nature of the element used in the practice.
3. Step through each process element. For each attribute of that element, determine the range of possibilities (set of alternatives) for that attribute. For example, in a practice stating that an agent (process element of role) accomplishes a task, the scope of the agent is an attribute. Thus, "the individual," "the team," or "the group/department" might each be appropriate agents. We recommend selecting a finite set of values or alternatives that projects can choose from.
4. For each alternative, determine the primary considerations that would lead or compel a project to select that alternative when performing their process tailoring. The major considerations are listed in Section 4.2, but may include other parameters such as size, complexity, criticality, or environment.

We recommend that the guidance provided be flexible. For example, if the consideration is project size, the guidance might indicate the appropriate alternatives to select for small, medium, large, and mega projects. The flexibility could be built in by including significant overlaps in the size categories. Thus, for a project in the overlap zone between small and medium, the project manager would have the freedom to treat his or her project as at "the top end of small" or at "the bottom end of medium."

The table is only a starting point. It helps to develop a general set of guidelines and to provide consistency across the tailoring effort. However, in order to be useful, the organization's tailoring guidelines need to address the specifics in their OSSP. The guidelines need to address the details of tailoring for specific procedures, work products, and other process elements. For example, an organization may have certain procedures or work products, for which tailoring is not permitted, or is at least strongly discouraged. (The procedures governing the operation of the configuration control board come to mind.) These instances and other restrictions need to be clearly spelled out.

4.4.2 Developing the Project's Specific Process

At the project level, tailoring is performed by using project needs to select an appropriate alternative from the recommended set. Figure 4-1 presents a diagram of this process. Note that the selection mechanism is human judgment.

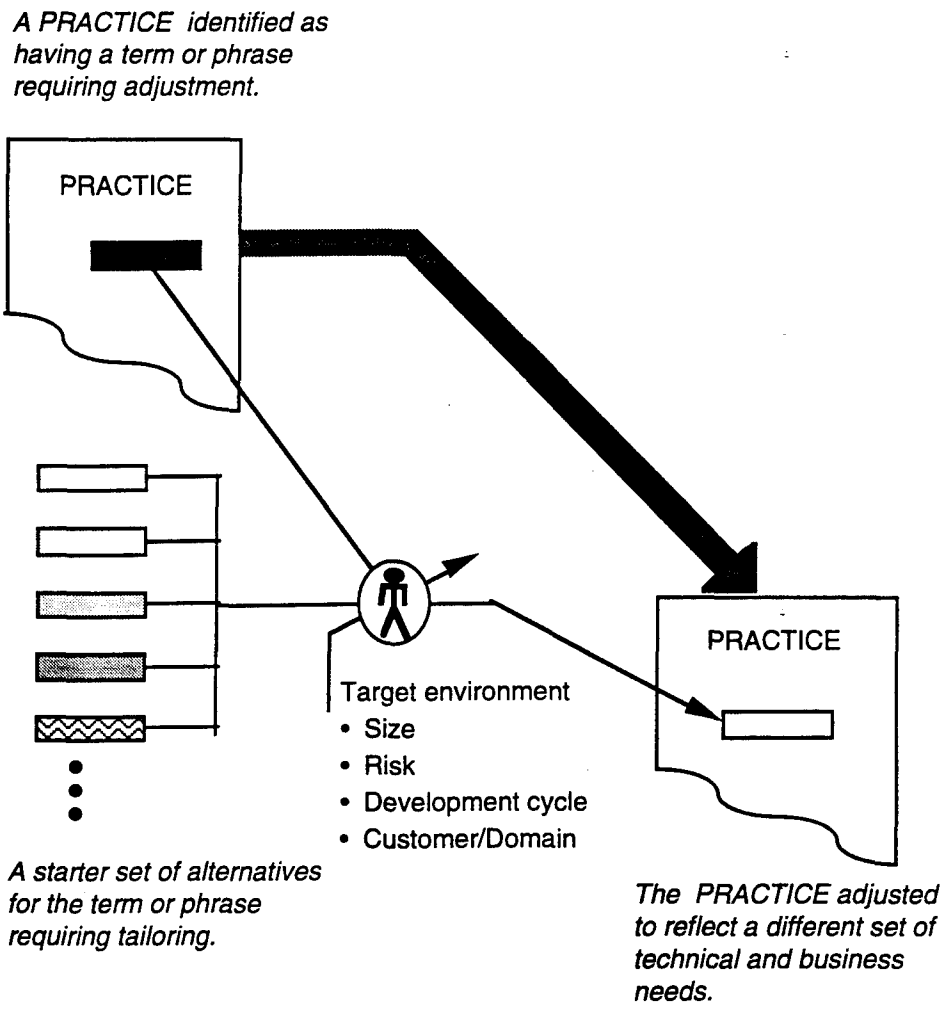

Figure 4-1: Project Tailoring

The project starts with the OSSP. It then uses the guidance in the tailoring guidelines to select appropriate alternatives. The selected alternatives are used to modify the OSSP into a project-specific process. The project-specific process may be captured in a number of ways. Many organizations have checklists embedded in their OSSP. The checklists typically have spaces to record the selected tailoring. Other organizations permit projects to edit a process template, and create their project-specific processes in that manner. Whatever mechanism your organizations uses, we recommend that it be used consistently across all projects. Also, we recommend that the SEPG and/or SQA be involved in the tailoring efforts.

4.5 The Software Development Plan

A project's software development plan is a key element in the management of the project. The schedules contained in the software development plan represent an application of the project's defined software process to the development of one specific set of software products (see Figure 2-2).

The project's defined software process contains the tailored version of the OSSP to be used on the project. It may be similar or identical to other projects' defined software processes in the organization. The specific software configuration to be developed and the schedule for the development are not part of the project's defined software process. These are derived form the project-specific business and technical requirements and combined with the project's defined software process to form a major part of the project's software development plan.

Tailoring is a key element in the organization's ability to develop reasonable and efficient software development plans for use in managing and controlling projects within the organization.

5 Data Sources

The following publications are good sources for data that are useful in tailoring and/or interpretation:

A Software Process Framework for the Capability Maturity Model (CMU/SEI-94-HB-1), 1994; Olson et. al. This SEI Handbook contains a series of checklists to help an organization determine if its software policies, standards, processes, procedures, training, and tools are consistent with the SW-CMM [Olson94].

The Capability Maturity Model: Guidelines for Improving the Software Process, Addison-Wesley, 1995; Paulk et. al. This book integrates and elaborates the description of the SW-CMM and how to interpret its practices. Chapter 4, "Interpreting the SW-CMM," should be particularly useful for those engaged in tailoring work.

A Discipline For Software Engineering, Addison-Wesley, 1995; Humphrey. This book summarizes the costs and benefits of a Personal Software Process (PSP). It scales down industrial software practices to fit the needs of small-scale program development [Humphrey95].

6 Summary

The Software Capability Maturity Model is a publicly owned model that is commonly used to support software process improvement and software capability evaluation. The key practices of the SW-CMM are stated in general terms so they can be used in a wide variety of environments. The general nature of the model and its specific assumptions require that each organization tailor the key practices to fit its own specific environment, business needs, and goals.

A framework for tailoring the SW-CMM and a set of techniques for tailoring the key practices are essential to ensure that the results of tailoring are consistent with the intent of the model. Using well-defined tailoring techniques will also aid the organization in developing its organization's standard software process and tailoring guidelines, so that these documents support the often conflicting goals of achieving project efficiency while maintaining process commonality across projects.

References

[Paulk93a] Paulk, Mark C.; Curtis, Bill; Chrissis, Mary Beth; and Weber, Charles V. *Capability Maturity Model for Software, Version 1.1* (CMU/SEI-93-TR-24, ADA 263403). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, February 1993.

[Paulk93b] Paulk, Mark C.; Weber, Charles V.; Garcia, Suzanne M.; Chrissis, Mary Beth, and Bush, Marilyn W. *Key Practices of the Capability Maturity Model, Version 1.1* (CMU/SEI-93-TR-25, ADA 263432). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, February 1993.

[Olson94] Olson, Timothy G.; Reizer, Neal R.; Over, James W. *A Software Process Framework for the Capability Maturity Model* (CMU/SEI-94-HB-1, ADA 285595). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1994.

[Humphrey95] Humphrey, Watts S. *A Discipline For Software Engineering.* Reading, MA: Addison-Wesley Publishing Company, 1995.

Appendix A. Glossary

Note: these definitions are taken verbatim from the SW-CMM, except as noted by an asterisk.

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

activities performed: (See *common features.*)

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.)

* *activity role:* All the explicit and implied players needed to execute an activity successfully. Typically, the activity roles include supplier, agent, customer, verifier, and reviewer.

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.

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

common features - The subdivision categories of the SW-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 SW-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 in order 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 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]

defined level: (See *maturity level.*)

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.

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.

* *interpret*: The process of analyzing the definitions and/or terms of a general process description and comparing them to an existing instantiation of the description in order to facilitate the understanding of the relationship between the description and the instantiation, e.g., analyzing the key practices of a key process area of the SW-CMM and comparing them to the processes present at a potential software contractor.

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

managed level: (See *maturity level.*)

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

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.

optimizing level: (See *maturity level*.)

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 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.

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

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

* *process definition criteria:* The set of information that must be included in a software process description for it to be usable by the people performing the process. [Olson94]

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 element*: Those portions of a process description that satisfy the process definition criteria. Process definition criteria are typically stated as questions (who, what, when, why). Process elements therefore, include purpose, input, output, role, activity, entry and exit criteria, and procedure to name a few. (See the Software Process Framework [Olson94] .)

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

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.

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.*)

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

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

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 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 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 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 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 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 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.

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 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.)

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.

statement of work: A description of all the work required to complete a project, which is provided by the customer.

system: A collection of components organized to accomplish a specific function or set of functions. [IEEE-STD-610]

tailor: To modify a process, standard, or procedure to better match process or product requirements.

* *tailor by degree*: To modify a process element (inputs, outputs, activities, entry/exit criteria, etc.) by changing an attribute of that element. Attributes may include formality, frequency,

granularity, and scope, to name a few. The change does not alter the intent of the process element.

* *tailoring*: The act of adjusting the definitions and/or particularizing the terms of a general process description to derive a description that is applicable to an alternate (specific) environment, e.g., tailoring the key practices of the SCM key process area for use on a small project or tailoring the key practices of the SW-CMM to produce process requirements for an organization's standard software process.

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.)

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.

user: (See *end user.*)

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]

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.*)

Appendix B. Acronyms

SW-CMM	Software Capability Maturity Model
KP	Key practice
KPA	Key process area
OSSP	Organization's standard software process
PSP	Personal Software Process
SCE	Software capability evaluation
SCM	Software configuration management
SDP	Software development plan
SEI	Software Engineering Institute
SEPG	Software engineering process group
SPA	Software process assessment
SPF	Software Process Framework

Appendix C. Example Codes for Tailoring Tables

Activity Disposition Codes

A	Accept
E	Expand
T	Tailoring recommended
O	Optional
NR	Not recommended

Activity Role Codes

—	An activity role is not referenced in the key practice.
O	The activity role is not performed or not defined by the organization.
N/A	The activity role is not appropriate for the organization.
X	(in the N/A column) The practice does not apply to the organization.

Work Product Codes

C	Complete coverage
S	Shared coverage
P	Partial coverage
–	No coverage

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS None
2a. SECURITY CLASSIFICATION AUTHORITY N/A 2b. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for Public Release Distribution Unlimited
4. PERFORMING ORGANIZATION REPORT NUMBER(S) y practivCMU/SEI-94-TR-024	5. MONITORING ORGANIZATION REPORT NUMBER(S) ESC-TR-024
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 F19628-95-C-0003
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)
Process Tailoring and the Software Capability Maturity Model

12. PERSONAL AUTHOR(S)
Mark Ginsberg

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (year, month, day)	15. PAGE COUNT
Final	FROM TO	November 1995	53

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
interpretation
key practices
software process framework
software process improvement

19. ABSTRACT (continue on reverse if necessary and identify by block number)

The Software Capability Maturity Model (sm) (SW-CMM[sm]) is serving as the foundation for a major portion of the proces improvement being undertaken in the software industry. It is composed of two volumes: the Capability Maturity Model for Software and the Key Practices of the Capability Maturity Model. The key practices of the SW-CMM are expressed in terms that reflect normal practices of organizations that wortk on large, government contracts. There is, however, a significant population of software-producing and -acquiring organizations, operating in different environments. for which the key practices require significant interpretation and/otr tailoring prior to application. This report

(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

presents a tailoring framework that identifies process artifacts, tailoring processes, and their relationships to project artifacts, and explores the nature of various kinds of tailoring used in the definition and development of software process descriptions. Techniques appropriate to each type of tailoring are then discussed. The general approach utilizes and builds upon the Software Process Framework, whose purpose is to provide guidance for designing, analyzing, and reviewing software processes for consistency with the SW-CMM.