Technical Report
CMU/SEI-94-TR-6
ESC-TR-94-006

Software Capabilty Evaluation Version 2.0 Method Description

CMM-Based Appraisal Project

June 1994

Technical Report
CMU/SEI-94-TR-6
ESC-TR-94-006
June 1994

Software Capability Evaluation Version 2.0 Method Description

CMM-Based Appraisal Project

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

This report was prepared for the

SEI Joint Program Office
HQ ESC/AXS
5 Eglin Street
Hanscom AFB, MA 01731-2116

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

FOR THE COMMANDER

(signature on file)

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

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

This document is available through Research Access, Inc., 800 Vinial Street, Pittsburgh, PA 15212.
Phone: 1-800-685-6510. FAX: (412) 321-2994. RAI also maintains a World Wide Web home page. The URL is http://www.rai.com

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

This document is also available through the Defense Technical Information Center (DTIC). DTIC provides access to and transfer of scientific and technical information for DoD personnel, DoD contractors and potential contractors, and other U.S. Government agency personnel and their contractors. To obtain a copy, please contact DTIC directly: Defense Technical Information Center, Attn: FDRA, Cameron Station, Alexandria, VA 22304-6145. Phone: (703) 274-7633.

Contents

List of Figures

List of Tables

Acknowledgments

This document represents the next evolutionary improvement of the SCE method. Most of the development work to bring the SCE Method up to CMM v1.1 was done by Edward Averill, Pat Hanavan, and John Maphis. Extensive help, guidance, and support were provided by Ernie Bauder, Carol Bothwell, and Mark Ginsberg.

This document was written by a team consisting of F. Michael Dedolph, John Maphis, and Bob Lang. Ernie Bauder provided a stream of encouragement, advice, and good counsel.

Paul Byrnes, the SCE Project leader for the past three years, has directed and sponsored the development and evolution of the SCE method and was the driving force behind publication of this document.

This document is based largely on the previous work that led to the *SCE Method Description, Version 1.5*. Thanks are again offered to all of the folks who contributed to developing, writing and reviewing that document. Many reviewers have contributed to the development and revision of this document; their valuable insights added a great deal to the quality of the finished product.

A very special thank you to John Maphis. In a one-year tenure as a resident affiliate from the Navy, he quietly produced more work than anyone could reasonably expect from three ordinary mortals.

Part 1

Section 1 Introduction

Abstract: This report describes Version 2.0 of the Software Capability Evaluation (SCE) Method, as taught at the Software Engineering Institute (SEI) from fourth quarter 1993. This version of the SCE Method is based on the Capability Maturity Model defined in *Capability Maturity Model for Software, Version 1.1* [Paulk 93a]. The document includes an overview of the SCE Method and its evolution, a detailed description of the activities performed during an SCE, and a discussion of the characteristics of the method and their implications for the use of the method. This document provides a new baseline for future evolution of the SCE Method.

This section of the document contains the following subsections:

This report documents Version 2.0 of the Software Capability Evaluation (SCE) Method, as taught to SCE teams from fourth quarter 1993. This document incorporates CMM v1.1 [Paulk 93a] and the key practices of CMM v1.1 [Paulk 93b] into the SCE Method. Before the release of this document the SCE Method was not CMM based.

The report provides a step-by-step explanation of the method and contextual information for understanding its use in government acquisition and other areas. The primary focus of this report is on *what* is done; less attention is given to *how* it is done. (SCE team training provides this how-to information.)

Some of the objectives for the SCE Method are that it should be reliable, repeatable, trainable, and consistent. This document is part of ongoing efforts at the SEI to meet those objectives and to improve the method.

The purposes of this document are

- To publicly document the CMM-based version of the SCE Method.
- To provide a baseline for the future evolution of the SCE Method.
- To provide an in-depth introduction to the method.

Achieving these purposes will clarify misunderstandings about the SCE Method, motivate community “ownership” of the SCE Method, and help improve consistency in how SCEs are conducted.

The report will help software acquisition managers and software development managers to understand the details of the SCE Method. It will also help SCE teams and software engineering process groups gain a deeper insight into the SCE Method. It is assumed that the reader has some knowledge of the SEI’s ➠ **Capability Maturity Model** (CMM) [Paulk 93a] and the associated document, *Key Practices of the Capability Maturity Model, Version 1.1* [Paulk 93b]. It is also assumed the reader has some knowledge of system or software acquisition practices within the government, and experience with software development or acquisition.

This report has two main sections and several appendices. Section 1 provides background information, a high level description of the major SCE activities, a conceptual model for CMM-based SCE data collection, and information about the evolution of the SCE Method. This section provides background for Section 2, but can also be used as a stand-alone description of the method for management or other personnel who don’t need to know the details of the SCE Method. Section 2 is the bulk of the document, and describes in detail what is done in each step of the SCE Method. The appendices provide additional detail to supplement the text.

1.1 Background and Context

➠ **Software Capability Evaluation (SCE)**[1] is a method for evaluating the software process of an organization to gain insight into its software development capability. This insight can be a valuable input to process improvement activities.

Hence, the SCE Method helps evaluate the ➠ **software process capability** of a software ➠ **development organization** (an organization that develops and/or maintains software products). Software process capability refers to the range of expected results that can be achieved by following a process.

The processes evaluated by SCE include decision-making processes (such as project management), communication processes (such as design reviews and peer reviews), and technical support processes (such as integration and test)—but not technical production processes (such as processes required by a particular design methodology). The SCE Method does *not* evaluate technical production processes such as requirements analysis, specification, and design, but instead focuses on the management of the technical production processes and on other key processes, as shown in Figure 1-1.

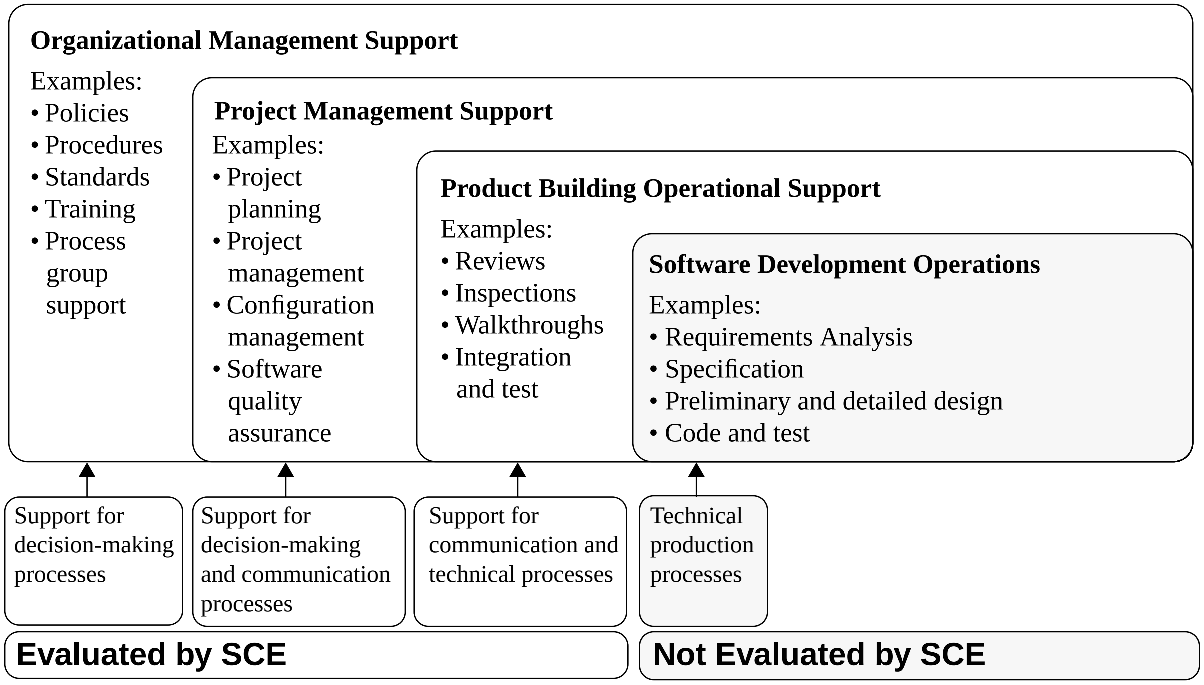

Organizational Management Support
Examples:
• Policies
• Procedures
• Standards
• Training
• Process group support
Project Management Support
Examples:
• Project planning
• Project management
• Configuration management
• Software quality assurance
Product Building Operational Support
Examples:
• Reviews
• Inspections
• Walkthroughs
• Integration and test
Software Development Operations
Examples:
• Requirements Analysis
• Specification
• Preliminary and detailed design
• Code and test
Support for decision-making processes
Support for decision-making and communication processes
Support for communication and technical processes
Technical production processes
Evaluated by SCE
Not Evaluated by SCE

Figure 1-1: Processes Evaluated by SCE

[1] An arrow (➠) preceding a term in **boldface** type indicates that the term is defined in the glossary on page 197.

The SEI's software process principles are derived from the works of Deming, Juran, and others [Deming 86], [Juran 88], [Juran 89], [Crosby 79], who promoted the idea that close attention to the processes used to create products leads to improved product quality—i.e., the product will fully satisfy the customer's requirements and will be produced within existing constraints such as cost and schedule. There are many examples of this principle and its successful application in the manufacturing domain, but the principle can be applied anywhere management and communication processes play an important role in the success of an organization's mission.

The SEI's Capability Maturity Model (CMM) applies this principle to the software development arena. The CMM defines several ➠ **key process areas** (KPAs); each KPA "identifies a cluster of related activities that, when performed collectively, achieve a set of goals considered important for enhancing process capability."[1] Each KPA contributes to the environment in which development organizations create software products. Within the CMM, the KPAs are organized into five basic levels of process maturity to describe the progression from an ad hoc software process to one that is well defined and can act as a stable foundation for continuous process improvement.

By evaluating the development organization's software projects against the KPAs in the CMM, the SCE team determines whether the development organization follows a stable, predictable software process. Although mature processes do not guarantee a successful product, the likelihood of success should increase as the software processes mature toward the Optimizing level. In other words, mature processes reduce the risk associated with the planned development.

[1] Mark Paulk, et al. *Capability Maturity Model for Software, Version 1.1* [Paulk 93a], page A-10.

1.2 Overview of the SCE Method

As mentioned before, SCE is a method for evaluating the software process of an organization to gain insight into its software development capability. An SCE probes into the development organization’s process implementation to establish the ➠ **strengths** and ➠ **weaknesses** in the processes used to support development of software products; thus, SCE helps provide insight into the development organization’s software process capability.

To evaluate software process capability, a team of four to six trained and experienced people from the sponsoring organization use the SCE Method to sample and analyze information about the development organization’s implementation of the software processes. There are two major ways information is collected: ➠ **interviewing** and ➠ **document review**. The choice of information to be sampled is determined by the ➠ **Target Process Capability**—that is, the process capability that is most appropriate for the planned development. The Target Process Capability consists of a set of KPAs that will be evaluated, and establishes the boundaries of the evaluation.

SCE is a model-based method that provides a structure for collecting information at varying levels of detail. A brief overview of the structure is included in the discussion below. A more detailed discussion is provided in Section 1.3 on page 20.

Although the boundaries of the SCE are determined by the KPAs in the Target Process Capability, the evaluation is done at a more detailed level. Within the Target Process Capability KPAs ➠ **subprocess areas** are examined. A subprocess area is a set of activities in an implemented process that, acting together, helps an organization to achieve one of the goals of a KPA. There are two or more goals for each KPA; each goal describes something that should be achieved by implementing the KPA. Subprocess areas are mapped directly to KPA goals: each goal represents a desired state, while each subprocess area describes the activities needed to achieve that state. The table below uses the Software Project Planning KPA to show the relationship between KPAs, goals, and subprocess areas.

KPA	KPA goal	Subprocess area
Software Project Planning	1. Software estimates are documented for use in planning and tracking the software project.	Develop estimates
	2. Software project activities and commitments are planned and documented.	Plan software activities
	3. Affected groups and individuals agree to their commitments related to the software project.	Make commitments

Table 1-1: Relationship Between KPAs, Goals, and Subprocess Areas

Subprocess areas represent a finer level of detail than KPAs. However, in order to conduct an SCE investigation—that is, in order to determine what documents to review, whom to interview, and what kinds of questions to ask—teams need yet a further level of detail.

The level of detail at which an SCE is conducted is the ➠ **topic**. A topic defines a subject that will be probed during an SCE investigation. A rule of thumb for a topic is that it can be transformed into an open-ended question that can be readily answered by a person or document. For example, within the *develop estimates* subprocess area, the team might want to investigate how estimates are derived. Thus, they would ask the question, “What are the procedures used to develop software size estimates?” (This is a highly simplified version of topic selection. Step 9 Develop Topic Lists on page 68 describes the process in more detail.)

The analysis and summary of the information collected on an SCE become the ➠ **findings** of the team. Findings document the software process strengths, weaknesses, and observed improvement activities in the KPAs evaluated by the team. An ➠ **improvement activity** is a process improvement that is not yet institutionalized—for example, a pilot of a new process put in place to address a weakness identified by the organization.

Findings are used to determine risk from the implemented processes relative to the planned development effort. How the findings are used represents the ➠ **results** of the evaluation.

The findings generated during an SCE are primarily used to determine risk for a particular development, although the findings could also be used to pinpoint specific areas for software improvement activities. This is a subtle but important difference between the SCE Method and other appraisal methods such as Software Process Assessment (SPA). During a Software Process Assessment, one of the main objectives is to get organizational buy-in and support for organization-wide improvement efforts. This is *not* an objective for an SCE, although the findings from an SCE may be factored into an organization’s process improvement plan. Also, the results of a Software Process Assessment are not normally used to determine risk for a particular development effort, as they are in SCE.

The process of conducting an SCE is independent of the way the findings are used. Specifically, conducting an SCE leads to a set of findings based on strengths, weaknesses, and improvement activities observed during interviewing and document review. The findings are independent of how they are used. There are two primary ways that the SCE Method has been used.

1. ➠ **Source selection**. This was the original reason SCE was created and has been the major use of the method. In source selection, the results of the SCE are used to characterize the software process-related risk of awarding a contract to an offeror.[1] SCE is only one criterion among many used to select software contractors in government acquisitions.

2. ➠ **Contract monitoring**. SCE has been used in the monitoring of an acquisition after contract award by serving as an input for an incentive/award fee decision. SCE has also been used to help the sponsoring organization tailor their contract monitoring efforts by allowing them to prioritize their efforts based on the observed strengths and weaknesses of the development organization's processes. Both of these uses are new but show great promise for the future, because they focus on a long-term relationship with a development organization and encourage the development organization to invest in software process improvement.

For example, suppose that the Software Configuration Management (SCM) KPA was investigated during an SCE, and that the following observations were made about the processes in use at a particular development organization site:

- The investigation revealed well documented procedures for the SCM change control process.
- The investigation noted that no training was available for software development personnel in the change control procedures.
- The investigation revealed an automated library system in use (but only on one project) that supported and enforced the procedures.
- The investigation revealed that there was a plan in place for implementing the library system across all of the projects.

The findings for this KPA might be that there was a strength (the well-documented procedures), a weakness (the lack of available training), and an improvement activity (the automated library system and the plan for implementing it across the organization).

The results would then depend on the ➠ **use of the SCE Method**. The findings belong to the sponsoring organization and could be used in many different ways—that is, the results could be different. For example, in a source selection, the findings might be factored into a risk determination. The development organization might be given a "moderate" risk rating for software Configuration Management based on the findings, and assigned a color code of "Yellow" for

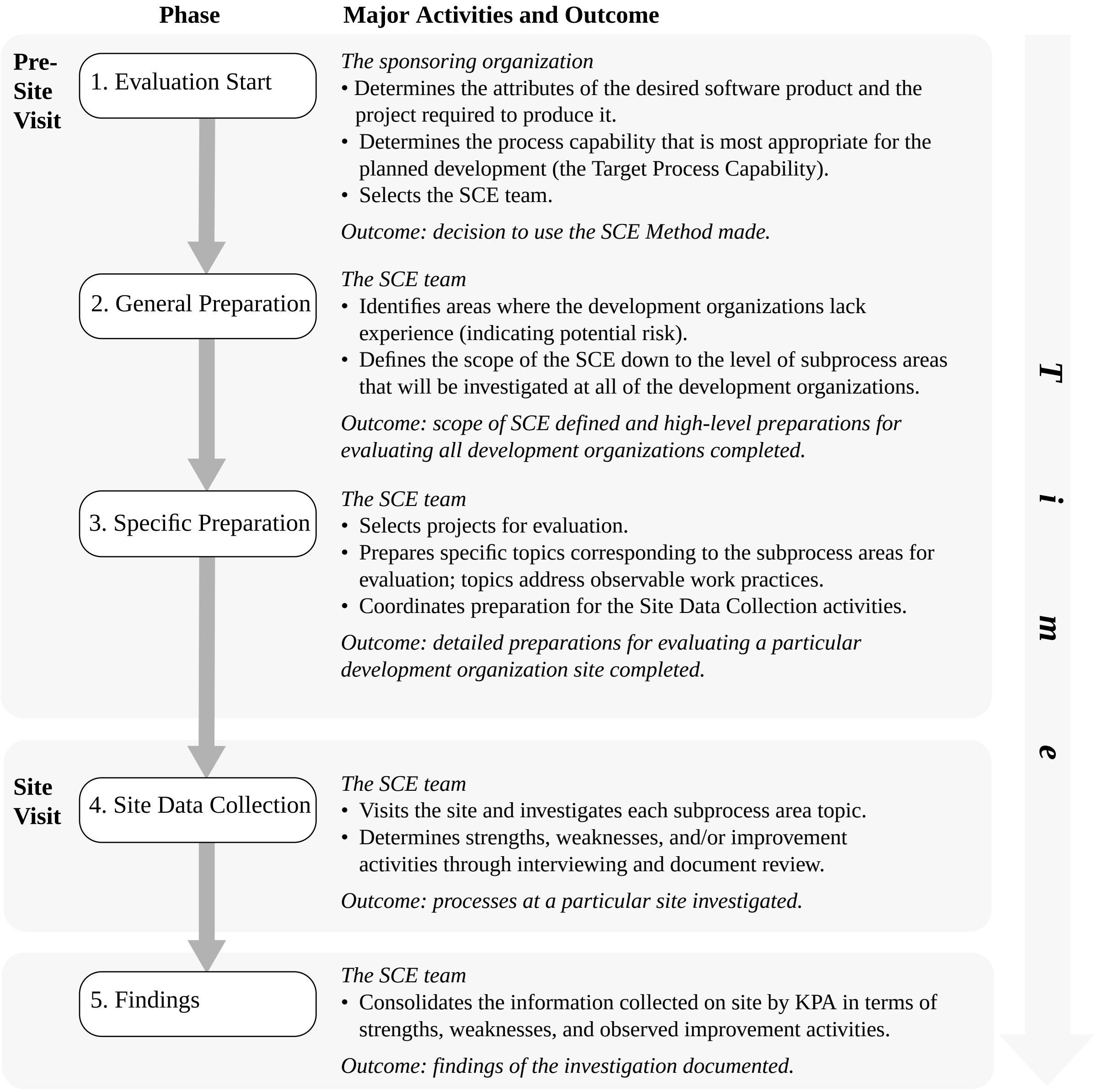
[1] Because SCE has been used extensively in source selection, in the SCE team training handouts and case study materials the terms *offeror* and *contractor* are often used to denote the development organization. The development organization is the recipient of the SCE. Similarly, in the training materials the term ➠**acquisition agency** is often used to denote the ➠**sponsoring organization**, which is the organization conducting the SCE. This document uses the terms *development organization* and *sponsoring organization* almost exclusively in anticipation of wider use of the method in other contexts.

this category. (The yellow color code might be defined as, “fails to meet evaluation standards or has low probability of meeting the requirement; or has significant but correctable deficiencies.”)[1]

The individual risk ratings for all the KPAs evaluated during an SCE would result in a composite SCE risk rating. This factor would be considered along with many others (such as cost, technical evaluations, prior performance, etc.) when awarding the contract. On the other hand, in a contract monitoring situation, the same findings might lead the sponsoring organization to insist that the automated library system be implemented on their development project, and some portion of an award fee might be tied to successful implementation of a training program in the procedures for SCM change control.

[1] This is one example of how the findings might be scored and consolidated with the results of other source selection activities. Many other scoring mechanisms have been used in source selection, and the relative weight of each factor is unique to the particular source selection.

The SCE Method consists of five major activity phases: Phase 1, Evaluation Start; Phase 2, General Preparation; Phase 3, Specific Preparation; Phase 4, Site Data Collection (or Site Visit); and Phase 5, Findings. The remainder of this overview briefly explains each phase. The phases and major activities within each SCE phase are shown in Figure 1-2.



The sponsoring organization

- Determines the attributes of the desired software product and the project required to produce it.
- Determines the process capability that is most appropriate for the planned development (the Target Process Capability).
- Selects the SCE team.

Outcome: decision to use the SCE Method made.

The SCE team

- Identifies areas where the development organizations lack experience (indicating potential risk).
- Defines the scope of the SCE down to the level of subprocess areas that will be investigated at all of the development organizations.

Outcome: scope of SCE defined and high-level preparations for evaluating all development organizations completed.

The SCE team

- Selects projects for evaluation.
- Prepares specific topics corresponding to the subprocess areas for evaluation; topics address observable work practices.
- Coordinates preparation for the Site Data Collection activities.

Outcome: detailed preparations for evaluating a particular development organization site completed.

The SCE team

- Visits the site and investigates each subprocess area topic.
- Determines strengths, weaknesses, and/or improvement activities through interviewing and document review.

Outcome: processes at a particular site investigated.

The SCE team

- Consolidates the information collected on site by KPA in terms of strengths, weaknesses, and observed improvement activities.

Outcome: findings of the investigation documented.

Figure 1-2: Phases and Major Activities of an SCE

Phase 1: Evaluation Start

In this phase, the sponsoring organization decides to use the SCE Method, and begins preparing to conduct the SCE. This phase is performed by the sponsoring organization. In all of the remaining phases the activities are conducted by the SCE team.

The purposes of the Evaluation Start phase are to determine the role of SCE, to determine the ➠ **attributes** of the desired software product and the project required to produce it, to determine the process capability that is most appropriate for the planned development, and to select the SCE team.

The SCE Method is performed within the context of a “larger” process such as source selection or contract monitoring. The Evaluation Start phase is when the relationships are established between the SCE Method and the process that uses the SCE findings. The first steps toward use of the SCE Method begin sometime during the preliminary planning for product development, when the role of the SCE is determined.

To determine the role of SCE, the sponsoring organization considers how SCE can be used in conjunction with other technical and managerial activities to identify and mitigate risks associated with the planned product development. The focus of the SCE is on risks associated with software process capability. Risks not associated with software process capability need to be addressed by other methods.

With this in mind, the sponsoring organization should define how SCE results will be used and should determine the resources required to perform the SCEs. At some point, the sponsoring organization decides that the potential risks to the planned development from software process capability warrant using the SCE Method, and the decision to use the SCE Method is made.

Planning for the SCE starts with the decision that the SCE Method should be used. During this phase, planning for the SCE should consider

- Funding for personnel, training, and travel.
- Coordinating SCE ➠ **site visits** and requests for information with the development organization(s).
- Scheduling time for the SCE activities within the context of the use of the method (e.g., source selection or contract monitoring).

As a result of this planning, the sponsoring organization commits resources to conducting the SCE.

Determining the role of SCE and planning for the use of the method may not be done by the SCE team, but these activities are critical to the successful use of the SCE Method. For example, to use SCE as part of the technical proposal evaluation in source selection,[1] the Source Selection Authority (SSA) would have to: (1) determine the relative weight of SCE results compared to results of other technical evaluations, (2) insert the requirements for conducting the SCEs into the ➠ **Request For Proposal** (RFP), and (3) allow enough time in the source selection schedule to train the teams and perform the evaluations.

Once the decision to use SCE is made, the sponsoring organization determines the software process capabilities needed to minimize the risk coming from the processes likely to be used for the planned development. This is accomplished by analyzing the attributes of the desired software product and then determining the process capability that is most appropriate for the planned development. The processes examined by an SCE always fall within the KPAs of the CMM. The Target Process Capability establishes the boundaries of the SCE investigation—a KPA is evaluated if and only if it is part of the Target Process Capability.

The sponsoring organization defines the Target Process Capability by considering the desired software product and determining the software process capabilities required to build it. In other words, the preliminary analysis of the product attributes puts bounds on the processes the SCE will examine.

The sponsoring organization also selects the SCE team. A team consists of four to six experienced people from the sponsoring organization who have completed SCE team training currently available at the SEI. The same team is used to conduct all of the evaluations for a particular use of the SCE Method.

The people involved with the decision to use SCE are senior software project managers or acquisition managers and staff with software engineering experience. Senior management or acquisition management should select the SCE team and assign the personnel resources, and should assess the potential impact on schedule. Staff with software engineering experience should establish the Target Process Capability for the SCE.

When the Evaluation Start phase is complete the decision to use the SCE Method is made, the role of SCE is established, and resources have been committed to the effort by the sponsoring organization. In addition, analysis of the attributes of the desired software product and the project required to produce it are complete, the process capability desired for the planned development is established as the Target Process Capability, and the SCE team has been selected and trained.

[1] Guidance for most acquisition applications/implementations can be found in the *SCE Version 2.0 Implementation Guide* [SCE 94].

The SCE team will be responsible for all of the subsequent work in the following phases.

Phase 2: General Preparation

The General Preparation phase consists of site visit preparation activities that pertain to all of the development organizations equally.

In this phase, the SCE team completes high level preparations for evaluating all of the development organizations that are involved with a particular use of the method; these development organizations are collectively referred to as the ➠ **development organization community**.

The purpose of the General Preparation phase is to define the scope of the investigation for all of the development organizations. The ➠ **scope** of the SCE consists of subprocess areas within the KPAs that make up the Target Process Capability, and that will be used to evaluate all development organizations.

To achieve this purpose, the SCE team identifies those software processes that contribute most to the potential development risk throughout the development organization community. To do this, the team examines information[1] from each development organization about their view of the product to be built and information about the software projects they are submitting as candidates for evaluation.[2]

The attributes of the product to be built are compared to the attributes of products developed by the projects that have been submitted as candidates for evaluation. These comparisons identify areas in which the development organization may lack experience, indicating potential risk.

The experience shortfalls of the individual development organizations are then consolidated for the development organization community. The experience shortfalls indicate areas that may have higher risk and should be investigated.

Based on the experience shortfalls in the development organization community (and other factors described in Phase 2: General Preparation, Section 2.3 on page 49), the SCE team selects one or more subprocess areas within each of the Target Process Capability KPAs for evaluation. These subprocess areas

[1] Other information is collected at the same time, including organization charts and information about the CMM related software processes in use. Information about the processes in use is collected using a CMM-based maturity questionnaire (available in SCE team training). This information is used in Phase 3, Specific Preparation. In source selection, the information is normally requested as part of the RFP.

[2] The projects are submitted by the development organization based on instructions provided by the sponsoring organization. In source selection, the requirements for selecting and submitting projects as candidates for evaluation are usually contained in the RFP.

are called ➠**critical subprocess areas**, and will be investigated at all development organization sites. Collectively, they make up the Critical Subprocess Area List and define the scope of the SCE.

The critical subprocess areas define the scope of the SCE. In Evaluation Start, the product was used to establish the boundaries of the investigation in terms of KPAs; in General Preparation the collective experience of the development organization community is used to define and tailor the scope of the SCE down to the subprocess area level. This tailoring is necessary because of site visit time limitations. During a site visit, it is not possible to investigate all of the subprocess areas within the KPAs in the Target Process Capability, so a sample is used.

The activities in the General Preparation phase establish the context for the Specific Preparation phase (Phase 3). General Preparation as described here applies primarily to use of the SCE Method in a source selection, where multiple development organizations are evaluated using the same critical subprocess areas. In contract monitoring, the same steps should be followed for the initial evaluation. Subsequent evaluations of the same organization would be tailored to reflect the special needs of the contract to be monitored and the weaknesses observed during the first evaluation.

When the General Preparation phase is complete, the SCE team will have identified areas where the development organization community lacks experience, and will have determined the critical subprocess areas that will be investigated for each development organization.

Phase 3: Specific Preparation

The Specific Preparation phase extends and refines General Preparation phase activities to a particular development organization.

In the Specific Preparation phase, the SCE team completes detailed preparations for evaluating a particular development organization site. The activities in the Specific Preparation phase are repeated for each development organization being evaluated.

The purpose of the Specific Preparation phase is to prepare the SCE team for a specific site visit. To prepare for the visit, the SCE team selects projects to evaluate and selects detailed topics for investigation.

The critical subprocess areas selected during the General Preparation phase are investigated for each development organization; however, subprocess areas are too broad to be probed directly. Topics address observable work practices and are used to probe the process implementation that corresponds to the critical subprocess areas. A ➠ **topic** defines a specific subject that will be probed during the investigation. For example, a topic might be, “investigate whether the organization has standard procedures for the software configuration management change control process.”

Topics are developed by considering ➠ **features**; features are implementation characteristics that are common to every subprocess area. The features used in the SCE Method are derived from the ➠**common features** of CMM v1.1 [Paulk 93a] and are defined in Appendix A.5 on page 142. For example, every process should have corresponding training and should also have documented plans and procedures; “training” and “plans and procedures” are two of the features that can be used to develop topics for investigation.

After selecting evaluation topics, the team plans an interview strategy and identifies documentation for the preliminary document review. The team then works closely with the development organization’s site visit coordinator to coordinate interview schedules, request documentation for review, and to arrange for the facilities the team will require during the site visit.

When the Specific Preparation phase is finished, the SCE team will be ready to perform the activities in the Site Data Collection phase. The team will have determined what topics will be investigated (and to what level), whom they need to talk to, what questions they need to ask during exploratory interviews, and which documents they will review first. The development organization will have prepared the facility for the team, will have the requested documentation on hand, and will have ensured that the interviewees are available.

Thorough preparation is essential, because the amount of information to be considered during the brief Site Data Collection phase will overwhelm the SCE team members if they are not sufficiently prepared.

Phase 4: Site Data Collection (Site Visit)

The Site Data Collection phase is the crux of the SCE Method. During the Site Data Collection Phase, the SCE team investigates the processes at a particular development organization site.

The purpose of Site Data Collection is to investigate the topics associated with each critical subprocess area in enough depth to determine the strengths, weaknesses and improvement activities for the corresponding subprocess area. Although the purpose is simple, this is the most complicated activity during an SCE, and puts the team in direct contact with many of the development organization’s personnel.

To successfully complete the investigation, the team needs to have a good working relationship with the development organization’s site visit coordinator. This relationship builds on the previous contacts with the site visit coordinator made during the preparation activities. The team should also maintain high standards of professional conduct; this helps to establish their credibility and to increase the level of cooperation they receive from development organization personnel.

After setting expectations for the site visit with an entry briefing, the team starts the data collection activities. Site data collection has two basic components: investigation of the topics and decision making about the information collected. These components are applied iteratively until a decision has been made about each topic under investigation; this is summarized in Figure 1-3 below.

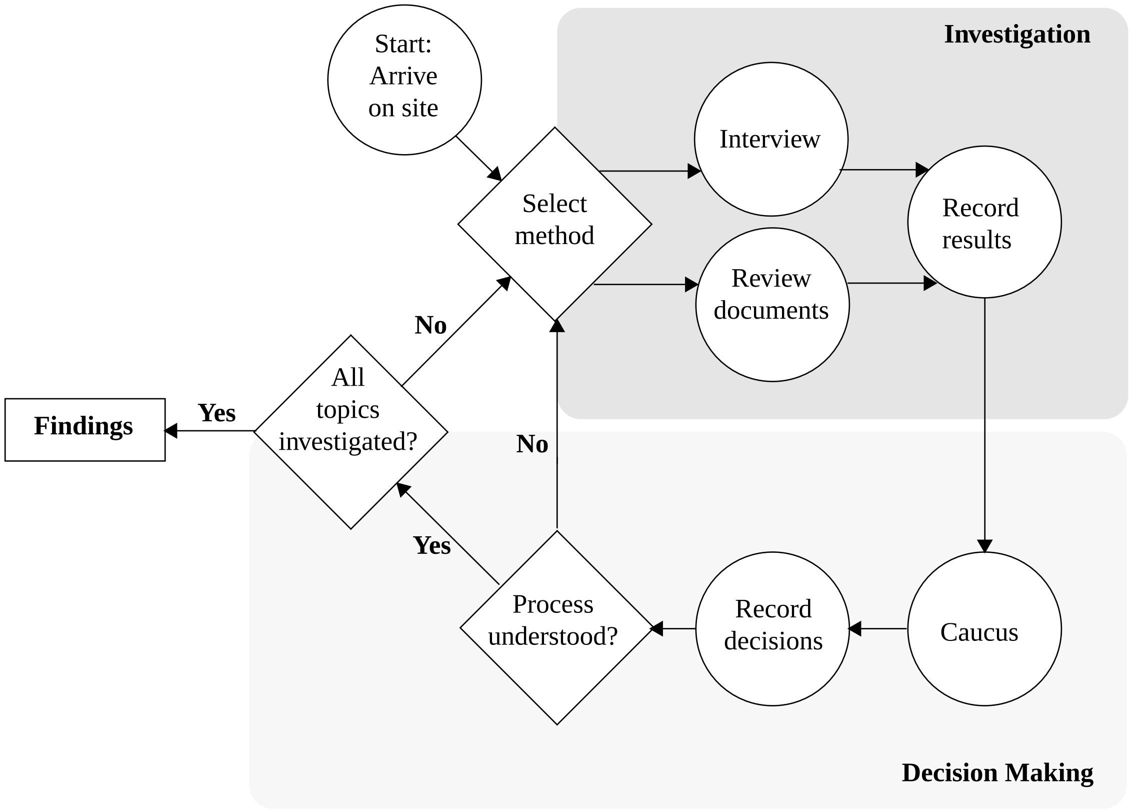


Figure 1-3: A Flow Chart of the Site Data Collection Activities

The SCE team uses two complementary mechanisms to investigate a topic: document review and interviews.

Documents can be used to define and standardize processes, indicate commitment to use the processes, provide an audit trail of processes that were used, and collect data about process performance. Reviewing documents can provide objective evidence of the processes used. A fundamental assumption of the SCE Method is that if a process is not documented, there is no guarantee that it will be followed.

Interviews give insight into how the processes are implemented in practice and show the extent that processes are internalized and understood by the development organization staff. There are two types of interviews used during an SCE: *exploratory interviews* and *consolidation interviews*. During exploratory interviews the questions and answers reveal the actual processes practiced and guide the team to the supporting documentation. Consolidation interviews focus on corroboration and clarification of evidence.

The team members record the results of the investigations into each topic for use in decision making.

Decision making is done by consensus in an ongoing team ➠ **caucus**. In caucus, the team asks the question “Do we have enough information to reach a consensus about this topic yet?” The team must agree that there are at least two pieces of evidence supporting the decision. If the evidence is not conclusive, a new round of interviewing and/or document review is planned and initiated. Decisions resulting in a determination that there is a strength, weakness, or improvement activity associated with one of the topics under investigation are recorded for use in the Findings phase.

When the Site Data Collection phase is finished, the SCE team members are ready to generate their consolidated findings. The information recorded during Site Data Collection is the support for the findings.

Phase 5: Findings

The Findings phase completes the SCE. During the Findings phase, the SCE team documents the results of the investigation.

The findings are actually generated during the site visit, although the final report of the findings may be done later. The Findings phase is treated separately to clearly indicate the end of the SCE activity and to separate the SCE Method activities from the use of the findings in a source selection or contract monitoring context.

The purpose of the Findings phase is to consolidate the decisions made during the Site Data Collection phase. This purpose is accomplished by “rolling up” the decisions that were made about specific topics and subprocess areas into findings at the KPA level.

Findings are expressed in terms of the strengths, weaknesses, and improvement activities that were observed by the team. Ideally, the SCE team presents the findings to the development organization during an exit briefing.[1] Because of the importance of the SCE findings to process improvement, efforts should be made to provide feedback in a timely manner.

When the Findings phase is complete, the detailed decisions made during the Site Data Collection phase about the subprocess area topics will be consolidated and summarized by KPA. A formal final report will be generated for the sponsoring organization to use; how the findings report is used depends on the context.

[1] In some cases the source selection authority may not allow the findings to be presented to the development organization, or may specify that findings be presented after contract award.

1.3 CMM-Based SCE Data Collection Model

The current SCE Method (version 2.0) uses CMM v1.1 [Paulk 93a] as the basis for investigating and making judgements about a development organization's software processes.[1] In addition to CMM v1.1, the SCE Method uses the associated key practices found in *Key Practices of the Capability Maturity Model, Version 1.1* [Paulk 93b].

Collectively, these materials provide a robust structure for collecting information; this structure is diagrammed at a high level in Figure 1-4 below. As indicated, the structure is not strictly hierarchical; subprocess areas include features and key practices, while features and practices may be associated with more than one subprocess area. This section summarizes the structural components of the data collection model; more information is given in Appendix A on page 129.

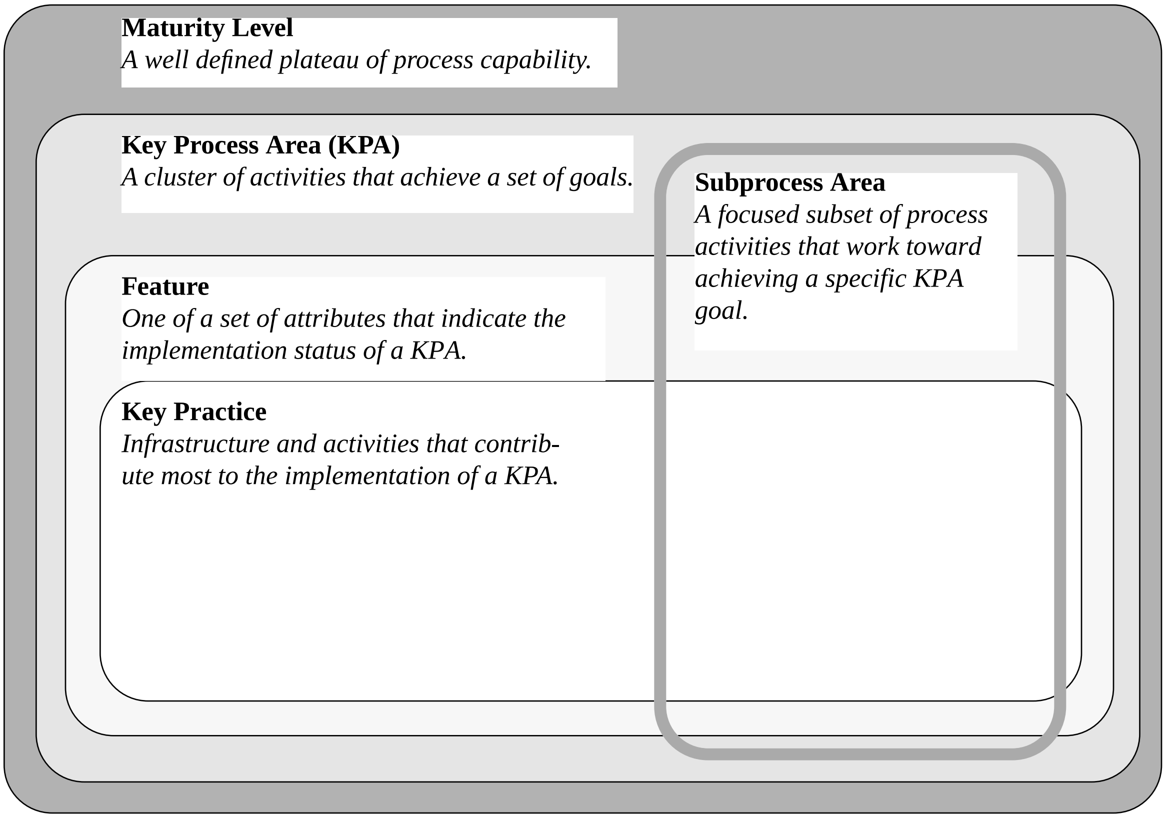

Maturity Level
A well defined plateau of process capability.
Key Process Area (KPA)
A cluster of activities that achieve a set of goals.
Subprocess Area
A focused subset of process activities that work toward achieving a specific KPA goal.
Feature
One of a set of attributes that indicate the implementation status of a KPA.
Key Practice
Infrastructure and activities that contribute most to the implementation of a KPA.

Figure 1-4: CMM–Based Data Collection Model

[1] SCE v1.5 used an older maturity model. A mapping of that maturity model to CMM v1.1 is shown in Appendix B on page 145.

Maturity Levels. CMM v1.1 provides a framework for organizing the evolutionary steps of process improvement into five maturity levels. A ➠**maturity level** is a “well-defined evolutionary plateau” toward achieving a mature software process. Each maturity level provides a layer in the foundation for continuous process improvement. Maturity levels are listed in Appendix A.1 on page 130.

Key process areas (KPAs). Except for the initial level, each maturity level is decomposed into several KPAs. Each KPA identifies a cluster of related activities that, when performed collectively, achieve a set of goals considered important for enhancing process capability.

The path to achieving the goals of a KPA may differ across projects based on differences in application domains or environments. Nevertheless, all of the goals of a KPA must be achieved for the organization to satisfy that KPA. When the goals of a KPA are accomplished on a continuing basis across projects, the organization can be said to have institutionalized the process capability characterized by the KPA. The KPAs are listed in Appendix A.2 on page 131.

Goals. The CMM defines a set of goals for each KPA. A goal describes in a general way what should be achieved by implementing a KPA.The goals can be used to determine whether an organization or project has effectively implemented the KPA. The goals signify the scope, boundaries, and intent of each KPA. When evaluating a specific implementation of a KPA, the goals can be used to determine if the implementation satisfies the intent of the KPA. The KPA goals are listed in Appendix A.3 on page 132.

Subprocess areas: The goal statements in the CMM represent desired states that an organization should try to achieve in its process. However, what the team observes are the activities that are performed to achieve those states. The SCE Method uses subprocess areas to help teams identify these activities. There is a one-to-one mapping of subprocess areas to goals. For example, one of the goals of the Software Project Planning KPA is “software estimates are documented for use in planning and tracking the software project” (a state). The subprocess area that corresponds to that goal is “develop estimates” (an activity). The subprocess areas are listed in Appendix A.4 on page 135.

Features. Within CMM v1.1, the KPAs are organized by a set of common features. A common feature is “an attribute that indicates whether the implementation and institutionalization of a key practice is effective, repeatable, and lasting” [Paulk 93b]. The common features represent the necessary attributes of any process.

An expanded list of common features was developed for use in the SCE Method. The expanded list, referred to simply as *features*, is derived from the definitions and examples of the common features in the CMM. A feature is one of a set of attributes that provides a view of “whether the implementation and institutionalization of a key practice are effective, repeatable, and lasting” [Paulk 93b]. The features are more appropriate for defining a single topic of investigation than the common features. The features used in SCE are listed in Appendix A.5 on page 142.

➠ **Key practices** are the infrastructure and activities that contribute most to the effective implementation and institutionalization of a key process area [Paulk 93b].

The key practices serve as examples of “what” is to be done, but they should not be interpreted as mandating “how” the goals should be achieved. Alternative practices may accomplish the goals of the KPA. The key practices should be interpreted to judge whether the goals of the KPA are achieved.

In SCE data collection, teams don’t look for key practices directly; however, the key practices do serve as examples of things that a team might see, particularly in the *activities performed* key practice. The guidance in the look for tables includes cross-references to applicable key practices.

Generally, the key practices are at a level too low to be used directly in an SCE evaluation. There are more key practices than a team can investigate thoroughly. Also, if SCE teams were to work at that level of detail, the teams might tend to look for specific implementations of a practice rather than investigating the existing practices. The key practices are useful for adding context to the goals, and give SCE teams clues about what to look for during data collection. The key practices are described in *Key Practices of the Capability Maturity Model, Version 1.1* [Paulk 93b]. Guidance about the key practices is embedded in the “look-for” tables used in SCE. (Look-for tables can be found in the *SCE Team Member’s Guide*, which is currently available only in the SCE team training class.)

Example. The following example uses the Software Project Planning KPA to illustrate the relationships among the concepts described above.

When it has implemented the *Software Project Planning KPA*, an organization will be able to establish reasonable plans for performing software engineering and for managing the software project.

Within this KPA, one of the things the organization hopes to accomplish is to make sure that *affected groups and individuals agree to their commitments related to the software project (a goal).*

The SCE Method uses the term *make commitments (a subprocess area)* to describe the activities performed to achieve this goal.

An example of one of these activities is ensuring that *software project commitments made to individuals and groups external to the organization are reviewed with senior management according to a documented procedure (a key practice*).

This key practice is categorized as one of the *Activities Performed (a feature)* within the Software Project Planning KPA.

Summary of the data collection model. There are too many potential topics for a team to investigate thoroughly, so a large part of the General and Specific Preparation phases are spent systematically narrowing the sample space of topics in a manner that is fair to all development organizations.

First, the KPAs are selected, then the critical subprocess areas. After selecting critical subprocess areas, the team uses the features to develop topics for investigation. Subprocess areas represent the activities performed. Features are the necessary components of a process. The combination of a subprocess area and a feature make up a single topic of investigation.

1.4 Evolution of the SCE Method

The version of the SCE Method documented here is based on CMM v1.1 [Paulk 93a]. This is the third published version of the SCE Method.

The original version of the method is described in *A Method for Assessing the Software Engineering Capability of Contractors* [Humphrey 87b]. The original SCE Method was developed to support source selection in major government software acquisitions. While the major activities of interviewing and document review remained the same, other aspects of the SCE Method evolved significantly as a result of feedback from users of the method, observing the effect of SCEs on industry, and the evolution of the CMM. This led to public baselining of the SCE Method in the *Software Capability Evaluation Version 1.5 Method Description* [SCE 93], and to the changes contained in this document.

The major changes in the method to date are the following:

- Elimination of maturity level scores.
- Shift from a “question-based” to a “model-based” method.
- Refinement of the KPAs to include subprocess areas.
- Focusing SCEs based on risk for a specific development.
- Decomposition of the method into discrete phase and steps.
- Public baselining of the SCE Method through publishing the Software Capability Evaluation Version 1.5 Method Description [SCE 93].
- Updating the SCE Method baseline by incorporating CMM v1.1 into the method and publishing this document, the *Software Capability Evaluation Version 2.0 Method Description*.

Each of the major changes is described below, along with a brief rationale for the change. It is important to note that before publication of *Software Capability Evaluation Version 1.5 Method Description* [SCE 93], these changes did not occur in a strict sequence; often the changes happened concurrently.

Elimination of maturity level scores

The SCE Method no longer calculates a maturity level “score”—that is, development organizations are not rated as a “Level 1” or “Level 2” organization during an SCE.

Maturity level scores can be useful to describe goals, or for process improvement efforts, especially when a development organization is initiating process improvement efforts. However, feedback from SCE teams indicated a need for more specific information about the underlying process capabilities of the development organizations. This type of information was needed because the sponsoring organizations needed to understand the detailed aspects of the underlying processes that might indicate potential risk to a planned development. Furthermore, there was a temptation for the sponsoring organizations to use the maturity level score as a “grade,” obscuring pertinent information about process-related strengths, weaknesses, and improvement activities associated with the planned development.

These considerations led to the current method of reporting strengths, weaknesses, and improvement activities in the SCE findings.

Shift from a “question-based” to a “model-based” method

Originally, the goal of an SCE was to validate the development organization’s responses to the questions on the Maturity Questionnaire,[1] but now the goal of an SCE is to evaluate the underlying KPAs. This change shifted the emphasis of an SCE from the questions in the Maturity Questionnaire to the KPAs in the ➠ **maturity model**.

The original SCE Method relied on the Maturity Questionnaire to “sample” a development organization’s software process. Information was collected to verify that the organization’s responses to the questionnaire were based upon actual practice, and then to determine the organization’s software process capability by scoring the validated responses. This was the “question-based” SCE Method.

Feedback from SCE users indicated that a more versatile and robust sampling mechanism was needed to ensure adequate coverage of the key processes. For example, not all of the KPAs were covered adequately by questions, and some questions were not based on KPAs at all. There were two issues: (1) teams needed a broader range of processes to draw the sample from, and (2) the sample had to be drawn from a finite set of process areas that were known to contribute to software process capability. These issues were addressed by using the KPAs in the maturity model as a basis for selecting processes to evaluate.

By 1989, the SCE emphasis had started to shift away from validating the Maturity Questionnaire responses to a more direct evaluation of the underlying KPAs. The current SCE Method uses a CMM-based questionnaire as one tool

[1] The “Maturity Questionnaire” refers to the “Assessment Recording Form” and the questions associated with it that are defined in *A Method for Assessing the Software Engineering Capability of Contractors* [Humphrey 87b]. This questionnaire was used through SCE Method Version 1.5. As of Version 2.0 (this document), teams are being trained to use a CMM-based maturity questionnaire. As future versions of CMM-based questionnaires are developed, they will be incorporated into the SCE Method.

to provide the teams with some initial information before the site visit. The information is one of the inputs considered when the team selects the topics for investigation on site.

SCE teams currently use KPAs to define the boundaries of an SCE at the highest level. Using KPAs gives the SCE teams a stable and robust framework for evaluating software process capability. The KPAs to be evaluated can be selected and tailored based on the needs of the planned development.

The shift from a question-based to a model-based method and the elimination of maturity level scores represent a major “paradigm shift” in the SCE Method—from validating answers to a fixed set of questions in order to assign a maturity level score to selectively sampling KPAs and determining the associated strengths, weaknesses, and improvement activities.

Refinement of the KPAs to include subprocess areas

The KPAs in the maturity model were refined to include subprocess areas.[1] Also, a set of elements [2] (now called *features*) common to all of the subprocess areas was defined to help the teams select topics to be evaluated.

Because of the shift away from the question-based method, SCE teams no longer evaluated information related to specific questions on the Maturity Questionnaire. Rather, they evaluated each KPA by observing and collecting information about the process implementations being used.

The KPAs are general in nature, and each KPA represents many possible process implementations. To evaluate the KPAs, the SCE teams needed a method to focus the investigation down to the level of observable work practices. The method for focusing the investigation had to help the teams select specific topics to be evaluated, yet ensure that the resulting evaluation stayed within the boundaries of the KPAs.

To help SCE teams evaluate KPAs effectively, each KPA was further refined into a set of subprocess areas. A set of elements common to every subprocess area (regardless of KPA) was defined; these elements were used to generate specific topics for evaluation. The topics derived in this way focused the

[1] In version 2.0 of the SCE Method (the version described in this document), the subprocess areas are derived from the goals of CMM v1.1 [Paulk 93a]. Previous versions of the SCE Method were not CMM-based.

[2] Previous versions of SCE, described in this section, used the term *element*. In version 2.0 of SCE (the version described in this document), the term *features* is used in place of *elements*. The term has changed; the concept has not.

common elements investigation of the subprocess area. The results were (and still are) “rolled up” into strengths, weaknesses, and improvement activities associated with the corresponding KPA.

For example, a topic for investigation might be investigating the “training” feature of the “control changes” subprocess area within the Software Configuration Management (SCM) KPA.

Collectively, subprocess areas and their common elements (now called *features*) improved the SCE team’s ability to probe specific software process capabilities.

Focusing SCEs based on risk for a specific development

The preparatory steps conducted before using the SCE Method were changed to focus each SCE on the software processes that contribute the most to risk for the planned development.

When the method was question based, SCE teams looked at essentially the same software processes each time the method was used. But by emphasizing the KPAs and subprocess areas, the method now allows an SCE team to select the areas for evaluation that are most important for the given use of the method.

The way an SCE is focused is through selection of the KPAs and subprocess areas for evaluation. The KPAs and subprocess areas are selected based on the attributes of the desired product and experience shortfalls within the development organization community relative to the planned development (and other factors described in Phase 2: General Preparation, Section 2.3 on page 49). The product attributes can indicate inherent risks, and experience shortfalls can indicate potential process-related risk.

The strengths and weaknesses observed in the process implementation form a picture of the software process-related risk to the planned development.

Decomposition of the method into discrete phases and steps

The method was decomposed into five activity phases, each containing several discrete steps.

This change was the result of a desire for greater consistency in the SCE Method, and of ongoing efforts to improve the SCE team training. The other evolutions of the method discussed earlier improved the versatility and utility of the SCE Method, but also increased the complexity of the method.

Decomposition of the SCE Method into phases and steps clarified several issues related to the transition from a “question-based” method to the current “model-based” method.

Both versions 1.5 and 2.0 of the SCE Method have the same 24 discrete steps, which are divided into the 5 activity phases introduced earlier. The steps in the SCE Method are described in detail in Section 2 of this document.

Public baselining of the SCE Version 1.5 Method Description

Publication of the *SCE Version 1.5 Method Description* [SCE 93] provided the first public description of the method since publication of *A Method for Assessing the Software Engineering Capability of Contractors* [Humphrey 87b].

Before the SCE v1.5 document was published, detailed information about the SCE Method was available only through SCE team training, which was available only to government teams.

Feedback from both industry and government indicated the need for an SCE Method baseline, and for “stakeholder” involvement in the future evolution of the SCE Method. *SCE Version 1.5 Method Description* [SCE 93] provided that baseline.

Publication of a public baseline gives the SCE Method a basis for controlled, public evolution in the future, and will help to make the SCE Method more consistent.

Public baselining of the CMM-based SCE Version 2.0 Method Description

This document incorporates CMM v1.1 [Paulk 93a] and the key practices of CMM v1.1 [Paulk 93b] into the SCE Method.

Here are the major changes from version 1.5 found in this document:

- The subprocess areas used are based on the goals of the CMM in a 1-for-1 manner.
- Guidance was developed to help team members select the CMM-based subprocess areas for evaluation.
- “Elements” used to select investigation topics have been replaced by “features” derived from the common features of CMM v1.1.
- Guidance has been developed to map the activities of *Key Practices of the Capability Maturity Model*, *Version 1.1* [Paulk 93b] to the subprocess areas, in the form of look-for tables.

CMM v1.1 as used in SCE provides a rich structure for data collection and consolidation, as described in Section 1.3 on page 20.

Impact of major changes

The changes described above document the evolution of the SCE Method from a “question based” to a more general “model based” evaluation method, and finally to a method based on CMM v1.1. The changes have made it easier to tailor an SCE to the needs of the product being developed. They improve the utility and versatility of the method by providing more thorough and detailed guidance to users of the method. Finally, the changes provide a baseline for orderly public evolution of the method in the future.

Part 2

Section 2 The SCE Process

2.1 Introduction

This section describes the steps in the SCE Method in detail, with the primary focus on *what* is done; less attention is given to *how* it is done. This section contains the following subsections:

Section name	Section and page number
Phase 1: Evaluation Start	Section 2.2, page 38
Phase 2: General Preparation	Section 2.3, page 49
Phase 3: Specific Preparation	Section 2.4, page 62
Phase 4: Site Data Collection (Site Visit)	Section 2.5, page 81
Phase 5: Findings	Section 2.6, page 100
Coordination of SCE Activities	Section 2.7, page 107

In SCE team training, several forms are used as examples of how to capture and preserve information during an SCE; copies of most of these forms are provided in Appendix C on page 155. The forms are conceptual in nature; they indicate information needed to conduct an SCE, but they are not mandatory.

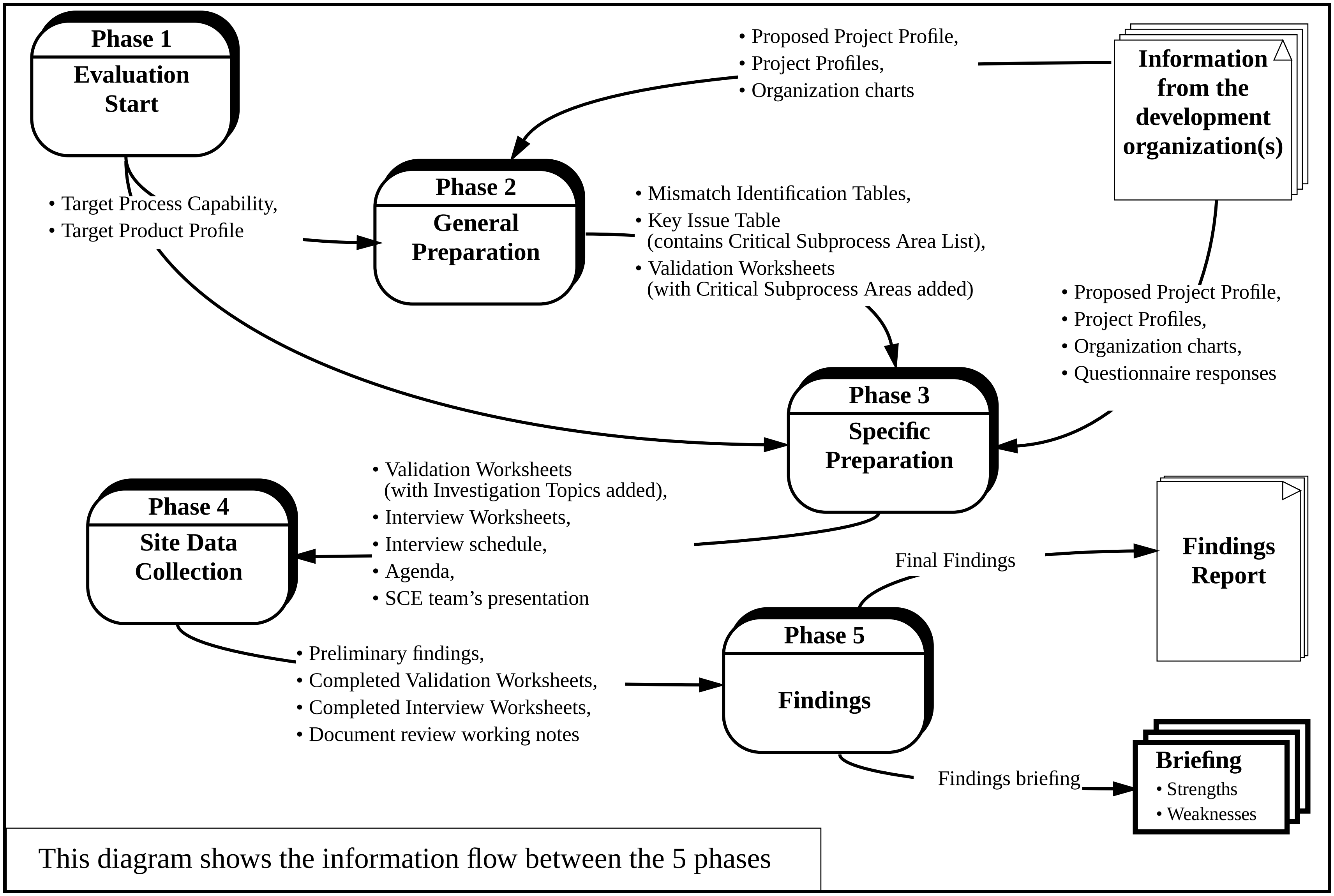
This section is intended for people who need more detail about the SCE Method than was provided in Section 1. The purposes of this section are to publicly document the SCE Method and to provide an in-depth introduction to the method. Realizing these purposes will help to clarify misunderstandings about the SCE Method and will help improve consistency in how SCEs are conducted.

The SCE Method has 24 steps, which are grouped into the 5 phases of activity introduced in Section 1.2 (see Table 2-1 on page 36). The structure of this section parallels the structure of the method, but emphasizes the steps within the phases rather than the phases. Discussion of each phase begins with a short overview of the phase that includes a diagram and a table of the steps in the phase. The discussion continues with a detailed description of the steps within the phase, and ends with a short summary that illustrates how the steps fit together.

Each step is described by providing a common set of information: the step name and number (for reference), inputs (or requirements for starting), actions taken and people involved (who does what), the purpose of the step (why), expected outcome (including outputs), and notes (to provide additional detail, caveats, special instructions, and so on).

There are some activities that span multiple phases or steps in the SCE Method. For example, information is requested from the development organization(s) in several of the phases, and proper scheduling of the site visit is crucial to the success of Phase 4, Site Data Collection. These activities are critical for integration of the steps in the SCE Method that are described in this section. Because the discussion requires some knowledge of the activities in the steps but doesn’t fit within the description of a single step, these items are discussed in Coordination of SCE Activities, Section 2.7 on page 107.

A high-level summary of the 5 phases and their major information interfaces is shown in Figure 2-1 on page 35. This diagram sets the stage for the remaining discussion, and is followed by a table that lists the steps within the phases (Table 2-1 on page 36).



This diagram shows the information flow between the 5 phases

Figure 2-1: Overview of Phases in SCE

A more detailed summary of the method is provided in Table 2-1, below. The table lists the phases, the steps within the phases, the primary purpose for each step, and a page number for easy reference. The first three phases collectively define the activities conducted before the site visit, the last two phases include the site visit and post site visit activities.

Phase	Step	Purpose	Page
Phase 1: Evaluation Start	1. Develop Target Product Profile	Understand attributes of the software product and the project required to produce it.	page 41
	2. Determine Target Process Capability	Determine the process capability that is most appropriate for the planned development—the Target Process Capability.	page 42
	3. Select Team	Have a trained team in place to execute the SCE.	page 44
Phase 2: General Preparation	4. Create Experience Table	Identify areas where the development organizations lack experience, indicating a potential for risk.	page 52
	5. Create Critical Subprocess Area List	Define and document the scope of the SCE, in terms of critical subprocess areas within the Target Process Capability KPAs.	page 55
	6. Originate Validation Worksheets	Record the set of critical subprocess areas for all development organizations on forms that can be used in subsequent information collection efforts.	page 59
Phase 3: Specific Preparation	7. Select Projects to Investigate	Select projects for evaluation that give the most insight into the processes that will be used.	page 64
	8. Develop Key Issue Worksheet	Create a consolidated list of key issues for investigation at the development organization site.	page 66
	9. Develop Topic Lists	Select topics for probing the process implementation; topics define observable work practices that map to the critical subprocess areas.	page 68
	10. Add Topics to Validation Worksheet	Capture the consolidated topic list for use at a particular site.	page 75
	11. Prepare for Exploratory Interviews	Develop detailed interview strategy, including the team’s decisions on who will be interviewed, when they will be interviewed, and what they will be asked.	page 75
	12. Prepare Entry Briefing	Establish the agenda for the initial organization meeting and set initial expectations for the site visit.	page 78

Table 2-1: Summary of Phases and Steps in an SCE

Phase	Step	Purpose	Page
Phase 4: Site Data Collection	13. Conduct Initial Organization Meeting	Clarify expectations of the SCE site visit.	page 84
	14. Conduct Initial Document Review	Determine the degree to which the organization and project-level documentation define and support standard processes for the KPAs and subprocess areas under investigation.	page 85
	15. Conduct Exploratory Interviews	Provide insight into how the subprocess areas are implemented in practice; determine the extent that processes have been internalized by the development organizations; identify critical implementation-level documents.	page 87
	16. Hold Team Caucus	Analyze, share, and consolidate information in order to reach conclusions about topics.	page 88
	17. Conduct Document Review	Search for objective evidence of how processes are implemented at the working level.	page 89
	18. Develop Preliminary Findings	Articulate conclusions about the subprocess areas based on the information available; guide subsequent information-gathering efforts.	page 91
	19. Create Consolidation Plan	Plan and initiate further data collection.	page 94
	20. Conduct Consolidation Interviews	Clarify any remaining issues by confirming or negating candidate findings through further interviews.	page 95
	21. Conduct Final Document Review	Clarify any remaining issues by confirming or negating candidate findings through further document review.	page 96
Phase 5: Findings	22. Determine Findings	Validate the preliminary findings and consolidate them by KPA.	page 102
	23. Produce Findings Report	Document the SCE activities and provide a formal record of the findings.	page 103
	24. Conduct Exit Briefing	Provide feedback to the recipient and conclude the SCE.	page 104

Table 2-1: Summary of Phases and Steps in an SCE

2.2 Phase 1: Evaluation Start

In this phase, the sponsoring organization decides to use the SCE Method and begins preparing to conduct the SCE. This phase is performed by the sponsoring organization; in all of the remaining phases the activities are conducted by the SCE team.

The purposes of the Evaluation Start phase are to determine the role of SCE, to determine the attributes of the desired software product and the project required to produce it, to determine the process capability that is most appropriate for the planned development, and to select the SCE team.

The SCE Method is performed within the context of a “larger” process such as source selection or contract monitoring. The Evaluation Start phase is where the relationships are established between the SCE Method and the process that uses the SCE findings. The first steps toward use of the SCE Method begin sometime during the preliminary planning for product development, when the role of the SCE is determined.

Determining the role of SCE consists of defining how the results can be used, deciding how SCE will fit in with any other technical and management evaluations of the development organization(s), and making a decision to use the SCE Method.

Planning for the SCE starts with the decision that the SCE Method should be used. During this phase, the people planning for the SCE should consider

- Funding for personnel, training, and travel.
- Coordinating SCE site visits and requests for information with the development organization(s) (see Sample Site Visit Schedules on page 118 and Information Request Timetable on page 121).
- Scheduling time for the SCE activities within the context of the use of the method (e.g., source selection or contract monitoring).

The information from the development organization is not used during this phase, but must be available when needed in the later phases. The information requested includes a Proposed Project Profile, six to eight Project Profiles for the projects that are candidates for evaluation, and organization charts and information for the projects and the organization. Questionnaire responses are usually requested at this time. In source selection, the information is usually requested in the RFP. In contract monitoring, an official request for the information is made.

Once the decision to use SCE is made, the sponsoring organization determines the software process capabilities needed to minimize the risk coming from the processes likely to be used for the planned development. This is accomplished by analyzing the attributes of the desired software product (Step 1), then determining the process capability that is most appropriate for the planned development (Step 2). The desired process capability is documented as the Target Process Capability and establishes the boundaries of the investigation—a KPA is evaluated if and only if it is part of the Target Process Capability. The sponsoring organization must also select the SCE team (Step 3). It is recommended but not necessary that these steps be performed in sequential order.

The people involved with the decision to use SCE are senior software project managers or acquisition managers and staff with software engineering experience. Establishing the Target Process Capability for the SCE should be done by staff with software engineering experience, possibly with help from SCE team members. The SCE team will be responsible for all of the subsequent work in the following phases.

Figure 2-2 on page 40 provides a high-level diagram of the steps in this phase.

This diagram shows the information flow between steps, not the sequence of activity.

Note: The inputs to Step 1 are omitted from this diagram, as is the output of Step 3 (the SCE team).

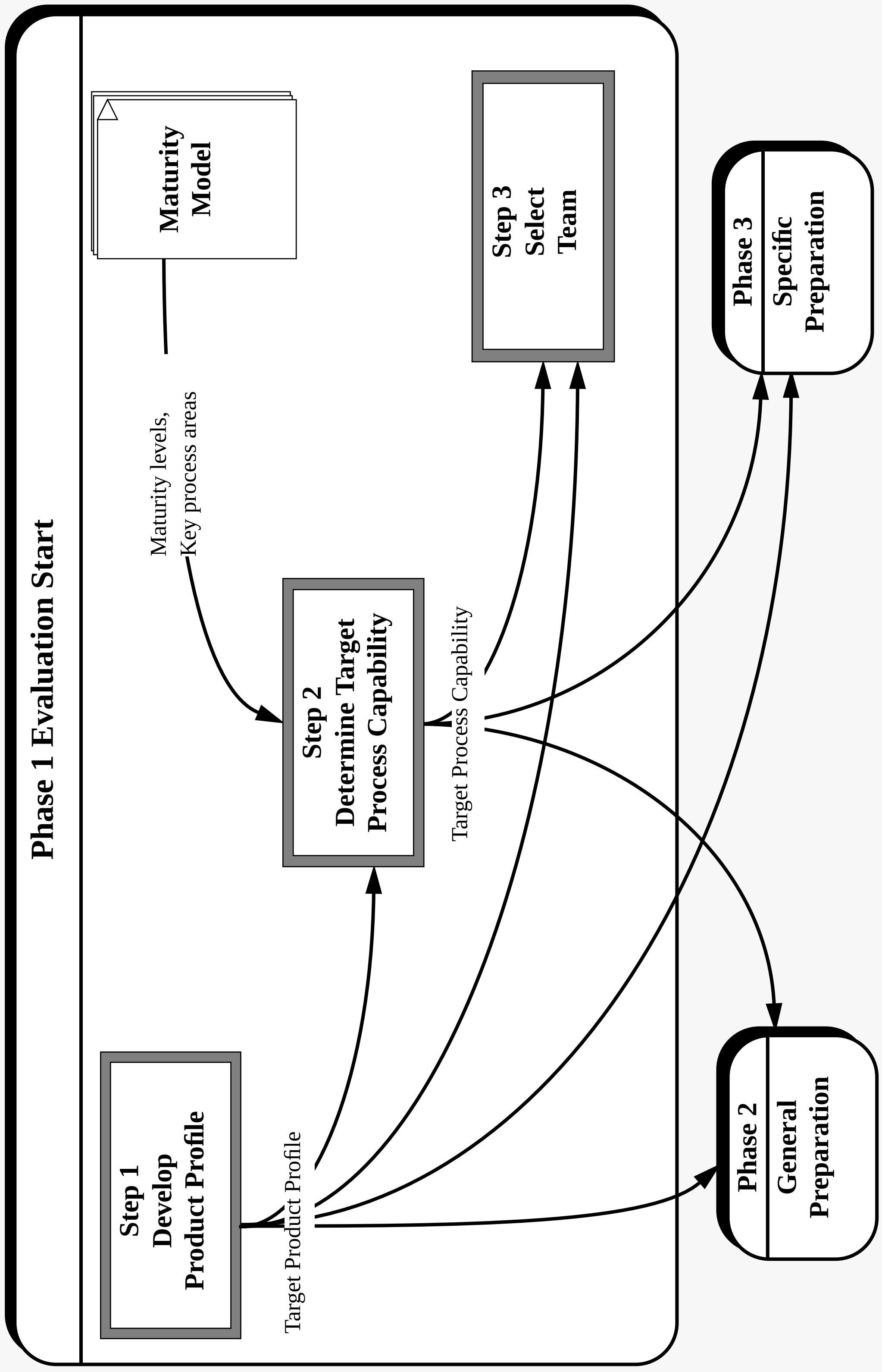

Phase 1 Evaluation Start
Maturity Model
Maturity levels, Key process areas
Step 1 Develop Product Profile
Target Product Profile
Step 2 Determine Target Process Capability
Target Process Capability
Step 3 Select Team
Phase 2 General Preparation
Phase 3 Specific Preparation

Figure 2-2: Diagram of Steps in Phase 1, Evaluation Start

The table below provides an overview of the steps in this phase.

Phase	Step	Purpose	Page
Phase 1: Evaluation Start	1. Develop Target Product Profile	Understand attributes of the software product and the project required to produce it.	page 41
	2. Determine Target Process Capability	Determine the process capability that is most appropriate for the planned development—the Target Process Capability.	page 42
	3. Select Team	Have a trained team in place to execute the SCE.	page 44

Table 2-2: Overview of Phase 1

Step 1 Develop Target Product Profile

Input

The inputs to Step 1 are the decision to use the SCE Method, and the context in which the method is used. Here are examples of inputs that depend on the context:[1]

- In source selection, the information known about the desired software product before release of the RFP.
- In contract monitoring, the established attributes of the project to be monitored.
- In a process improvement effort, a conceptual idea of the typical product or range of products that the development organization desires the capability to produce.

Action

The sponsoring organization generates a ➠ **profile** of product attributes (the Target Product Profile) for the product to be developed. The attributes used in SCE are defined in Appendix D on page 179.

Since the product has not yet been developed, the sponsoring organization estimates most of the attributes. An example Target Product Profile is shown in Table 2-3 (also see Appendix C.2 on page 158).

The Target Product Profile should be developed by people with software engineering experience, possibly with inputs from systems engineers. If the SCE team members have been selected (see Step 3), they should help with this effort.

Purpose

The purpose of this step is for the sponsoring organization to understand the attributes of the software product to be developed and the project required to produce it. The sponsoring organization must understand the nature of the development product before the development organization’s process can be evaluated in the proper context.

[1] For simplicity, these inputs are omitted from the step diagrams (e.g., Figure 2-2 on page 40).

Outcome

The direct output is the Target Product Profile, used as an input to Steps 2, 3, 5, and sometimes Steps 4 and 7. The primary outcome is a better understanding on the part of the sponsoring organization of the product to be developed. That understanding is communicated to the SCE team through the Target Product Profile.

Notes

The first column in Table 2-3 shows the attribute type. For a Target Product Profile, the attributes are organized into major and minor categories (see Appendix D on page 179 for a list of major and minor attributes and their definitions). The second column lists attribute names. The third column lists an example of a Target Product Profile.

Major attributes significantly impact the implementation of the software process environment that supports product development.

Minor attributes primarily impact implementation details within the environment that supports the software developers.

Attribute Type	Attribute Name	Target Product Profile
Major	Application Domain Product Type Size Type of Work Operational Precedence	Command and control ASW helicopter/sono-buoys 24 months, 100 software engineers, 300 KSLOC Full development No - replacement of existing system

Minor	Language(s)
	Target(s)
	Applicable Standards
	Customer
		Ada
		M68000
		DoD-STD-2167A, 2168
		NAVAIR

Table 2-3: Attributes and Target Product Profile

Step 2 Determine Target Process Capability

Input

The Target Product Profile from Step 1 is used along with the key process areas (KPAs) from the maturity model (see Figure 2-3 on page 43, also see Appendix A.2 on page 131).

Action

The sponsoring organization determines the key process areas (KPAs) to be evaluated at all development organization sites. These KPAs form the boundary, or Target Process Capability, of the evaluation.

The recommended Target Process Capability encompasses the KPAs within the Repeatable and the Defined levels as shown in Figure 2-3 on page 43; this is the default. At a minimum, the Target Process Capability must include at least the Repeatable level KPAs.

This step should be done by senior people with software engineering experience who have a good understanding of the Target Product Profile attributes and software process concepts. Since SCE team members use the output of this step, they should help with this effort if they have been selected (Step 3).

This activity establishes the boundaries of the evaluation at a high level. The same KPAs are the basis for evaluation at all development organization sites—a KPA is evaluated if and only if it is in the Target Process Capability. Development organizations must understand what to expect when an SCE is conducted; the Target Process Capability can be used to communicate the boundaries of the SCE to the development organizations.

	Maturity Level	**Key Process Area**
	Optimizing	Defect Prevention Technology Change Management Process Change Management
	Managed	Quantitative Process Management Software Quality Management
Recommended (default) Target Process Capability: Repeatable and Defined KPAs	*Defined*	Organization Process Focus Organization Process Definition Training Program Integrated Software Management Software Product Engineering Intergroup Coordination Peer Reviews
Minimum Target Process Capability: Repeatable KPAs	*Repeatable*	Requirements Management Software Project Planning Software Project Tracking and Oversight Software Subcontract Management Software Quality Assurance Software Configuration Management
	Initial	(none)

Figure 2-3: Key Process Areas and Target Process Capability

Purpose The purpose of this step is to determine the process capability that is most appropriate for the planned development and to document the desired capability as the Target Process Capability.

Outcome The output of this step is the Target Process Capability, which defines the boundaries of the evaluation at the KPA level. The Target Process Capability is an input to Steps 3, 5, 8, and 12.

Notes In source selection, Steps 1 and 2 are accomplished before the RFP is released.

A considerable amount of information is collected from the development organization(s) between the time Steps 1 and 2 are completed and the first site visit occurs (see Section 2.7, Coordination of SCE Activities, page 121.) Planning this data collection effort and defining how the SCE teams will interact with the development organizations is critical to successful use of the SCE Method.

To be assured of performance at a particular maturity level, all of the KPAs at all levels through the highest level desired must be included. Although maturity level scores are no longer part of an SCE—that is, organizations are no longer rated as Level 1, Level 2, and so on—the requirement to use at least the Repeatable level KPAs as the Target Process Capability has been kept. This was done because there is little chance of benefit from Defined level capability if the Repeatable level KPAs are not implemented effectively, while lack of Repeatable level capability significantly increases risk. There are few[1] organizations that effectively implement all of the KPAs in the recommended default Target Process Capability (Repeatable and Defined level KPAs); since very few development organizations demonstrate the higher levels of maturity, the recommended Target Process Capability is sufficient at this time for most applications of the SCE Method.

Step 3 Select Team

Input The Target Product Profile from Step 1 and the Target Process Capability from Step 2 may be inputs.

Action The sponsoring organization selects the individuals who will conduct the SCE. The selection of an SCE team must be completed in this phase so that the individuals can be assigned to the team and trained in the SCE Method, and can go through normal team building activities prior to planning and conducting the steps in the remaining SCE phases.

[1] According to *An Analysis of SEI Software Process Assessment Results: 1987–1991*, by David H. Kitson and Steve Masters [Kitson 92].

All team members must be trained. If the sponsoring organization selects someone who is not trained, they must schedule training and allow enough time for completion of training before conducting the remaining steps of the SCE.

Team selection should be accomplished by someone senior enough in the organization to commit the resources for the duration of the period that SCEs will be performed.

The Target Product Profile and Target Process Capability help define the expertise the team needs. The team requires expertise in each of the KPAs in the Target Process Capability and should have expertise with the product type and application domain from the Target Product Profile.

Purpose The purpose of team selection is to have a trained team in place to plan and execute the remaining steps of the SCE.

Outcome The desired outcome is a skilled and compatible team to be trained in the SCE Method. The SCE team can be considered to be the output of this step.[1]

Notes The SCE team can be selected before completion of Steps 1 and 2. If this is done, the team members can assist with those activities. Alternatively, the team leader can be selected and take responsibility for working the other Evaluation Start phase activities, including selection of the other team members.

The same team should conduct all SCEs for a particular use of the method, especially in source selection, where consistency of results across all of the development organizations is essential. Once the team is established, the team should be left intact for continuity of effort.

SCEs are conducted by a team to avoid individual bias, and SCE findings are made by consensus. There is no rank associated with team members during team deliberations; in this respect the team is like a jury.

Each individual on the team is important to the success of the SCE. The individuals must possess the right qualifications to participate on the SCE team, and the team must have the right balance of skills and experience.

For a team to be successful, several criteria must be met. These criteria are discussed below; they include training, team composition, team leadership, team member experience and knowledge, individual skills, and team development skills.

[1] The SCE team is not listed on the step diagrams as an output.

Training. All team members must be trained in the SCE Method. Team members trained previously may require additional training, particularly if the training they attended was conducted before 1992.

Team Composition. At a minimum, the SCE team members must have an average of seven years of software development or software management experience; software acquisition experience is also helpful. At least two team members should have participated in previous SCEs. No more than one team member should have less than two years of professional software experience.

Leadership. Ideally, the team leader should be an experienced individual who has participated in two or more SCEs as a team member.

Team member experience and knowledge. Collectively, the team must have knowledge of and experience with

- The application domain and product type.
- The management processes required to create an effective environment for the engineering and development of a software product.
- The major phases that engineering and development of a software product must go through.
- The support processes and management environment required to reduce or eliminate unnecessary rework within the engineering and development of a software product.
- The relationship between technology (in the form of methods and tools) and the support processes.

Individual skills. Each SCE team member must have the practiced skill to

- Perform all the roles required of an SCE team member (e.g., facilitator, recorder, and participant).
- Conduct SCE interviews (e.g., make an interviewee feel at ease, ask open-ended yet focused questions, keep the interviewee on track).
- Separate what an interviewee says from what the listener hears (i.e., to be consciously aware of their own paradigms which act as filters and translators of what is said).

Team development skills. All of the SCE team members must actively work at the initial team building and, once built, at continued development of the team. This requires skills in consensus building, conflict resolution, negotiation, and decision making.

The criteria described above are necessary, but do not guarantee success. The team must work well together under stress. Whenever possible, the team should engage in extensive team building activities before the first site visit, possibly including a practice SCE site visit.

Summary of Phase 1 Evaluation Start

The Evaluation Start phase is where the SCE Method interfaces with the acquisition or contract monitoring process.

When the Evaluation Start phase is complete the decision to use SCE is made, the role of SCE is established, the boundaries of the SCE are established at the KPA level, and the SCE team is trained and in place.

In Steps 1 and 2 collectively, the sponsoring organization determines the software process capabilities needed to minimize the risk related to the processes likely to be used for the planned development. In Step 1, the desired software product is analyzed, and the Target Product Profile is created. The Target Product Profile is an estimate of the basic software product attributes of the product to be developed and the project required to produce it. In Step 2, personnel with software engineering experience use the Target Product Profile information and their knowledge of software processes to determine the Target Process Capability—that is, the process capability that is most appropriate for the planned development. The Target Process Capability defines the boundaries of the SCE at the KPA level; these (and only these) KPAs will be evaluated at all development organization sites.

Selecting an SCE Team (Step 3) requires planning and an organizational commitment to use SCE. Commitment is shown by allocating personnel resources to the SCE team; planning is needed to ensure that adequate time is factored into the schedule for training and for performing the SCE site visits.

The SCE team will be responsible for all of the subsequent work in the following phases. The Target Product Profile and Target Process Capability defined in this phase are used in Phase 2, General Preparation.

Before Steps 4 and 5 in the General Preparation phase, the team will need information from the development organization, including the Proposed Project Profile, Project Profiles for the projects that are candidates for evaluation, and organization charts and information. Usually, questionnaire responses are also requested at this time (see Information Request Timetable on page 121). This information provides the development organization’s view of the product to be built and provides information about the projects that the development organization is submitting for evaluation. The sponsoring organization must request the information during this phase so it is available when needed.

2.3 Phase 2: General Preparation

The General Preparation phase consists of site visit preparation activities that pertain to all of the development organizations equally.

In this phase, the SCE team completes high-level preparations for evaluating all of the development organizations that are involved with this use of the method. The General Preparation phase starts when the SCE team has received all of the information requested from the development organization(s) during Phase 1, Evaluation Start.

The purpose of the General Preparation phase is to define the scope of the investigation for all of the development organizations. The scope of the SCE consists of subprocess areas within the KPAs that make up the Target Process Capability, and will be used to evaluate all development organizations.

To achieve this purpose, the SCE team identifies those software processes that contribute most to the potential development risk throughout the development organization community. To do this, the team examines information from each development organization about their view of the product to be built (in the form of a Proposed Project Profile). The team also examines preliminary information from each development organization about the software projects they are submitting for evaluation (in the form of Project Profiles).

The various profiles from the development organization are compared to identify areas where the development organization may lack experience, indicating potential risk. The experience shortfalls of the individual development organizations are then consolidated for the development organization community (Step 4). The experience shortfalls indicate areas that may have higher risk, and should be investigated.

Based on the experience shortfalls in the development organization community (and other factors described in Step 5), the SCE team selects subprocess areas within each of the Target Process Capability KPAs for evaluation. These subprocess areas are called critical subprocess areas and will be investigated at all development organization sites. Collectively, they make up the Critical Subprocess Area List and define the scope of the SCE.

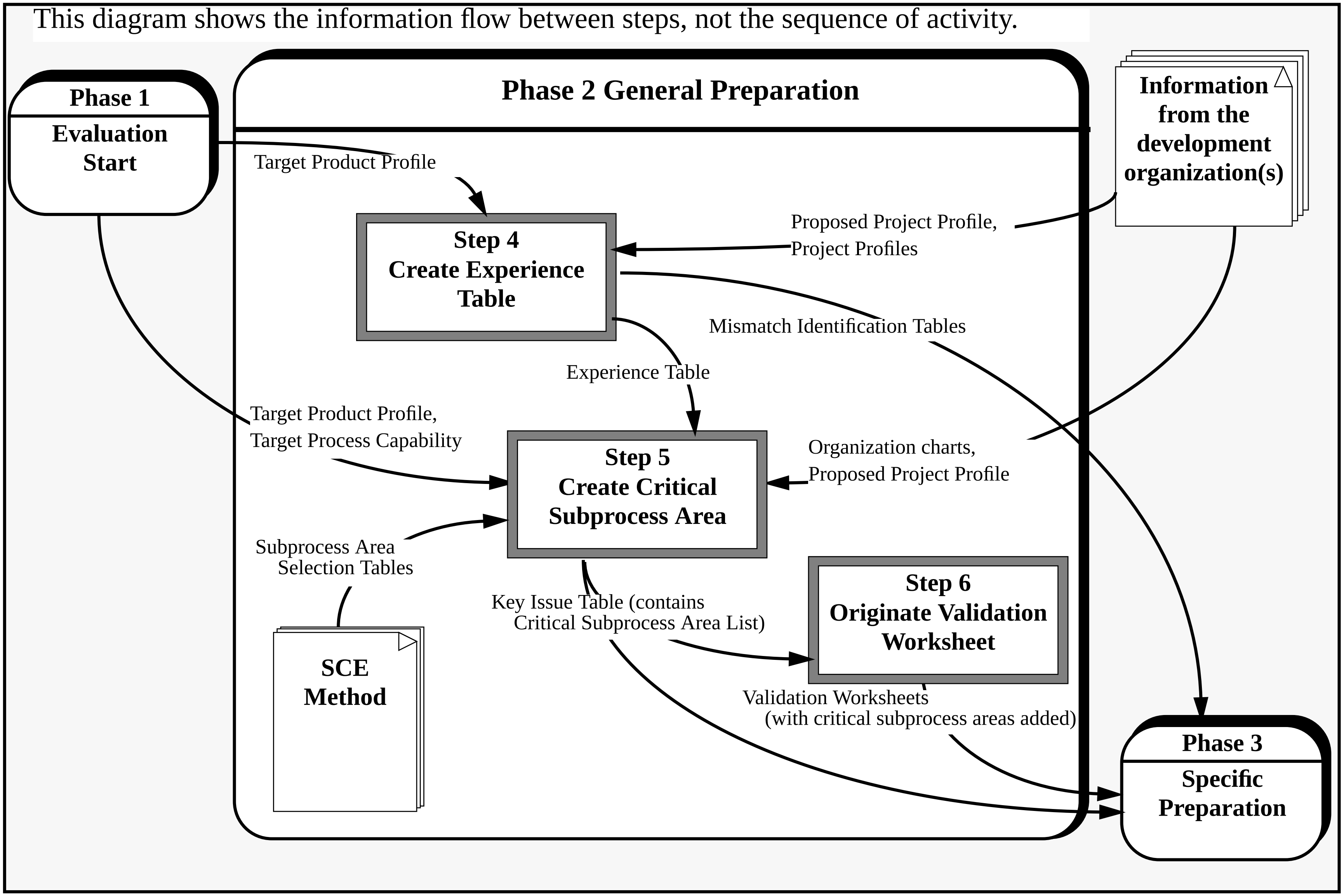
A set of Validation Worksheets is created for each development organization, one for each subprocess area on the Critical Subprocess Area List (Step 6).

The information in the Proposed Project Profile and the Project Profiles must be requested sufficiently in advance of Step 4 to allow the development organization(s) time to respond (see Information Request Timetable on page 121). In source selection, these are typically requested as part of the RFP.

The activities in the General Preparation phase establish the context for the Specific Preparation phase (Phase 3). General Preparation as described here applies primarily to use of the SCE Method in a source selection context, where multiple organizations will be evaluated using the same subprocess areas.

In a contract monitoring effort, the same steps should be followed for the initial evaluation. Subsequent evaluations could use a tailored subset of the Critical Subprocess Area List developed for the initial evaluation. During preparations for a subsequent evaluation, the team should concentrate on changes implemented since the previous evaluation. The team should also consider the special needs of the contract to be monitored and the weaknesses observed during the previous evaluation. Each evaluation in turn acts as a baseline when preparing for the next one.

Figure 2-4 on page 51 provides a high-level diagram of the steps in this phase.


This diagram shows the information flow between steps, not the sequence of activity.

Phase 1
Evaluation Start
Phase 2 General Preparation
Information from the development organization(s)
Target Product Profile
Step 4 Create Experience Table
Proposed Project Profile, Project Profiles
Mismatch Identification Tables
Experience Table
Target Product Profile, Target Process Capability
Step 5 Create Critical Subprocess Area
Organization charts, Proposed Project Profile
Subprocess Area Selection Tables
SCE Method
Key Issue Table (contains Critical Subprocess Area List)
Step 6 Originate Validation Worksheet
Validation Worksheets (with critical subprocess areas added)
Phase 3
Specific Preparation

Figure 2-4: Diagram of Steps in Phase 2, General Preparation

The table below provides an overview of the steps in this phase.

Phase	Step	Purpose	Page
Phase 2: General Preparation	4. Create Experience Table	Identify areas where the development organizations lack experience, indicating a potential for risk.	page 52
	5. Create Critical Subprocess Area List	Define and document the scope of the SCE, in terms of critical subprocess areas within the Target Process Capability KPAs.	page 55
	6. Originate Validation Worksheets	Record the set of critical subprocess areas for all development organizations on forms that can be used in subsequent information collection efforts.	page 59

Table 2-4: Overview of Phase 2

Step 4 Create Experience Table

Input

The inputs to this activity are

- The Target Product Profile from Step 1.
- The Proposed Project Profile from the development organization.
- A Project Profile for each of the projects that has been submitted for evaluation by the development organization.

The Proposed Project Profile is created by the development organization, and consists of an attribute profile similar to the Target Product Profile created by the sponsoring organization in Step 1, except the Operational Precedence attribute is replaced by the Subcontractors attribute. The Proposed Project Profile describes the development organization’s view of the proposed project. A Proposed Project Profile is shown in Appendix C.2 on page 158.

The development organization also submits a Project Profile for each project that is a candidate for evaluation. Typically, six to eight Project Profiles are submitted from projects at the development site where the work will be done. The Project Profiles contain information about the same attributes as the Proposed Project Profile; additionally, they contain information about the development project such as the current development phase, how many months since the project started, etc. The Project Profiles capture information about products the development organization has already developed or is in process of developing, and they indicate development experience that is pertinent to the proposed product. (In source selection, the projects submitted as candidates for evaluation must also meet any other requirements of the RFP that pertain to project selection.) A sample Project Profile is shown in Appendix C.3 on page 160.

The Proposed Project Profile and Project Profiles must be requested during Phase 1 Evaluation Start, and may be combined with requests for other documentation (see Information Request Timetable on page 121.)

Action

The SCE team identifies areas in which the development organizations may lack the experience required to build the proposed product by examining the attributes in the various profiles submitted by the development organization. This activity is performed in two stages: (1) the team identifies mismatched attributes for each development organization, and (2) the team tabulates and summarizes the experience mismatches across all of the development organizations on a single table. To illustrate this concept, an example of each of these activities follows.

Identifying mismatched attributes for each development organization. Mismatched attributes are identified by comparing the attributes in the Proposed Project Profile to the Project Profiles submitted by the same organization. The additional attributes in the Project Profiles are not used in this step (i.e., development phase and schedule status). The purpose of comparing is to look for similarities, not for exact matches. For example, a 98 KSLOC system would be considered a match for a 105 KSLOC system. Judgment and team consensus are used to resolve any questionable comparisons.

A mismatch is defined to exist for an attribute *only if none of the Project Profiles match* the Proposed Project Profile for that attribute (see Table 2-5). A Mismatch Identification Table (see Appendix C.4 on page 161) is created to consolidate the information resulting from comparing the profiles submitted by a development organization.

Major attributes	Project Able	Project Baker	Project Charley	Project Delta	Project Foxtrot	Project Gamma	Org "A" Result
Application Domain	0	0	1	0	0	0	
Size	0	0	0	0	0	0	Ps

Ps = Size (attribute used in SCE)

Table 2-5: Identifying Mismatched Attributes for Organization A

Matches for the projects are shown by a 1, mismatches by a 0. The Application Domain is acoustic signal processing, and the "Estimated Software Size" part of the Size attribute for the proposed system is 1,000 KSLOC. Every project submitted (except Charley) was a command and control system, and the size of each was under 300 KSLOC. Because there are no matches anywhere in the Size row, the result is "Ps" (an abbreviation for product size), indicating a mismatch for Organization A.

Tabulating and summarizing the experience mismatches. After identifying the mismatches for each development organization, the team tabulates and summarizes the experience mismatches across all of the development organizations. The information is recorded on a single Experience Table (see Appendix C.5 on page 164).

An experience mismatch is defined to exist for an attribute *if any one of the development organizations has a mismatch* for that attribute (see Table 2-6). This ensures that any potential risk area within the development organization community is addressed during the evaluation. The development organizations are not compared to each other. Instead, the potential risk for any of the development organizations is translated into an indication of processes that should be investigated across the development organization community.

Major Attributes	Org A	Org B	Org C	Org D	Result
Application Domain					
Size	Ps				Ps

Ps = Size (attribute used in SCE)

Table 2-6: Tabulating and Summarizing Experience Mismatches

Experience mismatches are indicated by an entry in the corresponding row. The Application Domain is acoustic signal processing, and the "Estimated Software Size" part of the Size attribute for the proposed system is 1,000 KSLOC. Every organization has Application Domain experience, but Organization A has not developed any projects this large. This is shown by the "Ps" (an abbreviation for product size) under Organization A in the Size row. Hence the development organization community as a whole does not have experience with large projects. This is indicated in the result column.

Purpose

The purpose of this step is for the SCE team to identify areas in which the development organizations may lack experience, indicating a potential for risk. A development organization must have well-defined processes to mitigate the risk, especially if the development organization lacks product type or application domain experience.

Outcome

Direct outputs are the Experience Table (used in Step 5) and the Mismatch Identification Tables (used in Steps 7 and 9). The Experience Table tabulates and summarizes the experience mismatches for all of the development organizations. The Mismatch Identification Table consolidates the information resulting from comparing the Proposed Project Profile and the Project Profiles submitted by a development organization. The Proposed Project Profile and Project Profiles from the development organization(s) are kept for use in later steps.

Notes

In general, the development organization’s Project Profiles are drawn from projects at the site where the development work will be performed. There are two reasons for using profiles from the site where the work will be done: (1) it reduces the expense of performing the evaluation, because all of the documentation and personnel are on site, and (2) projects developed at a particular site are better indicators of the way work is likely to be done at that site—processes used at other sites may vary, and are much less likely to be used effectively at the proposed development site than processes already in place.

If the development organization has not submitted a Proposed Project Profile to the sponsoring organization, the Target Product Profile from Step 1 may be used to identify the mismatches instead, except the Operational Precedence attribute is not used.

The Target Product Profile (from Step 1) represents a “customer view” of the product to be built, and the Proposed Project Profile represents a “developer view.” Both of these give insight into the planned development processes, but both are estimates because the product hasn’t been built yet. The Project Profiles for the projects that are candidates for evaluation are not estimates; they represent real projects with actual processes that can be evaluated. If there is a close match between the planned project and the development organization’s actual projects, then the actual development processes currently in use are good indicators of the processes that will be used for the new development.

Step 5 Create Critical Subprocess Area List

Input

The inputs to this step are

- The Target Product Profile from Step 1.
- The Target Process Capability from Step 2.
- The Experience Table from Step 4.
- The Proposed Project Profile from the development organization.
- Organization charts and information from the development organization.
- The CMM v1.1 Subprocess Area Selection Tables (see Appendix E on page 185).

The organization charts and information from the development organization must be requested during Phase 1, Evaluation Start, and may be combined with requests for other documentation (see Information Request Timetable on page 121).

Action

There are three activities performed in this step: (1) selecting the critical subprocess areas to be investigated (the Critical Subprocess Area List), (2) documenting the critical subprocess areas as the ➠ **key issues** on the Key Issue Table (see Appendix C.6 on page 166), and (3) comparing the sponsoring organization’s view of the planned development (the Target Product Profile) to the development organization’s view (the Proposed Project Profile). Each of these activities is discussed below.

Selecting the critical subprocess areas to be investigated. The SCE team uses all of the information available to determine the subprocess areas that will be investigated at each site. The subprocess areas selected for investigation are called critical subprocess areas. Collectively, the critical subprocess areas define the scope of the SCE; the same critical subprocess areas are used to investigate the processes in use at each development organization. The set of critical subprocess areas is collectively referred to as the Critical Subprocess Area List. The Critical Subprocess Area List is not a distinct product; it is conceptual in nature. The critical subprocess areas are annotated on the Key Issue Table in Step 6.

Selecting the critical subprocess areas is a complex activity and is performed in several stages. First, a preliminary list is constructed one (or both) of two ways: by considering the product size, or by considering other factors such as experience mismatches, operational precedence, and a recommended “nucleus capability.”

To select critical subprocess areas by product size, the team looks at the size of the project, expressed in terms of the management structure proposed for the project. The team uses the subprocess areas recommended in the table in Appendix E.1 on page 187 as the initial set of subprocess areas.

Selecting subprocess areas based on other factors is done by performing a table “look up,” using the Subprocess Area Selection Tables (Appendix E.2 on page 188). The rows of the table are subprocess areas. The columns of the table are the following

- The major attributes from the Experience Table (Application Domain, Product Type, Size, Type of Work, Subcontractors).
- The Operational Precedence attribute from the Target Product Profile.
- A “nucleus capability” column indicating subprocess areas that are important for every development.

Within each column, rows are marked to identify relevant subprocess areas for evaluation. These subprocess areas address potential risk associated with the attribute and are intended as a guide.

The table “look up” starts by using any mismatched major attributes indicated in the Experience Table. The corresponding column in the Subprocess Area Selection Table identifies a set of relevant subprocess areas for evaluation; these are added to the list. After using the Experience Table mismatches, the Operational Precedence column is used (if applicable). The nucleus capability column is then used to complete the preliminary Critical Subprocess Area List.

Either (or both) of the methods discussed is used to generate a preliminary Critical Subprocess Area List; the list is refined using these additional considerations:

- Critical subprocess areas are limited by the boundaries of the Target Process Capability. Any subprocess area from a KPA that is not in the Target Process Capability is removed from the list.
- At least one subprocess area must be selected as critical for each KPA in the Target Process Capability. If a KPA does not have a corresponding subprocess area selected for evaluation, the team adds at least one of the subprocess areas for that KPA to the list.
- The SCE team may select additional subprocess areas for any KPA within the Target Process Capability based on their own experience and judgment.
- If multiple subprocess areas have been selected for a given KPA, the team may choose to eliminate a subprocess area to reduce the number of items that will be investigated. (However, at least one subprocess area must be investigated for each KPA in the Target Process Capability.)

As noted above, team judgment is used to select additional subprocess areas for evaluation. All of the information available to the team is used to make these judgments. Factors that might be considered include

- A mismatch in the minor attributes (such as Language).
- The various organizational structures—for example, an organization without a separate SQA function.
- Mismatches between attributes in the Target Product Profile and the Proposed Project Profile (discussed below).

The result of the table look up and the list refinement described above is the Critical Subprocess Area List. The boundaries of the SCE are defined by the Target Process Capability; but the Critical Subprocess Area List provides an additional level of detail. Each of the subprocess areas on the Critical Subprocess Area List will be investigated at each development organization.

Documenting the critical subprocess areas as key issues. The Key Issue Table (see Appendix C.6 on page 166) is used to document the Critical Subprocess Area List and the reason for adding each subprocess area to the list. The Critical Subprocess Area List pertains to the development organization community as a whole; the Key Issue Table documents the list and summarizes each organization's contribution to the list.

The team records the Critical Subprocess Area List on the Key Issue Table. The critical subprocesses are sorted by their associated KPA within the Target Process Capability. Each critical subprocess area on the list defines a row of the Key Issue Table.

Each development organization has a column in the table. The team annotates the Key Issue Table with information relating the critical subprocess area to the development organization in the corresponding column; this information identifies "key issues" for the development organization. For example, if the subprocess area was selected because it was part of the nucleus capability, this would be annotated in the table for each development organization. If a subprocess area was selected because of a Size attribute mismatch for organizations A and C, then the columns for those organizations would be annotated to show the Size mismatch.

Comparing the sponsoring organization's view of the planned development to the development organization's view. Another activity performed during this step is a comparison of the Proposed Project Profile from each development organization to the Target Product Profile generated by the sponsoring organization in Step 1. This comparison is one of the factors the team could consider when adding subprocess areas to the Critical Subprocess Area List. For example, if there are major differences in the two profiles, the team could treat it as a mismatch and add subprocess areas to the Critical Subprocess Area List accordingly.

Purpose

The purpose of this step is to define and document the scope of the SCE, in terms of critical subprocess areas within the Target Process Capability KPAs.

Outcome

The direct outputs are the Critical Subprocess Area List and the Key Issue Table. The Critical Subprocess Area List is documented in the Key Issue Table (used in Steps 6 and 8).

Notes

If a development organization lacks experience in one or more attributes, and if the relevant subprocess areas are not well defined within a development organization's operations, the development may be at risk of not meeting cost, schedule, or quality targets. In the interests of source selection fairness, the same critical subprocess areas will be investigated for each development organization. Once the critical subprocess areas are selected, the scope of the SCE is established–subprocess areas cannot be added or deleted after the SCE begins to investigate individual development organizations.

As mentioned, one of the tasks performed is checking if the development organization's view of the product to be built is similar to the sponsoring organization's view. This is done by comparing the Target Product Profile to the Proposed Project Profile. Usually the Target Product Profile and the Proposed Project Profile will be nearly identical—if they differ greatly, it should be investigated. This is a topic of concern because it indicates a major difference in understanding about what the development project entails. Resolving these differences in understanding is not part of the SCE investigation, but should be brought to the attention of the sponsoring organization as a concern to be resolved through other channels.

The Target Product Profile represents a "customer view" of the product to be built, while the Proposed Project Profile represents a "developer view." Major differences in these points of view can indicate innovation or a lack of understanding. For example, assume the sponsoring organization estimates 1,000 KSLOC for size and the development organization estimates 300 KSLOC. The development organization may be planning to reuse code from a previous project, or one of the organizations may not understand the magnitude of the required development effort. Understanding why the estimates differ is essential.

Step 6 Originate Validation Worksheets

Input

The input to this step is the Critical Subprocess Area List, as documented on the Key Issue Table in Step 5.

Action

The SCE team creates a Validation Worksheet for each subprocess area on the Critical Subprocess Area List. This is done by entering the KPA and the subprocess area on the top of the Validation Worksheet. This records the Critical Subprocess Area List on a set of forms that can be used throughout the SCE to record the results of the investigation. The set of Validation Worksheets is replicated for each development organization, and is used for each SCE conducted at a development organization site. A sample Validation Worksheet is shown in Appendix C.7 on page 169.

Purpose

The purpose of this step is to record the set of critical subprocess areas for all development organizations on forms that can be used in subsequent information collection efforts. Later, the team will use the validation worksheets to guide information collection efforts for all the critical subprocess areas. When completed, these worksheets are used to generate and support the findings at the end of the SCE.

Outcome

The output of this stage is a set of Validation Worksheets, used throughout the rest of the SCE to guide information collection efforts.

Summary of Phase 2 General Preparation

When the General Preparation phase is complete, the SCE team will have identified areas in which the development organization community lacks experience, and will have determined the critical subprocess areas that will be investigated for all development organizations; this defines the scope of the SCE.

In Phase 1 Evaluation Start, the Target Product Profile was used to identify critical processes at the KPA level, forming the Target Process Capability. In the General Preparation phase, the collective experience of the development organization community is used to define and tailor the scope of the SCE down to the subprocess area level within the KPAs. This is done by analyzing information about the planned development and project information to identify areas in which the development organizations may lack experience (Step 4). The SCE team selects critical subprocess areas for investigation based on the experience shortfalls in the development organization community and other factors (Step 5).

These subprocess areas comprise the Critical Subprocess Area List. The critical subprocess areas will be evaluated across all of the development organizations; collectively, they define the scope of the SCE. The Critical Subprocess Area List is not kept as a separate document; instead, the critical subprocess areas are entered on a set of Validation Worksheets that are used throughout the SCE (Step 6).

Much of the information used during the General Preparation phase is also used extensively during the Specific Preparation phase. For example, the Mismatch Identification Tables created in Step 4 are used in Step 7 to select the projects that will be investigated at the development organization’s site.

General Preparation as described here applies primarily to use of the SCE Method in a source selection context, where multiple development organizations will be evaluated using the same subprocess areas. In a contract monitoring effort, the same steps should be followed for the initial evaluation, but subsequent evaluations could use a tailored subset of the Critical Subprocess Area List developed for the initial evaluation. During preparations for a subsequent evaluation, the team should concentrate on changes implemented since the previous evaluation. The team should also consider the special needs of the contract to be monitored and the weaknesses observed during the previous evaluation. Each evaluation in turn acts as a baseline when preparing for the next one.

The activities during the General Preparation phase establish the context for the activities in Phase 3, Specific Preparation. The General Preparation activities define the Critical Subprocess Area List. The Specific Preparation phase will take the critical subprocess areas on the list and use them to prepare topics for investigation at a development organization site.

2.4 Phase 3: Specific Preparation

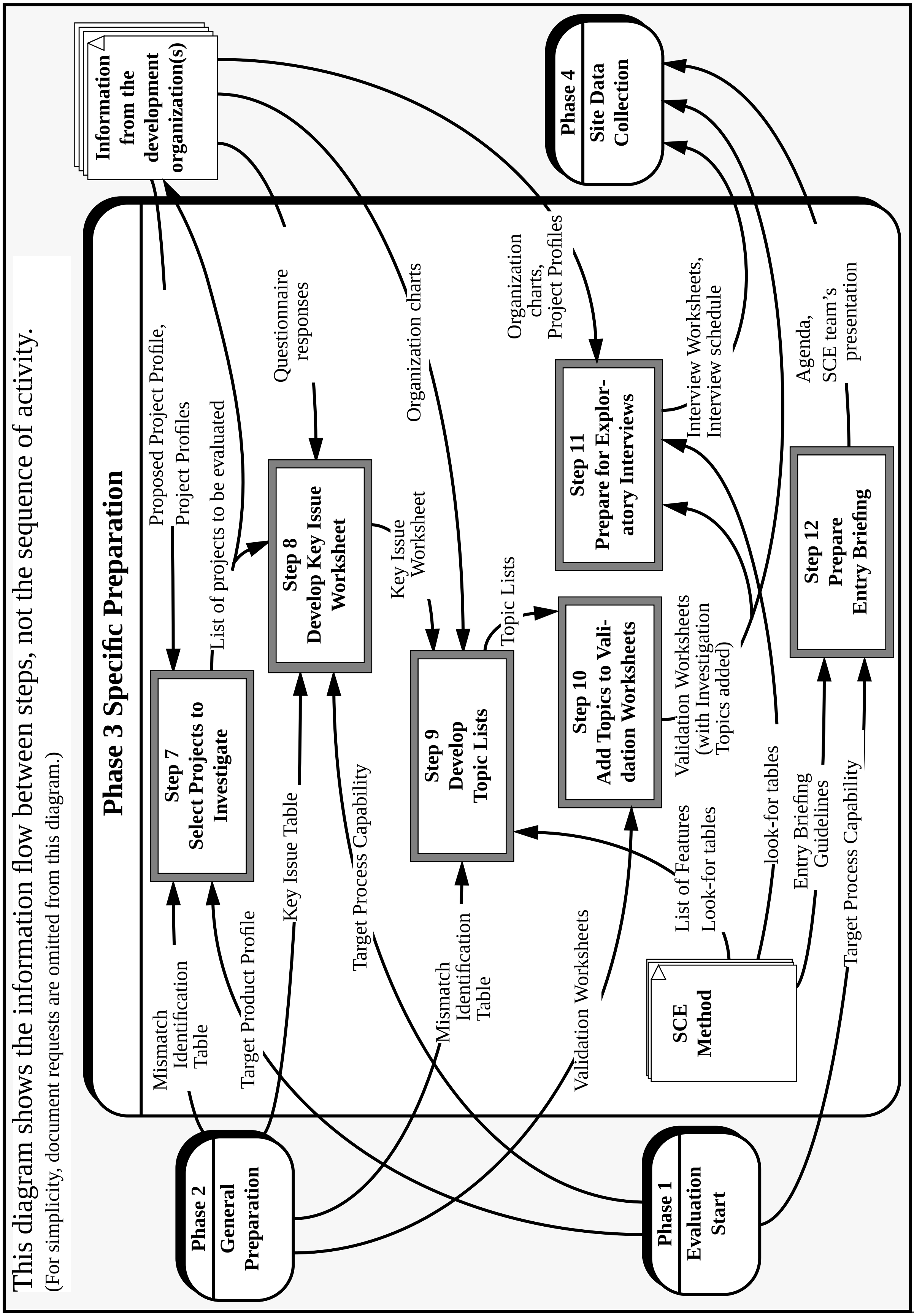
In the Specific Preparation phase, the SCE team completes detailed preparations for evaluating a development organization site. The activities in the Specific Preparation phase are repeated for each development organization being evaluated.

During the General Preparation phase, the SCE team decided which subprocess areas would be investigated at all of the development organization sites. During the Specific Preparation phase, the team translates those decisions into specific, detailed topics to be investigated at a development organization site.

The purpose of the Specific Preparation phase is to prepare the SCE team for a specific site visit. To achieve this, the SCE team selects projects to evaluate (Step 7), determines the key issues to be investigated (Step 8), and selects detailed topics for evaluation (Step 9). After selecting evaluation topics, the team records the topics on the Validation Worksheets (Step 10). The topics are used to plan the preliminary interview strategy and develop an interview schedule (Step 11). The interview schedule is closely coordinated with the development organization’s SCE site visit coordinator. The team also prepares an entry briefing (Step 12); the entry briefing is used to set the development organization’s expectations for the site visit.

During this phase the SCE team also identifies the documents for use during the initial document review and requests them from the development organization’s site visit coordinator. Other critical preparation activities include identifying the facilities the team will require during the site visit and arranging for their availability with the site visit coordinator.

Figure 2-5 on page 63 provides a high-level diagram of the steps in this phase.



This diagram shows the information flow between steps, not the sequence of activity.
(For simplicity, document requests are omitted from this diagram.)

Phase 3 Specific Preparation
Phase 2 General Preparation
Phase 1 Evaluation Start
Phase 4 Site Data Collection
Information from the development organization(s)
SCE Method
Step 7 Select Projects to Investigate
Step 8 Develop Key Issue Worksheet
Step 9 Develop Topic Lists
Step 10 Add Topics to Validation Worksheets
Step 11 Prepare for Exploratory Interviews
Step 12 Prepare Entry Briefing
Mismatch Identification Table
Target Product Profile
Key Issue Table
Target Process Capability
Mismatch Identification Table
Validation Worksheets
List of Features
Look-for tables
look-for tables
Entry Briefing Guidelines
Target Process Capability
Proposed Project Profile, Project Profiles
List of projects to be evaluated
Questionnaire responses
Key Issue Worksheet
Organization charts
Topic Lists
Validation Worksheets (with Investigation Topics added)
Organization charts, Project Profiles
Interview Worksheets, Interview schedule
Agenda, SCE team's presentation

Figure 2-5: Diagram of Steps in Phase 3, Specific Preparation

The table below provides an overview of the steps in this phase.

Phase	Step	Purpose	Page
Phase 3: Specific Preparation	7. Select Projects to Investigate	Select projects for evaluation that give the most insight into the processes that will be used.	page 64
	8. Develop Key Issue Worksheet	Create a consolidated list of key issues for investigation at the development organization site.	page 66
	9. Develop Topic Lists	Select topics for probing the process implementation; topics define observable work practices that map to the critical subprocess areas.	page 68
	10. Add Topics to Validation Worksheet	Capture the consolidated topic list for use at a particular site.	page 75
	11. Prepare for Exploratory Interviews	Develop detailed interview strategy, including the team's decisions on who will be interviewed, when they will be interviewed, and what they will be asked.	page 75
	12. Prepare Entry Briefing	Establish the agenda for the initial organization meeting and set initial expectations for the site visit.	page 78

Table 2-7: Overview of Phase 3

Step 7 Select Projects to Investigate

Input

The inputs to this section are

- The Target Product Profile from Step 1.
- The Mismatch Identification Table for the development organization from Step 4.
- The Proposed Project Profile from the development organization.
- The Project Profiles that were submitted by the development organization.

The Proposed Project Profile and the Project Profiles must be requested during Phase 1, Evaluation Start, and may be combined with requests for other documentation (see Information Request Timetable on page 121).

Action

The SCE team selects three or four projects for investigation whose attributes most closely match the planned development, as shown by the Proposed Project Profile. The Mismatch Identification Table that was created in Step 4 shows the matches by attribute.

The team first uses the major attribute matches to select projects for evaluation. If this does not reduce the number of projects to the required number, the team uses all of the available information about the projects to decide. Tiebreaking factors might include the following:

- Mismatches in minor attributes such as Language. For example, if Ada usage is mandated, a project with Ada experience would be preferred to one done in another language.
- The detailed attribute descriptions in the Proposed Project Profile and the Project Profiles. For example, if the Proposed Project Profile had a size estimate of 2,500 KSLOC, a project with 2,350 KSLOC might be preferred to one with 2,200 KSLOC, although the team might have recorded both as a match in the Mismatch Identification Table. (If the Proposed Project Profile is unavailable, the Target Product Profile from Step 1 can be used instead.)
- The schedule and status information in the Project Profiles. For example, a project that was completed three years ago is not a good choice for evaluation because the personnel may not be readily available for interviews, and a project that is still in the requirements phase would not be a good choice for evaluation if the planned development was solely a design and code effort.

Once the projects are selected, the SCE team requests documents for the initial document review (Step 14) from the development organization (see Information Request Timetable on page 121). The names for the documentation will vary from organization to organization, but preliminary identification of the documentation that will be reviewed is critical. Typically, the team requests copies of pertinent organizational ➠ **policies**, ➠ **standards**, ➠**procedures**, and ➠ **directives** relating to software development. The team also requests project-level procedures, standards, and directives for the projects selected for review. This documentation defines both the organization-level processes and the high-level processes used on the selected projects.

If they have not already done so, the team will request a completed questionnaire for each of the projects selected for evaluation (see Information Request Timetable on page 121).

Purpose

The purpose of this step is to select projects for evaluation that give the most insight into the processes that will be used on the planned development project. Because the team is interested in identifying risks pertinent to the processes that will be used on the planned development, it selects the projects that are most similar to the planned development. By evaluating the actual processes used on similar projects, the team obtains a clearer picture of the processes that will probably be used on the planned development.

Outcome The output of this step is a list of projects to be evaluated at the development organization site and requests for documentation from the development organization.

Notes In general, projects are selected from the same development organization site that will manage and develop the software, as described in Step 4.

As mentioned, each project selected for evaluation in Step 7 must provide documentation for review. This documentation will be examined during the initial document review (Step 14). In source selection, each development organization is given the same amount of time to prepare for the site visit. In this case, the timing of requests for documentation about the projects will be dictated by the site visit schedule (see Sample Site Visit Schedules on page 118).

Classified (so-called “black”) projects cause special problems. The team may not have the required clearance level to examine the project information, access to information will be much more difficult and time consuming, and special facilities may be required for the interviews. If the team will evaluate black projects, they must address these issues.

Step 8 Develop Key Issue Worksheet

Input The inputs to this step are

- The Target Process Capability from Step 2.
- The Key Issue Table from Step 5, (which includes the Critical Subprocess Area List for all development organizations).
- The development organization’s responses to the questionnaires for each of the projects to be investigated.

Action The SCE team identifies key issues for a development organization and integrates the information about that development organization on a single worksheet (the Key Issue Worksheet, see Appendix C.9 on page 174). This is done in two parts: (1) consolidating answers to the questionnaires from the projects, and (2) creating a consolidated list of key issues for the development organization.

Consolidating answers to the questionnaires. The team prepares an SCE Questionnaire Worksheet (see Appendix C.8 on page 171) to summarize the questionnaire responses from the projects selected for evaluation.[1]

[1] As of this version of the SCE method (SCE v2.0), teams are being trained to use a questionnaire based on CMM v1.1. As future versions of CMM-based questionnaires are developed, they will be incorporated into the SCE Method. During 1992 and 1993 SCE teams used the “Maturity Questionnaire” from *A Method for Assessing the Software Engineering Capability of Contractors* [Humphrey 87b].

Creating a consolidated list of key issues for the development organization. The key issues for a development organization are listed on a Key Issue Worksheet (see Appendix C.9 on page 174).

It is important to distinguish between the Key Issue *Table* (from Step 5) and the Key Issue *Worksheet* created in this step. There is one Key Issue Table, and it contains information about all of the development organizations. There is one Key Issue Worksheet for each development organization. The Key Issue Worksheet is used to focus on a development organization by consolidating all of the information known about the organization and relating it to the critical subprocess areas. The relationships define the key issues for the organization.

The key issues help to determine the level of investigation required for each critical subprocess area at a development organization site.

The critical subprocess areas are the same for each development organization, namely the Critical Subprocess Area List that was recorded on the Key Issue Table during Step 5. However, the information relating to those subprocess areas will differ from organization to organization.

A Key Issue Worksheet is created in three steps: (1) the team copies the Critical Subprocess Area List from the Key Issue Table to the Key Issue Worksheet, (2) the team copies the mismatch information specific to this development organization from the Key Issue Table to the Key Issue Worksheet, and (3) the team reviews the SCE Questionnaire Worksheet to identify inconsistencies and anomalies in the responses to the questionnaires, and records them on the Key Issue Worksheet.

An *inconsistency* is an apparently contradictory response from the same project to two (or more) questions on the questionnaire that relate to the same subprocess area. An *anomaly* is a contradictory response to the same question by two projects. An anomaly would be noted for both projects in the appropriate places on the Key Issue Worksheet. Both types of responses may indicate that the related key issues (critical subprocess areas) should be probed in depth.

When completed, the Key Issue Worksheet consolidates all of the information that is available about the key issues (critical subprocess areas) for a development organization. The information is captured in a form that can be used later to help prioritize the amount of time spent investigating each subprocess area.

Purpose

The purpose of these activities is to create a consolidated list of key issues for investigation at the development organization site, in a form that can be used later to help prioritize the amount of time spent investigating the issues. Although the same critical subprocess areas are investigated for all development organizations, each development organization has unique strengths and weaknesses. More time or effort should be spent investigating the key issues (critical subprocess areas) that correspond to known weaknesses, because they indicate potential risks for the planned development. The team's goal is *not* to validate the development organization's response to the questionnaires; rather, the goal is to investigate the related subprocess areas to identify strengths, weaknesses, and improvement activities.

Outcome The output of this step is the Key Issue Worksheet (used in Step 9), which guides the SCE team in selecting topics for investigation for the particular development organization site. An SCE Questionnaire Worksheet is also created, but is not used outside of this step.

Notes During the site visit, documentation will be reviewed for each project selected for evaluation (see Step 7; also see Document Review During an SCE Site Visit on page 108). Some teams request that the comments column of the questionnaires be annotated to indicate what documentation exists to support the answers to the questions. This information can be used to tailor the requests for documentation to be reviewed during the initial document review (Step 14 on page 85). If this was done, the documents for initial document review are requested at this time. The request that the questionnaires be annotated in this way should be made as early as possible because of the extra preparation time it requires.

Only subprocess areas within the KPAs in the Target Process Capability are used in the Key Issue Worksheet. If the team identifies inconsistencies or anomalies in other areas, they should document them and forward the information to the sponsoring organization. However, they will not be investigated as part of the SCE because they are beyond the scope of the SCE established during Phase 2, General Preparation.

Inconsistencies and anomalies can help to identify potential weaknesses within a subprocess area that should be investigated in more depth. However, even if the questionnaire responses are all "yes," indicating no inconsistencies or anomalies in a critical subprocess area, it is *not* sufficient grounds for removing a key issue (critical subprocess area) from consideration during the investigation of the development organization.

Step 9 Develop Topic Lists

Input Inputs to this step are

- The Mismatch Identification Table from Step 4.

- The Key Issue Worksheet from Step 8.
- The organization charts and information from the development organization.
- The list of subprocess area features shown below in Table 2-8 on page 71.
- The look-for tables from the SCE Method.

Action

For each critical subprocess area on the Key Issue Worksheet, the SCE team generates one or more topics for investigation. A topic defines a subject that will be probed during the investigation; topics are the level of detail at which an SCE is conducted.

A topic is an abstraction of a work practice that corresponds to a portion of the process implementation for the development organization. Topics are intended to be detailed enough to focus the investigation on observable, documented work practices, but sufficiently abstract that they avoid prescribing how the topic is implemented.

Each topic focuses on one feature within a subprocess area. *Features* are characteristics common to all subprocess areas; each subprocess area has the following features:

- *leadership* - the assignment of responsibility and the presence of sponsorship.
- *organizational policies* - there are written policies governing the subprocess area.
- *resources* - the adequacy of resources (e.g., staff, funds, facilities, tools).
- *organizational structures* - the organizational structure provides support for the process activities (e.g., job descriptions, defined relationships between entities on the organization chart).
- *training* - availability of pertinent training and orientation, and its timeliness for the people who carry out the activities in the implementation of the subprocess area (e.g., curriculum content, training schedule, records).
- *plans and procedures* - plans and procedures exist and are prepared according to a documented procedure.
- *work performed* - the objective evidence of the use of plans, procedures, and standards in the work done by the organization (i.e., the track record and “paper or electronic trail”).
- *tracking* - how the work is tracked and how problems are identified.
- *corrective actions* - the identification and resolution of problems.

- *measure process* - the measurements of activities performed (e.g., resources consumed, problems encountered, work product characteristics, and status of activities).
- *analyze measurements* - the analysis and use of measurements taken.
- *reviews* - management reviews.
- *audits* - there are audits undertaken of activities and work products.

Features indicate whether the implementation and institutionalization of a key process area is effective, repeatable, and lasting [Paulk 93]. (The features used in SCE were derived from the common features defined in CMM v1.1. Appendix A.5 on page 142 shows the relationship between the common features used in CMM v.1.1 and the features used in SCE.) A feature as applied to a subprocess area constitutes a topic for investigation. The SCE investigation uses document review and interviewing to probe the development organization's process implementation for these features. If these features exist for a subprocess area, the team can conclude that the development organization has implemented the subprocess area.

The look-for tables provide information that can be useful during topic selection. Recall that topics are the level of detail at which the SCE is conducted and that the two primary means for collecting data are interviewing and document review. In order to generate topics on which to base their interviews and document reviews, SCE teams will need to know whom to interview, what kinds of documents to review, and what kinds of activities to look for. The look-for tables help provide this guidance. There are three kinds of look for tables. The first of these tables uses the terms *agents*, *artifacts*, and *relationships* to help teams identify people, documents, and activities. A partial example is shown below.

Key Process Area: Software Project Planning

Subprocess area: Develop estimates
Action: Develop documented estimates

Examples of Agents	*Examples of Artifacts*
Senior software engineers Software managers System engineers Testing managers	Allocated requirements Estimate package (e.g., bases of estimate with assumptions, task descriptions, labor spread over schedule)
Senior software engineers Software managers Accounting staff	Productivity coefficients and parameters Historical database Estimation tool Cost package

Examples of relationships between agents and artifacts[1]

- *Senior software engineers* have estimated the software components to be built and included the type of effort for each component in the *estimate package*.
- *Senior software engineer* analysis of the development work, and use of level of effort historical data, are recorded on *task descriptions*, a *basis of estimate* for each task, together with an assessment of type of *labor required over the proposed schedul*e.
- *Senior software engineers* are involved in selecting the parameters that are appropriate for the development of the *cost package*.
- *Accounting staff* has an *estimation tool* and/or an equivalent process to translate size estimates into cost estimates.

Table 2-8: Example Look-For Table Showing Agents, Artifacts, and Relationships

An *agent* is a defined role; an agent may perform an activity, provide an input for the activity, receive the output, define how the activity is to be performed, or verify that the activity is performed. *Artifact*s are the work products which are part of the activity. Artifacts may be either process artifacts or product artifacts. *Relationships* indicate the roles agents and artifacts play in the activities performed by an organization, and serve as examples of possible activities that teams might look for.

During topic selection, teams should also consider the guidance found in the “probing guides” tables available in SCE team training. The guidance in these tables contains many examples of activities that teams might observe. The guidance is phrased in terms of the end results needed for findings, but the

[1] The examples of relationships are just that—examples. Other relationships may exist; therefore, the relationships shown in this table may not be the same as the ones that are observed.

tables provide a useful perspective to consider when selecting topics for the investigation that will lead to those findings. An example of a “probing guides for key process areas” table is shown below. (The guidance in the tables is based on the key practices of the CMM v1.1 [Paulk 93b].)

Key Process Area: Software Project Planning	
Subprocess Area	**Develop estimates**
Goal	Software estimates are documented for use in planning and tracking the software project.
Probing Guides	Evidence exists that there is a defined process for deriving and recording the estimates used in software planning: • Software product size, cost/effort, critical computer resource, and schedule estimates are derived. • Risks associated with estimates are identified. • This software planning data is recorded.

Table 2-9: Example Look-For Table Showing Probing Guides for a Key Process Area

A rule of thumb for topics is that they can be transformed into an open-ended question that can be answered readily by a person or document. For example, consider the question “What are the procedures used to develop software size estimates?” This question investigates the “plans and procedures” feature within the “develop estimates” subprocess area of the Software Project Planning KPA. Similarly, to investigate the Tracking feature for the same subprocess area, the team might ask the question “What mechanism(s) ensure the software size estimating procedures are followed?” An example of topic selection is included below in the *Notes* paragraph of this step.

Working individually, each team member decides which features should be investigated to validate the subprocess area for this particular development organization. The team then caucuses to develop a single, consolidated topic list that represents team consensus. All of the available information is used to create the consolidated topic list, including the information on the Key Issue Worksheet, organization charts, and the individual expertise of the team members.

The individual topic lists are merged into a single, consolidated team topic list using any method preferred by the team; the goal is consensus. Consensus means that the result is acceptable to all team members, but not necessarily optimal.

Purpose

The purpose of this step is to select topics for probing the process implementation; topics define observable work practices that map to the critical subprocess areas. A subprocess area is too broad to be directly observable;

each topic defines an observable work practice. Topics help the SCE team to structure the investigation, providing consistency across subprocess areas and KPAs. Each topic serves as the basis of a set of questions that the team can ask of an interviewee to probe the implementation of a subprocess area. Topics also focus the documentation review.

Outcome

The output of this step is a single list of topics for each critical subprocess area, used to generate interview questions in Step 11 and to guide document review in Steps 14 and 17.

Notes

Topics are the level at which the actual SCE investigation is conducted. Before selecting topics, considerable preparatory activity occurs and several interrelated products are produced. Here is a quick summary of these products:

- The Target Process Capability defines the KPAs to be investigated.
- The Critical Subprocess Area List applies to all of the development organizations and adds another level of detail to the KPAs in the Target Process Capability.
- The Key Issue Worksheet tabulates all of the information available about a development organization and arranges it by subprocess area on the Critical Subprocess Area List.
- The features common to the subprocess areas are used to build a consolidated topic list for the actual investigation.

Here is a hypothetical example that describes how topics might be selected for “Organization A,” starting with the first step in the SCE and carrying it through the preparatory steps discussed so far:

1. In Step 2, the Target Process Capability was selected. The decision was made to use the default Target Process Capability for this development. One of the KPAs in the Target Process Capability is Software Project Planning.
2. During Step 4, a mismatch was noted for Organization A in the Size attribute. When the Critical Subprocess Area List was created in Step 5, one of the subprocess areas selected as critical was “develop estimates.” This subprocess area was selected as critical because of the mismatched attribute.
3. When the Key Issue Worksheet was created in Step 8, the team noted the Size attribute mismatch for organization A, and also noted an anomaly between projects in response to questions about software size and cost estimation for Organization A.
4. In Step 9, the team created the topic list. First, the team members created individual lists of topics.

- One member selected the *plans and procedures* and *work performed* features for investigation, and developed topics based on these features.
- Another team member selected *organizational policies*, *training*, *work performed*, and *reviews*.
- A third member selected *plans and procedures*, *work performed, tracking* and *reviews.*
- When the team caucused, they created a combined topic list using these five features: *organizational policies, plans and procedures, training*, *work performed* and *reviews*.

In this example, five topics were selected because the team viewed the subprocess area as very important, especially in view of the factors noted on the Key Issue Worksheet. The *tracking* feature was dropped because it was not considered as good a topic as the others to start the investigation with.

The SCE team must select topics that provide the most significant information for the purposes of the SCE; it is not possible to investigate all of the topics because of the limited amount of time spent on site. There are 37 subprocess areas within the default Target Process Capability. Each has 13 possible features for investigation,[1] which gives a total of 481 possible topics. A typical 3-day site visit has between 19 and 21 hours for interviews and document review (for sample schedules, see Table 2-13 on page 118 and Table 2-14 on page 119). If all 481 topics were investigated, this would allow less than 3 minutes per topic. Topic selection is a critical activity; the team must balance adequate coverage of the critical subprocesses against the overwhelming amount of information that could be examined.

Individual expertise of the team members in a subprocess area might reduce the number of topics the team would have to consider within the subprocess area.

Topics selected may vary between development organizations, based on questionnaire responses (as summarized on the Key Issue Worksheet), information about the organization’s structure, and the mismatches on the Mismatch Identification Table. For example, if a development organization has a very well defined software size estimation method, a separately staffed functional area within the organization dedicated to performing size and cost estimates for all projects, and experience with projects of similar size to the planned development, then the team might check only for work performed. Because of mismatches, at another development organization the team might

[1] This assumes that all of the features are applicable, and at least one topic can be derived from each feature.

check for organizational policies, documented software size estimation procedures, training in the software size estimation process, the track record of the actual work performed, and whether management reviews were done.

Step 10 Add Topics to Validation Worksheet

Input The inputs to this step are

- The Validation Worksheets initiated in Step 6.
- The consolidated list of topics from Step 9.

Action The SCE team adds the topics from the consolidated list of topics to the Validation Worksheets for the corresponding subprocess areas. The Validation Worksheets are now ready for use during the site visit. Appendix C.7 on page 169 shows an example of a Validation Worksheet with topics included.

Purpose The purpose of this step is to capture the consolidated topic list for use at a particular site. During the site visit, the SCE team members will use the Validation Worksheets to document their observations about each topic. The worksheets offer a convenient way to consolidate information about the selected topics for each subprocess area.

Outcome The output of this step is an updated set of Validation Worksheets; each includes the topics, critical subprocess areas, and KPAs that will be investigated for a development organization. These are used throughout the Site Visit, and also in Step 11.

Step 11 Prepare for Exploratory Interviews

Input The inputs to this step are

- The Validation Worksheets that were updated in Step 10.
- The Project Profiles from the development organization.
- The organization charts and information collected from the development organization during the General Preparation phase.
- The look-for tables from the SCE method.

Action The team develops a high-level interview strategy and prepares materials to guide them during the interviews. This activity has four major components: (1) allocating time for the site visit, (2) selecting interviewees, (3) creating interview worksheets, and (4) coordinating the interview schedule.

Allocating time for the site visit. Based on the topics from the Validation Worksheets, the team estimates the amount of time needed for interviewing and document review; this translates into how much time the team needs to allocate to the site visit.

Selecting interviewees. The Validation Worksheet topics and the information about the organization's structure is used to decide who will be interviewed about each topic. Interviewees are not selected as individuals, but instead by position in the organization or by their functional area (e.g., CM, SQA, project manager). The look-for tables contain examples of agents likely to fill a given role with regard to a particular subprocess area.

Creating interview worksheets. Interview Worksheets are prepared with questions derived from the topics for each interviewee; this keeps the interview focused (a sample Interview Worksheet is shown in Appendix C.10 on page 177). The look-for tables contain examples of artifacts that may be used to document the processes in use, and also have examples of many process activities. This information can help team members develop focused questions about a particular subprocess area.

Coordinating the interview schedule. Finally, the team decides the preferred order for the interviews and coordinates with the development organization's site visit coordinator to set up an interview schedule.

Another critical activity that must be performed by the team at this time is arranging any other factors relating to the visit with the site visit coordinator. The team must also arrange for access to the facility, adequate working space, a conference room, telephone and copier access, and so on. The documents for initial document review were specified previously (in Step 7), but the team should ensure that the documents will be available in the working space assigned to them. (Some teams have developed logistics checklists or worksheets to help with this planning effort.)

Purpose

The purpose of this step is to develop a detailed initial interview strategy, which should include the team's decisions on who will be interviewed, when they will be interviewed, and what they will be asked. The Interview Worksheets help the team to organize and plan the interview strategy. The worksheets help focus the team on the relevant issues during the interview, increasing the chances of gathering the relevant information during the interview. A second purpose is to make the final arrangements for the site visit.

Outcome

The outputs of this step are completed Interview Worksheets for each interviewee, used during Step 15, and a coordinated interview schedule for the site visit.

Notes

Interviewing is a learned skill—interviews are difficult to conduct and manage. Interview worksheets do not guarantee a successful interview; however, they help to ensure coverage of all the topics.

This activity has four major components: (1) allocating time for the site visit, (2) selecting interviewees, (3) creating interview worksheets, and (4) coordinating the interview schedule. Each is discussed in more detail below.

Allocate site visit time. The team decides how much time it will allocate for interviews and document review. The strategy depends on unique circumstances for each development organization. If, for example, organizational policies, procedures, roles, and responsibilities are easy to identify from documentation, the team may require less interview time and may prefer to spend time gathering "audit trail" information. On the other hand, if documentation is complex or unorganized, more interview time may be needed to clarify the organization's processes. The team must be able to react to unforeseen circumstances.

Select interviewees. For each topic from the Validation Worksheets, the team selects interviewees from the development organization. The selection is denoted by organizational function (e.g., project manager, project engineer) or unit, not by the individual's name. More than one person may be interviewed about a single topic, and one person may be asked about multiple topics. The team may not be sure who to ask, because they don't know the organization well. The strategy in that case is to "ask whom to ask" at the most senior level appropriate to the topic, and conduct a follow-up interview with the indicated person.

The look-for tables contain examples of agents likely to fill a given role with regard to a particular subprocess area. This can be a starting point for selecting interviewees, but teams should remember that the job titles will vary from organization to organization.

Create interview worksheets. For each interviewee, the team creates a worksheet, which identifies the interviewee by role (e.g., Project SQA Staff). For each topic to be addressed to that interviewee, the team generates questions that are relevant to the interviewee's role in the organization. The questions should validate that topic or indicate organizational documents for review. For each question, the team also notes the related KPA, subprocess area, and topic on the worksheet; this helps when transferring the information back to the Validation Worksheet.

For topic refinement into questions, the SCE teams can use the probing guides found in the look-for tables as a starting point. The probing guides are "generic" statements about what to look for relative to software processes. The teams can also use the examples of agents, artifacts, and relationships.

Coordinate interview schedule. The team leader coordinates with each site visit development organization to set up a schedule that is as convenient as possible to all parties. To maintain fairness, each development organization must be given the same amount of time to prepare for their site visit. Therefore, the exact schedule for a development organization will not be determined until that development organization has been notified of the site visit dates (the interview schedule may change slightly as a result of conflicts). There are two strategies for establishing the interview schedule: working “top down” through the organization, and interviewing project by project. Examples of both of these strategies are shown in the sample site visit schedules (Table 2-13 on page 118 and Table 2-14 on page 119). These strategies are often combined to develop the actual schedule.

Step 12 Prepare Entry Briefing

Input

The inputs to this activity are

- The Target Process Capability from Step 2.
- Entry briefing guidelines for the development organization’s briefing to the team (described below in the *Notes* section),
- Coordination contacts between the SCE team leader and the organization’s site visit coordinator.

Action

The team prepares an entry briefing and negotiates an agenda for the Initial Organization Meeting (Step 13 in Phase 4, Site Data Collection).

This activity requires extensive coordination with the development organization’s site visit coordinator. Collaboration on the on-site agenda will help the development organization receiving the SCE be more comfortable with the process.

The team must decide what information they will provide to the development organization to prepare for the site visit. The Target Process Capability should be presented to the development organization so they will understand the boundaries of the investigation at the KPA level.

The team must also decide what guidance to give the development organization about their presentation to the team and provide appropriate guidance to the site visit coordinator.

Purpose

The purpose of this step is to establish the agenda for the Initial Organization Meeting (Step 13 in Phase 4, Site Data Collection) and set initial expectations for the site visit. The team also creates their entry briefing to present to the development organization.

Outcome

The SCE team's presentation will be prepared. The agenda for the Initial Organizational Meeting will be collaboratively developed with the development organization. Both are used in Step 13, Conduct Initial Organization Meeting.

Notes

This step marks the end of the Specific Preparation phase, and sets the stage for the Site Data Collection phase activities.

When developing the agenda, there are two concerns: what the team will present and what the team expects from the development organization's presentation. The total length of the initial meeting should be no more than 60–90 minutes.

The team's briefing will set expectations for the on-site period. Topics may include introducing the team members, describing the major on-site activities, discussing how interviews will be conducted, how (or if) the team will present their findings to the development organization, and a short question and answer session. The team briefing should be standardized for all development organizations.

The SCE team may suggest that the organization demonstrate their processes or give a presentation of their methods.

Here are the entry briefing guidelines for the development organization's presentation to the team.

The development organization should explain to the team

- What the organization does (without giving a "marketing pitch" or an in-depth recital of their standard processes).
- The organizational structure, (who does what), especially any changes that have occurred since the initial organization charts and information that was provided in Step 4.
- How responsibility, accountability, and authority are managed, particularly in regard to such items as software configuration management, software quality assurance, integration and test, requirements definition and management, systems test, and software development.
- How the organization's process integrates responsibility, accountability, and authority through the development life cycle; the organization's description should be focused on the projects selected for review.
- The ➠ **organization-level documents** (policies, procedures, etc.) and present a roadmap of how the documents are organized.

Summary of Phase 3 Specific Preparation

When the Specific Preparation phase is finished, the SCE team will be ready to perform the Site Data Collection phase activities. The team will have determined what topics will be investigated (and to what level), whom they need to talk to, what questions they need to ask during exploratory interviews, and which documents they will review at first. The development organization will have prepared the facility for the team, will have the requested project and organization documentation on hand, and will have ensured that the interviewees are available.

This phase further refines the scope of the SCE by using the critical subprocess areas to develop topics that address the development organization's observable work practices. In Step 7, the team selects projects for evaluation that provide the most insight into the processes that are likely to be used during the proposed development. Next, in Step 8, Key Issue Worksheets are developed that identify the key issues that need to be investigated at this development organization site. The key issues are used to guide selection of investigation topics in Step 9. Topic selection is a critical activity; the team must balance adequate coverage of the key issues against the overwhelming amount of information that could be examined. During Step 10, the team documents the selected topics on the Validation Worksheets for use during the site visit and also for use when the team develops the detailed interview strategy in Step 11. Finally, the team sets expectations for the upcoming site visit by preparing an entry briefing (Step 12) and coordinating the agenda for the Initial Organization Meeting (Step 13 in Phase 4).

During this phase the SCE team also identifies the documents for initial document review and requests them from the development organization (see Information Request Timetable on page 121). The requests are made after projects are selected for review in Step 7. The team coordinates the documentation requests with the SCE site visit coordinator.

Another critical preparation activity is identifying the facilities the team will require during the site visit and arranging for their availability. It takes considerable time and effort to coordinate an SCE with the development organization.

As mentioned before, thorough preparation is essential because the amount of information to be considered during the Site Visit can be overwhelming.

2.5 Phase 4: Site Data Collection (Site Visit)

The Site Data Collection phase is the crux of the SCE Method. During the Site Data Collection phase, the SCE team investigates the processes at a development organization site.

The purpose of Site Data Collection is to investigate the topics associated with each critical subprocess area in enough depth to determine the strengths, weaknesses, and improvement activities for the corresponding subprocess area. This is the most complicated and intense activity phase during an SCE.

The team presents the entry briefing that was prepared during Phase 3 to the development organization during the initial organizational meeting (Step 13). After setting expectations for the site visit, the team starts the data collection activities. Site data collection has two basic components: investigation and decision making about the information collected. These components are applied iteratively until a decision has been made about each topic under investigation; this is summarized visually in Figure 2-6. (Figure 2-6 was introduced earlier as Figure 1-3 on page 17.)

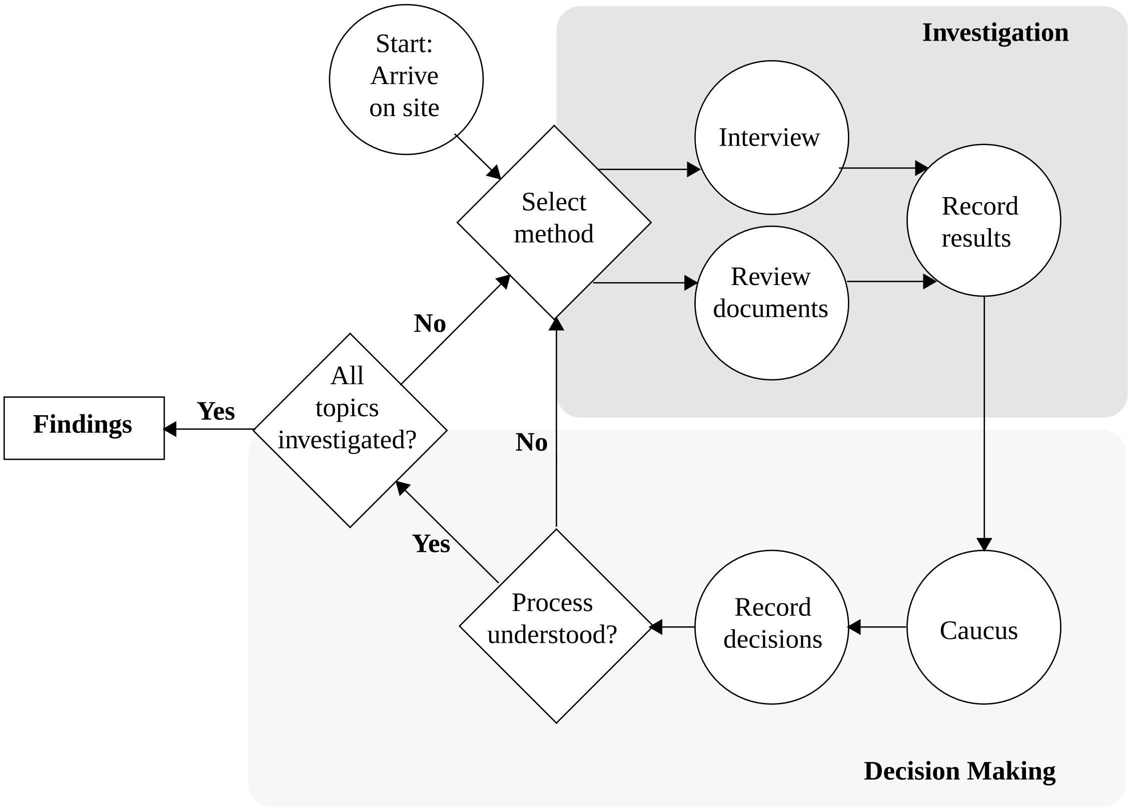


Figure 2-6: A Flow Chart of the Site Data Collection Activities

The SCE team uses two complementary mechanisms to investigate a topic: document reviews (Steps 14, 17, and 21) and interviewing (Steps 15 and 20). The team members record the results of their investigations into each topic for use in decision making. Decisions are made by consensus in an ongoing team caucus (Step 16). In caucus, the team asks the question “Do we have enough information to reach a consensus about this topic yet?” If the team agrees that there are at least two pieces of evidence supporting the decision, the decision is documented as a ➠ **preliminary finding** (Step 18). If the evidence is not conclusive, a new round of interviewing and/or document review is planned (Step 19) and initiated. The preliminary findings are the source documents used in the Findings phase.

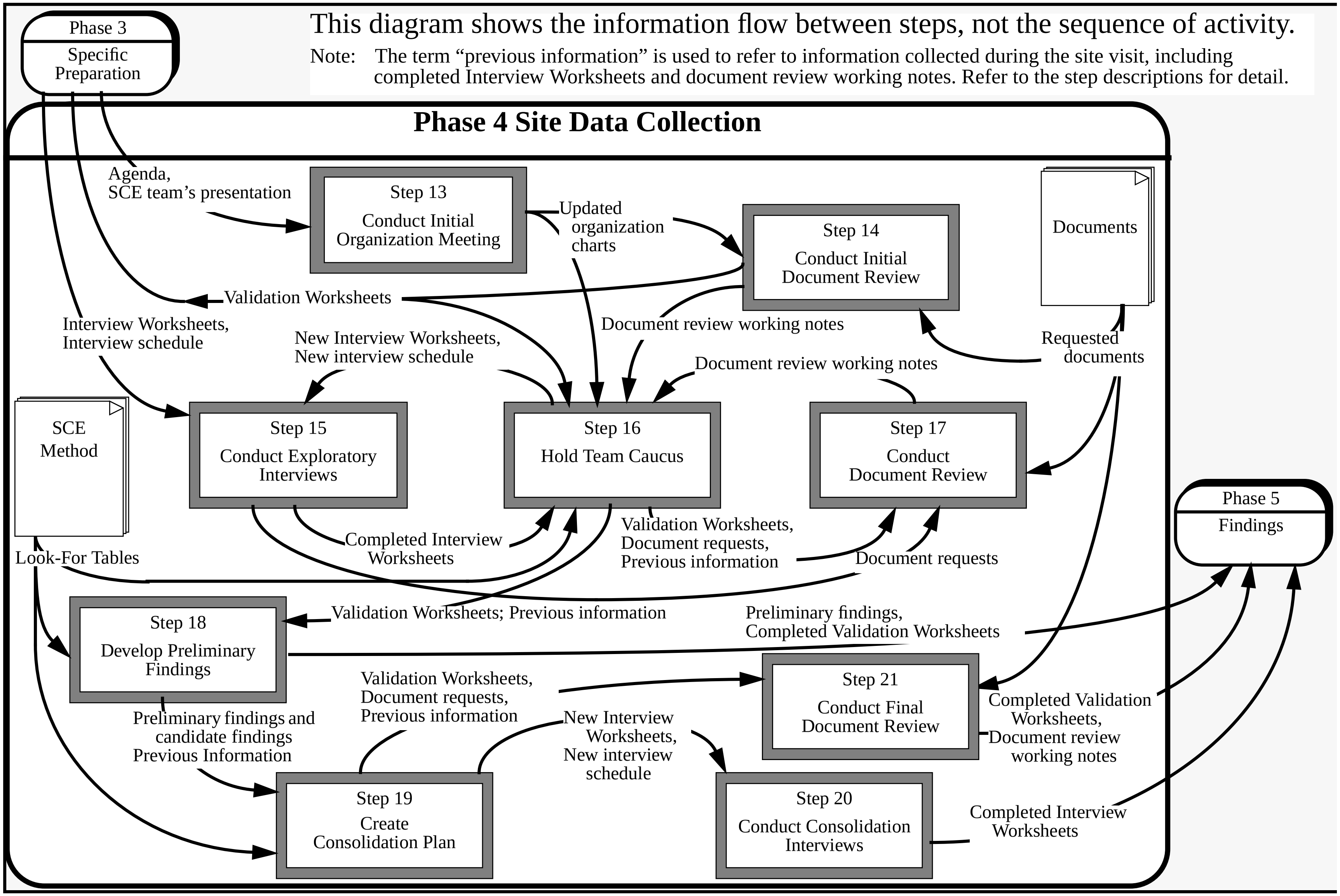
The steps listed above are *not* strictly sequential; document reviews are interspersed with interviews, and the consensus process is ongoing throughout the site visit. The different interviewing and document review steps listed reflect different data collection emphases at different points during the site visit.

Because of the iterative, interlaced nature of the fundamental activities during this phase and their central importance to several of the steps, a consolidated description of the basic concepts of document review and interviewing (as applied to the SCE Method) is provided in Section 2.7 on page 108 and page 115, respectively. This description provides a framework for the activities performed during the individual, related steps of the method and supplements the discussion provided within the descriptions of the steps in this section.

The team will request additional documentation for review throughout the Site Data Collection phase. These requests must be coordinated with the development organization’s site visit coordinator.

When the Site Data Collection phase is complete, the SCE team members are ready to generate their consolidated findings. The information recorded during Site Data Collection is the support for the findings.

Figure 2-7 on page 83 provides a high-level diagram of the steps in this phase.



This diagram shows the information flow between steps, not the sequence of activity.

Note: The term “previous information” is used to refer to information collected during the site visit, including completed Interview Worksheets and document review working notes. Refer to the step descriptions for detail.

Phase 3
Specific Preparation
Phase 4 Site Data Collection
Agenda, SCE team’s presentation
Step 13
Conduct Initial Organization Meeting
Updated organization charts
Step 14
Conduct Initial Document Review
Documents
Validation Worksheets
Interview Worksheets, Interview schedule
New Interview Worksheets, New interview schedule
Document review working notes
Requested documents
Document review working notes
SCE Method
Step 15
Conduct Exploratory Interviews
Step 16
Hold Team Caucus
Step 17
Conduct Document Review
Phase 5
Findings
Completed Interview Worksheets
Validation Worksheets, Document requests, Previous information
Document requests
Look-For Tables
Step 18
Develop Preliminary Findings
Validation Worksheets; Previous information
Preliminary findings, Completed Validation Worksheets
Validation Worksheets, Document requests, Previous information
Step 21
Conduct Final Document Review
Completed Validation Worksheets, Document review working notes
Preliminary findings and candidate findings Previous Information
New Interview Worksheets, New interview schedule
Step 19
Create Consolidation Plan
Step 20
Conduct Consolidation Interviews
Completed Interview Worksheets

Figure 2-7: Diagram of Steps in Phase 4, Site Data Collection

The table below provides an overview of the steps in this phase.

Phase	Step	Purpose	Page
Phase 4: Site Data Collection	13. Conduct Initial Organization Meeting	Clarify expectations of the SCE site visit.	page 84
	14. Conduct Initial Document Review	Determine the degree to which the organization and project-level documentation define and support standard processes for the KPAs and subprocess areas under investigation.	page 85
	15. Conduct Exploratory Interviews	Provide insight into how the subprocess areas are implemented in practice; determine the extent that processes have been internalized by the development organizations; identify critical implementation-level documents.	page 87
	16. Hold Team Caucus	Analyze, share, and consolidate information in order to reach conclusions about topics.	page 88
	17. Conduct Document Review	Search for objective evidence of how processes are implemented at the working level.	page 89
	18. Develop Preliminary Findings	Articulate conclusions about the subprocess areas based on the information available; guide subsequent information gathering efforts.	page 91
	19. Create Consolidation Plan	Plan and initiate further data collection.	page 94
	20. Conduct Consolidation Interviews	Clarify any remaining issues by confirming or negating candidate findings through further interviews.	page 95
	21. Conduct Final Document Review	Clarify any remaining issues by confirming or negating candidate findings through further document review.	page 96

Table 2-10: Overview of Phase 4

Step 13 Conduct Initial Organization Meeting

Input

The inputs to this meeting are

- The agenda from Step 12.
- The SCE team presentation from Step 12.
- The organization charts and information from the development organization (not shown in the diagram on the previous page).
- The development organization’s presentation (not shown in the diagram on the previous page).

The organization charts and information from the development organization must be requested during Phase 1, Evaluation Start, and are first used during the General Preparation phase (see Information Request Timetable on page 121).

Action Using the agenda set in Step 12, the team describes to the development organization personnel what it hopes to accomplish and what ground rules apply, and briefly introduces the team members. The development organization then makes its presentation to the SCE team. The team updates any organization charts and information with the latest information.

Purpose The purpose is clarifying expectations of the SCE site visit for both parties. The team gains information about the organization, and the development organization gains information about the team's purpose and method. Building a good working relationship is critical for the success of the site visit; this meeting sets the tone.

Outcome The direct outputs are updated organization charts and information, used throughout the remaining steps in the site visit. Expectations for on-site SCE process and SCE schedule are established.

Notes At this time, the team should confirm that the previously negotiated arrangements for facilities have been made correctly (e.g., working space, meeting rooms, telephone access), that requested documentation is available, and that the right people are available for preliminary interviews. These items were requested from the site visit coordinator during the Specific Preparation phase.

Step 14 Conduct Initial Document Review

Input Inputs to this step are

- The Validation Worksheets from Step 10 and the Interview Worksheets from Step 11.
- Documents for initial document review.
- Updated organization charts and information from Step 13.

The documents for initial document review were requested during Phase 3, after Step 7 (see Information Request Timetable on page 121). These include both project- and organization-level documents. The request must be previously coordinated with the development organization's site visit coordinator (during the Specific Preparation phase) so that the documents will be readily available in the team's assigned work area.

Action

The team examines each item on their Interview Worksheets to determine what information the documents can provide. The Interview Worksheet topics were extracted from the Validation Worksheets in Step 11; the Validation Worksheets are also a source of document review topics.

The team then reviews the initial document set and the organization charts and information provided in Step 13. Initial document review is focused on organization-level documents and high-level project documents.

During the initial document review, the team gains further insight into each scheduled interviewee’s proper role in the organization’s operations. Information is collected informally, as document review working notes. During the team caucus (Step 16), the team can use the information collected to modify the questions it has prepared on the Interview Worksheets and to modify the interview schedule.

Initial document review is usually done before starting interviews, but may be interspersed with interviews.

Purpose

The purpose of this step is for the team to determine the degree to which the organization-level documents and ➠ **project-level documents** define and support standard processes for the KPAs and subprocess areas that are under investigation.

From this activity, the team gains a better understanding of the development organization’s organizational structure and process, and is better prepared for exploratory interviews. By providing further insight into the policies and procedures that guide the organization’s processes, the team can sometimes eliminate the need for a question during the interviews or sharpen the focus for a question.

Another objective of this step is to identify documents from the development organization that do not have a clear purpose; this lets the team seek clarification through the site visit coordinator or from an organization employee during interviews.

Outcome

The direct output is a set of working notes from the informal document review. The team will understand the purpose and content of each relevant document and the document’s relation to the topics that the team wants to evaluate. The subsequent interviews will be better focused; the team should have a better idea of which employees to interview about each topic and what to ask them.

Notes

Guidelines for document review are provided in Document Review During an SCE Site Visit on page 108.

Occasionally, some topics may be partly or fully validated through the initial document review (remember, validation requires team consensus that at least two pieces of evidence support the finding).

Three levels of documents are reviewed: organization-level documents, project-level documents, and implementation (or track record) documents. Initial document review focuses on only the first two levels. The “track record” documents are primarily reviewed in Steps 17 and 20. However, review of organization- and project-level documentation is not limited to the initial document review period.

Step 15 Conduct Exploratory Interviews

Input The inputs to this step are the Interview Worksheets and the interview schedule (developed in Step 11, or updated in Step 16).

Action The interviews are conducted as listed on the interview schedule.

One strategy is for the team to interview the organizational employees with responsibilities at the organization level, and then the employees with responsibilities at the project level. This is a “top down” strategy. An alternative strategy is to interview project by project and follow up by interviewing people with organization-level responsibilities. Both strategies are represented in the sample site visit schedules provided in Section 2.7 (page 118 and page 119, respectively).

This step is part of an iterative process that includes the ongoing Team Caucus and Document Review (Steps 16 and 17). The team members note all relevant information on the Interview Worksheet; the caucus is used to consolidate, corroborate and reach consensus on the information.

Purpose Exploratory interviews serve these purposes:

- Provide insight into how the subprocess areas are implemented in practice.
- Determine the extent to which processes have been internalized by the development organizations.
- Identify critical ➠ **implementation-level documents**.

Outcome Information is gathered from the interviews and recorded on the Interview Worksheets, where the information can be used to validate topics. A list of documents to be reviewed is also created and is used to request and locate the implementation record of the development organization’s processes.

Notes Guidelines for interviewing are provided on page 115.

Interviews help the team determine the extent to which the documented procedures and policies have been implemented throughout the projects. By asking project personnel about specific practices (e.g., design and code reviews), the team can evaluate whether the organization and project-level policies and procedures have been communicated to the people who need to implement them and if they are understood.

Exploratory interviews also point the SCE team to the implementation-level documentation for a project and guide the document review at that level. This documentation is used to validate both the interview responses and the higher level procedures during subsequent document reviews.

Every piece of information obtained during an interview can lead to identification of a strength, a weakness, or an improvement activity. Before the information can become part of the SCE findings, it must be validated against the track record documents (see Step 17).

Step 16 Hold Team Caucus

Input

The inputs to this step include all of the information gathered during previous information gathering efforts, such as

- Updated organization charts and information (from Step 13).
- Document review working notes (from Steps 14 and 17).
- The Interview Worksheets (from Step 15).
- The Validation Worksheets (from Step 10).
- The look-for tables from the SCE method.

Action

This is the decision making step in the iterative information gathering and decision making process. During caucus, the team assesses their progress toward the goal of validating topics by evaluating the information gathered so far. No particular format is specified for the caucus, but the following steps are typical:

- The team reviews the topics that were the focus of the most recent investigations.
- The team reviews any new information, and identifies areas that require further clarification.
- If the team consensus shows that the information is sufficient for a preliminary finding, the preliminary finding is appropriately defined and then entered on the Validation Worksheet for review during Step 18, Develop Preliminary Findings.
- If the team cannot reach consensus, they identify information that will settle the outstanding issues.

- If enough information has been gathered to make a determination of “no finding” about a topic, it is dropped from further consideration. (For example, if no subcontractors are used on the projects, findings related to subcontractor management would not be applicable, leading to a determination of “no finding.”)

The look-for tables provide guidance the teams may reference during the caucus. The benefit of the look-for tables is that they associate specific activities that the team may have observed with particular KPAs and subprocess areas in a concise format. Properly associating the information gathered with the appropriate subprocess area and KPA is essential for the accuracy of the findings.

Purpose The purpose of the team caucus is to analyze, share, and consolidate the available information in order to reach conclusions about the topics. The SCE team gathers a large quantity of data; caucusing helps the team sift through the information. Caucusing also provides a chance for the team to share diverse perspectives on the data, which helps prevent misinterpretations and premature decisions. The caucus keeps the team focused on the objectives of the SCE.

Outcome Direct outputs from the caucus include annotations of information about the topics on the Validation Worksheets, requests for documentation, new Interview Worksheets, and updated interview schedules. (The requests for documentation and updated interview schedules must be coordinated with the development organization’s site visit coordinator.) This information is used to generate preliminary findings in Step 18. Any or all of several outcomes are possible after a particular team caucus:

- The team reaches consensus on strengths, weaknesses, or improvement activities based on the available information and annotates the information on the Validation Worksheets.
- The team validates one or more topics based on what was heard or seen since the last caucus.
- The team identifies a need for more data to confirm or negate an observation about one or more topics. This may generate a request for additional documentation to review or for further interviews.

Notes This step occurs throughout the site visit. Successful caucusing depends on the team’s consensus-building ability.

Step 17 Conduct Document Review

Input The input to this step is the information gathered previously through interviews and document review activities, such as

- The Interview Worksheets (originally generated in Step 15).

- The documents to be reviewed from the development organization, requested in Steps 15 and 16.
- Document review working notes (from previous iterations of document review).
- The Validation Worksheets.

Action

The team reviews project-level and implementation-level documents for a project to validate information gathered through other sources such as interviews and higher level document review. The topics on the Validation Worksheets and the results of the interviews as recorded on the Interview Worksheets are used to focus the review.

Informal document review working notes are kept to use during the caucus; the relevant information is entered onto the Validation Worksheet after caucusing (Step 16).

Documents on this level provide an audit trail of the processes used and the work performed. Through these reviews, the team confirms or negates the proposition that the actual work practices implement the processes described in the organization- and project-level documents.

Purpose

The purpose of this step is to search for objective evidence of how the processes are implemented at the working level—this provides support for findings. In other words, the team determines whether the processes defined on paper and elicited from the interviews correspond to what the people on the projects are actually doing.

Outcome

After each document review iteration, the SCE team has new information for caucus. Usually the information gained confirms or negates several topics. The direct output is the document review working notes used during the caucus.

Notes

Guidelines for document review are provided in Document Review During an SCE Site Visit on page 108.

This level of document review focuses on implementation-level documents; but some project- and organization-level documents may also be referenced.

Pairs of team members may visit the organization’s document library, if one exists. Some team members may prefer to select the documents they review from the library of documents rather asking for specific documents. However, in any document review, the objective is to collect objective evidence about the critical subprocess areas by investigating the topics on the Validation Worksheets.

Many teams develop simple checklists and forms to facilitate document review. For example, a team may have simple checklists and forms for

- Capturing the author, scope, and revision date of each document reviewed.
- Listing the document type, whether it was found, and comments.
- Consolidating information found in different documents by subprocess area and KPA.

Step 18 Develop Preliminary Findings

Input

The inputs to developing preliminary findings are all the information collected so far, including

- The Interview Worksheets (originally generated in Step 15).
- Document review working notes (from Steps 14 and 17).
- The Validation Worksheets (from Step 16).
- The look-for tables from the SCE method.

Action

This is a special caucus that is focused on drawing conclusions about the subprocess areas under investigation, and is one of the steps in the iterative decision making process.

In Steps 9 and 10, the SCE team translated the critical subprocess areas into topics for investigation. Subsequently, interviews and document reviews were used to gather information about the topics, and team caucuses established consensus about the critical subprocess areas by considering the information gathered.

The SCE team now develops conclusions about the topics listed on the Validation Worksheets. To do this, the team consolidates all of the available information about the topics pertaining to a subprocess area. The team then abstracts the information and makes a conclusion about processes the SCE sponsor can expect to be applied to the next project by the development organization.

Based on the conclusions, the team develops preliminary findings in terms of strengths, weaknesses, and improvement activities for subprocess areas within the KPAs. The team also identifies ➠ **candidate findings** for which there is not yet enough objective evidence to make a decision. Candidate findings become the subjects of consolidation interviews or subsequent document reviews, as shown in Figure 2-8 on page 92.

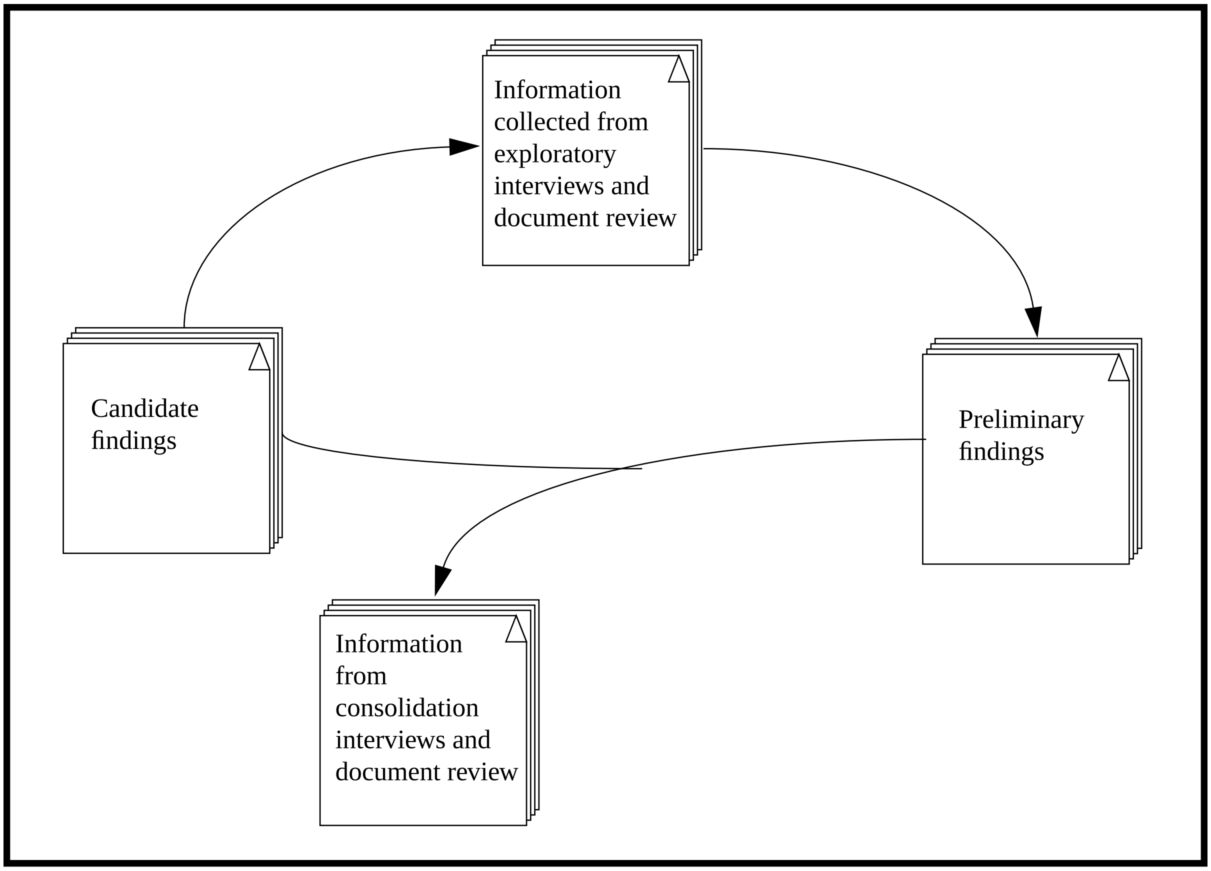

Information collected from exploratory interviews and document review
Candidate findings
Preliminary findings
Information from consolidation interviews and document review

Figure 2-8: Transformation of Information into Findings

Once a preliminary finding has been made, the subprocess area is dropped from further consideration. That does not mean that new evidence will not be considered, but rather that the team will not spend any more time looking for data relative to the issue; this helps the team to use its time on site most effectively.

The look-for tables provide guidance the teams may reference when developing findings. The look-for tables associate specific activities that the team may have observed with particular KPAs and subprocess areas in a concise format. Properly associating the information gathered with the appropriate subprocess area and KPA is essential for accuracy.

The preliminary findings are the primary input to Phase 5, Findings.

Purpose

The purposes of this step are

- To articulate, based on the information available, conclusions about the development organization’s implementations that map to the subprocess areas.

- To guide subsequent information gathering efforts.

Outcome

The outputs of this iterative step are preliminary findings and candidate findings about the critical subprocess areas; these are annotated on the Validation Worksheets. When a preliminary finding has been annotated on the Validation Worksheet, the Validation Worksheet is "completed."

If the Validation Worksheets are kept up to date, an experienced team will complete the preliminary findings caucus in approximately 45–90 minutes.

Throughout the site visit, the team creates a picture of the organization's software practices. Preliminary findings are the articulation of the team's picture, based on observations that are documented on the Validation Worksheets. Because findings require objective evidence and corroboration from multiple sources, early articulation of the findings gives the team enough time to identify missing evidence before the conclusion of the site visit.

In order for an SCE finding (strength, weakness, or observed improvement activity) to exist, the following guidelines must be met:

- There must be objective evidence in the form of documentation to support the finding.
- The team must observe supporting evidence in two or more independent sources.
- The team must generate the findings through a consensus process. That is, there are no minority opinions opposed to the finding.
- The evidence must support the findings.

All judgements made by the team should be correlated by at least two separate pieces of information. As the significance of the judgement increases, the correlation may require three or more separate sources of information. As a general rule, if there is any doubt at all about whether a finding is valid, the team should defer it to the consolidation step and should initiate additional data collection efforts.

The information collected that does not become a preliminary finding may become a candidate finding, as shown in Figure 2-8 on page 92. Candidate findings are subject to further investigation during consolidation interviews or during further document review. If a finding cannot be validated, if doubt remains, or if consensus is not achieved despite additional documentation or interviews, then there can be no finding in that instance and the candidate finding should be discarded.

If the team identifies a possible weakness, the development organization (through the site visit coordinator or in subsequent interviews) should be given an opportunity to produce evidence that might mitigate or eliminate the weakness. By double checking, the team avoids making findings based on anomalous responses. No direct mention is made of the preliminary finding. The request for clarification should define the subject matter and ask whether what the team observed or heard is representative. For example, the team might ask “We were not able to determine if the estimates for project size were based on actual data. Did we miss something?”

It is possible for a given subprocess area to have strengths, weaknesses and improvement activities—for example, well-defined procedures (a strength), no training in the procedures (a weakness), and an ongoing course development effort for the new procedures sponsored by the organization (an improvement activity).

At some point every subprocess area on the Critical Subprocess Area List (as documented by the Validation Worksheets) must have a finding or an explicit annotation of “no finding,” meaning that there was not enough observed evidence to make a finding. A “no finding” annotation completes the Validation Worksheet.

The team must be cautious about associating findings with a particular subprocess area within a KPA. For example, some activities relate to more than one goal of the KPA, hence to more than one subprocess area. If a team “rolls up” the information to a subprocess area that is outside of the scope of the SCE, the findings can not be used.

Also, if a topic was not investigated, the team should be careful not to reference the topic as a weakness in the findings. For example, suppose the team did not investigate the “organizational policies” feature of the “develop estimates” subprocess area of the Software Project Planning KPA. To say that the team found “no evidence of policies requiring estimates” would be invalid—the team did not look for this.

Step 19 Create Consolidation Plan

Input The inputs to this step are

- Candidate findings from Step 18 (as annotated on the Validation Worksheets); they represent subprocess areas where there was not enough evidence to make a decision.
- The updated organization charts and information (originally from Step 13).
- The look-for tables from the SCE method (avaiable in SCE team training).

Action

This step is part of the iterative decision making process. The team decides what data or further objective evidence it needs to finalize the candidate findings, and plans how they will gather the information. The team then initiates the next round of interviews and/or document review. All interviews and document reviews must be completed within the remaining time of the site visit.

The Validation Worksheets contain the preliminary findings; they are also used to identify the project and topic that the subsequent investigation should focus on.

If further interviews are needed, the team prepares new Interview Worksheets and coordinates the interview schedule with the development organization's site visit coordinator. If additional documentation is needed, the team coordinates the request with the site visit coordinator.

The look-for tables may be used to prepare for interviews and document reviews by suggesting potential agents and artifacts that the team might look for, or by reminding the team of activities that they did not probe for relative to a particular subprocess area.

The Consolidation Plan is not a separate document; rather, the plan is contained in the Interview Worksheets and the requests for further documentation.

Purpose

The purpose of this step is planning and initiating further data collection efforts.

Outcome

Outputs are new Interview Worksheets and interview schedules (used in Step 20), and further requests for documents (used in Step 21).

Step 20 Conduct Consolidation Interviews

Input

The inputs to this step are

- New Interview Worksheets from Step 19.
- New interview schedules from Step 19.
- The development organization's documentation.

Action

The team interviews personnel who may be able to provide additional objective evidence required by the SCE team to finalize their findings. The team may ask development organization personnel to help them locate evidence in the documentation.

The second (and any subsequent) round of interviews are consolidation interviews. These interviews follow at least two iterations of document review and serve to validate the candidate findings about the selected projects.

Purpose The purpose of these interviews is to clarify any remaining issues by confirming or negating candidate findings through further interviews. If documentation is required to substantiate a finding, the team will ask development organization personnel to indicate the location of information within documents for the team to review.

Outcome The output of this step can be either interview-based evidence that supports or negates the team's candidate findings, or the location of evidence in documentation that can be used to validate the findings. Evidence based on interviews is annotated on the Interview Worksheets.

Notes Guidelines for interviewing are provided in Interviewing During an SCE Site Visit on page 115.

The main difference between consolidation interviews and exploratory interviews is in the amount of information the team already has to guide it through consolidation. Consolidation interviews usually focus on one or two questions and are aimed at eliciting information related to a discrepancy, i.e., resolving an issue that remains open after the exploratory interviews and the document reviews.

Step 21 Conduct Final Document Review

Input The input to this step is the previous information generated, including

- Document review working notes (originally from Steps 14 and 17).
- The Interview Worksheets from (originally from Step 15).
- The Validation Worksheets.
- List of documents to be reviewed from Step 19, and the corresponding documents from the development organization.

Action The team performs the final document search for specific information that will confirm or negate the candidate findings.

Purpose The purpose of this activity is to clarify any remaining issues by confirming or negating candidate findings through further document review.

Outcome New evidence is gathered to support or negate the team's preliminary findings, and recorded in the form of document review working notes, or annotated on the Validation Worksheets. When annotated with a preliminary finding or "no finding," the Validation worksheets are completed.

Notes Guidelines for document review are provided in Document Review During an SCE Site Visit on page 108.

Final document review focuses on locating a specific piece of information that the team needs to confirm a candidate finding. Usually the team will request documents that contain the information the team needs but has been unable to locate. It is possible that the information exists in an unexpected location; the focus is verifying the existence of the information, regardless of where it is.

Summary of Phase 4 Site Data Collection

When the Site Data Collection phase is complete, the processes at the site are investigated, and the SCE team is ready to generate their consolidated findings. The information recorded on the Validation Worksheets, Interview Worksheets and the objective evidence collected during Site Data Collection is the support for the findings.

The overall purpose of the site visit is for the team to investigate the topics associated with each critical subprocess area in enough depth to determine the strengths, weaknesses, and/or improvement activities of the corresponding subprocess area.

To start the visit, the team presented an entry briefing at the initial organization meeting (Step 13). The initial organization meeting was used to explain the on-site activities and to set the tone for the visit.

Next, the team started their data collection activities by performing the initial document review (Step 14). There are three levels of documents reviewed during an SCE; the initial document review concentrated on the top two levels—organizational and project. The initial document review was used to tailor and focus the subsequent exploratory interviews (Step 15). Steps 14 and 15 represent the first iteration of the two complimentary mechanisms used to investigate a topic during an SCE. These techniques were applied iteratively during the site visit until enough information was gathered to make a decision about each topic.

Decisions were made in an ongoing team caucus (Step 16). The caucus gives the team a chance to share information and to build consensus when enough information exists to make a decision about a topic.

After the first round of interviews, the team returned to Document Review (Step 17), this time concentrating primarily on implementation-level documents. Because of the iterative nature of the process, the detailed document review activity was interspersed with interviews and caucuses.

A special caucus was held to develop the Preliminary Findings (Step 18). Preliminary findings are expressed in terms of the subprocess areas and consolidate the information gathered on the topic level. This portion of the decision-making process eliminated some topics from further consideration, and identified areas in which more data was needed. This information was used to create the Consolidation Plan in Step 19, which guides the consolidation interviews (Step 20). The consolidation interviews and Final Document Review (Step 21) implemented the plan. The final two steps resolved open issues and located specific documentation support.

The site visit activities culminate when every subprocess area has an associated preliminary finding or an elicited determination of “no finding”; the team is then ready for Phase 5, Findings.

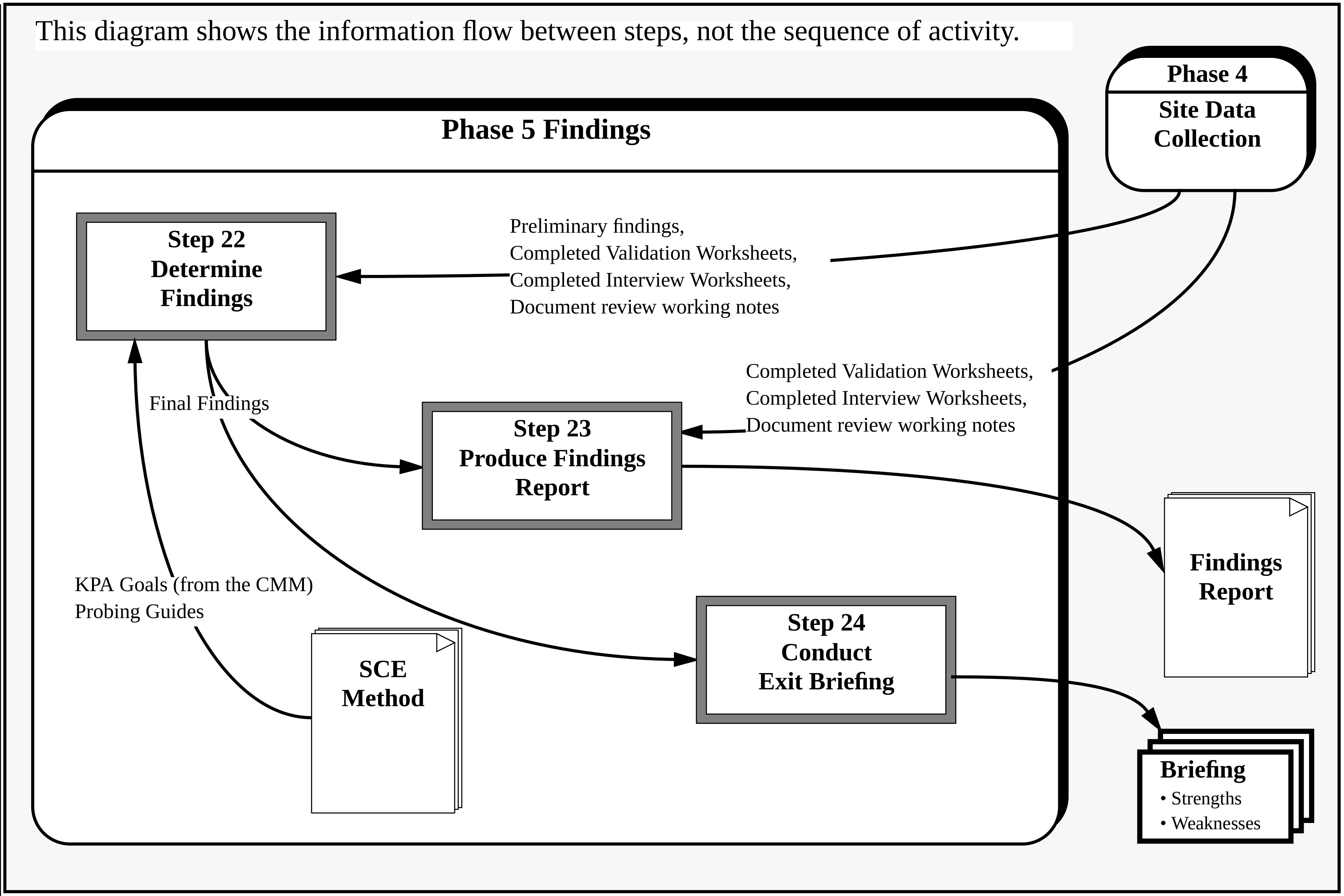
2.6 Phase 5: Findings

The Findings phase completes the SCE.

The purpose of the Findings phase is to consolidate the decisions made during the Site Data Collection phase. This purpose is accomplished by “rolling up” the decisions that were made about specific topics and subprocess areas into findings at the KPA level (Step 22). Findings are expressed in terms of the strengths, weaknesses, and improvement activities that were observed by the team. Ideally, the SCE team presents the findings to the development organization during an exit briefing (Step 24).

The findings are actually generated during the site visit, although the final report (Step 23) of the findings may be done later. The Findings phase is treated separately to clearly indicate the end of the SCE activity and to separate the SCE Method activities from the use of the findings in a source selection or contract monitoring context.

On the next page, Figure 2-9 provides a high level diagram of the steps in this phase.



This diagram shows the information flow between steps, not the sequence of activity.

Figure 2-9: Diagram of Steps in Phase 5, Findings

The table below provides an overview of the steps in this phase.

Phase	Step	Purpose	Page
Phase 5: Findings	22. Determine Findings	Validate the preliminary findings and consolidate them by KPA.	page 102
	23. Produce Findings Report	Document the SCE activities and provide a formal record of the findings.	page 103
	24. Conduct Exit Briefing	Provide feedback to the recipient and conclude the SCE.	page 104

Table 2-11: Overview of Phase 5

Step 22 Determine Findings

Input The inputs to this step are

- Preliminary findings from Step 18.
- Interview Worksheets from Steps 15 and 20.
- Document review working notes from Steps 14, 17, and 21.
- Completed Validation Worksheets from Steps 18 and 21.
- The KPA goals and the probing guides. (These are in the look-for tables available in SCE team training).

Action In a final series of caucuses, the team analyzes the information learned from the consolidation interviews and final document review to determine whether the information confirms or negates any of the preliminary findings and also whether the information supports or negates any of the candidate findings.

As mentioned previously, the look-for tables provide guidance the teams may reference when developing findings. The benefit of the look-for tables is that they associate specific activities that the team may have observed with particular KPAs and subprocess areas in a concise format.

A finding is essentially a judgment about the degree to which a KPA goal has been met; the phrasing of the probing guides facilitates considering whether the evidence gathered is sufficient for a finding.

The validated preliminary findings become ➠ **final findings**, while the negated findings are dropped from consideration. The preliminary findings were made relative to subprocess areas; the final step is to regroup them by KPA. Final findings consist of strengths, weaknesses, and improvement activities for each KPA investigated by the team.

All the steps after Step 18 (Develop Preliminary Findings) focused on resolving outstanding issues identified during Step 18. This step represents the final resolution of those issues; this is the final step in the iterative process of evaluating the organization's process capability.

Purpose The purposes of this step are to validate the preliminary findings and to consolidate them by KPA.

Outcome The output is a set of final findings, summarizing the results of the information-gathering activities. At this point, all data collection activities stop.

Step 23 Produce Findings Report

Input The Findings Report may incorporate all of the information gathered, including

- Final findings from Step 22 are the primary input.
- Document review working notes.
- Interview Worksheets.
- Validation Worksheets.
- Information from the development organization (such as the Project Profiles and organization charts).

Action The team prepares a formal report of the SCE containing a standard set of information. The information specified allows comparison of all SCEs performed for the development. The report documents the major steps of the SCE and the objective evidence that supports the findings.

Some portions of the report are generated during the visit, such as the findings. For accuracy, the remainder of the report should be generated as soon as possible after the site visit.

Purpose The purpose of this step is to document the SCE activities and provide a formal record of the findings.

Outcome The direct output is the findings report. This report ensures that the SCE activities are fully documented and the findings are formally recorded for future use.

Notes In most cases, the conclusion of the site visit represents the conclusion of the SCE team's activities.

In a source selection, the findings report must be complete enough so that sponsoring organization officials can understand all judgements made by the SCE team in case the SCE team is not available to explain them. In contract monitoring, the report must be complete enough so they can be compared to subsequent evaluations in a meaningful way.

In source selection, the acquisition agency will specify the exact information to be provided; in any case, the format will be standardized across the development organizations.

The findings report should contain the following information:[1]

- **Information common to all development organizations**, including the Target Product Profile, the Target Process Capability, the Critical Subprocess Area List, etc.
- **Information provided by the individual development organization**, including Project Profiles, the Proposed Project Profile, organization charts and information, and questionnaire responses.
- **All worksheets**, including Key Issue Worksheet, Validation Worksheet, and Interview Worksheets.
- **Objective evidence** which serves as a basis for findings. (This section should be a formal description of the evidence supporting the team's findings rather than the actual evidence. The team will not be allowed to take the evidence with them.)
- **Findings**, including a separate sheet(s) for each KPA. The findings sheets should include references to the objective evidence which support them.

Step 24 Conduct Exit Briefing

Input The final findings from Step 22 are the only input to this step.

Action The team prepares a short debriefing and delivers it to the development organization before leaving the site. The content of the briefing may vary from a simple "courtesy call" to a formal presentation of the final findings. The depth of the Exit Briefing will depend on the application of the SCE and on source selection considerations.

The findings should be presented to the development organization at the time of the site visit, so they can use the findings in their process improvement activities. However, some source selection authorities insist that the briefing be deferred until after contract award.

Purpose The purposes of this step are to provide feedback to the recipient and to conclude the SCE.

Outcome The team concludes the SCE. The findings briefing is the only output.

[1] In source selection, all materials pertaining to unsuccessful bidders are kept segregated from materials pertaining to the selected contractor. If the SCE findings are to be used for process improvements by the selected development organization, the findings report would not include any information that referred to the other development organizations, such as the Experience Table generated in Step 4.

Notes

In a source selection, the Procuring Contracting Officer (PCO) must agree to the agenda of the exit briefing. The acquisition process is controlled by regulations that puts severe constraints on “discussions” with development organizations. The PCO may decide that a debriefing of findings in any form constitutes a discussion. Customer feedback indicates that most source selection authorities are reluctant to let the SCE team present findings before contract award.

Summary of Phase 5 Findings

When the Findings phase is complete, the detailed decisions made during the Site Data Collection phase about the subprocess area topics will be consolidated and summarized by KPA in terms of strengths, weaknesses, and observed improvement activities.

The activities in this phase are conducted both on site and off site. The team generates the final findings in Step 22. The final findings are then used to prepare a formal Findings Report in Step 23. The Findings Report is used by the sponsoring organization; how the findings in the report are used depends on the context and represents the results of the SCE. The team prepares and conducts an Exit Briefing (Step 24) before leaving the site. The exit briefing varies in content, but the team should use the exit briefing as a forum for presenting the final findings to the development organization.

2.7 Coordination of SCE Activities

This section contains information that is useful for coordinating activities across multiple steps (or phases), or for understanding the relationships between the steps. Because these activities (or relationships) are not confined to a single step, their description is distributed over several steps in the previous discussion of the activities during an SCE. In this section, the information is consolidated for easier reference.

For example, there are several points during an SCE at which information is requested from the development organization. The information requests are referenced within the descriptions of the steps, but the references within the description of the steps do not provide a consolidated picture of when information requests are made.

This discussion is divided into two major subsections: Coordination of Site Data Collection Activities, and Coordination of Information Flow During an SCE. Coordination of Site Data Collection Activities is concerned with the activities during the site visit, while Coordination of Information Flow During an SCE is concerned with information flow within and between the five activity phases.

Coordination of Site Data Collection Activities covers these topics:

- Document Review During an SCE Site Visit (page 108).
- Interviewing During an SCE Site Visit (page 115).
- Sample Site Visit Schedules (Table 2-13 on page 118, and Table 2-14 on page 119).

Coordination of Information Flow During an SCE covers these areas:

- Information Request Timetable (Table 2-15 on page 122).
- Primary Inputs and Outputs for Each Step (Table 2-16 on page 124).

Coordination of Site Data Collection Activities

The SCE team uses two complementary mechanisms to investigate a topic: document reviews (Steps 14, 17, and 21), and interviewing (Steps 15 and 20). The steps are not strictly sequential; document reviews are interspersed with interviews, and the consensus-building process is ongoing throughout the site visit.

Because of the iterative, interlaced nature of these fundamental activities during the Site Data Collection phase and their central importance to several of the steps, a consolidated description of the basic concepts of document review and interviewing (as applied to the SCE Method) is provided here. After discussing the basics of document review and interviewing, sample site visit schedules are provided. The schedules demonstrate the iterative nature of the investigation activities and the team caucuses.

This discussion provides a framework for the activities performed during the individual, related steps of the method, and supplements the discussion provided previously within the descriptions of the steps.

Document Review During an SCE Site Visit

This section provides a consolidated description of the basic concepts of document review, as applied to the SCE Method.

Documents can be used to

- Define and standardize processes.
- Indicate commitment to use the processes.
- Provide an audit trail of processes that were used.
- Collect data about process performance.

Documents can provide objective evidence of the processes used. A fundamental assumption of the SCE Method is that if a process is not documented, there is no guarantee that it will be followed.

Documents may be in paper or electronic form, and they vary widely in name, content, and format. They are not arranged by topic within subprocess area within KPA; the team needs a broad range of professional experience to determine whether a document or set of documents satisfies a topic. Because of time constraints, document review during an SCE is broad rather than deep; the development organization should be strongly encouraged to organize and cross reference the documentation provided to the team.

Document reviews during an SCE do not check whether the project’s work products are consistent with the project’s development objectives. For example, no checks are made to see that the software requirements specification is complete and accurate when compared to the system requirements that are allocated to software; or that the schedule information showing progress made accurately portrays the progress actually made. In other words, no direct judgment is made about how effective the processes are based on the products they produce. However, the team will decide how well the processes are defined and implemented.

Many teams develop simple checklists and forms to facilitate document review. For example, a team may have simple checklists and forms for

- Capturing the author, scope, and revision date of each document reviewed.
- Listing the document type, whether it was found, and comments.
- Consolidating information found in different documents by subprocess area and KPA.

Three “levels” of documents are reviewed during an SCE: *organization-level* documents, *project-level* documents, and *implementation-* (or “track record”) *level* documents. Each document level and the corresponding review is described in more detail below.

Organization-Level Documents

At the top level are the organization-level documents—the policies and procedures which establish the development environment for all company project activities. They define the process guidelines that management expects *all* projects to follow.

Ideally, this level of documentation ties the need for software development processes such as software configuration management and software quality assurance to defined “business needs” in the form of policies.

For all development projects, organization-level documents define the process and management constraints the organization places on projects by

- Demonstrating commitment to perform activities.
- Defining organizational structures that support the activities.
- Defining roles and responsibilities.
- Specifying default procedures and standards to be used on all development projects.
- Defining organization-wide support for training.

The purpose of reviewing this level of documentation is determining the degree to which the organization supports the project's development and maintenance of software products by defining standard processes.

Organization-level documents show what management thinks will happen with planned projects and can be used as an indicator of planned process improvements.

This review is usually carried out in Step 14, before the exploratory interviews (Step 15), although in some cases organization-level document review can begin during the preparation phases. For a given subprocess area, the features that may be partially or completely validated during this review include leadership, organizational policies, resources, organizational structures, training, plans and procedures, corrective actions, analysis of measurements, and reviews by management.

The scope of review for organization-level documents may include (but is not limited to) checking for items such as

- Organizationally controlled size and costing procedures.
- Standard status reporting practices that are required across the organization.
- Defined default plans and procedures for a project.
- Tailoring guidelines and waiver procedures.
- Training provided by the organization.
- Peer reviews that are required as part of product development work.
- Independent reporting channels for project software quality assurance, software configuration management, and testing activities.
- Defined organizational roles for software configuration management, software quality assurance, and software subcontract management.

Project-Level Documents

The next level of documents are project-level documents; these documents define the development processes in use for a project.

Ideally, project-level documents should be traceable to the organization-level documents—that is, project-level procedures should be consistent with organizational policies, and, whenever possible, should be tailored versions of the organization-level documents.

For a current project, these documents define the detailed processes that are used to manage, coordinate, and integrate the engineering activities required for the development. This level of documentation gives structure to the development by

- Translating high level organizational policies and procedures into detailed procedures, plans, and guidelines.
- Establishing the required organizational entities on the project level to support the defined processes.
- Defining specific project roles and responsibilities.
- Allocating human and other resources to fulfill the process related responsibilities on the project level.
- Defining detailed procedures to supplement and enhance the organization-level procedures.
- Specifying and planning for required training.
- Defining how adherence will be tracked.
- Defining measurements that will be used to manage project activities.

Although process measurement and tracking are often associated with implementation-level documents, the project-level documents should provide scope for the implementation-level documents by defining the measurements that should be made and how project-level processes will be tracked.

The purpose of reviewing this level of documentation is determining the degree to which the project-level processes support project activities. To do this, the team determines what processes are defined for the project and how the project-level processes relate to the organization-level documents. The development support environment is achieved by explicit specification of work practices that integrate the different engineering disciplines; project members should not have to create the processes they use. Comparing documentation from older and newer projects is a good indicator of commitment to process improvement.

Project-level document review is initiated during initial document review (Step 14), before the exploratory interviews (Step 15), and may continue throughout the Site Visit. In some SCE applications this review might start during the preparation phases.

The topic areas that may be partially or completely validated during this review include leadership, organizational policies, resources, organizational structures, training, plans and procedures, corrective actions, analysis of measurements, management reviews, and audits. Adherence tracking methods should be defined at this level, along with measurements that will be taken to monitor and improve the software processes. Validation of these topics, however, is usually completed on the implementation document level.

The scope of review for project-level documents could include (but is not limited to) checking for items such as

- A software quality assurance plan.
- A software configuration management plan.
- Indication that processes referenced in the ➠ **software development plan**[1] are effectively defined in other documents.
- Indication that processes defined only in the software development plan provide sufficient detail to guide actual work practices.
- Indication of how changes to the schedule or requirements are handled over the life of the project.
- Existence of a software manager or lead engineer who directly supports the project manager.
- Independence of software integration and testing from software development.
- Configuration management control over software in test.
- Project notebooks or directives that define how the project collectively understands and integrates the engineering processes.

Implementation-Level Documents

The third level of documents to be reviewed are the implementation (i.e., track record) documents such as status reports, minutes, schedules, etc. These documents provide an audit trail of processes that were used.

Ideally, the purpose, format, and content of implementation-level documents should be traceable to organizational or project-level procedures and standards. Implementation-level documents should capture actions that are necessary for work performance, should be easy to use, and should collect real information about the work accomplished.

This level of documentation can provide

- Evidence of conformance with organizational and project standards.
- Evidence of actual practices used.
- A record of resources used.

[1] A software development plan is “the collection of plans that describe the activities to be performed for the software project” (Mark Paulk, et al. *Key Practices of the Capability Maturity Model* [Paulk 93b], page A-18), not necessarily the document referred to in DoD-STD-2167A.

- Data for process improvement efforts.

The purpose of reviewing this level of documentation is to determine whether the processes defined on paper and elicited from the interviews correspond to what the people on the projects are actually doing. This review is initiated in Step 17, and continues iteratively throughout the Site Visit.

The scope of review for project implementation-level documents could include (but is not limited to) checking for items such as

- Meeting minutes (e.g., from project management meetings or configuration control boards).
- Project status reports and schedules.
- Software change request forms.
- Test records.
- Training records.
- Software development folders.
- Historical data derived by comparing past schedules and status reports to determine “planned versus actual.”
- Analyses of resource consumption and trends.

Document review summary

Document review is a complex process that is conducted throughout the site visit. For the purposes of an SCE, there are three levels of documents. Each level addresses a different set of subprocess area features (these are defined in Appendix A on page 129). Table 2-12 lists the subprocess area features used to generate investigation topics and the corresponding document level.

Feature (used in SCE Method)	Associated Common Feature (from CMM v1.1)	Document Level
Leadership	Commitment to perform	Organization, Project
Organizational policies	Commitment to perform	Organization, Project
Resources	Ability to perform	Organization, Project
Organizational structures	Ability to perform	Organization, Project
Training	Ability to perform	Organization, Project, Implementation
Plans and procedures	Activities performed	Organization, Project
Work performed	Activities performed	Project, Implementation
Tracking	Activities performed	Project, Implementation
Corrective actions	Activities performed	Project, Implementation
Measure process	Measurement and analysis	Project, Implementation
Analyze measurements	Measurement and analysis	Organization, Project
Reviews	Verifying implementation	Organization, Project
Audits	Verifying implementation	Project, Implementation

Table 2-12: Features and Document Level

Interviewing During an SCE Site Visit

This section provides a consolidated description of the basic concepts of interviewing, as applied to the SCE Method.

Interviews give insight into how the processes are implemented in practice and show the extent to which processes are internalized and understood by the development organization staff. A fundamental assumption of the SCE Method is that if a process is not understood by the people implementing it, there is no guarantee that it will be followed.

Interviews also point the SCE team to the implementation-level documentation for a project and guide the document review on that level.

Interviews during an SCE site visit typically involve one of the development organization’s personnel and the entire SCE team. One advantage to this approach is that the employee will probably speak more freely without his or her supervisor or a company representative present. Another advantage is that the data collection is likely to be more effective than if only one team member were conducting the interview. Because the situation may make the interviewee nervous every effort must be made to make the interviewee comfortable. The guidelines for interviewing that start on page 116 include several items that specifically address this need.

There are two types of interviews used during an SCE site visit: *exploratory interviews* and *consolidation interviews*.

During exploratory interviews the questions and answers reveal the actual processes practiced and guide the team to the supporting documentation. The purposes of exploratory interviews are to

- Provide insight into how the subprocess areas are implemented in practice.
- Determine the extent that processes have been internalized by the development organizations.
- Identify critical implementation-level documents.

Consolidation interviews focus on corroboration and clarification of evidence. The purpose of a Consolidation Interview is to clarify any remaining issues by confirming or negating candidate findings.

Interviews are structured by the Interview Worksheets and interview schedule (initiated in Step 11), and should focus on topics within subprocess areas. Initial questions should be framed to elicit a descriptive response. They should not provide the interviewee with ideas about what the team may want to hear.

There are two basic strategies for establishing the interview schedule: working “top down” through the organization, and interviewing project by project. These strategies are used separately or combined to develop the interview schedule.

Schedule changes should be minimized. Availability of the interviewees can cause schedule changes, but schedule changes are disruptive to the orderly analysis of the topics and may prolong the site visit time.

The team must listen well—often there are subtle differences in how terminology is used that must be detected and clarified. Because of time constraints, the team must be willing to cut the interviewee off when the team has collected the data according to their plan.

Initial questions should be “open-ended” rather than leading to a simple “yes” or “no” answer. Questions leading to a “yes” or “no” answer should be used only to confirm information the team has seen or heard previously. Also, if questions are phrased in a leading manner, the interviewee will be likely to try to fulfill a perceived expectation rather than providing information about how the work is actually performed.

Interview data requires corroboration—sometimes a person will tell the team what he or she thinks the team wants to hear, and an individual employee may not know or follow the standard processes for a variety of reasons. No single interview should be the basis for deciding that there is a strength, weakness, or improvement activity in an area.

These considerations are summarized in the following interviewing guidelines. The last four items suggest ways to help to make the interviewee less nervous.

Guidelines for interviewing

- The SCE team leader sets up the interview in cooperation with the development organization’s site visit coordinator.
- The team prepares for the interview by preparing Interview Worksheets (originally done in Step 11, and as needed later); the team should always be aware of the specific information they are seeking.
- Each question is derived from a specific topic on the Validation Worksheets (Step 10).
- Throughout the interview, the team members ask questions to identify documents that will be needed to validate the information.
- One person is interviewed at a time.
- Ask open-ended questions (e.g., please describe how size estimates for this project were determined).

- Allow the interviewee time to clarify responses or ask questions. Interviewees can use this opportunity to ensure that the SCE team clearly understood their responses.
- Introduce all team members and explain the nature and purpose of the interview at the start.
- Emphasize the non-attribution policy and confidentiality of the interview. No information presented to the sponsoring organization or to the development organization's management will be attributed to specific individuals by name.
- Ask the interviewee to briefly describe his or her role in the organization.
- Use polite interruptions to keep the conversation focused on the SCE team's objectives.

Sample Site Visit Schedules

This section contains two sample “strawman” site visit schedules. The schedules clearly demonstrate the interrelationships between the document review, interviewing, and caucusing activities during the Site Data Collection phase. For both of the example schedules, it is assumed that the formal Findings Report (Step 23) is prepared later, after the site visit is completed.

The first schedule assumes that interviews are structured “top down,” interviewing all of the project managers, then the software supervisors, etc. The second assumes that the projects are interviewed sequentially—first Project A, then Project B, and so on.

Day	Activity	Steps	Hours
Day 1	Initial organization meeting with site management SCE team in-brief (15 minutes) Development organization in-brief/ Selected project presentations (60 minutes) SCE team caucus (15 minutes)	13	1.5
	Initial document review and caucus on documentation. (Documents should be available in assigned meeting room).	14, 16	3.0
	Exploratory interviews with project managers and software managers, with caucuses between each.	15, 16	3.0
Evening	Document Review and Caucus	17, 16	3.0
Day 2	Exploratory interviews continue with software supervisors, SQA engineers, SCM personnel, test personnel, and software engineers	15, 16	3.5
	Review of documents requested during exploratory interviews	17	2.0
	Caucus on information gained, possibly with interviews of people who create track record-level documentation.	16, 15	2.0
Evening	Preparation of Preliminary Findings	18	1.5
	Development of Consolidation Plan	19	1.5
Day 3	Consolidation interviews	20	2.0
	Final Document Review	21	2.5
	Determination of Findings	22	1.5
	Exit Briefing	24	2.0

Table 2-13: Site Visit Schedule, Example 1—Interviews Conducted “Top - Down”

Table 2-14 shows second of two “strawman” site visit schedules. The assumption here is that the projects are interviewed sequentially—first Project A, then Project B, and so on. Subsequent interviews address people who have specialty roles or an organization-wide focus.

Day	Activity	Steps	Hours
Day 1	Initial organization meeting with site management SCE team in-brief (30 minutes) Development organization in-brief/ Selected project presentations (60 minutes)	13	1.5
	Initial document review, caucus on documents	14, 16	2.0
	Exploratory interviews with Project A, caucus	15, 16	1.5
	Exploratory interviews with Project B, caucus	15, 16	1.5
	Exploratory interviews with Project C, caucus	15, 16	1.5
Evening	Document review and caucus	17, 16	3.0
Day 2	Document review and caucus	17,16	1.0
	Exploratory interviews with Project D, caucus	15, 16	1.5
	Site SQA interview and caucus	15,16	0.75
	Site Software CM interview and caucus	15,16	0.75
	Corporate management interview and caucus	15,16	0.75
	Site SEPG interview and caucus	15,16	0.75
	Document review and caucus	17,16	2.5
Evening	Preparation of Preliminary Findings	18	1.5
	Development of Consolidation Plan	19	1.5
Day 3	Consolidation interviews	20	2.0
	Final Document Review and caucus	21,16	1.5
	Determination of Findings	22	1.5
	Exit Briefing	24	2.0

Table 2-14: Site Visit Schedule, Example 2—Interviews Conducted One Project at a Time

The times for both of the schedules are approximate. A detailed plan for the exploratory interviews should be made before the visit (Step 11), and coordinated with the company's site visit coordinator. The SCE team must adhere to the interview times or risk appearing unprofessional.

It is necessary to leave sufficient time between the scheduled interviews to allow for

- The team to caucus and reach consensus on what has been learned.
- Additional interviews to obtain information the managers or lead personnel could not provide.
- Some document review.

Coordination of Information Flow During an SCE

There are two topics covered here. The Information Request Timetable indicates when documents must be requested from the development organization(s). The next section covers the inputs and outputs for each step. The inputs listed include inputs that come from the development organization, the SCE Method, or from work done by the team in previous steps.

Information Request Timetable

At several times during an SCE, the sponsoring organization must request documents or information from the development organization. Table 2-15 on the following page lists the information required from a development organization during an SCE, when it needs to be asked for, when the information is needed (required not later than), and which steps it is used in.

Information requested	Asked for in	Required not later than	Used in
Proposed Project Profile	Phase 1[1]	Step 4, Create Experience Table	Steps 4, 5, 7
Project Profiles from projects that are candidates for evaluation	Phase 1[2]	Step 4, Create Experience Table	Steps 4, 7, 11
Organization charts and information	Phase 1[3]	Step 5, Create Critical Subprocess Area List	Steps 5, 9, 11, 13
Questionnaire responses	Phase 1 or 3[4]	Step 8, Develop Key Issue Worksheet	Step 8
Documents for initial document review	Phase 3, after Step 7 [5] [6]	Step 14, Conduct Initial Document Review	Steps 14, 17, 21
Updated organization charts and information	Phase 4, after Step 13	Step 14, Conduct Initial Document Review	Step 14, and throughout the remaining steps
Documents for Document Review	Phase 4, Steps 15 and 16	Step 17, Conduct Document Review	Steps 17, 21
Documents for Final Document Review	Phase 4, Steps 15, 16, and 19	Step 21, Conduct Final Document Review	Step 21

Table 2-15: Information Request Timetable

[1] In source selection, the logical time to request this is when the RFP is sent out. In contract monitoring mode, this request should be made as soon as possible in Phase 1 before the first evaluation. Subsequent evaluations may ask for updates, if any apply.

[2] In source selection, the logical time to request these is in the RFP. In contract monitoring mode, this request should be made as soon as possible in Phase 1.

[3] This request can be combined with the requests for the Proposed Project Profile and the Project Profiles, although the information is used later in the evaluation.

[4] There are two strategies for collecting questionnaire responses:

(1) they can be requested during Phase 1, Evaluation Start for each project submitted by the development organization as a candidate for evaluation, which means the request would be made before Step 4 along with the request for the Project Profiles and the Proposed Project Profile, or,

(2) they can be requested after the projects are selected for evaluation in Phase 3, Step 7.

The first strategy is usually used because it is easier for the SCE team to fit into the schedule—the information is available before it is needed. The second strategy involves less work for the development organization and provides more current information about the projects to the team; however, it can be very difficult to implement because of time constraints, and because timing of the information requests and site visits can be difficult.

5. For initial document review, the team typically requests that copies of all organizational policies, standards, procedures and directives relating to software development be made available in the team's caucus room. The team also requests the project-level procedures, standards, and directives for the projects selected for review in Step 7. This documentation defines organization-level processes and the high-level processes used on the selected projects.

   In a source selection, it is important to allow each development organization the same amount of time to prepare for the site visit. This means that requests for the documents should be coordinated with the site visit schedule.

6. During the site visit, documentation will be reviewed for each project selected for evaluation in Step 7. Some teams request that the comments column in the questionnaires be annotated to indicate what documentation exists to support the answers to the questions. This information can be used to tailor the request for documentation to be reviewed during the initial document review (Step 14). In this case, the documents for initial document review may be requested after Step 8.

   If this is going to be done, the development organization should be notified as far in advance of the site visit as possible because of the extra preparation which will be required from the development organization. Ideally, the requirement that the documentation be annotated on the questionnaires should be spelled out in the RFP for source selection, and as early as possible for contract monitoring.

Primary Inputs and Outputs for Each Step

Table 2-16 lists the primary inputs and outputs for each of the defined steps in the method, including information that is part of the SCE Method, information from the development organization, and information that is generated by the team based on their investigations.

Many of the inputs and outputs correspond to forms that are listed in Appendix C on page 155. These forms are conceptual in nature; they indicate information needed to conduct an SCE, but they are not mandatory. Other forms could be used by a team, provided the forms contain at least the same information set.

Items marked with (†) are not shown on the step diagrams as inputs and outputs, but are included here for completeness. (For example, Figure 2-2 on page 40 does not show the SCE team as an output).

Step	Inputs	Outputs
1. Develop Product Profile	(†) Decision to use SCE and context for the evaluation	Target Product Profile
2. Determine Target Process Capability	Target Product Profile Key process areas and Maturity Levels from the maturity model	Target Process Capability
3. Select SCE Team	Target Product Profile Target Process Capability	(†) SCE Team
4. Create Experience Table	Target Product Profile Proposed Project Profile Project Profiles for projects submitted for evaluation	Mismatch Identification Tables Experience Table
5. Create Critical Subprocess Area List	Target Product Profile Target Process Capability Experience Table Proposed Project Profile Organization charts and information Subprocess Area Selection Tables	Key Issue Table: contains the Critical Subprocess Area List
6. Originate Validation Worksheets	Critical Subprocess Area List (from Key Issue Table)	Validation Worksheets: adds the Critical Subprocess Areas

Table 2-16: Primary Inputs and Outputs for Each Step

Step	Inputs	Outputs
7. Select Projects to Investigate	Target Product Profile Mismatch Identification Table Proposed Project Profile Project Profiles for projects submitted for evaluation	List of projects to be evaluated Document requests
8. Develop Key Issue Worksheet	Target Process Capability Key Issue Table Questionnaire responses List of projects to be evaluated	Key Issue Worksheet
9. Develop Topic Lists	Mismatch Identification Table Key Issue Worksheet Organization charts and information List of Features Look-for tables	Topic Lists
10. Add Topics to Validation Worksheet	Validation Worksheets Topic Lists	Validation Worksheets: adds the Investigation Topics and project names
11. Prepare for Exploratory Interviews	Validation Worksheets Project Profiles for projects selected for evaluation Organization charts and information Look-for tables	Interview schedule Interview Worksheets
12. Prepare Entry Briefing	Target Process Capability Entry Briefing Guidelines from the SCE Method.	SCE team’s presentation Agenda for organizational meeting
13. Conduct Initial Organization Meeting	SCE team’s presentation Agenda	Updated organization charts and information
14. Conduct Initial Document Review	Validation Worksheets Updated organization charts and information Documents for initial document review	Document review working notes
15. Conduct Exploratory Interviews	Interview Worksheets Interview schedule	Completed Interview Worksheets Document requests
16. Hold Team Caucus	Interview Worksheets Validation Worksheets Document review working notes Updated organization charts and information Look-for tables	Updated Validation Worksheets New Interview Worksheets New or revised interview schedule Document requests

Table 2-16: Primary Inputs and Outputs for Each Step

Step	Inputs	Outputs
17. Conduct Document Review	Interview Worksheets Validation Worksheets Document review working notes Documents requested in Steps 15 and 16	Document review working notes
18. Develop Preliminary Findings	Interview Worksheets Validation Worksheets Document review working notes Look-for tables	Preliminary findings and candidate findings, Completed Validation Worksheets
19. Create Consolidation Plan	Validation Worksheets Candidate findings Updated organization charts and information Look-for tables	New Interview Worksheets Revised Interview Schedule Document requests
20. Conduct Consolidation Interviews	New Interview Worksheets Revised Interview Schedule	Completed Interview Worksheets
21. Conduct Final Document Review	Interview Worksheets Validation Worksheets Document review working notes Documents requested in Step 19	Document review working notes Completed Validation Worksheets
22. Determine Findings	Preliminary findings Completed Interview Worksheets Completed Validation Worksheets Document review working notes KPA goals and Probing guides from the look-for tables	Final Findings
23. Produce Findings Report	Final Findings Completed Interview Worksheets Completed Validation Worksheets Document review working notes	Findings Report
24. Conduct Exit Briefing	Final Findings	Findings briefing

Table 2-16: Primary Inputs and Outputs for Each Step

Part 3

Appendix A Overview of the Capbility Maturity Model V1.1 Used in SCE Training

This version of the SCE Method (Version 2.0) uses the process maturity model defined in the *Capability Maturity Model for Software, Version1.1* (CMM) [Paulk 93a]. This appendix provides a summary of essential information contained in the CMM or derived from the CMM which is used in the SCE Method. The information is repeated in this document for easy reference.

This appendix contains the following sections:

Section name	**Section and page number**
CMM V1.1 Process Maturity Levels	Section A.1, page 130
CMM V1.1 Key Process Areas (KPAs)	Section A.2, page 131
CMM V1.1 KPA Goals	Section A.3, page 132
Subprocess Areas	Section A.4, page 135
Features	Section A.5, page 142

The first three sections summarize the levels, KPAs, and KPA goals from the CMM v1.1. The last two sections contain subprocess areas and features. Both of these sections are extracted from the common rating framework for CMM-based appraisals which is under development at the SEI.

The subprocess areas used in this version of the SCE method are derived from the goals of the CMM; each subprocess area corresponds to a single goal, and each goal has a single subprocess area. The features include the common features from the CMM and additional subfeatures.

The KPAs and subprocess areas listed here differ from those described in the baseline SCE Method Description [SCE 93]. A mapping between CMM V1.1 and the maturity model used by the previous SCE method is provided in Appendix B on page 145.

A.1 CMM V1.1 Process Maturity Levels

A maturity level is “a well-defined evolutionary plateau toward achieving a mature software process” [Paulk 93b]. The SCE Method uses these definitions of process maturity levels, which are extracted from *Capability Maturity Model for Software, Version 1.1* [Paulk 93a]:

1. **Initial:** The software process is characterized as ad hoc, and occasionally even chaotic. Few processes are defined, and success depends on individual effort.
2. **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.
3. **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.
4. **Managed:** Detailed measures of the software process and product quality are collected. Both the software process and products are quantitatively understood and controlled.
5. **Optimized:** Continuous process improvement is enabled by quantitative feedback from the process and from piloting innovative ideas and technologies.

A.2 CMM V1.1 Key Process Areas (KPAs)

A key process area (KPA) “identifies a cluster of related activities that, when performed collectively, achieve a set of goals considered important for enhancing process capability” [Paulk 93b]. The KPAs used in Version 2.0 of the SCE Method are from *Capability Maturity Model for Software, Version 1.1* [Paulk 93a].

The KPAs listed here differ from the KPAs in the baseline SCE Method Description [SCE 93]. A mapping between the CMM V1.1 KPAs and the ones used in the previous SCE method is provided in Appendix B.1 on page 146.

Process Maturity Levels	KPAs
5 - Optimized	Defect Prevention
	Technology Change Management
	Process Change Management
4 - Managed	Quantitative Process Management
	Software Quality Management
3 - Defined	Organization Process Focus
	Organization Process Definition
	Training Program
	Integrated Software Management
	Software Product Engineering
	Intergroup Coordination
	Peer Reviews
2 - Repeatable	Requirements Management
	Software Project Planning
	Software Project Tracking and Oversight
	Software Subcontract Management
	Software Quality Assurance
	Software Configuration Management
1 - Initial	

Table A-1: CMM V1.1 KPAs

A.3 CMM V1.1 KPA Goals

The goals of a KPA are “a summary of the key practices of a KPA;” goals “can be used to determine whether an organization or project has effectively implemented the KPA. The goals signify the scope, boundaries, and intent of each KPA” [Paulk 93b]. The KPA goals used in this version of the SCE Method are from *Capability Maturity Model for Software, Version 1.1* [Paulk 93a].

Goals for the Repeatable Level (Level 2) KPAs

Requirements Management

Goal 1 System requirements allocated to software are controlled to establish a baseline for software engineering and management use.

Goal 2 Software plans, products, and activities are kept consistent with the system requirements allocated to software.

Software Project Planning

Goal 1 Software estimates are documented for use in planning and tracking the software project.

Goal 2 Software project activities and commitments are planned and documented.

Goal 3 Affected groups and individuals agree to their commitments related to the software project.

Software Project Tracking and Oversight

Goal 1 Actual results and performances are tracked against the software plans.

Goal 2 Corrective actions are taken and managed to closure when actual results and performance deviate significantly from the software plans.

Goal 3 Changes to software commitments are agreed to by the affected groups and individuals.

Software Subcontract Management

Goal 1 The prime contractor selects qualified software subcontractors.

Goal 2 The prime contractor and the software subcontractor agree to their commitments to each other.

Goal 3 The prime contractor and the software subcontractor maintain ongoing communications.

Goal 4 The prime contractor tracks the software subcontractor’s actual results and performance against its commitments.

Software Quality Assurance

Goal 1 Software quality assurance activities are planned.

Goal 2 Adherence of software products and activities to the applicable standards, procedures, and requirements is verified objectively.

Goal 3 Affected groups and individuals are informed of software quality assurance activities and results.

Goal 4 Noncompliance issues that cannot be resolved within the software project are addressed by senior management.

Software Configuration Management

Goal 1 Software configuration management activities are planned.

Goal 2 Selected software work products are identified, controlled, and available.

Goal 3 Changes to identified software work products are controlled.

Goal 4 Affected groups and individuals are informed of the status and content of software baselines.

Goals for the Defined Level (Level 3) KPAs

Organization Process Focus

Goal 1 Software process development and improvement activities are coordinated across the organization.

Goal 2 The strengths and weaknesses of the software processes used are identified relative to a process standard.

Goal 3 Organization-level process development and improvement activities are planned.

Organization Process Definition

Goal 1 A standard software process for the organization is developed and maintained.

Goal 2 Information related to the use of the organization’s standard software process by the software projects is collected, reviewed, and made available.

Training Program

Goal 1 Training activities are planned.

Goal 2 Training for developing the skills and knowledge needed to perform software management and technical roles is provided.

Goal 3 Individuals in the software engineering group and software-related groups receive the training necessary to perform their roles.

Integrated Software Management

Goal 1 The project’s defined software process is a tailored version of the organization’s standard software process.

Goal 2 The project is planned and managed according to the project’s defined software process.

Software Product Engineering

Goal 1 The software engineering tasks are defined, integrated, and consistently performed to produce the software.

Goal 2 Software work products are kept consistent with each other.

Intergroup Coordination

Goal 1 The customer’s requirements are agreed to by all affected groups.

Goal 2 The commitments between the engineering groups are agreed to by the affected groups.

Goal 3 The engineering groups identify, track, and resolve intergroup issues.

Peer Reviews

Goal 1 Peer review activities are planned.

Goal 2 Defects in the software work products are identified and removed.

Goals for the Managed Level (Level 4) KPAs

Quantitative Process Management

Goal 1 The quantitative process management activities are planned.

Goal 2 The process performance of the project’s defined software process is controlled quantitatively.

Goal 3 The process capability of the organization’s standard software process is known in quantitative terms.

Software Quality Management

Goal 1 The project’s software quality management activities are planned.

Goal 2 Measurable goals for software product quality and their priorities are defined.

Goal 3 Actual progress toward achieving the quality goals for the software products is quantified and managed.

Goals for the Optimized Level (Level 5) KPAs

Defect Prevention

Goal 1 Defect prevention activities are planned.

Goal 2 Common causes of defects are sought out and identified.

Goal 3 Common causes of defects are prioritized and systematically eliminated.

Technology Change Management

Goal 1 Incorporation of technology changes are planned.

Goal 2 New technologies are evaluated to determine their effect on quality and productivity.

Goal 3 Appropriate new technologies are transferred into normal practice across the organization.

Process Change Management

Goal 1 Continuous process improvement is planned.

Goal 2 Participation in the organization’s software process improvement activities is organization wide.

Goal 3 The organization’s standard software process and the projects’ defined software processes are improved continuously.

A.4 Subprocess Areas

The KPAs defined in the CMM are large clusters of activities with multiple goals. In order to understand the processes implemented by an organization and to make judgments about them, it is convenient to divide the KPAs into smaller chunks of activities. The SCE Method uses subprocess areas for this purpose. The subprocess areas listed here were developed as part of the common rating framework development at the SEI.

A subprocess area is a set of activities in an implemented process that, acting together, helps an organization to achieve one of the goals of a KPA. There is a one-to-one correspondence between the subprocess areas and the KPA goals. The subprocess area definitions are derived from the goal statement.

The subprocess areas listed here differ from those described in the baseline SCE Method Description [SCE 93]. A mapping to the KPAs and subprocess areas used in the previous SCE method is provided in Appendix B.2 on page 148.

The following tables list the KPAs, subprocess areas, actions taken in accordance with the subprocess areas, and the KPA goals which correspond to the subprocess areas. The tables are organized by maturity level.

Appendix E on page 185 defines the relationship between the subprocess areas listed below and the attributes in the profiles used in SCE (such as the Proposed Product Profile and the Project Profiles from projects that are candidates for evaluation).

KPA	Subprocess Area	Corresponding KPA Goal
Requirements Management	*Establish and maintain requirements baseline*	1. System requirements allocated to software are controlled to establish a baseline for software engineering and management use.
	Manage requirements-driven changes	2. Software plans, products, and activities are kept consistent with the system requirements allocated to software.
Software Project Planning	*Develop estimates*	1. Software estimates are documented for use in planning and tracking the software project.
	Plan software activities	2. Software project activities and commitments are planned and documented.
	Make commitments	3. Affected groups and individuals agree to their commitments related to the software project.
Software Project Tracking and Oversight	*Track progress*	1. Actual results and performances are tracked against the software plans.
	Take corrective action	2. Corrective actions are taken and managed to closure when actual results and performance deviate significantly from the software plans.
	Manage commitment changes	3. Changes to software commitments are agreed to by the affected groups and individuals.
Software Subcontract Management	*Select subcontractors*	1. The prime contractor selects qualified software subcontractors.
	Establish and maintain commitments	2. The prime contractor and the software subcontractor agree to their commitments to each other.
	Maintain communications	3. The prime contractor and the software subcontractor maintain ongoing communications.
	Track progress	4. The prime contractor tracks the software subcontractor’s actual results and performance against its commitments.

Table A-2: Subprocess Areas for the Repeatable Level KPAs

KPA	Subprocess Area	Corresponding KPA Goal
Software Quality Assurance	*Plan SQA*	1. Software quality assurance activities are planned.
	Perform SQA	2. Adherence of software products and activities to the applicable standards, procedures, and requirements is verified objectively.
	Communicate results	3. Affected groups and individuals are informed of software quality assurance activities and results.
	Address noncompliance	4. Noncompliance issues that cannot be resolved within the software project are addressed by senior management.
Software Configuration Management	*Plan SCM*	1. Software configuration management activities are planned.
	Create software work products baselines	2. Selected software work products are identified, controlled, and available.
	Control changes	3. Changes to identified software work products are controlled.
	Report status	4. Affected groups and individuals are informed of the status and content of software baselines.

Table A-2: Subprocess Areas for the Repeatable Level KPAs

KPA	Subprocess Area	Corresponding KPA Goal
Organization Process Focus	*Coordinate software process activities*	1. Software process development and improvement activities are coordinated across the organization.
	Assess software processes used	2. The strengths and weaknesses of the software processes used are identified relative to a process standard.
	Plan SPI	3. Organization-level process development and improvement activities are planned.
Organization Process Definition	*Provide standard process*	1. A standard software process for the organization is developed and maintained.
	Retain software process information	2. Information related to the use of the organization's standard software process by the software projects is collected, reviewed, and made available.
Training Program	*Plan training*	1. Training activities are planned.
	Provide training.	2. Training for developing the skills and knowledge needed to perform software management and technical roles is provided.
	Receive necessary training.	3. Individuals in the software engineering group and software-related groups receive the training necessary to perform their roles.
Integrated Software Management	*Define project process*	1. The project's defined software process is a tailored version of the organization's standard software process.
	Manage according to process	2. The project is planned and managed according to the project's defined software process.
Software Product Engineering	*Build software*	1. The software engineering tasks are defined, integrated, and consistently performed to produce the software.
	Ensure consistency	2. Software work products are kept consistent with each other.

Table A-3: Subprocess Areas for the Defined Level KPAs

KPA	Subprocess Area	Corresponding KPA Goal
Intergroup Coordination	*Agree on customer’s requirements*	1. The customer’s requirements are agreed to by all affected groups.
	Coordinate intergroup commitments	2. The commitments between the engineering groups are agreed to by the affected groups.
	Manage intergroup issues	3. The engineering groups identify, track, and resolve intergroup issues.
Peer Reviews	*Plan peer reviews*	1. Peer review activities are planned.
	Identify and remove defects	2. Defects in the software work products are identified and removed.

Table A-3: Subprocess Areas for the Defined Level KPAs

KPA	Subprocess Area	Corresponding KPA Goal
Quantitative Process Management	*Plan QPM*	1. The quantitative process management activities are planned.
	Control process quantitatively	2. The process performance of the project's defined software process is controlled quantitatively.
	Establish organization's process capability	3. The process capability of the organization's standard software process is known in quantitative terms.
Software Quality Management	*Plan quality management*	1. The project's software quality management activities are planned.
	Define software quality goals	2. Measurable goals for software product quality and their priorities are defined.
	Track quality progress	3. Actual progress toward achieving the quality goals for the software products is quantified and managed.

Table A-4: Subprocess Areas for the Managed Level KPAs

KPA	Subprocess Area Name	Corresponding KPA Goal
Defect Prevention	*Plan defect prevention*	1. Defect prevention activities are planned.
	Identify defect causes	2. Common causes of defects are sought out and identified.
	Eliminate defect causes	3. Common causes of defects are prioritized and systematically eliminated.
Technology Change Management	*Plan technology changes*	1. Incorporation of technology changes are planned.
	Evaluate new technologies	2. New technologies are evaluated to determine their effect on quality and productivity.
	Adopt new technology	3. Appropriate new technologies are transferred into normal practice across the organization.
Process Change Management	*Plan process improvement*	1. Continuous process improvement is planned.
	Empower everyone	2. Participation in the organization’s software process improvement activities is organization wide.
	Continuously improve	3. The organization’s standard software process and the projects’ defined software processes are improved continuously.

Table A-5: Subprocess Areas for the Optimizing Level KPAs

A.5 Features

A subprocess area is inherently too broad to investigate within the constraints of a site visit. However, each subprocess area has common features. Common features are “attributes that indicate whether the implementation and institutionalization of a key process is effective, repeatable and lasting.” In other words, a common feature is an implementation characteristic common to all subprocess areas.

The features used in this version of the SCE Method are based on the definitions of the common features from the *Capability Maturity Model for Software, Version 1.1* [Paulk 93a]. The previous version of the SCE method [SCE 93] used a subset of these features; however, the term “element” was used to denote them.

The features used in SCE are at a finer level of detail than the CMM common features. The table below shows the definitions of the features used in SCE and shows their relationship to the CMM common features.

Common Feature (from CMM v1.1)	Feature (used in SCE Method)
Commitment to Perform: the actions taken to ensure that the subprocess ares is implemented and will endure	*Leadership* - the assignment of responsibility and the presence of sponsorship
	Organizatinal policies - there are written policies governing the subprocess area
Ability to Perform: the preconditions to implement the subprocess area competently exist in the project or organization	*Resources* - the adequacy of resources (r.g., staff, funds, facilities, tools)
	Organizational structures - the organizational strucuture provides support for the process activities (e.g., job descriptions, defined relationships between entities on the organization chart)
	Training - availability of pertinent training and orientation, and its timeliness for the people who carry out the activities in the implemenation of the subprocess area (e.g., curriculum conent, training schedule, records)
Activities Performed: the roles and procedures necessary for implenentation of the processes	*Plans and procedures* - plans and procesures exist and are prepared according to a documented procedure
	Work performed - the objective evidence of the use of plans, procedures, and standards in the work done by the organization (i.e., track record and “paper or electronic trail”)
	Tracking - how the work is tracked and how problems are identified
	Corrective actions - the identification and resolution of problems

Table A-6: Features Used in SCE

Common Feature (from CMM v1.1)	**Feature (used in SCE Method)**
Measurement and Analysis: the determination if te status ansd effectiveness of the activities	*Measure process* - the measurements of activities performed (e.g., resources consumed, problems encountered, work product characteristics, and status of activities)
	Analyze measurements - the analysis and use of measurements taken
Verifying Implementation: the actions that ensure compliance to established practice	*Reveiws* - management reviews
	Audits - audits of activities and work products

Table A-6: Features Used in SCE

Features provide a level of structure that enables teams to ask specifically focused yet open-ended questions during interviews and document reviews.

When a feature is tied to a specific subprocess area it becomes a topic for investigation. A *topic* is an abstraction of a work practice. Topics are intended to be detailed enough to focus the investigation on observable, documented work practices, but sufficiently abstract that they avoid prescribing how the topic is implemented.

The features are used in Step 9, Develop Topic Lists (along with the “Look For” tables) to specify the topics which will be investigated for each subprocess area on the Critical Subprocess Area List.

Appendix B Comparison to the Maturity Model Used in Earlier Versions of SCE

This appendix shows how the KPAs and subprocess areas used in earlier versions of the SCE method correspond to the KPAs and subprocess areas used in the current method, which is based on version 1.1 of the CMM [Paulk 93a].

This appendix contains the following sections:

Section name	Section and page number
Comparison of Key Process Areas (KPAs)	Section B.1, page 146
Subprocess Areas	Section B.2, page 148

The purpose of this section is to help team members who were trained in the earlier versions of the method to relate their training to the new model.

The original version of the SCE method was based upon *A Method for Assessing the Software Engineering Capability of Contractors* [Humphrey 87b] The method relied on the Maturity Questionnaire[1] and the maturity framework contained in the report. The questionnaire was used to collect information about a development organization’s software process. Site visits were used primarily to validate the responses on the questionnaire.

Building on experience with the Maturity Questionnaire, the SEI extended the software process maturity framework into a maturity model. The model incorporated knowledge acquired from software process assessments and feedback from both industry and government. The maturity model provided more effective guidance for understanding and evaluating an organization’s software development processes. The *Capability Maturity Model for Software V1.1* (CMM) [Paulk 93a] evolved from the earlier maturity model which has been used in the SCE method.

The maturity model used in the previous version of the SCE method is described in detail in Appendix A of the SCE version 1.1 method description [SCE 93].

[1] The “Maturity Questionnaire” refers to the “Assessment Recording Form” and the questions associated with it that are defined in *A Method for Assessing the Software Engineering Capability of Contractors [Humphrey 87b].*

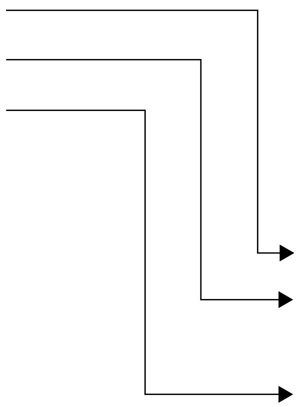
B.1 Comparison of Key Process Areas (KPAs)

The maturity model used in the previous version of the SCE method only described KPAs for the Repeatable and Defined maturity levels [SCE 93]. At the time, teams were not taught to evaluate higher maturity levels. The current SCE method includes the CMM v1.1 KPAs for all maturity levels including the highest levels, Managed and Optimizing (levels 4 and 5). The KPAs at these levels are described in Appendix A.1 on page 130, but no comparison to the previous KPAs can be made for these KPAs.

CMM v1.1 repartitioned the 8 KPAs used in the previous SCE team training into the 13 KPAs currently found at the Repeatable and Defined maturity levels. With this change, some of the names were changed, and there was some realignment of what was at each level; this is shown in Table B-1 on page 147.

The SCE training materials anticipated the change in the number of KPAs by partitioning the *Project Management* and *Software Engineering Process Group* KPAs into “major” subprocess areas. Major subprocess areas contained other subprocess areas; some of these major subprocess areas became KPAs in the CMM. For example, Requirements Management was a major subprocess area in SCE version 1.5 under the “Project Management” KPA that is now a KPA in the CMM. The major subprocess areas either correspond directly to KPAs in CMM v1.1 or were incorporated into other KPAs in CMM v1.1.

Table B-1 illustrates how the SCE Version 1.5 KPAs map to those specified in CMM V1.1 [Paulk 93a]. The major subprocess areas of the older model are listed below the KPA names and are indicated by bullets. Each KPA or subprocess area is directly across from the corresponding KPA in CMM V1.1, excepting those that changed levels; these changes are indicated by arrows.



B.2 Subprocess Areas

The subprocess areas used in version 1.5 of the SCE method [SCE 93] were created by the SCE project members at the SEI. They were used to provide guidance to the teams about areas to investigate within each KPA. The teams were not restricted to the set of subprocess areas provided.

The subprocess areas used in the current SCE method (version 2.0) are derived from the KPA goals defined in CMM V1.1 [Paulk 93a].

The following tables list the subprocess areas from version 1.5 of the SCE method next to the subprocess area derived from the CMM that most closely corresponds to it. The subprocess areas are grouped by KPA. The KPAs are arranged by maturity level as they appeared in version 1.5. At the time, Intergroup Coordination, Integrated Software Management, and Testing occurred as subprocess areas under the Project Management KPA at the Repeatable level, but they are now incorporated into KPAs at the Defined level of CMM V1.1.

SCE Version 1.5	SCE Version 2.0
Project Management	**Software Project Tracking and Oversight**
General Management Functions	
• tracking; actual vs. estimate comparison; commitment evidenced by reviews of compliance • reviewing and oversight; oversight by senior management, and management reviews • usage and collection of performance data	*Track progress*
• taking corrective action; issue/action item tracking	*Take corrective action*
• commitment management process • customer interface	*Manage commitment changes*
• compliance to organizational standards	
Requirements Management	**Requirements Management**
• requirements allocation • requirements implication evaluation	*Establish and maintain requirements baseline*
• requirements change • matching software architecture to requirements; transforming requirement into top-level design	*Manage requirements-driven changes*
Integrated Software Management	**Integrated Software Management** [Now at Defined Level]
• tailoring and selection of project process and its support environment	*Define project process*
• risk management; recognition of risk events; cost, software technology, resources, and schedule • maintenance of process performance database	*Manage according to process*
Intergroup Coordination	**Intergroup Coordination** [Now at Defined Level]
• replanning the project's system plans	*Agree on customer's requirements*
• communicating/ obtaining consensus on the project's system development plans	*Coordinate intergroup commitments*
• coordination between project groups	*Manage intergroup issues*

Table B-2: Repeatable Level KPAs and Subprocess Areas: SCE Version 1.5 Mapped to SCE Method Version 2.0

SCE Version 1.5	SCE Version 2.0
Subcontracting	**Software Subcontract Management**
• subcontractor selection	*Select subcontractors*
• contracting; subcontract process	*Establish and maintain commitments*
• coordination of work with subcontractor	*Maintain communications*
• subcontractor monitoring	*Track progress*
Testing	[Absorbed into Software Product Engineering at the Defined Level]
• preparing to carry out testing; test procedures • carrying out test operations • reviewing test scenarios, testbeds, and test cases • regression testing	
Project Planning	**Software Project Planning**
Size estimation; software development resources, costs, and critical target and host computer resources; the scope of work and effort has a basis in reality *Cost estimation; cost has a documented correspondence to estimated size and schedule; software responsibility, software engineering technical direction*	*Develop estimates*
Planning; resource panning and management for project's software size, cost, and schedule, software development plan, the software life cycle model, planning schedules, software schedules *Project manager's participation with the project proposal team*	*Plan software activities*
Commitment process during change *Integration of technical direction, engineering tools and methods into planning process, engineering and technical reviews of plans*	*Make commitments*
Usage of software process database *Product capacity tracking, critical target computer resources*	

Table B-2: Repeatable Level KPAs and Subprocess Areas: SCE Version 1.5 Mapped to SCE Method Version 2.0

SCE Version 1.5	SCE Version 2.0
Configuration Management	**Software Configuration Management**
SCM plan; baselining of software engineering products and process specifications; a configuration management repository for the software baselines; software baseline audits	*Plan SCM*
Release of software baseline products *Library support system*	*Create software work products baselines*
Change control process, standard forms for reporting errors *Configuration control board*	*Control changes*
Status report, monitoring, configuration responsibility	*Report status*
Software Quality Assurance	**Software Quality Assurance**
SQA plan *Reporting chain; SQA group reports, independent authority*	*Plan SQA*
Auditing; SQA objective evidence of audits *SQA group participation*	*Perform SQA*
SQA concurrence on milestone progress *Oversight for all process support systems; e.g., corrective action system; data collection of defects; earned value of system deviation handling*	*Communicate results*
Noncompliance resolution	*Address noncompliance*

Table B-2: Repeatable Level KPAs and Subprocess Areas: SCE Version 1.5 Mapped to SCE Method Version 2.0

SCE Version 1.5	SCE Version 2.0
Software Engineering Process Group	**Organization Process Focus**
General Functions	
• coordination of review with senior project technical staff, analysis, and evaluation of software process definition, responsibility assignment	*Coordinate software process activities*
• planning systems and software process improvement; review of existing and proposed process standards • defining training requirements	*Assess software processes used*
• assignment of full-time resources, establishing and supporting	*Plan SPI*
Software Product Engineering	**Software Product Engineering**
• integrating the project's process with the SW architecture; process change and technology transition review • investigating software engineering tools and methods; tool selection and use with gathering of performance data • new technologies	*Build software*
• developing and maintaining the project's software architecture • reviewing the system/software testing	*Ensure consistency*
Standards and Procedures	**Organization Process Definition**
Planning standard software process development *implementing standard software process development*	*Provide standard process*
Process assets; a process library system; library of software process specifications; software process database maintenance; tailoring the organization's standard software process *Standards for software development folders* *Review standards* *Human-machine interface standards*	*Retain software process information*

Table B-3: Defined Level KPAs and Subprocess Areas: SCE Version 1.5 Mapped to SCE Version 2.0

SCE Version 1.5	SCE Version 2.0
Training	**Training Program**
Planning/procuring training courses for training curriculum, courses	*Plan training*
Job analysis to support each project's training needs *Communicating and keeping track of delivered training; schedules for all professional and technical staff; records of training*	*Provide training*
Delivering training; management support *The organization's training program; training requirements*	*Receive necessary training*
Peer Reviews	**Peer Reviews**
Planning/assigning peer reviews; technical review schedule, process for technical reviews *review assignments*	*Plan peer reviews*
Conducting peer reviews *Peer review performance; organizational database of review activities; cost; peer review result handling.*	*Identify and remove defects*

Table B-3: Defined Level KPAs and Subprocess Areas: SCE Version 1.5 Mapped to SCE Version 2.0

Appendix C Sample Forms for Use in SCE

This appendix provides examples of the forms used for planning, analysis, and data collection throughout the SCE process. The forms included here are based on the ones used during the SCE team training; in some cases they have been resized to fit in this document better.[1] These forms are conceptual in nature; they indicate information needed to conduct an SCE, but their use is not mandatory.

Examples of the following forms are shown in this appendix:

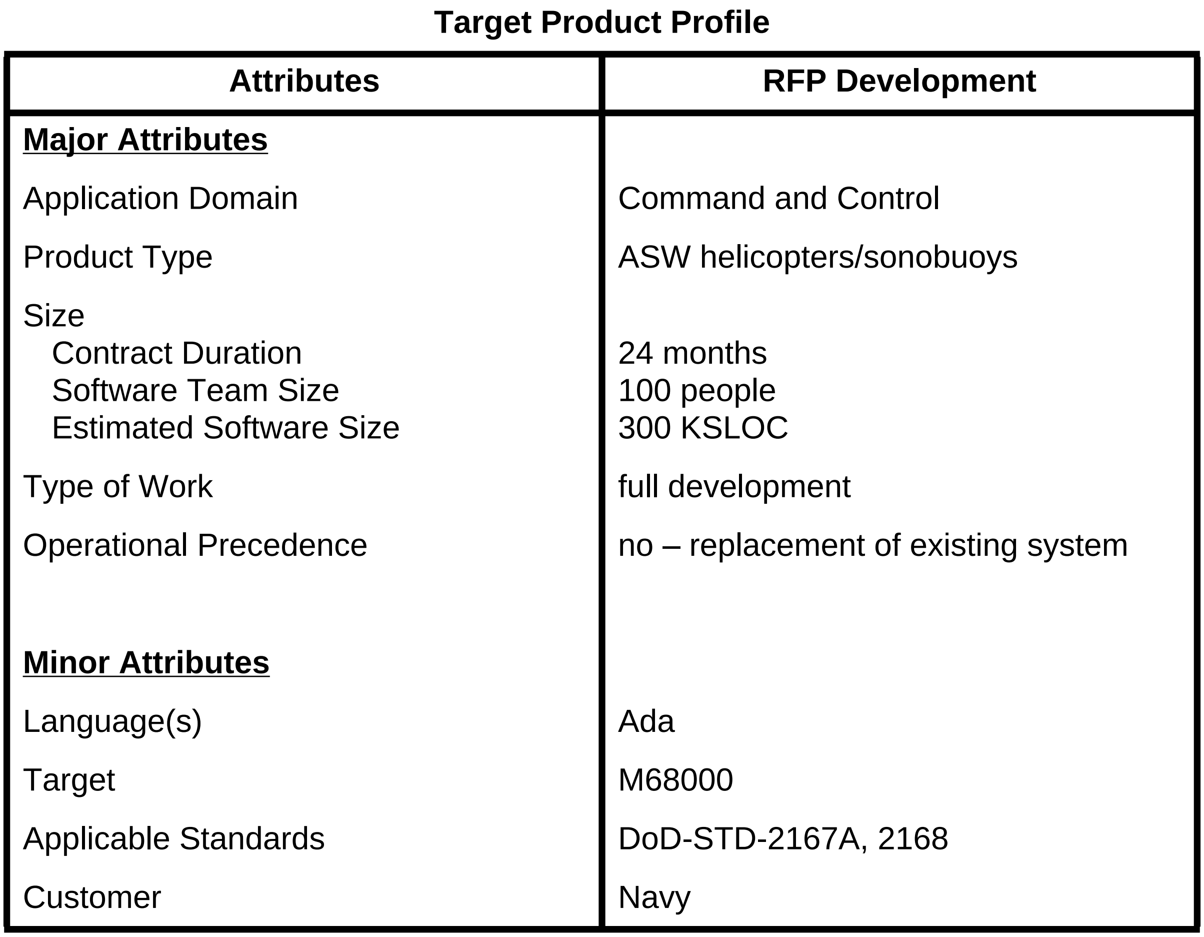
Form	**Section and page**
Target Product Profile	Section C.1, page 156
Proposed Project Profile	Section C.2, page 158
Project Profiles	Section C.3, page 160
Mismatch Identification Table	Section C.4, page 161
Experience Table	Section C.5, page 164
Key Issue Table	Section C.6, page 166
Validation Worksheet	Section C.7, page 169
SCE Questionnaire Worksheet	Section C.8, page 171
Key Issue Worksheet	Section C.9, page 174
Interview Worksheet	Section C.10, page 177

A sample copy of each form is included along with the purpose of the form, a summary of how the form is used, and a description of the data recorded on the form.

[1] The terminology of the forms is acquisition oriented because that was the focus of the initial training, and is still the primary use of the SCE method. For example, “offeror” is used for “development organization” on some of the forms.

C.1 Target Product Profile

The Target Product Profile is used to specify the characteristics of the product to be developed in terms of a standard set of attributes (the attributes are defined in Appendix D on page 179). The Target Product Profile represents a “customer view” of the product to be built. The Target Product Profile is used to identify risk areas that should be given special attention during the evaluation, to define expertise needed on the SCE team, and to provide a the team with a basic understanding of the desired product. Figure C-1 shows a Target Product Profile form with sample data.



Target Product Profile

Attributes	RFP Development
Major Attributes	
Application Domain	Command and Control
Product Type	ASW helicopters/sonobuoys
Size	
Contract Duration	24 months
Software Team Size	100 people
Estimated Software Size	300 KSLOC
Type of Work	full development
Operational Precedence	no – replacement of existing system
Minor Attributes	
Language(s)	Ada
Target	M68000
Applicable Standards	DoD-STD-2167A, 2168
Customer	Navy

Figure C-1: Sample Target Product Profile Form

The Target Product Profile is developed in Step 1 at the start of the SCE process. It is created by the sponsoring organization. The data for the form is based on the sponsoring organization's independent cost and schedule estimates. In source selection, most of the Target Product Profile information is contained in the Request For Proposal (RFP). One Target Product Profile is developed during an SCE.

The Target Product Profile is used in Step 2 to determine the Target Process Capability. It is used in Step 3 to show the types of experience and background to look for when selecting team members. The *operational precedence* attribute from the form is also used in Step 5 for selecting critical subprocess areas. In Step 5, the Target Product Profile is also used to compare the sponsoring organization's view of the product to be built with the development organization's view. The Target Product Profile may also be used as an additional input in Step 4 for creating the Experience Table and in Step 7 for selecting projects for evaluation.

A Target Product Profile lists the names of the attributes and the characteristics of the product in terms of the attributes. The Target Product Profile uses all the major attributes except *subcontractors* and all the minor attributes except *host development system* and *configuration management tool*.[1]

[1] The *host development system* and *configuration management tool* attributes are normally not specified by the sponsoring organization, and may be different for each development organization.

C.2 Proposed Project Profile

The Proposed Project Profile is developed by a development organization to describe the planned development. The Proposed Project Profile provides a “developer view” of the planned development. The information is specified in terms of a standard set of attributes (the attributes are defined in Appendix D on page 179). The information is used to help evaluate a development organization’s previous experience relative to the product being procured in order to identify risk areas that should be given special attention during the evaluation. The information is also used to help select projects for evaluation. Figure C-2 shows a Proposed Project Profile form with sample data.

Proposed Project Profile

Attributes	**Proposed Development**
Major Attributes	
Application Domain	Command and Control
Product Type	ASW helicopters/sonobuoys
Size	
Contract Duration	24 months
Software Team Size	100 people
Estimated Software Size	350 KSLOC (310 new, 40 port/mod)
Type of Work	full development
Subcontractors	none expected
Minor Attributes	
Language(s)	Ada (new), FORTRAN and Assembly (ported)
Target	M68000
Applicable Standards	DoD-STD-2167A, DoD-STD-2168
Customer	Navy
Host Development System	VAX/VMS
Configuration Management Tool	CMS/ MMS

Figure C-2: Sample Proposed Project Profile Form

When the decision to use SCE has been made, the sponsoring organization will request that each of the development organizations prepare a Proposed Project Profile. There will be one Proposed Project Profile for each development organization.

In source selection, the data required for the Proposed Project Profile should be described in the RFP.

The Proposed Project Profile is used in Step 4 along with the Project Profiles to create the Experience Table. The Proposed Project Profile is also used in Step 7 as a guide for selecting projects for evaluation.

A Proposed Project Profile lists the names of the attributes and the characteristics of the project in terms of the attributes. The Proposed Project Profile uses all the major attributes, except for *operational precedence,*[1] and all of the minor attributes.

[1] *Operational precedence* is an indication of whether the end user has previous experience with the type of system to be built. It does not depend on the experience of the development agency.

C.3 Project Profiles

The Project Profiles are similar to the Target Product Profile and the Proposed Project Profile, but are derived from information about actual projects rather than estimates about planned efforts. They are used to gather high level project information from a development organization about previous and current projects. The information shows experience that is relevant to the planned development. The Project Profiles are used along with the Proposed Project Profile to compare a development organization's previous experience to the planned development effort in order to identify risk areas that should be given special attention during the evaluation. The information is also used to help select projects for evaluation. Figure C-3 below shows Project Profiles for three projects with sample data.

The sponsoring organization will request that each development organization prepare Project Profiles for six to eight projects which are similar to the proposed project. The Project Profiles are used in Step 4 along with the Proposed Project Profile to create the Experience Table. They are also used in Step 7 as a guide for selecting projects for evaluation and in Step 11 to help generate the detailed interview plan.

The first column of the Project Profile lists the names of the attributes. A Project Profile uses all the major attributes, except for *operational precedence*,[1] all of the minor attributes, and the schedule attributes. (The attributes are defined in Appendix D on page 179.)

Next, the Project Profile contains a column for each project that lists the characteristics of the projects in terms of the attributes.

[1] *Operational precedence* is an indication of whether the end user has previous experience with the type of system to be built. It does not depend on the experience of the development organization.

C.4 Mismatch Identification Table

Project Profiles

Project	Able	Baker	Charlie
Major Attributes			
Application Domain	acoustic signal processing	acoustic signal processing	command and control
Product Type	sonar navigation (upgrade)	sonar signal analysis (upgrade)	helicopter drone (subcontractor to Mega Corp)
Size			
Contract Duration	27 months	27 months	29 months
Software Team Size	37 people	34 people	27 people
Estimated Software Size	160 KSLOC (80 new, 80 port/mod)	150 KSLOC (110 new, 40 port/mod)	125 KSLOC (all new)
Type of Work	full development	full development	code development
Subcontractors	none	none	none
Minor Attributes			
Language(s)	CMS-2, assembly	Ada, Fortran	Ada
Target	UYK-43	VAX	M68000
Applicable Standards	DoD-STD-1679A	DoD-STD-2167	DoD-STD-2167A
Customer	Navy	Navy	Navy
Host Development System	Univac 1100	VAX/VMS	VAX/VMS
Configuration Management Tool	Sigma Tech Tool	CMS and MMS (VAX tools)	CMS and MMS (VAX tools)
Schedule Data			
Current Phase	system testing	integration and test	coding
Current Month	25	21	18
Start	month 0	month 0	month 0
Design Ends	month 13	month 13	month 15, slipped to month 17
Coding Ends	month 20	month 20	month 22

Figure C-3: Sample Project Profiles Form

The Mismatch Identification Table is a tool used to analyze the experience of a specific development organization relative to the product being procured. A Mismatch Identification Table is prepared for each specific development organization. Figure C-4 is a sample Mismatch Identification Table.

The Mismatch Identification Table is created by the SCE team members in Step 4. The information to generate the form comes from the Proposed Project Profile and the Project Profiles submitted by the specific development organization. The team members compare the attributes of each project on the Proposed Project Profile to the attributes on the Project Profiles.

The Mismatch Identification Table is used by the SCE team in Step 4 to prepare the Experience Table. It is also used by the team members in Step 7 as a guide to help select projects to investigate.

Mismatch Identification Table

Projects	Able	Baker	Charlie	Delta	Enigma	Fiesta	Result
Major Attributes							
Application Domain	0	0	1	0	0	0	
Product Type	1	1	1	0	0	0	
Size	0	0	0	0	0	0	Ps
Type of Work	1	1	0	1	1	0	
Subcontractors	1	1	1	1	1	1	
Minor Attributes							
Language(s)	0	1	1	0	0	0	
Target(s)	0	0	1	0	0	1	
Applicable Standards	0	1	1	0	0	0	
Customer	1	1	1	0	1	1	

0 = experience mismatch, 1 = experience match

Figure C-4: Sample Mismatch Identification Table

The Mismatch Identification Table lists the names of the attributes from the Proposed Project Profile form. Each row of the table corresponds to an attribute. Refer to Appendix D on page 179 for a description of the attributes.

The form has a column for each project that is a candidate for evaluation. These columns show the result of comparing the attributes of each project that are listed on the Project Profile with the attributes of the product being developed, as listed on the Proposed Project Profile. A “1” is placed in the table when the attributes match and a “0” when there is a mismatch.

The last column is the *Result* column. It shows the attributes of the product being procured where the development organization lacks experience. The abbreviation of the attribute is entered in the *Result* column if zeros are entered across the entire row. If there is at least one “1” in the row (i.e., there is previous experience) then the *Result* column is left blank.[1]

[1] On this form, the abbreviation “Ps” stands for “Product Size.” This is used to represent the “Size” attribute.

C.5 Experience Table

The Experience Table is used to determine the attributes of the product to be developed for which any of the development organizations may lack previous experience. These attributes indicate areas of risk that should be given special attention during the evaluation. Figure C-5 is a sample Experience Table form.

The Experience Table is created by the SCE team members in Step 4. It is created by consolidating the *Result* columns of each of the Mismatch Identification Tables for each specific development organization an SCE will be applied to.[1]

The Experience Table is used by the SCE team members in Step 5 to help select the subprocess areas that will be looked at during the evaluation. The subprocess areas selected for evaluation are referred to as critical subprocess

Experience Table

Attribute Name	**Offerors**			
	Sigma Tech	**Beverly Ind**	**Crystal City**	**Result**
Major Attributes				
Application Domain				
Product Type		Pt		Pt
Size	Ps	Ps	Ps	Ps
Type of Work				
Subcontractors				
Minor Attributes				
Language(s)				
Target(s)				
Applicable Standards		Stds	Stds	Stds
Customer				

Figure C-5: Sample Experience Table

[1] On this form, the abbreviation “Ps” stands for “Product Size.” This is used to represent the “Size” attribute.

areas; collectively these subprocess areas make up the Critical Subprocess Area List. The critical subprocess areas are the basis against which all development organizations are evaluated.

The Experience Table lists the names of the attributes from the Proposed Project Profile form. Each row of the table corresponds to an attribute. Refer to Appendix D on page 179 for a description of the attributes.

The Experience Table also contains a column for each of the development organizations to be evaluated. Each column is a copy of the *Result* column from the Mismatch Identification Table for that development organization.

The last column is the *Result* column. It shows whether the development organizations, considered as a community, lack relevant experience in any of the attributes of the product being developed. Each row of the *Result* column contains the abbreviation for the attribute if the corresponding row of any other column contains an entry. Otherwise the entry is blank.

C.6 Key Issue Table

The Key Issue Table is used to record the Critical Subprocess Area List that will be used to evaluate all development organizations. The table also indicates which of the critical subprocess areas should be given special attention for a specific development organization because of a lack of experience in that area (a Key Issue for that development organization). Figure C-6 on page 167 shows a sample Key Issue Table.

The Key Issue Table is created by the SCE team in Step 5. The information for the Key Issue Table comes from the tables provided as guidance in Appendix E on page 185, from the Target Product Profile created in Step 1, from the Experience Table created in Step 4, and from the Critical Subprocess Area List created in Step 5.

In Step 5, the team members select critical subprocess areas based on the experience of the development organizations and on whether the end user has experience with similar systems (*operational precedence*). The team also selects subprocess areas that represent basic processes that a development organization would need for any software development effort. This is referred to as a *nucleus capability*. Additional factors (such as the size of the undertaking) are used to extend and refine the list of subprocess areas. Collectively, these subprocess areas form the Critical Subprocess Area List. The Critical Subprocess Area List does not have a separate form—the Key Issue Table is used to document the list.

There is one Key Issue Table created for an SCE. The table will probably have multiple pages. The Key Issue Table is used in Step 8 to develop the Key Issue Worksheet.

The Key Issue Table lists all the Key Process Areas (KPAs) included in the Target Process Capability. The critical subprocess areas are listed under the KPA with which they are associated. There will be at least one subprocess area selected for each KPA in the Target Process Capability.

The table also contains a column for each development organization. These columns show why a subprocess area was selected and whether the subprocess area needs to be given special attention for a specific development organization. (This indicates that the subprocess area is a key issue for the organization.) The following criteria are used to indicate the relationships between the development organizations and the subprocess areas

- If a subprocess area was selected because of a lack of experience for a particular project attribute, as indicated in the Experience Table, the abbreviation for the attribute is entered in the column for each development organization that lacked experience.[1]
- If the subprocess area was selected because the end user lacks

Key Issue Table

Critical Subprocess Area List	Offerors		
	Sigma Tech	**Beverly Ind**	**Crystal City**
Requirements Management			
Establish and maintain requirements baseline	Ps	Ps	Ps
Manage requirements-driven changes	Ps, *	Pt, Ps, *	Ps, *
Software Project Planning			
Develop estimates	Ps	Pt, Ps	Ps
Plan software activities	Ps	Ps	Ps
Make commitments	Ps, *	Ps, *	Ps, *
Software Project Tracking and Oversight			
Manage commitment changes			
Track progress	Ps, *	Pt, Ps, *	Ps, *
Take corrective action	Ps, *	Ps, *	Ps, *
Software Quality Assurance			
Plan SQA	Ps	Ps	Ps
Perform SQA	Ps, *	Ps, *	Ps, *
Address noncompliance	Ps, *	Pt, Ps, *	Ps, *
Software Configuration Management			
Create software work products baseline	*	Pt, *	*
Control changes	Ps, *	Pt, Ps, *	Ps, *

Figure C-6: Sample Page of a Key Issue Table

operational precedence with similar systems (as indicated on the Target Product Profile), the column contains the abbreviation “Op”.

- If the subprocess area was selected because it is associated with a *nucleus capability*, the column contains an asterisk (“*”).
- If there is no entry in the column, it means the subprocess area was selected because of a lack of experience elsewhere in the development organization community, or added to the list because of team judgment. This subprocess area will be investigated, but the team may decide to spend more time on other subprocess areas.

[1] On this form, the abbreviation “Ps” stands for “Product Size.” This is used to represent the “Size” attribute.

C.7 Validation Worksheet

The Validation Worksheet contains the topics that will be explored during the site visit for a specific development organization. The worksheet is used to record the team's consensus on the data they have collected for each topic. Figure C-7 on page 170 below shows a sample Validation Worksheet.

The Validation Worksheet is prepared by the SCE team. In Step 6, the team members create a set of Validation Worksheets. One worksheet is created for each subprocess area in the Critical Subprocess Area List, as documented on the Key Issue Table. A copy of the set of worksheets is made to be used for each development organization. In Step 10, the team members add topics for each subprocess area to the worksheets (the consolidated topic list is created in Step 9).

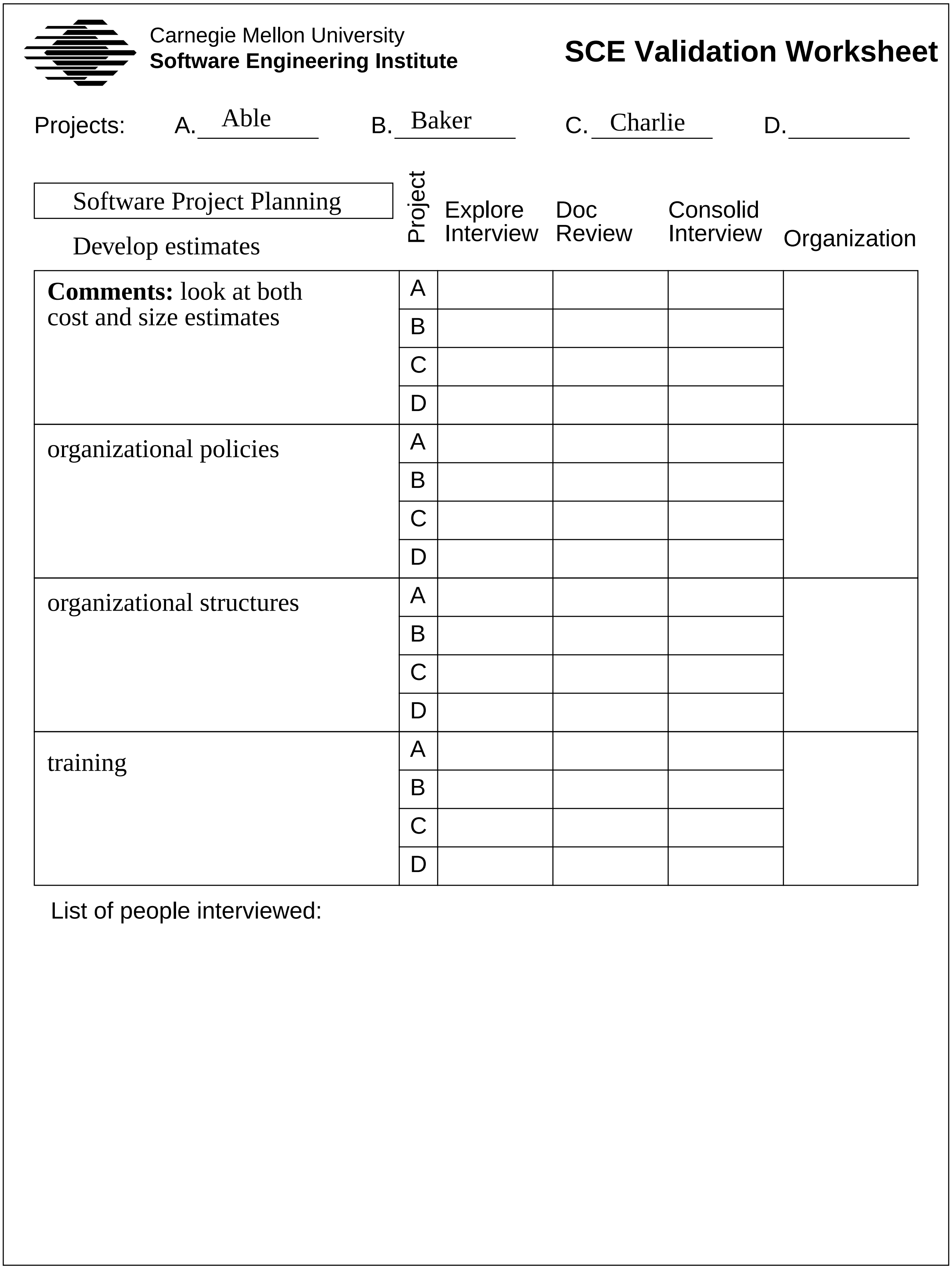
In Step 11, the Validation Worksheets are used to generate interview questions. The Validation Worksheets are used throughout the site visit to record when consensus has been reached on a topic and to determine what topics need to be pursued in follow-on interviews and document reviews.

The top of each page of the form contains the name of the KPA and subprocess area, and a space for the name of each project being evaluated. The names of the projects are preceded by a letter that is used to identify the information for a project.

The form contains a row for each topic associated with the subprocess area. The topics are listed in the first column.

The next four columns are subdivided into rows for each of the projects being evaluated. The first of these columns contains the letter to indicate which project the information in the row is associated with. The other three columns are used to record whether the team reaches a consensus on a topic for a project as a result of exploratory interviews, documentation reviews, or consolidation interviews.

The last column of each row is used to record the composite finding on the topic for the organization.



Carnegie Mellon University
Software Engineering Institute

SCE Validation Worksheet

Projects: A. Able B. Baker C. Charlie D.__________

Software Project Planning

Develop estimates

	Project	Explore Interview	Doc Review	Consolid Interview	Organization
Comments: look at both cost and size estimates	A				
	B				
	C				
	D				
organizational policies	A				
	B				
	C				
	D				
organizational structures	A				
	B				
	C				
	D				
training	A				
	B				
	C				
	D				

List of people interviewed:

Figure C-7: Sample Page of a Validation Worksheet

C.8 SCE Questionnaire Worksheet

Each development organization completes a questionnaire for the projects that the team may evaluate. Usually questionnaires are completed for all of the six to eight projects that are candidates for evaluation. (These are the same projects listed on the Project Profile form.) In some cases, the questionnaire will only be required for the three to four projects selected for evaluation (see Information Request Timetable on page 121).

The questionnaire is used to collect information about the software development processes used on the projects that will be evaluated. The questionnaire provides an initial data input to the SCE team about the processes in use.

Until recently, the questionnaire used for SCEs was the Maturity Questionnaire contained in *A Method for Assessing the Software Engineering Capability of Contractors* [Humphrey 87b]. Questionnaires based on CMM V1.1 have been developed and incorporated into the SCE team training corresponding to this version of the SCE Method.[1]

The SCE Questionnaire Worksheet is used to summarize the questionnaire responses submitted by a development organization. Figure C-9 is a sample page from an SCE Questionnaire Worksheet. The example questions shown on the form are drawn from the CMM based questionnaire.

The SCE Questionnaire Worksheets are prepared in Step 8. A worksheet is prepared for each development organization. The worksheet will have multiple pages. The SCE team members copy the information from the questionnaire for the projects selected for evaluation. The worksheets make it possible to compare results from all projects and to map question responses to subprocess areas to be investigated.

The SCE Questionnaire Worksheets are used by the SCE teams in Step 8 in the preparation of the Key Issue Worksheet. The SCE Questionnaire Worksheets are reviewed for inconsistencies and anomalies that indicate critical subprocess areas that should receive special attention for a specific development organization.

[1] Both the CMM based questionnaire and the Maturity Questionnaire from *A Method for Assessing the Software Engineering Capability of Contractors* [Humphrey 87b] are likely to be used in the field until all source selections using the previous SCE method are completed.

The top line of the SCE Questionnaire Worksheet contains the names of the projects that are being evaluated. The names of the projects are preceded by a letter that is used to indicate which project a response corresponds to.

The leftmost column of the SCE Questionnaire Worksheet is grouped by KPA and subprocess area. The name of the KPA is listed at the top of the column. An abbreviation for the subprocess area is listed above each question. The abbreviation uses the following format:

<KPA abbreviation>.<number of a KPA goal>

<number of a KPA goal> is the KPA goal that corresponds to the subprocess area investigated by the question below the abbreviation (recall that there is a one-to-one mapping of subprocess areas to KPA goals). For example, the question listed under PTO.1 investigates Goal **1** under the Software **P**roject **T**racking and **O**versight KPA; the subprocess area that corresponds to that goal is *track progress*. (Appendix A.4 on page 135 lists KPA goals and shows how they map to subprocess areas.)

There may be more than one subprocess area group on a page.

The next two columns are subdivided into rows for each of the projects evaluated. The first of these columns contains a letter to indicate which project the response is for. The second of the two columns is used to record the responses to the questions from the questionnaire.

The last column of each row is used to record any comments from the questionnaire.

Carnegie Mellon University
Software Engineering Institute

SCE Questionnaire Worksheet 3.1

Projects:	**A.**	**B.**	**C.**	**D.**

Software Project Tracking and Oversight	**Project**	**Questionnaire Responses**	**Comments**
Subprocess Areas			
PTO .1 Are the project's actual results (e.g., schedule, size, cost) compared with estimated in the software plans?	**A**		
	B		
	C		
	D		
PTO.2 Is corrective action taken when actual results differ significantly from the project's software plans?	**A**		
	B		
	C		
	D		
PTO.3 Are changes in the software commitments agreed to by all affected groups and individuals?	**A**		
	B		
	C		
	D		

Y = Yes / N = No / DK = Don't know

Figure C-8: Sample Page of A Questionnaire Worksheet

C.9 Key Issue Worksheet

The Key Issue Worksheet is used to collect all of the information available about a development organization in one place so the team can determine the relative amount of time to spend investigating each of the critical subprocess areas during the site visit.

The Key Issue Worksheet also supports analysis of the Questionnaire Worksheets prepared for each development organization. This analysis may indicate subprocess areas that should receive special attention during the evaluation because of apparent inconsistencies or anomalies in the questionnaire responses. An anomaly occurs when the response to one question by one or more projects is different. An inconsistency occurs when responses to two questions for the same project are apparently in conflict.

Taken by itself, any questionnaire is limited by the focus of the questions asked. However, the standard SEI questionnaires can point the SCE team to a specific part of the critical subprocess area by identifying anomalies and inconsistencies. Figure C-6 on page 167 shows a sample Key Issue Worksheet.

The Key Issue Worksheet is created by the SCE team in Step 8. The information comes from the Key Issue Table and from the Questionnaire Worksheet.

The Key Issue Worksheet is used in Step 9 to develop the topic lists that will guide the interviews and document reviews for a specific development organization.

The *Critical Subprocess Areas* column of the Key Issue Worksheet is taken from the Key Issue Table (Figure C-6 on page 167). It lists the KPAs and the critical subprocess areas that will be investigated.

The second column shows why each subprocess area is important with regard to the specific development organization.[1] It is the same as the column from the Key Issue Table that shows the experience mismatches for the development organization (see Appendix C.6 on page 166).

There is also one column for each of the projects selected for evaluation. This column is used to record the results of reviewing the SCE Questionnaire Worksheet for inconsistencies and anomalies.

[1] On this form, the abbreviation “Ps” stands for “Product Size.” This is used to represent the “Size” attribute.

Anomalies and inconsistencies are recorded in the rows corresponding to the subprocess areas to which the question applies (see Figure C-9 on page 175). When an anomaly or inconsistency is found, an abbreviated summary of the response is recorded. It is sometimes handy to annotate the question numbers as well. An example of an anomaly and an inconsistency follow.

Key Issue Worksheet

Critical Subprocess Areas	**Sigma**	**Projects**		
	Tech	Able	Baker	Charlie
Requirements Management				
Establish and maintain requirements baseline	Ps			
Manage requirements-driven changes	Ps, *			
Software Project Planning				
Develop estimates	Ps			Inc: est. training
Plan software activities	Ps			
Make commitments	Ps, *			
Software Project Tracking and Oversight				
Manage commitment changes		Customer I/F	Customer I/F	
Track progress	Ps, *			
Take corrective action	Ps, *	issue trking		issue trking
Software Quality Assurance				
Plan SQA	Ps			
Perform SQA	Ps, *	An:CDRLs	An:CDRLs	An:CDRLs
Address noncompliance	Ps, *			
Software Configuration Management				
Create software work products baseline	*			
Control changes	Ps, *	CCB	CCB	CCB

Figure C-9: Sample Page of a Key Issue Worksheet

Anomaly: Consider the following question within the Software Quality Assurance key process area:

- Do SQA activities provide objective verification that software products and activities adhere to applicable standards, procedures, and requirements?

If this question is answered “yes” for three projects and “no” for one of the selected projects then that can be considered to be an *anomaly* in the organization in that SQA does not appear to be the same for all projects. This is what the “An: CDRL” entry refers to in the “perform SQA” subprocess area in Figure C-9 on page 175.

Inconsistency: Consider the following questions within the Software Configuration Management key process area:

- Does the project follow a documented procedure to control changes to configuration items/units?
- Are project personnel trained to perform the software configuration management activities for which they are responsible?

If one project responds “no” to the first question, and “yes” to the second question, then the team may consider this an inconsistency in that they may wonder about the quality and content of the training if there is no documented procedure to guide the change control activities.

C.10 Interview Worksheet

The Interview Worksheet is used as a guide for an interview with a specific person. It contains the KPAs and subprocess areas that are to be investigated for that person with questions that will be asked. The worksheet is used to record the responses to the interview questions. Figure C-6 below shows a sample Interview Worksheet.

Interview Worksheet

Interviewee’s Name: Position:	Date: Time:
Question	**Response**
Requirements Management Establish and maintain requirements baseline What is your role in maintaining the baseline requirements? How is the requirements baseline managed? Possible documents: policy and procedures for a CCB position description	
Requirements Management Manage requirements driven changes How are changes resulting from new requirements managed? How are changes tracked? Possible documents: CCB minutes, revised size and cost estimates, traceability matrix	
<KPA> <subprocess area> question 5 question 6 <possible document types>	

Figure C-10: Sample Page of an Interview Worksheet

The Interview Worksheets for the exploratory interviews are prepared by the SCE team in Step 11. Additional Interview Worksheets may be prepared during the team caucus sessions at the site interview as the need for follow-on interviews is determined. The information for the Interview Worksheets comes from the Validation Worksheets.

The Interview Worksheets are used by the team members to record the interview responses throughout the site visit.

The Interview Worksheet contains two columns. The first column contains the questions to be asked and any notes, such as types of documentation to request, that may be needed during the interview. This column also contains the KPA and subprocess area that the question is associated with. The second column is used to record the responses to the questions.

The header for the Interview Worksheet contains the position of the person being interviewed, the name of the person, and the time of the interview.

Appendix D Attribute Definitions

This appendix contains the definitions of the standard product and project attributes as they are used during the first three phases of the SCE method (Evaluation Start, General Preparation, and Specific Preparation). The attributes are used to specify important characteristics of a product or project so that comparisons can be made in a systematic way.

This appendix contains the following sections:

D.1 Major Attributes

The major attributes are used to compare previous experience on the part of the development organization and end user to the experience needed for the current development. This comparison is used to identify potential risk areas that should be looked at during the SCE. The major attributes are also given first consideration when selecting projects for evaluation.

The major attributes are used in creating the Target Product Profile, the Proposed Project Profile, the Project Profiles, the Mismatch Identification Table, the Experience Table, the Key Issue Table, and the Key Issue Worksheet. They are also used as a guide for selecting subprocess areas from the Subprocess Area Selection Tables shown in Appendix E.2 on page 188.

Application domain

The *application domain* attribute indicates the area of subject matter expertise needed to translate system requirements into software requirements.

There is no accepted taxonomy of application domains; however, the concept is widely understood and used. Information systems, command and control systems, weapon systems, simulation systems, training systems, avionic systems, sensing systems, and so on are all recognized and accepted as different application domains. What makes application domains different is the operational environment that uses the system. The unique characteristics of the operational environment are

- The mission for which the system is needed.

- The roles and responsibilities of the people who interface with the system.
- The resources that the system depends upon, which defines the potential limit of the services that the system can provide the people in the operational environment.

Product type

The *product type* attribute refers to the particular aspect of the application domain which the system will support or to the type of service which the system will provide. It may be considered a subset of the application domain.

For example, communications or displays could be product types in a command and control system, a weapons system or other application domain. Although there may be similarities in the communications subsystem in the various application domains, they each have their own set of unique problems which must be addressed.

Size

The size attribute is composed of three related attributes. The *contract duration* is the estimated or required length of time for the development of the software product. The *software team size* is the number of software developers who will be involved in the project. The *estimated software size* is the amount of code to be developed.

There is no standard way of measuring the size attributes. For the purposes of an SCE, the specific method used is not important as long as the method is used consistently so that comparisons will be meaningful.

This attribute was previously referred to as “Product Size,” and abbreviated “Ps”; in some of the materials the abbreviation “Ps” is still used.

Type of work

The *type of work* attribute is used to indicate the portion of the development life cycle which will be performed by the development organization. The life cycle can be an important consideration. For example, consider a maintenance shop planning a new software development that starts with requirements analysis and design. Because the development organization is proposing development activities for a portion of the life cycle that the organization does not have extensive experience with, there may be increased risk for the planned development.

The type of work attribute may indicate subprocess areas that should receive more or less emphasis during an SCE. Similar factors might apply if an organization was going to use a new life cycle model (or development methodology) for a planned development.

The following are examples of different types of work that may be required:

- full software development: The development organization is required to build a product based upon the system requirements. The development organization will typically be required to complete software requirements, top level design, detailed design, code and unit test, and acceptance testing at the development organization's site. The development scope is the same as or similar to the phases described in DoD-STD-2167A.
- code development only: The development organization is required to develop code according to the system requirements and software top level design provided by the issuing authority. This type of development might be done under a delivery order contract. The development organization may do the detailed design, coding, integration, and testing, but the system testing may be done by the customer.
- system development without coding: The development organization may be required to do all the work except the software detailed design and development.
- a prime contract acquisition: In a large system acquisition there may be many organizations who subcontract significant parts of the system, especially software parts. The prime contractor allocates system requirements to the subcontractor, integrates the components, and conducts acceptance tests.

Subcontractors

The *subcontractors* attribute is used to indicate whether the development organization plans to use subcontractors. If the development organization intends to use subcontractors for the planned development and does not have demonstrated experience using subcontractors, then this attribute is a potential risk. The lack of experience indicates that there may be risk in areas such as requirements management and software configuration management because of the additional coordination of effort required. If there are no plans to use subcontractors, then the lack of experience in subcontract management does not need to be considered.

The subcontractors attribute does not replace the Software Subcontract Management KPA of the CMM. The Software Subcontract Management KPA applies any time the development organization plans to use subcontractors for a major, separately managed portion of the software development, regardless of the development organization's experience with handling subcontractors. If the development organization lacks experience, the subcontractors attribute is used to indicate an even greater potential risk that applies to other KPAs and subprocess areas as well.

Operational precedence

The *operational precedence* attribute indicates whether the end user has previous experience with the type of system to be built. The values for this attribute are *no* (meaning operational precedence is not a factor—the end user has experience with similar systems), or *yes* (meaning the system is unprecedented to the end user.) Systems that are providing a new capability tend to have more changes to the requirements than systems that are replacing existing systems. The more unprecedented a system is, the more dynamic the requirements will be.

D.2 Minor Attributes

The minor attributes are used on the Target Product Profile, the Proposed Project Profile, the Project Profiles, the Mismatch Identification Table, and the Experience Table. They provide additional information which may be used in selecting projects for evaluation.

Language(s)

The *language* attribute indicates the programming languages in which the code is to be written, or in which it has been written.

Target

The *target* attribute indicates the hardware configuration that the developed software will run on when operational.

Applicable standards

The *applicable standards* attribute indicates the development standards that are imposed on the project such as DoD-STD-2167A, DoD-STD-2168, or MIL-STD-1521B.

Customer

The *customer* attribute indicates who the development is being done for. Examples include one of the DoD services or a particular market within industry.

Host development system

The *host system* attribute refers to the computer environment which will be used for the software development.

Configuration management tool

The *configuration management tool* attribute defines the tool set used on the host development system for supporting such activities as the software build process, baselining, and version control.

D.3 Schedule Attributes

The schedule attributes are used on the Project Profiles. They identify where the development organization is in relation to the project's schedule. The schedule attributes are used in selecting projects to be evaluated.

Current Phase

The *current phase* attribute refers to the life cycle phase of the development which the project is currently in, such as design, coding, integration, or acceptance testing.

Current Month

The *current month* attribute is the number of months since the start of the project.

Start

The *start* attribute shows when the project actually begins relative to the start of the contract.

Design Ends

The *design ends* attribute shows how long after the start of the project the design phase was completed or is scheduled to be completed.

Coding Ends

The *coding ends* attribute shows how long after the start of the project the coding phase was completed or is expected to be completed.

Appendix E CMM V1.1 Subprocess Area Selection Tables

This appendix contains information used to help SCE teams select critical subprocess areas for evaluation. Critical subprocess areas are selected in Step 5 Create Critical Subprocess Area List on page 55.

This appendix contains the following sections:

Section name	Section and page number
Selecting Critical Subprocess Areas Based on Size of the Development Undertaking	Section E.1, page 187
Selecting Critical Subprocess Areas Based on Experience Mismatches	Section E.2, page 188

There are several things the team should consider when selecting subprocess areas. General factors that should be considered in selecting critical subprocess areas include the following

- What processes would an organization need to manage the aspects of the project which are new to the organization?
- If the product being developed is new to the end user, what processes will the development organization need to manage the anticipated requirements changes?
- What are the basic processes that a development organization would need for any software development effort?

This appendix contains tables the teams can use to help select critical subprocess areas. The tables were created by SCE project members at the SEI for guidance only. SCE teams are expected to use their experience and judgement to select critical subprocess areas based on the requirements of the particular development.

There are two sets of tables, respectively based on

- The size of the development undertaking (Appendix E.1 on page 187.)
- Mismatches indicating a lack of experience either in the development organization or the end user of the system (Appendix E.2 on page 188.)

The size of the development undertaking can be used to select subprocess areas as critical, as described in Appendix E.2 on page 188.

Appendix E.2 on page 188 contains information that can be used two ways. First, the project profiles and the proposed project profile may indicate that a particular subprocess area is significant for the product to be acquired because of lack of experience in some attribute associated with developing the product to be acquired. These tables also indicate a recommended nucleus capability of subprocess areas that should be considered for every SCE.

E.1 Selecting Critical Subprocess Areas Based on Size of the Development Undertaking

This section contains tables that show the relationship between the number of levels of management within the development undertaking and candidate critical subprocess areas. The size of the development undertaking is indicated by the proposed project profile; information about the levels of management required for the project is found by examining information provided about the organizational structure. Table E-1 shows the relationship between subprocess areas and the size of the development undertaking.

Size of Development Undertaking	KPA	Subprocess Area Action
Major Undertaking (software manager has reports from two or more second-line software managers)	*Software Project Planning*	Develop documented estimates.
		Obtain agreement on planned commitments.
	Software Configuration Management	Identify selected software work products for a baseline, which is controlled and made available.
		Control changes to software baselines.
	Intergroup Coordination	Obtain agreement by affected groups on commitments between engineering groups.
	Integrated Software Management	Define project's software process by tailoring the organization's standard software process.
	Peer Reviews	Identify and remove defects in software work products.
Medium Undertaking (software manager has reports from two or more supervisors)	*Software Quality Assurance*	Verify adherence of activities and products to applicable standards
		Address non-compliance issues.
	Software Project Planning	Plan, and document software activities and commitments.
	Peer Reviews	Plan peer review activities.
	Software Project Tracking and Oversight	Take and manage corrective actions to reduce variance from plans.
		Obtain agreement on commitment changes.
Small Undertaking (software manager has reports only from software leads)	*Software Project Tracking and Oversight*	Track progress against software plans.
	Software Quality Assurance	Plan software quality assurance activities.
		Communicate SQA results.

Table E-1: Critical Subprocess Areas Based on Size of the Development

E.2 Selecting Critical Subprocess Areas Based on Experience Mismatches

The entries in this table represent consensus judgment from a group of experienced practitioners at the SEI. Selection of subprocess areas using these tables should be tempered by team judgment, experience, and detailed knowledge of the planned development.

How To Read the Tables

This section contains a table for each key process area (KPA) in the *Repeatable* and *Defined* levels. The tables contain the following columns.

KPA and Subprocess Areas Column
Each row under this column corresponds to a KPA or a subprocess area associated with the KPA. The KPAs are indicated by boldface type.

Major Attributes Columns (ApD, **Pt**, **Ps**, **Tw**, and **Sub)**
A black square (■) in the column for an attribute indicates that the subprocess area listed in that row may be important to the development organization for managing the risk associated with a lack of experience relative to that attribute. These columns correspond to the five major attributes from the Experience Table created in Step 4. The Experience Table shows where any of the development organizations may lack experience with regard to some attribute of the new project. Refer to Appendix D.1 on page 179 for a definition of each attribute.

Operational Precedence (Op) Column
A black square (■) in this column indicates that the subprocess area listed in that row may be important for managing the level of requirements changes which may be anticipated if end users do not have experience with similar products. The **Op** column corresponds to the *operational precedence* attribute from the Target Product Profile developed by the sponsor. This attribute indicates the degree to which the product being developed may be new to the end user. Refer to page 182 for a definition of the *operational precedence* attribute.

Nucleus Capability (*) Column
A black square (■) in this column indicates that the subprocess area listed in that row is part of the recommended *nucleus capability*. Nucleus capability refers to a basic set of subprocesses which are needed for almost any software development.

	Major Attributes					
Key Process Areas and Subprocess Areas	**ApD**	**Pt**	**Ps**	**Tw**	**Sub**	**Op**	*****
Requirements Management							
Establish and maintain requirements baseline	■		■	■	■	■	
Manage requirements-driven changes	■	■	■	■	■	■	■
Software Project Planning							
Develop estimates	■	■	■	■			
Plan software activities			■	■			
Make commitments			■	■	■		■
Software Project Tracking and Oversight							
Track progress			■	■			■
Take corrective action	■	■	■	■	■		■
Manage commitment changes							
Software Subcontract Management							
Select subcontractors					■		
Establish and maintain commitments		■		■	■		■
Maintain communications							
Track progress				■	■		■
Software Quality Assurance							
Plan SQA			■	■			
Perform SQA			■	■			■
Communicate results					■		
Address noncompliance	■	■	■		■		■

Table E-2: Subprocess Area Selection Table for the Repeatable Level KPAs

Key					
Bold	Key process area	*Italics*	Subprocess area	ApD	Application Domain
Pt	Product Type	Ps	Size	Tw	Type of Work
Sub	Subcontracting	Op	Operational Precedence	*	Nucleus Capability

	Major Attributes						
Key Process Areas and Subprocess Areas	**ApD**	**Pt**	**Ps**	**Tw**	**Sub**	**Op**	*****
Software Configuration Management							
Plan SCM				■			
Create software work products baselines	■	■		■	■	■	■
Control changes	■	■	■		■	■	■
Report status							

Table E-2: Subprocess Area Selection Table for the Repeatable Level KPAs

	Major Attributes						
Key Process Areas and Subprocess Areas	**ApD**	**Pt**	**Ps**	**Tw**	**Sub**	**Op**	*****
Organization Process Focus							
Coordinate software process activities	■		■	■	■		
Assess software processes used							■
Plan SPI	■			■			
Organization Process Definition							
Provide standard process							
Retain software process information							■
Software Product Engineering							
Build software	■	■	■	■	■	■	■
Ensure consistency							
Integrated Software Management							
Define project process	■	■	■	■			
Manage according to process							■
Intergroup Coordination							
Agree on customer’s requirements	■	■	■	■	■	■	
Coordinate intergroup commitments			■	■			
Manage intergroup issues							■
Peer Reviews							
Plan peer review							■
Identify and remove defects	■	■	■	■			■

Table E-3: Subprocess Area Selection Table for the Defined Level KPAs

Key					
Bold	Key process area	*Italics*	Subprocess area	ApD	Application Domain
Pt	Product Type	Ps	Size	Tw	Type of Work
Sub	Subcontracting	Op	Operational Precedence	*	Nucleus Capability

	Major Attributes						
Key Process Areas and Subprocess Areas	**ApD**	**Pt**	**Ps**	**Tw**	**Sub**	**Op**	*****
Training Program							
Plan training							■
Provide training							
Receive necessary training	■	■	■	■			■

Table E-3: Subprocess Area Selection Table for the Defined Level KPAs

	Major Attributes						
Key Process Areas and Subprocess Areas	**ApD**	**Pt**	**Ps**	**Tw**	**Sub**	**Op**	*****
Quantitative Process Management							
Plan QPM							
Control process quantitatively	■	■	■				
Establish organization's process capability							
Software Quality Management							
Plan quality management							
Define software quality goals							
Track quality progress	■	■	■				

Table E-4: Subprocess Area Selection Table for the Managed Level KPAs

	Major Attributes						
Key Process Areas and Subprocess Areas	**ApD**	**Pt**	**Ps**	**Tw**	**Sub**	**Op**	*****
Defect Prevention							
Plan defect prevention							■
Identify defect causes							
Eliminate defect causes	■	■	■	■			
Technology Change Management							
Plan technology changes	■	■					
Evaluate new technologies	■	■					
Adopt new technology							
Process Change Management							
Plan process improvement							
Empower everyone							
Continuously improve							

Table E-5: Subprocess Area Selection Table for the Optimizing Level KPAs

Key					
Bold	Key process area	*Italics*	Subprocess area	ApD	Application Domain
Pt	Product Type	Ps	Size	Tw	Type of Work
Sub	Subcontracting	Op	Operational Precedence	*	Nucleus Capability

Appendix F References

[Crosby 79] Crosby, P. B. *Quality is Free*. New York: McGraw-Hill, 1979.

[Deming 86] Deming, W. Edwards. *Out of the Crisis.* Cambridge, Ma.: MIT Center for Advanced Engineering Study, 1986.

[Humphrey 87a] Humphrey, W. S. *Characterizing the Software Process: A Maturity Framework* (CMU/SEI-87-TR-11, ADA182895). Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University, 1987.

[Humphrey 87b] Humphrey, W. S., and W. L. Sweet. *A Method for Assessing the Software Engineering Capability of Contractors* (CMU/SEI-87-TR-23, ADA187230). Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University, 1987.

[Humphrey 89] Humphrey, W. S. *Managing the Software Process.* Reading, Ma.: Addison-Wesley, 1989.

[IEEE 91] ANSI/IEEE Std 610.12-1990, *IEEE Standard Glossary of Software Engineering Terminology*. February 1991.

[Juran 88] Juran, J. M. *Juran on Planning for Quality*. New York: Macmillan, 1988.

[Juran 89] Juran, J. M. *Juran on Leadership for Quality.* New York: The Free Press, 1989.

[Paulk 91] Paulk, Mark C., et al. *Capability Maturity Model for Software* (CMU/SEI-91-TR-24, ADA240603). Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University, 1991

[Paulk 93a] Paulk, Mark C., et al. *Capability Maturity Model for Software, Version 1.1* (CMU/SEI-93-TR-24). Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University, 1993.

[Paulk 93b] Paulk, Mark C., et al. *Key Practices of the Capability Maturity Model, Version 1.1* (CMU/SEI-93-TR-25). Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University, 1993.

[SCE 93] SCE Project, *Software Capability Evaluation Version 1.5 Method Description* (CMU/SEI-93-TR-17). Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University, 1993.

[SCE 94] Members of the CMM-Based Appraisal Project, *Software Capability Evaluation Version 2.0 Implementation Guide* (CMU/SEI-94-TR-05). Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University, 1994.

Appendix G Glossary

Acquisition agency: an organization in charge of a government procurement effort. For purposes of this document, an acquisition agency is the sponsoring organization using the SCE method for a source selection.

Applicable standards: a minor attribute used in SCE. This attribute indicates the development standards that are imposed on the project such as DoD-STD-2167A, DoD-STD-2168, or MIL-STD-1521B.

Application of the SCE method: synonym for *use of the SCE method*.

Application domain: a major attribute used in SCE. An application domain is “a bounded set of related systems (i.e., systems that address a particular type of problem). Development and maintenance in an application domain usually requires special skills and/or resources. Examples include payroll and personnel systems, command and control systems, compilers, and expert systems” [Paulk 93b]. For SCE, this is a major attribute used within the various profiles. *The application domain* attribute indicates the area of subject matter expertise needed to translate system requirements into software requirements, and indicates significant differences in the engineering practices which transform the software requirements into accepted code.

Attributes: characteristics of a software product or project. For purposes of an SCE, there are three categories of attributes: major attributes, minor attributes, and schedule attributes. The attributes used in SCE are defined and discussed in Appendix D on page 179.

Candidate findings: findings for which there is not yet enough objective evidence to make a decision.

Caucus: SCE teams participate in three types of caucuses, or meetings, during an SCE:

Ongoing team caucus (Step 16): a meeting in which SCE team members analyze, share, and consolidate information in order to reach conclusions about what was seen and heard as a result of probing the implementation of a subprocess area.

Preliminary findings caucus (Step 18): a meeting in which team members articulate conclusions about the subprocess areas based on the information available.

Findings caucus (Step 22): a meeting in which the team analyzes information they have learned to date, including the consolidation interviews and Final Document Review to determine whether the information confirms or negates any of the preliminary findings.

Capability Maturity Model (CMM): “a description of the stages through which software organizations evolve as they define, implement, measure, control, and improve their software processes” [Paulk 93b]. For SCE this is a model consisting of five maturity levels and associated key process areas (KPAs) which are used for evaluating a development organization’s software process capability. (See also *maturity model*.)

Common feature: “an attribute that indicates whether the implementation and institutionalization of a key practice is effective, repeatable, and lasting” [Paulk 93b]. There are five common features defined for CMM v1.1: commitment to perform, ability to perform, activities performed, measurement and analysis, and verifying implementation.

Configuration management tool: a minor attribute in SCE. This attribute defines the tool set used on the host development system for supporting such activities as the software build process, baselining, and version control.

Contract monitoring: one of the two primary applications of the SCE method. In contract monitoring, SCE can serve as an input for an incentive/award fee or can be used to help the sponsoring organization tailor its contract monitoring efforts based on the observed strengths and weaknesses of the development organization’s processes.

Critical subprocess area: a subprocess area that is selected by the team for evaluation. A critical subprocess area is selected from within a Target Process Capability KPA. The set of all critical subprocess areas is the Critical Subprocess Area List, and will be investigated at all development organization sites. Collectively, the critical subprocess areas define the scope of the SCE.

Customer: a minor attribute in SCE. This attribute indicates who the development is being done for. Examples include one of the DoD services or a particular market within industry**.**

Development organization: an organization that develops and/or maintains software products, which is also the recipient of an SCE.

Development organization community: all of the development organizations that are involved with a particular use of the method. In a source selection these are the offerors (or all of the offerors remaining after a competitive range determination), and possibly their subcontractors.

Directive: an order or instruction describing actions that must be performed and authorizing their performance.

Document review: the process of examining documents to find evidence of the processes used by a development organization. Documents can define and standardize processes, can indicate commitment to use the processes, can provide an audit trail of processes that were used, and can collect data about process performance. Three levels of documents are reviewed during an SCE: *organization-level*, *project-level*, and *implementation-level*.

Feature: one of a set of attributes that provide a view of “whether the implementation and institutionalization of a key practice are effective, repeatable, and lasting” [Paulk 93b]. The features used in SCE are a refinement of the common features of CMM v1.1; they add a level of detail that is more appropriate to the SCE Method. Examples of features are ability to perform, organizational structures, training, plans and procedures, etc. Features are defined in Appendix A on page 129. (See *common feature*.)

Final findings: output from executing the SCE method. Final findings are used to develop the formal findings report.

Findings: includes preliminary findings, candidate findings and final findings. Findings are strengths, weaknesses, or improvement activities. In some cases, an explicit finding of “no finding” can be generated. For example, if there are no subcontractors planned to be used for a development, and no subcontractors are involved with the projects that are evaluated, then a “no finding” would result for the subprocess areas that deal with subcontractor management.

Host development system: a minor attribute in SCE. This attribute refers to the computer environment which will be used for the software development**.**

Implementation-level documents: the third of three levels of documents reviewed during an SCE. These are documents which provide an audit trail of processes that were used, and can be used by the development organization to collect data about process performance.

Improvement activity: a process improvement that is not yet institutionalized—for example, a pilot program that implements a new configuration management process. In SCE, it indicates potential mitigation of risk due to software process.

Interviewing: the process of questioning personnel from the development organization to find evidence of the processes used by the development organization. During an SCE, the SCE team typically interviews one person at a time. Interviews provide insight into how processes are implemented and show the extent to which processes have been internalized by members of the development organization.

Key issue: the relationship between a critical subprocess area on the Critical Subprocess Area List and a development organization or organizations. The subprocess area is a key issue for the development organization

- If there is information known about the development organization that relates it specifically to that critical subprocess area. As examples, this can happen because of a mismatch in the Mismatch Identification Table or because the organizational charts indicate a possible risk. These observations could cause the team to identify a particular subprocess area as a key issue that needs to be probed.
- If the subprocess area has been selected as a key issue for all development organizations. As examples, this could happen because the operational precedence attribute in the Target Product Profile caused the team to identify a subprocess area as a key issue that needed to be probed, or because the subprocess area was part of the nucleus capability.

Key process area (KPA): “a cluster of related activities that, when performed collectively, achieve a set of goals considered important for establishing process capability” [Paulk 93b]. Each KPA contributes to the environment in which development organizations create software products. Within the CMM, the KPAs are organized into five basic levels of process maturity to describe the progression from an ad hoc software process to one that is well defined and can act as a stable foundation for continuous process improvement.

Language(s): a minor attribute for SCE. This attribute indicates the programming languages in which the code is to be written, or in which it has been written.

Mapping: the relationship between actual practices in the software process implementation and the KPAs.

Maturity level: “a well-defined evolutionary plateau toward achieving a mature software process” [Paulk 93b]**.**

Maturity model: a model consisting of five maturity levels and associated Key Process Areas (KPAs) which are used for evaluating a development organization’s software process capability. The maturity model was used in previous versions of SCE, but is *not* used in version 2.0 (the version defined in this document). The maturity model was based on the process maturity framework defined in *Characterizing the Software Process: A Maturity Framework* [Humphrey 87b], and predates the published Capability Maturity Model (CMM) [Paulk 93a].

Operational Precedence: a major attribute used in SCE. This attribute indicates whether the end user has previous experience with the type of system to be built. Systems that are providing a new capability tend to have more changes to the requirements than do ones that are replacing existing systems.

Organization-level documents: the first (or top) level of three levels of documents reviewed during an SCE. These are the policies and procedures which establish the development environment for all company project activities. Organizational level documents define the process and management constraints the organization places on projects.

Policy: “a guiding principle, typically established by senior management, adopted by an organization to influence and determine decisions” [Paulk 93b].

Preliminary findings: findings for a subprocess area generated during caucus. These represent SCE team consensus about a subprocess area or *KPA*, and remove the area from further consideration during the site visit. These are the basis for the final findings.

Procedure: a written description of a course of action to be taken to perform a given task [IEEE 91].

Process capability: “the range of expected results that can be achieved by following a process” [Paulk 93b].

Product Type: a major attribute in SCE. The product type attribute refers to the particular aspect of the application domain which the system will support or to the type of service which the system will provide. For example, displays or communications could be product types in a command and control system, a weapons system, or another application domain. Although there may be similarities in the communications subsystem in the various application domains, they each have their own set of unique problems which must be addressed**.**

Profiles: a profile is the set of attributes (such as the major attributes Application Domain, Product Type, and Size) associated with a software product and the project that develops the product. There are three types of profiles used in SCE: Target Product Profiles, Proposed Project Profiles, and Project Profiles. The Target Product Profile represents the “customer view” of the product to be built, and captures the attributes of the desired product. The

Proposed Project Profile represents the development organization's view of the planned development. Project Profiles represent the actual attributes of ongoing or recently completed projects.

Project-level documents: the second of three levels of documents reviewed during an SCE. These are documents which define the development processes in use for a particular project. Project level documents define the detailed processes that are used to manage, coordinate, and integrate the engineering activities required for the development.

Project Profile: see *Profiles***.**

Proposed Project Profile: see *Profiles***.**

Request for Proposal (RFP): an acquisition document that describes characteristics of the system the government wants to acquire. This document is used to solicit proposals from commercial development organizations (offerors) and to communicate the characteristics of the desired system to the offerors. In source selection, this is the document that specifies that an SCE will be performed.

Results: how the findings are used by the sponsoring organization—for example, in risk determination for source selection.

Scope of SCE: the boundaries of the investigation, in terms of critical subprocess areas within the KPAs in the Target Process Capability. Items outside the defined scope of the SCE can't be looked at during source selection.

Site visit: an investigation conducted by four to six government personnel (the SCE team) over a three day period at a development organization's facility.

Size: a major attribute for SCE. The size attribute indicates the magnitude of the product (and hence the required project). Size is composed of three related attributes. The *contract duration* is the estimated or required length of time for the development of the software product. The *software team size* is the number of software developers who will be involved in the project. The *estimated software size* is the amount of code to be developed.

Software Capability Evaluation (SCE): a method for evaluating the software process of an organization to gain insight into its software development capability.

Software development plan (SDP): "the collection of plans that describe the activities to be performed for the software project" [Paulk 93b]. This could be, but is not necessarily the same document referred to in DoD-STD-2167A.

Software process capability: “the range of expected results that can be achieved by following a process” [Paulk 93b]. For purposes of an SCE, those CMM-related processes which provide a detailed environment for one or more development teams to produce software products. The processes evaluated include decision making processes (such as software project planning) and communication processes (such as peer reviews).

Software process implementation: a tailored set of practices that defines how software development work is supposed to be done.

Source selection: one of the two primary applications of the SCE method. In source selection, the results of the SCE are used by the sponsoring organization to characterize the software process-related risk of awarding a contract to an offeror. SCE is only one criterion among many used to select software contractors in government acquisitions.

Sponsoring organization: the organization that commissions the SCE to be performed and uses the findings.

Standard: “mandatory requirements employed and enforced to prescribe a disciplined, uniform approach to software development” [Paulk 93b].

Strength: in SCE, strength indicates a particular part of the software process capability that is sufficiently robust to mitigate the development risks due to software process.

Subcontractor: a development organization that is contracted to work for another development organization to produce software products.

Subcontractors: a major attribute in SCE. This attribute is used to indicate whether the development organization intends to use subcontractors in the development, and is a factor if they lack experience with subcontract management.

Subprocess area: a set of activities in an implemented process that, acting together, helps an organization to achieve one of the goals of a KPA. Alternatively, a focused subset of process activities that work toward achieving a specific KPA goal. This is a subdivision of a KPA that addresses a major process activity within the larger cluster of related activities that make up the KPA. The KPA goals represent desired states; subprocess areas encapsulate the activities needed to achieve those states. The Critical Subprocess Area List is a set of subprocess areas which collectively define the scope of the SCE.

Target: a minor attribute in SCE. This attribute indicates the hardware configuration that the developed software will run on when operational.

Target Process Capability: the process capability that is most appropriate for the planned development; the process capability desired by the sponsoring organization for the product to be developed. The Target Process capability consists of a set of KPAs, and establishes the boundaries of the SCE investigation—a KPA is evaluated if and only if it is part of the Target Process Capability.

Target Product Profile: see *Profiles***.**

Topic: a topic defines a subject that will be probed during the investigation. A topic is an abstraction of a work practice that corresponds to a portion of the process implementation for the development organization. Topics are intended to be detailed enough to focus the investigation on observable, documented work practices, but sufficiently abstract that they avoid prescribing how the subprocess area is implemented. Topics are selected by considering the features associated with each subprocess area.

Type of Work: a major attribute for SCE. This attribute indicates the portion of the development life cycle which will be performed. As examples of different types of work, in “full software development” a development organization is required to build a product based on the system requirements, while in “code development only” the development organization is required to develop code according to the system requirements and software top level design provided by the issuing authority.

Use of the SCE method: executing the SCE method within a particular context. To date, the two primary uses of the SCE method are in source selection and contract monitoring. This is sometimes referred to as the application of the method.

Weakness: In SCE, weakness indicates a particular part of the software process capability that has characteristics that increase the risks due to software process.

Index

Key
italicized page number: item appears in a table or figure
'n' preceding page number: item appears in a footnote

Key
italicized page number: item appears in a table or figure
'n' preceding page number: item appears in a footnote

Key
italicized page number: item appears in a table or figure
'n' preceding page number: item appears in a footnote

UNLIMITED, UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

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)
CMU/SEI-94-TR-6

5. MONITORING ORGANIZATION REPORT NUMBER(S)
ESC-TR-94-006

6a. NAME OF PERFORMING ORGANIZATION
Software Engineering Institute

6b. OFFICE SYMBOL (if applicable)
SEI

7a. NAME OF MONITORING ORGANIZATION
SEI Joint Program Office

6c. ADDRESS (city, state, and zip code)
Carnegie Mellon University
Pittsburgh PA 15213

7b. ADDRESS (city, state, and zip code)
HQ ESC/ENS
5 Eglin Street
Hanscom AFB, MA 01731-2116

8a. NAME OFFUNDING/SPONSORING ORGANIZATION
SEI Joint Program Office

8b. OFFICE SYMBOL (if applicable)
ESC/ENS

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
F1962890C0003

8c. ADDRESS (city, state, and zip code))
Carnegie Mellon University
Pittsburgh PA 15213

10. SOURCE OF FUNDING NOS.

PROGRAM ELEMENT NO	PROJECT NO.	TASK NO	WORK UNIT NO.
63756E	N/A	N/A	N/A

11. TITLE (Include Security Classification)
Software Capabilty Evaluation Version 2.0 Method Description

12. PERSONAL AUTHOR(S)
CMM-Based Appraisal Project

13a. TYPE OF REPORT
Final

13b. TIME COVERED
FROM TO

14. DATE OF REPORT (year, month, day)
June 1994

15. PAGE COUNT

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB. GR.

18. SUBJECT TERMS (continue on reverse of necessary and identify by block number)

assembly language
components
connectors
procedure calls
programming
software specifications

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

This report describes Version 2.0 of the Software Capability Evaluation (SCE) Method, as taught at the Software Engineering Institute (SEI) from fourth quarter 1993. This version of the SCE Method is based on the Capability Maturity Model defined in *Capability Maturity Model for Software, Version 1.1* [Paulk 93a]. The document includes an overview of the SCE Method and its evolution, a detailed description of the activities performed during an SCE, and a discussion of the characteristics of the method and their implications for the use of the method. This document provides a new baseline for future evolution of the SCE Method.

(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)

EDITION of 1 JAN 73 IS OBSOLETE

UNLIMITED, UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

ABSTRACT — continued from page one, block 19