Software Engineering Institute

A Method for Assessing the Software Engineering Capability of Contractors

Watts Humphrey
W.L. Sweet
R.K. Edwards
G.R. LaCroix
M.F. Owens
H.P. Schulz

September 1987

http://www.sei.cmu.edu

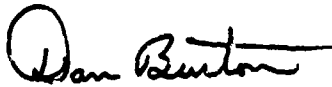**CarnegieMellon**

This technical report was prepared for the

SEI Joint Program Office
ESD/XRS
Hanscom AFB, MA 01731

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

Review and Approval

This report has been reviewed and is approved for publication.

FOR THE COMMANDER

SEI Joint Program Office

Daniel Burton
SEI Joint Program Office

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

Table of Contents

A Method for Assessing the Software Engineering Capability of Contractors

Abstract: This document provides guidelines and procedures for assessing the ability of potential DoD contractors to develop software in accordance with modern software engineering methods. It includes specific questions and a method for evaluating the results.

General Introduction

The purpose of this document is to facilitate objective and consistent assessments of the ability of potential DoD contractors to develop software in accordance with modern software engineering methods. Such assessments would be conducted either in the pre-solicitation qualification process, in the formal source selection process, or both. While this document is intended to guide the assessment of a contractor's overall software engineering capability, it can also be valuable in the assessment of a specific project team's software engineering capability.

Alternatively, this document can be used as an aid to software development organizations in conducting an internal assessment of their own software engineering capability. The document is designed to help an assessment team define the highest priority steps for the improvement of an organization's capability.

Because an understanding of proper software engineering practice is only now developing, standard, well-accepted measures do not yet exist. The assessment questions listed in the body of this document are phrased so that an affirmative answer indicates that an organization has a desirable characteristic. Some of the questions pertain to advanced concepts of software engineering that may not yet be sufficiently refined or disseminated to be incorporated in a contractor's standard practice; therefore, not all assessment questions need be answered affirmatively for an organization to be considered to have a modern software engineering capability.

The capability of a contractor to perform software engineering has been divided into three areas:

1. organization and resource management
2. software engineering process and its management
3. tools and technology.

The qualities that the questions assess are different for each of these areas and are described in the introductions to the questions for each area.

A full assessment of software engineering capability includes some evaluation of the experience level of the software development personnel. Addendum A contains suggested questions for use in this evaluation.

General Approach

This guideline was developed, at the request of the United States Air Force, by the Software Engineering Institute of Carnegie Mellon University with assistance from The MITRE Corporation. The motivation for this work was the increasing importance of software in DoD procurements and the need of all the services to more effectively evaluate the ability of their software contractors to competently perform on software engineering contracts.

A structured assessment approach has been developed to augment the current contractor evaluation methods. The primary objective has been to provide a standardized method that is documented, publicly available for review and comment, and periodically modified as experience is gained with its use.

A further objective is to provide a public process which is defined in advance and for which the contractors can prepare. This assessment guide has therefore been designed to assist software organizations in identifying areas where they should make improvements in their own capabilities. As contractors improve their ability to meet the needs of the services for quality software, the services will improve their ability to serve the national interest by awarding contracts to those with the best capability.

Assessment methodology is based on the principal that prior experience is a good predictor of future performance. Since there are exceptions to this principle, the guidelines suggest that procurement evaluations using this method consider both current capability and future plans for software process improvement.

This method should be used to augment the many steps currently involved in source selection. While the questionnaire structure provides a relatively simplistic numerical evaluation, it also indicates the strong and weak areas of a contractor's software process. This will provide the services with more information on which to base their procurement decisions.

Technical Approach

The assessment process is focused on defining and clarifying the positive attributes of good software engineering practices. It is further recognized that the state-of-the-practice of software engineering is steadily advancing and that additional criteria and a higher level of expectation will be appropriate for judging software engineering capability in the future.

Assessment questions are based on the following premises:

- The quality of a software product stems, in large part, from the quality of the process used to create it.

- Software engineering is a process that can be managed, measured, and progressively improved.

- The quality of a software process is affected by the technology used to support it.

- The level of technology used in software engineering should be appropriate to the maturity of the process.

- Software products developed by contractors for DoD use are acquired under contracts invoking DoD-STD-2167/A, Defense System Software Development, as tailored for each contract.

To provide a structure for assessment, five levels of process maturity and two stages of technology advancement have been postulated. (See Addenda B and C.)

Process Maturity Levels

1. Initial: The initial environment has ill-defined procedures and controls. The organization does not consistently apply software engineering management to the process, nor does it use modern tools and technology. Level 1 organizations may have serious cost and schedule problems.

2. Repeatable: At Level 2, the organization has generally learned to manage costs and schedules, and the process is now repeatable. The organization uses standard methods and practices for managing software development activities such as cost estimating, scheduling, requirements changes, code changes, and status reviews

3. Defined: In Level 3, the process is well characterized and reasonably well understood. The organization defines its process in terms of software engineering standards and methods, and it has made a series of organizational and methodological improvements. These specifically include design and code reviews, training programs for programmers and review leaders, and increased organizational focus on software engineering. A major improvement in this phase is the establishment and staffing of a software engineering process group that focuses on the software engineering process and the adequacy with which it is implemented.

4. Managed: In Level 4, the process is not only understood but it is quantified, measured, and reasonably well controlled. The organization typically bases its operating decisions on quantitative process data, and conducts extensive analyses of the data gathered during software engineering reviews and tests. Tools are used increasingly to control and manage the design process as well as to support data gathering and analysis. The organization is learning to project expected errors with reasonable accuracy.

5. Optimized: At Level 5, organizations have not only achieved a high degree of control over their process, they have a major focus on improving and optimizing its operation. This includes more sophisticated analyses of the error and cost data gathered during the process as well as the introduction of comprehensive error cause analysis and prevention studies. The data on the process are used iteratively to improve the process and achieve optimum performance.

Software Technology Stages

- A. Inefficient: Multiple implementations may be available and the practice may be in widespread use, but the technology is no longer effective. An organization that primarily employs inefficient software development technology is likely to be ineffective in developing software. Moreover, at this technology stage some important software engineering practices are not practical in large, complex developments.

- B. Basic: Multiple implementations are available, and they have been demonstrated to be effective. An organization that primarily employs basic software development technologies is likely to be moderately effective and, depending upon the maturity of its process, reasonably consistent in its performance.

Usage Guide

This document is intended for use by DoD development and procurement organizations to assess contractors' software engineering capabilities. When used as part of the formal DoD systems acquisition process, the questions are furnished, for information purposes, to potential contractors with the Request for Proposal (RFP). A qualified assessment team then visits each contractor to obtain responses to the assessment questions and assure accuracy and consistency of interpretation. The assessment results are included in the source selection process as information for the Source Selection Advisory Council.

The effectiveness of an assessment is critically dependent on the process used in the assessment and on the background and training of the personnel conducting it. The following guidelines are recommended for use oy procurement agencies for incorporating software capability assessments into the source selection process.

1. **Materials**
   The following basic documents are to be used:

   - "A Method for Assessing the Software Engineering Capability of Contractors"

   - the Assessment Recording Form (Addendum D)

   - the guideline for further questions (Addendum E)

   - available training guides and materials.

2. **RFP Content**
   When assessment results will be considered in source selection, a statement of this fact and the above materials must be included with the Request for Proposal.

3. **General Assessment Procedure**
   The answers to the assessment questions are not submitted with the proposal but are provided to an assessment team that visits each contending contractor during the proposal evaluation period. Using the follow-up questions in Addendum E as a guide, the assessment team clarifies what is meant by the responses to the questionnaire. Normally, at least three working days should be scheduled for an assessment to allow for reviewing the questions, obtaining and discussing back-up material, demonstrating support tools, and presenting conclusions. A single assessment team should visit all of the contending contractors to assure consistent interpretation of both the questions and the results.

4. **Selection of Assessment Team Members**
   The assessment team must have a mix of talents. Experienced professionals are required, including professionals knowledgeable in the software development process, the technology, the application area, and the specific procurement. All team members must have been trained in the assessment process.

5. **Assessment Training**
   The training program involves several days of classroom instruction to review the assessment questionnaire in detail and discuss the materials and support tools that should be available to demonstrate performance for each question.

6. **Contractor Preparation for Assessment**
   While making advance arrangements, the assessment team should ask each contractor to provide a listing of the major software development projects at the location, together with a brief indication of their status (e.g., design, implementation, development test, acceptance test). Projects recommended for assessment should also be noted. The assessment team and the contractor should agree in advance on several projects, in different stages of development and indicative of the standard practice in the organization, so that representatives of these projects can be available for participation in the assessment.

7. **Conduct of the Assessment**
   An on-site assessment begins with a briefing explaining the assessment process to the local management and the assessment participants and confirming the planned support for the assessment. The assessment team then goes through the questionnaire with the project representatives as a group, ensuring consistent interpretation of the questions and obtaining an initial set of answers for each project. Based on these initial results, the team makes a preliminary assessment of the organization's process maturity level and technology stage and then requests back-up materials and tool demonstrations to support the affirmative answers that determine the highest likely level and stage. For example, if the preliminary evaluation results (see the following section) indicate that an organization is at maturity level 3, the major focus should be directed to probing the affirmative responses to the maturity level 2 and 3 questions. In each case, the team should request evidence for a specific project at an appropriate phase of development.

8. **Assessment Conclusion**
   At the end of the assessment, the local management should be informed of the findings and given an opportunity to offer evidence to refute any disputed findings and to explain their plans for process improvement. Where such plans are material to the procurement, they should be documented and made part of the contract. It is important that the process be completely open because the complexity of the subject matter and the lack of common terms for many of the process elements could lead to confusion and misunderstanding.

9. **Utilization of Results**
   The results of the assessments will be made available to the Source Selection Advisory Council for consideration prior to final source selection.

Guidelines for Evaluation of Results

The questions in the body of this document have been designed to require only a "yes" or "no" answer. The method of evaluation presented here incorporates all the questions in this document except those in Addendum A. The questions in Addendum A are provided to assist in the assessment of a contractor's experience relevant to a particular procurement.

Level of Process Maturity

To determine a contractor's level of process maturity, the following procedure is used. This procedure requires successive qualifications at each level.

1. Determine the percentage of affirmative answers to all Level 2 questions and to the asterisked questions for Level 2. If the percentage* of affirmative answers to all questions is at least 80% and the percentage of affirmative answers to asterisked questions is at least 90%, the organization has qualified at Level 2; otherwise, it is at Level 1. If Level 2 is achieved, go on to the next step.

2. Determine the percentage of affirmative answers to all Level 2 and 3 questions combined and to the asterisked questions for Levels 2 and 3 combined. Again, if the percentage of affirmative answers to all questions is at least 80% and the percentage of affirmative answers to asterisked questions is at least 90%, the organization qualifies at Level 3, otherwise, it is at Level 2. If it qualifies at Level 3, this procedure is repeated combining Level 2, 3, and 4 answers, again requiring 80% for all questions and 90% for asterisked questions. If the organization qualifies at Level 4, the assessment for Level 5 combines Level 2, 3, 4, and 5 answers, again using 80% and 90% as the criteria.

3. Determine the level for the organization as a whole by averaging the levels of the projects assessed.

Software Technology Stages

To determine the technology stage of an organization, a similar procedure is used.

1. Determine the percentage of affirmative answers to all Stage B questions and to the asterisked questions for Stage B. If the percentage of affirmative answers to all questions is at least 80% and the percentage of affirmative answers to asterisked questions is at least 90%, the organization has qualified at Stage B; otherwise, it is at Stage A.

2. Determine the level for the organization as a whole by averaging the levels of the projects assessed.

Combined Process and Technology Evaluation

By placing the levels of process maturity and the stages of technology in a two dimensional matrix, an evaluation can now be made that combines both of these measures. Figure 1

---

*Threshold percentages have been arbitrarily established to promote consistency and objectivity.

presents process levels on the x-axis and technology stages on the y-axis, and indicates the target region toward which an organization should progress.

Figure 1: Process/Technology Matrix

Qualifying Considerations

As previously noted, the practice of software engineering is not only complex but is still evolving and is not yet fully defined. In using a specific procedure to assess software engineering capability, some qualifying factors should be considered.

It is recognized that there may be alternative methods to address a given problem, and it is possible that there may be acceptable alternatives to some of the positions taken in this document. Therefore, it is essential that this instrument be used by a competent and adequately trained assessment team if meaningful results are to be obtained. The SEI intends to provide, on a continuing basis, training and/or training materials to facilitate the training of assessment teams.

The process activities and data referred to in the questions are used as indicators of software engineering capability and are assumed to be of value to the internal operations of an organization that develops/maintains significant amounts of DoD software. It is not intended that either the process activities or data be identified as deliverable items in a procurement contract solely because they are referenced in this document. The cost-effectiveness of these activities may vary with different organizations; but available evidence clearly indicates that in the context of total life-cycle cost and performance, investment in these activities is well justified. In this document, software engineering capability is assumed to include the ability to perform large and complex software developments; therefore, the assessment process may not be fully applicable to small projects.

The authors of this document have established, on the basis of extensive experience in software development and acquisition, that the state-of-practice is measurable and that this state can be compared to a norm. This instrument will be used initially to establish the norm. The SEI intends to continue monitoring the use and evolution of this methodology to insure that it is consistent with best current software engineering practice and technology and to correct, whenever possible, those areas where its misuse may be causing problems.

The answers to the questions should reflect standard organizational practice.

Questions

In order to achieve clarity in the questions, many of the terms used have been given specific explanatory definitions in the glossary at the end of this document. Each use of a glossary term in the Questions section is italicized. Adherence to these definitions is essential for proper and consistent assessments. There is no significance to the order of the questions.

1. Organization and Resource Management

This section deals with functional responsibilities, personnel, and other resources and facilities. Its purpose is to define the magnitude, quality, and structure of the software engineering organization. The questions focus on responsibilities and the quality and quantity of resources.

The major responsibility concerns relate to quality assurance, process management, and configuration control. The intent is to ascertain whether these functional responsibilities are clearly delineated and assigned, not necessarily that an individual is assigned full time to each.

1.1. Organizational Structure

1.1.1.   For each project involving software development, is there a designated software manager?

1.1.2.   Does the project software manager report directly to the project (or project development) manager?

1.1.3.   Does the Software Quality Assurance (SQA) function have a management reporting channel separate from the software development project management?

1.1.4.   Is there a designated individual or team responsible for the control of software interfaces?

1.1.5.   Is software system engineering represented on the system design team?

1.1.6.   Is there a software configuration control function for each project that involves software development?

1.1.7.   Is there a software engineering *process group* function?

1.2. Resources, Personnel, and Training

The questions on resources concern software engineering training, process training, and adequacy of the support facilities.

1.2.1.   Does each software developer have a private computer-supported workstation/terminal?

1.2.2.   Is there a required training program for all newly appointed development managers
            designed to familiarize them with software project management?

1.2.3.   Is there a required software engineering training program for software developers?

1.2.4.   Is there a required software engineering training program for first-line supervisors
            of software development?

1.2.5.   Is a formal training program required for design and code *review leaders*?

1.3. Technology Management

The questions on technology management relate to the mechanisms used to introduce and
control new technologies.

1.3.1.   Is a *mechanism* used for maintaining awareness of the state-of-the-art in software
            engineering technology?

1.3.2.   Is a *mechanism* used for evaluating technologies used by the organization versus
            those externally available?

1.3.3.   Is a *mechanism* used for deciding when to insert new technology into the devel-
            opment *process*?

1.3.4.   Is a *mechanism* used for managing and supporting the introduction of new technol-
            ogies?

1.3.5.   Is a *mechanism* used for identifying and replacing obsolete technologies?

2. Software Engineering Process and its Management

This section concerns the scope, depth, and completeness of the software engineering process and how the process is measured, managed, and improved. The major topics are standards and procedures, metrics, data management and analysis, and process control.

2.1. Documented Standards and Procedures

The standards and procedures questions address the scope and usage of conventions, formats, procedures, and documentation during the various software development phases, i.e., requirements, design, code, and test.

2.1.1.   Does the software organization use a standardized and documented software development *process* on each project?

2.1.2.   Does the standard software development *process* documentation describe the use of tools and techniques?

2.1.3.   Is a *formal procedure* used in the management review of each software development prior to making contractual commitments?

2.1.4.   Is a *formal procedure* used to assure periodic management review of the status of each software development project?

2.1.5.   Is there a *mechanism* for assuring that software subcontractors, if any, follow a disciplined software development *process*?

2.1.6.   Are *standards* used for the content of software development files/folders?

2.1.7.   For each project, are independent audits conducted for each step of the software development *process*?

2.1.8.   Is a *mechanism* used for assessing existing designs and code for reuse in new applications?

2.1.9.   Are coding *standards* applied to each software development project?

2.1.10.   Are *standards* applied to the preparation of unit test cases?

2.1.11.   Are code maintainability *standards* applied?

2.1.12.   Are internal design review *standards* applied?

2.1.13.   Are code review *standards* applied?

2.1.14.   Is a *formal procedure* used to make estimates of software size?

2.1.15.   Is a *formal procedure* used to produce software development schedules?

2.1.16.   Are *formal procedures* applied to estimating software development cost?

2.1.17.   Is a *mechanism* used for ensuring that the software design teams understand each software requirement?

2.1.18.   Are man-machine interface *standards* applied to each appropriate software development project?

2.2. Process Metrics

The process metrics questions focus on the degree to which the software engineering process is quantified and measured. Typical metrics concern software quality, the amount of code developed, resources used, and such progress indicators as review coverage, test coverage, and test completion.

2.2.1.   Are software staffing profiles maintained of actual staffing versus planned staffing?

2.2.2.   Are profiles of software size maintained for each software configuration item, over time?

2.2.3.   Are statistics on software design errors gathered?

2.2.4.   Are statistics on software code and test errors gathered?

2.2.5.   Are design errors projected and compared to actuals?

2.2.6.   Are code and test errors projected and compared to actuals?

2.2.7.   Are profiles maintained of actual versus planned software units designed, over time?

2.2.8.   Are profiles maintained of actual versus planned software units completing unit testing, over time?

2.2.9.   Are profiles maintained of actual versus planned software units integrated, over time?

2.2.10.   Are target computer memory utilization estimates and actuals tracked?

2.2.11.   Are target computer throughput utilization estimates and actuals tracked?

2.2.12.   Is target computer I/O channel utilization tracked?

2.2.13.   Are design and code *review coverages* measured and recorded?

2.2.14.   Is *test coverage* measured and recorded for each phase of functional testing?

2.2.15.   Are the action items resulting from design reviews tracked to closure?

2.2.16.   Are software trouble reports resulting from testing tracked to closure?

2.2.17.   Are the action items resulting from code reviews tracked to closure?

2.2.18.   Is test progress tracked by deliverable software component and compared to the plan?

2.2.19.   Are profiles maintained of software build/release content versus time?

2.3. Data Management and Analysis

Data management deals with the gathering and retention of process metrics. Data management requires standardized data definitions, data management facilities, and a staff to ensure that data is promptly obtained, properly checked, accurately entered into the database, and effectively managed.

Analysis deals with the subsequent manipulation of the process data to answer questions such as, "Is there is a relatively high correlation between error densities found in test and those found in use?" Other types of analyses can assist in determining the optimum use of reviews and resources, the tools most needed, testing priorities, and needed education.

2.3.1.   Has a managed and controlled *process database* been established for *process metrics* data across a: projects?

2.3.2.   Are the *review data* gathered during design reviews analyzed?

2.3.3.   Is the error data from code reviews and tests analyzed to determine the likely distribution and characteristics of the errors remaining in the product?

2.3.4.   Are analyses of errors conducted to determine their *process* related causes?

2.3.5.   Is a *mechanism* used for error cause analysis?

2.3.6.   Are the error causes reviewed to determine the *process* changes required to prevent them?

2.3.7.   Is a *mechanism* used for initiating error prevention actions?

2.3.8.   Is *review efficiency* analyzed for each project?

2.3.9.   Is software productivity analyzed for major *process* steps?

2.4. Process Control

The process control questions concern the definition of the development process and the mechanisms for identifying process problems, correcting process deficiencies, and preventing their recurrence.

2.4.1.   Does senior management have a *mechanism* for the regular review of the status of software development projects?

2.4.2.   Is a *mechanism* used for periodically assessing the software engineering *process* and implementing indicated improvements?

2.4.3.   Is a *mechanism* used for identifying and resolving system engineering issues that affect software?

2.4.4.   Is a *mechanism* used for independently calling integration and test issues to the attention of the project manager?

2.4.5.   Is a *mechanism* used for regular technical interchanges with the customer?

2.4.6.   Is a *mechanism* used for ensuring compliance with the software engineering standards?

2.4.7.   Do software development first-line managers sign off on their schedules and cost estimates?

2.4.8.   Is a *mechanism* used for ensuring traceability between the software requirements and top-level design?

2.4.9.   Is a *mechanism* used for controlling changes to the software requirements?

2.4.10.   Is there a formal management *process* for determining if the prototyping of software functions is an appropriate part of the design *process*?

2.4.11.   Is a *mechanism* used for ensuring traceability between the software top-level and detailed designs?

2.4.12.   Are internal software design reviews conducted?

2.4.13.   Is a *mechanism* used for controlling changes to the software design?

2.4.14.   Is a *mechanism* used for ensuring traceability between the software detailed design and the code?

2.4.15.   Are formal records maintained of unit (module) development progress?

2.4.16.   Are software code reviews conducted?

2.4.17.   Is a *mechanism* used for controlling changes to the code? (Who can make changes and under which circumstances?)

2.4.18.   Is a *mechanism* used for configuration management of the software tools used in the development *process*?

2.4.19.   Is a *mechanism* used for verifying that the samples examined by Software Quality Assurance are truly representative of the work performed?

2.4.20.   Is there a *mechanism* for assuring that regression testing is routinely performed?

2.4.21.   Is there a *mechanism* for assuring the adequacy of regression testing?

2.4.22.   Are formal test case reviews conducted?

3. Tools and Technology

This section deals with the tools and technologies used in the software engineering process. It aims at ascertaining the degree to which the contractor's process employs basic tools and methodologies. (In subsequent revisions of this document, this section will be expanded as the applicability and effectiveness of advanced tools and methodologies become more fully established.)

3.1.   Is automated configuration control used to control and track change activity throughout the software development *process*?

3.2.   Are computer tools used to assist in tracing software requirements to software design?

3.3.   Are formal design notations such as PDL used in program design?

3.4.   Are computer tools used to assist in tracing the software design to the code?

3.5.   Is the majority of product development implemented in a high-order language?

3.6.   Are automated test input data generators used for testing?

3.7.   Are computer tools used to measure *test coverage*?

3.8.   Are computer tools used to track every required function and assure that it is tested/verified?

3.9.   Are automated tools used to analyze the size and change activity in software components?

3.10.   Are automated tools used to analyze software complexity?

3.11.   Are automated tools used to analyze cross references between modules?

3.12.   Are interactive source-level debuggers used?

3.13.   Are the software development and maintenance personnel provided with interactive documentation facilities?

3.14.   Are computer tools used for tracking and reporting the status of the software in the software development library?

3.15.   Are prototyping methods used in designing the critical performance elements of the software?

3.16.   Are prototyping methods used in designing the critical elements of the man-machine interface?

Addenda

Addendum A: Software Engineering Experience

A complete assessment of a contractor's capability to produce quality software at a particular facility should include an evaluation of the experience level of the software development personnel at that location. The experience level of the development staff significantly and directly influences the cost of software development projects. Information about experience level is normally obtained during source selection from proposals or from review team interviews. However, for the purpose of this evaluation, suggested questions are listed below.

A.1   What is the median number of years of applicable experience of software development managers?

A.2   What is the median number of years of applicable experience of software integration and test managers?

A.3   What percentage of the software development staff has a bachelor degree or higher in computer science or software engineering?

A.4   What is the median number of years of software development experience of the software staff?

A.5   What percentage of the software staff has at least one year of development experience with the design and implementation languages to be used?

A.6   Of those with such experience, what is the median number of years of experience with those languages?

A.7   What is the median size, in source lines of code, of software development projects completed in the last five years? The size of the smallest project? The largest?

A.8   What is the total size of the software development organization, including direct professionals, management, and support personnel?

A.9   What is the total number of software engineers in the organization?

Addendum B: Software Engineering Process Maturity Levels

Five levels of process maturity have been defined for the assessment of software engineering organizations.

- Level 1 Initial
- Level 2 Repeatable
- Level 3 Defined
- Level 4 Managed
- Level 5 Optimized

Level 1 - Initial Process

The initial environment has ill-defined procedures and controls. While positive responses to some of the organizational questions are likely, the organization does not consistently apply software engineering management to the process, nor does it use modern tools and technology.

Level 2 - Repeatable Process

At Maturity Level 2, the organization uses standard methods and practices for managing software development activities such as cost estimating, scheduling, requirements changes, code changes, and status reviews. The organization will provide positive responses to most of the following questions.

1.1.1	For each project involving software development, is there a designated software manager?
1.1.2	Does the project software manager report directly to the project (or project development) manager?
*1.1.3	Does the Software Quality Assurance (SQA) function have a management reporting channel separate from the software development project management?
*1.1.6	Is there a software configuration control function for each project that involves software development?
1.2.2	Is there a required training program for all newly appointed development managers designed to familiarize them with software project management?
1.3.1	Is a *mechanism* used for maintaining awareness of the state-of-the-art in software engineering technology?
*2.1.3	Is a *formal procedure* used in the management review of each software development prior to making contractual commitments?
2.1.4	Is a *formal procedure* used to assure periodic management review of the status of each software development project?
2.1.5	Is there a *mechanism* for assuring that software subcontractors, if any, follow a disciplined software development *process*?

2.1.7	For each project, are independent audits conducted for each step of the software development *process*?
2.1.9	Are coding *standards* applied to each software development project?
*2.1.14	Is a *formal procedure* used to make estimates of software size?
*2.1.15	Is a *formal procedure* used to produce software development schedules?
*2.1.16	Are *formal procedures* applied to estimating software development cost?
2.1.17	Is a *mechanism* used for ensuring that the software design teams understand each software requirement?
2.2.1	Are software staffing profiles maintained of actual staffing versus planned staffing?
*2.2.2	Are profiles of software size maintained for each software configuration item, over time?
*2.2.4	Are statistics on software code and test errors gathered?
2.2.7	Are profiles maintained of actual versus planned software units designed, over time?
2.2.8	Are profiles maintained of actual versus planned software units completing unit testing, over time?
2.2.9	Are profiles maintained of actual versus planned software units integrated, over time?
2.2.10	Are target computer memory utilization estimates and actuals tracked?
2.2.11	Are target computer throughput utilization estimates and actuals tracked?
2.2.12	Is target computer I/O channel utilization tracked?
2.2.16	Are software trouble reports resulting from testing tracked to closure?
2.2.18	Is test progress tracked by deliverable software component and compared to the plan?
2.2.19	Are profiles maintained of software build/release content versus time?
*2.4.1	Does senior management have a *mechanism* for the regular review of the status of software development projects?
2.4.5	Is a *mechanism* used for regular technical interchanges with the customer?
*2.4.7	Do software development first-line managers sign off on their schedules and cost estimates?
*2.4.9	Is a *mechanism* used for controlling changes to the software requirements?
*2.4.17	Is a *mechanism* used for controlling changes to the code? (Who can make changes and under which circumstances?)
2.4.20	Is there a *mechanism* for assuring that regression testing is routinely performed?

Level 3 - Defined Process

At Maturity Level 3, the organization not only defines its process in terms of software engineering standards and methods, it also has made a series of organizational and methodo-

---

logical improvements. These specifically include design and code reviews, training programs for programmers and review leaders, and increased organizational focus on software engineering. A major improvement in this phase is the establishment and staffing of a software engineering process group that focuses on the software engineering process and the adequacy with which it is implemented. In addition to the questions for Level 2, organizations at Level 3 will respond "yes" to most of the following questions.

1.1.4   Is there a designated individual or team responsible for the control of software interfaces?

1.1.5   Is software system engineering represented on the system design team?

*1.1.7   Is there a software engineering *process group* function?

1.2.1   Does each software developer have a private computer-supported workstation/terminal?

*1.2.3   Is there a required software engineering training program for software developers?

1.2.4   Is there a required software engineering training program for first-line supervisors of software development?

*1.2.5   Is a formal training program required for design and code *review leaders*?

1.3.2   Is a *mechanism* used for evaluating technologies used by the organization versus those externally available?

*2.1.1   Does the software organization use a standardized and documented software development *process* on each project?

2.1.2   Does the standard software development *process* documentation describe the use of tools and techniques?

2.1.6   Are *standards* used for the content of software development files/folders?

2.1.8   Is a *mechanism* used for assessing existing designs and code for reuse in new applications?

2.1.10   Are *standards* applied to the preparation of unit test cases?

2.1.11   Are code maintainability *standards* applied?

2.1.18   Are man-machine interface *standards* applied to each appropriate software development project?

*2.2.3   Are statistics on software design errors gathered?

*2.2.15 Are the action items resulting from design reviews tracked to closure?

*2.2.17 Are the action items resulting from code reviews tracked to closure?

2.4.3   Is a *mechanism* used for identifying and resolving system engineering issues that affect software?

2.4.4   Is a *mechanism* used for independently calling integration and test issues to the attention of the project manager?

*2.4.6   Is a *mechanism* used for ensuring compliance with the software engineering *standards*?

2.4.8   Is a *mechanism* used for ensuring traceability between the software requirements
            and top-level design?

2.4.11   Is a *mechanism* used for ensuring traceability between the software top-level and
            detailed designs?

*2.4.12   Are internal software design reviews conducted?

*2.4.13   Is a *mechanism* used for controlling changes to the software design?

2.4.14   Is a *mechanism* used for ensuring traceability between the software detailed design
            and the code?

2.4.15   Are formal records maintained of unit (module) development progress?

*2.4.16   Are software code reviews conducted?

2.4.18   Is a *mechanism* used for configuration management of the software tools used in
            the development *process*?

*2.4.19   Is a *mechanism* used for verifying that the samples examined by Software Quality
            Assurance are truly representative of the work performed?

*2.4.21   Is there a *mechanism* for assuring the adequacy of regression testing?

2.4.22   Are formal test case reviews conducted?

Level 4 - Managed Process

At Maturity Level 4, the organization typically bases its operating decisions on quantitative
process data, and conducts extensive analyses of the data gathered during software engi-
neering reviews and tests. Tools are used increasingly to control and manage the design
process as well as to support data gathering and analysis. The organization is learning to
project expected errors with reasonable accuracy. In addition to questions for Levels 2 and
3, organizations at Level 4 will respond "yes" to most of the following questions.

1.3.3   Is a *mechanism* used for deciding when to insert new technology into the devel-
            opment *process*?

*1.3.4   Is a *mechanism* used for managing and supporting the introduction of new technol-
            ogies?

2.1.12   Are internal design review *standards* applied?

*2.1.13   Are code review *standards* applied?

*2.2.5   Are design errors projected and compared to actuals?

*2.2.6   Are code and test errors projected and compared to actuals?

*2.2.13   Are design and code *review coverages* measured and recorded?

*2.2.14   Is *test coverage* measured and recorded for each phase of functional testing?

*2.3.1   Has a managed and controlled *process database* been established for *process
            metrics* data across all projects?

*2.3.2   Are the *review data* gathered during design reviews analyzed?

The answers to the questions should reflect standard organizational practice.

*2.3.3   Is the error data from code reviews and tests analyzed to determine the likely distribution and characteristics of the errors remaining in the product?

*2.3.4   Are analyses of errors conducted to determine their *process* related causes?

*2.3.8   Is *review efficiency* analyzed for each project?

2.3.9   Is software productivity analyzed for major *process* steps?

*2.4.2   Is a *mechanism* used for periodically assessing the software engineering *process* and implementing indicated improvements?

2.4.10   Is there a formal management *process* for determining if the prototyping of software functions is an appropriate part of the design *process*?

Level 5 - Optimized Process

At Maturity Level 5, organizations have not only achieved a high degree of control over their process, they have a major focus on improving and optimizing its operation. This includes more sophisticated analyses of the error and cost data gathered during the process as well as the introduction of comprehensive error cause analysis and prevention studies.

*1.3.5   Is a *mechanism* used for identifying and replacing obsolete technologies?

*2.3.5   Is a *mechanism* used for error cause analysis?

*2.3.6   Are the error causes reviewed to determine the *process* changes required to prevent them?

*2.3.7   Is a *mechanism* used for initiating error prevention actions?

Addendum C: Technology

This section defines a method for evaluating the software engineering technology of a contractor. The quality of a software process is affected by the stage of software technology employed. Two stages for describing the level of software technology have been defined.

Stage A - Inefficient Technology

An organization that primarily employs inefficient software development technology is likely to be ineffective in developing software. Many different implementations may be available and the practice may be in widespread use, but the technology is no longer effective. Moreover, at this technology stage some important software engineering practices are not practical in large, complex developments.

Stage B - Basic Technology

An organization that primarily employs basic software development technologies is likely to be moderately effective and, depending upon the maturity of its process, reasonably consistent in its performance. Multiple implementations are available, and they have been demonstrated to be effective. Organizations at Stage B will respond "yes" to most of the following questions.

3.1	Is automated configuration control used to control and track change activity throughout the software development *process*?
3.2	Are computer tools used to assist in tracing software requirements to software design?
3.3	Are formal design notations such as PDL used in program design?
3.4	Are computer tools used to assist in tracing the software design to the code?
*3.5	Is the majority of product development implemented in a high-order language?
3.6	Are automated test input data generators used for testing?
.	Are computer tools used to measure *test coverage*?
3.8	Are computer tools used to track every required function and assure that it is tested/verified?
3.9	Are automated tools used to analyze the size and change activity in software components?
3.10	Are automated tools used to analyze software complexity?
3.11	Are automated tools used to analyze cross references between modules?
*3.12	Are interactive source-level debuggers used?
*3.13	Are the software development and maintenance personnel provided with interactive documentation facilities?
*3.14	Are computer tools used for tracking and reporting the status of the software in the software development library?
3.15	Are prototyping methods used in designing the critical performance elements of the software?

3.16   Are prototyping methods used in designing the critical elements of the man-machine interface?

Addendum D: Assessment Recording Form
Contractor Software Engineering Capability

Contractor Code [   ]   Guide Version [093087]

The answers to these questions should reflect standard organizational practice as implemented by a single project.

| Question Number | Additional Information | | | Comments | Shade in Answer |
	Control Number	Level	Follow-up Question		
1.1.1	2	2	1		Y N
1.1.2	3	2	1		Y N
1.1.3	6	2	1		Y N
1.1.4	8	3	1		Y N
1.1.5	11	3	1		Y N
1.1.6	14	2	2		Y N
1.1.7	15	3	2		Y N
1.2.1	16	3	9		Y N
1.2.2	18	2	3		Y N
1.2.3	19	3	3		Y N
1.2.4	152	3	3		Y N
1.2.5	20	3	3		Y N
1.3.1	126	2	4		Y N
1.3.2	127	3	4		Y N
1.3.3	172	4	4		Y N
1.3.4	128	4	4		Y N
1.3.5	129	5	4		Y N
2.1.1	23	3	4		Y N
2.1.2	163	3	4		Y N
2.1.3	24	2	4		Y N
2.1.4	164	2	4		Y N
2.1.5	25	2	4		Y N
2.1.6	28	3	4		Y N
2.1.7	30	2	8		Y N
2.1.8	159	3	4		Y N
2.1.9	31	2	4		Y N
2.1.10	32	3	4		Y N
2.1.11	33	3	4		Y N
2.1.12	37	4	4		Y N
2.1.13	38	4	4		Y N

[▒▒▒▒] Of greater importance for indicated maturity level

Contractor Software Engineering Capability

Contractor Code [　　　　　]　　　　　　　　　Guide Version [C93087]

The answers to these questions should reflect standard organizational practice as implemented by a single project.

Question Number	Additional Information			Comments	Shade in Answer
	Control Number	Level	Follow-up Question		
2.1.14	42	2	4		Y N
2.1.15	123	3	4		Y N
2.1.16	44	2	4		Y N
2.1.17	124	2	4		Y N
2.1.18	130	3	4		Y N
2.2.1	45	2	5		Y N
2.2.2	46	2	5		Y N
2.2.3	47	3	5		Y N
2.2.4	48	2	5		Y N
2.2.5	49	4	5		Y N
2.2.6	50	4	5		Y N
2.2.7	51	2	5		Y N
2.2.8	52	2	5		Y N
2.2.9	53	2	5		Y N
2.2.10	54	2	5		Y N
2.2.11	55	2	5		Y N
2.2.12	56	2	5		Y N
2.2.13	57	4	5		Y N
2.2.14	58	4	5		Y N
2.2.15	59	3	5		Y N
2.2.16	60	2	5		Y N
2.2.17	61	3	5		Y N
2.2.18	62	2	5		Y N
2.2.19	63	2	5		Y N
2.3.1	64	4	9		Y N
2.3.2	67	4	6		Y N
2.3.3	68	4	6		Y N
2.3.4	70	4	6		Y N
2.3.5	69	5	4		Y N
2.3.6	71	5	7		Y N

[▒▒▒▒▒]　Of greater importance for indicated maturity level

Contractor Software Engineering Capability

Contractor Code [   ]   Guide Version [093087]

The answers to these questions should reflect standard organizational practice as implemented by a single project.

| Question Number | Additional Information | | | Comments | Shade in Answer |
	Control Number	Level	Follow-up Question		
2.3.7	72	5	4		Y N
2.3.8	75	4	6		Y N
2.3.9	76	4	6		Y N
2.4.1	77	2	4		Y N
2.4.2	78	4	4		Y N
2.4.3	79	3	4		Y N
2.4.4	81	3	4		Y N
2.4.5	82	2	4		Y N
2.4.6	83	3	4		Y N
2.4.7	84	2	8		Y N
2.4.8	86	3	4		Y N
2.4.9	87	2	4		Y N
2.4.10	88	4	4		Y N
2.4.11	90	3	4		Y N
2.4.12	91	3	8		Y N
2.4.13	92	3	4		Y N
2.4.14	93	3	4		Y N
2.4.15	94	3	5		Y N
2.4.16	95	3	8		Y N
2.4.17	96	2	4		Y N
2.4.18	165	3	4		Y N
2.4.19	98	3	4		Y N
2.4.20	171	2	4		Y N
2.4.21	99	3	8		Y N
2.4.22	162	3	8		Y N

[   ] Of greater importance for indicated maturity level

Contractor Software Engineering Capability

Contractor Code [   ]   Guide Version [093087]

The answers to these questions should reflect standard organizational practice as implemented by a single project.

| Question Number | Additional Information | | | Comments | Shade in Answer |
	Control Number	Stage	Follow-up Question		
3.1	160	B	10		Ⓨ Ⓝ
3.2	103	B	10		Ⓨ Ⓝ
3.3	142	B	9		Ⓨ Ⓝ
3.4	111	B	10		Ⓨ Ⓝ
3.5	151	B	9		Ⓨ Ⓝ
3.6	167	B	10		Ⓨ Ⓝ
3.7	113	B	10		Ⓨ Ⓝ
3.8	112	B	10		Ⓨ Ⓝ
3.9	140	B	10		Ⓨ Ⓝ
3.10	137	B	10		Ⓨ Ⓝ
3.11	143	B	10		Ⓨ Ⓝ
3.12	169	B	10		Ⓨ Ⓝ
3.13	146	B	9		Ⓨ Ⓝ
3.14	114	B	10		Ⓨ Ⓝ
3.15	131	B	9		Ⓨ Ⓝ
3.16	132	B	9		Ⓨ Ⓝ

� Of greater importance for indicated maturity level

Addendum E: Follow-up Questions

It is recommended that, when appropriate, the assessment team ask for amplification of responses to the assessment questions. The team should request actual data supporting the responses. Listed below are ten follow-up questions for amplifying data. The Assessment Recording Form indicates the number of the appropriate follow-up questions for each assessment question.

1. Where responsibility assignments are questioned, request the name of a specific individual, tenure in job, job description, and evidence of activity, such as monthly reports, meeting reports, control logs.

2. Where the existence of a group is questioned, request names of members, the organization represented, and recent meeting agendas and minutes.

3. Where the existence of education or training programs is questioned, request the schedule of recent courses offered, course outlines, names of attendees, and qualifications of instructors and students.

4. Where the existence of a mechanism, procedure, standard, criteria, or guideline is questioned, request a copy of the controlling document, its revision history, the name of individual(s) responsible for tracking, job description(s), and recent issue/activity reports.

5. Where the use of profiles, tracking reports, planned vs. actual comparisons, and measurements are questioned, request the three most recent reports, measurement summaries, or comparisons.

6. Where computations or analysis of data is questioned, request copies of the most recent computations, analysis reports, or summaries showing results or conclusions reached.

7. Where the initiation of actions are questioned, request copies of recent action tracking and/or summary reports.

8. Where the conduct of certain actions or use of facilities is questioned, request evidence in the form of procedures, responsibilities, or tracking systems to demonstrate performance.

9. Where the existence of a facility, capability, practice, or method is questioned, request supporting evidence in the form of inventory lists, tracking and usage reports, instruction manuals, education programs, etc.

10. Where the use of an automated tool or facility is questioned, request a demonstration of that tool or facility.

Glossary

This glossary should be used in conjunction with the *IEEE Standard Glossary of Software Engineering Terminology* (ANSI/IEEE STD729-1983) published by the Institute of Electrical and Electronic Engineers, February 18, 1983. Wherever possible, common software engineering terminology has been used. Where terms in this document are not included in the *IEEE Standard Glossary* or have special meaning in the context used here, they are described in this glossary.

contractor evaluation – A process by which a contracting organization uses the results of contractor assessments and other information to determine the relative capability of contractors.

error prevention analysis – A process that is typically conducted by a working group of software engineering professionals who developed the code in question. It is an objective assessment of each error, its potential cause, and the steps to be taken to prevent it. While placing blame is to be avoided, such questions as mistakes, adequacy of education and training, proper tools capability, and support effectiveness are appropriate areas for analysis.

formal procedure – A documented series of steps with guidelines for use.

mechanism – A means or technique whereby the performance of a task, procedure, or process is assured. The mechanism may involve several organizational elements, and its documentation may include some combination of function statements, operating plans, position descriptions, and/or formal procedures. The documentation defines what should be performed, how it should be performed, and who is accountable for the results.

process – A systematic series of mechanisms, tasks, and/or procedures directed towards an end. The software engineering process documentation defines the sequence of steps used to produce a finished product. Each step is described as a task that is performed by using a software engineering methodology or an administrative procedure, and it prescribes the automated tools and techniques to be used.

process data – The data that is gathered about the software engineering process. It typically includes review, test, and resource data by process phase and change activity. To be most meaningful, this data should be associated with the process documentation, the tools and methods used, and the characteristics of the product being produced.

process database – A repository into which all process data is entered. It is a centralized resource managed by the process group. Centralized control of this database ensures that the process data from all projects are permanently retained and protected.

process group – The software engineering process group is composed of specialists concerned with the process used by the development organization for software development. Its typical functions include defining and documenting the process, establishing and defining

39

metrics, gathering data, assisting projects in analyzing data, and advising management on areas requiring further attention. The process group typically conducts quarterly management reviews on process status and n.ay provide review leaders.

process metrics – Those measurements established for each step in the software engineering process that are used to determine its effectiveness. The metrics define the results of each process stage and relate them to the resources expended, errors introduced, errors removed, and various coverage, efficiency, and productivity indicators.

review coverage – The degree to which all code in a software product has been reviewed. It is typically stated as a percentage and measures the percentage of the lines of executable code or design statements evaluated by the review process.

review data – The data that is gathered from design or code reviews. This data is of two types. The first, concerning the review process, typically includes preparation time, lines of code per hour of preparation time, errors identified during preparation (by category), hours per error found in preparation, review time, lines of code (or design statements) reviewed, code (or design statements) reviewed per hour, and errors found per review man-hour (by category). The second type, product data from the review, typically includes errors found per line of code (or design statement), action items identified from each review, action items closed for each review, items needing re-review, re-reviews conducted.

review efficiency – The percentage of errors found through the review process. It is typically stated as a percentage and is calculated by dividing the total errors found during review by the total errors found by both review and test through the completion of product and system integration test. It does not include those errors found during acceptance test or field usage.

review leader – Typically a member of the process or assurance group who is thoroughly trained in the review process. The review leader's role is to ensure that the participants are properly prepared and that the review is efficiently and thoroughly conducted. The review leader is responsible for recording review data, making sure that the actions resulting from the review are completed, and for conducting re-reviews where appropriate.

standard – An approved, documented, and available set of criteria used to determine the adequacy of an action or object.

test coverage – The amount of code actually executed during the test process. It is stated as a percentage of the total instructions executed or paths traversed.

AD-A187 230

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION	1b. RESTRICTIVE MARKINGS
UNCLASSIFIED	NONE
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT
N/A	APPROVED FOR PUBLIC RELEASE
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	DISTRIBUTION UNLIMITED
N/A	

4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S)
CMU/SEI-87-TR-23	ESD-TR-87-186

6a. NAME OF PERFORMING ORGANIZATION	6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION
SOFTWARE ENGINEERING INSTITUTE	SEI	SEI JOINT PROGRAM OFFICE
6c. ADDRESS (City, State and ZIP Code)		7b. ADDRESS (City, State and ZIP Code)
CARNEGIE MELLON UNIVERSITY PITTSBURGH, PA 15213		ESD/XRS1 HANSCOM AIR FORCE BASE, MA 01731

8a. NAME OF FUNDING/SPONSORING ORGANIZATION	8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
SEI JOINT PROGRAM OFFICE	SEI JPO	F1962885C0003

8c. ADDRESS (City, State and ZIP Code)	10. SOURCE OF FUNDING NOS.			
CARNEGIE MELLON UNIVERSITY SOFTWARE ENGINEERING INSTITUTE JPO PITTSBURGH, PA 15213	PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.
		N/A	N/A	N/A

11. TITLE (Include Security Classification)
A METHOD FOR ASSESSING THE SOFTWARE ENGINEERING CAPABILITY OF CONTRACTORS ·Preliminary Version

12. PERSONAL AUTHOR(S)
W.S. Humphrey   W. L. Sweet

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Yr., Mo., Day)	15. PAGE COUNT
FINAL	FROM _____ TO _____	Sept. 1987	42

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB. GR.	contractor assessment
			software engineering maturity

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

This document provides guidelines and procedures for assessing the ability of potential DoD contractors to develop software in accordance with modern software engineering methods. It includes specific questions and a method for evaluating the results.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS ☒	UNCLASSIFIED, UNLIMITED

22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE NUMBER (Include Area Code)	22c. OFFICE SYMBOL
KARL SHINGLER	(412) 268-7630	SEI JPO

DD FORM 1473, 83 APR   EDITION OF 1 JAN 73 IS OBSOLETE.