



Using Contractor Capability Evaluations to Reduce Software Development Risk

Bonnie R. Troup
The Aerospace Corp.

Brian P. Gallagher
Software Engineering Institute

One aspect of software development risk is the development contractor's ability to deliver software within specified goals. One approach to mitigating this risk is to select the development contractor with the strongest software development capabilities. This paper will present and compare two methods used at Space and Missile Systems Center (SMC) and the National Reconnaissance Office (NRO) during source selection to evaluate a contractor's software development capability with the goal of selecting a mature software development contractor.

DESPITE THE BEST EFFORTS of software acquisition professionals, and the increased commitment of maturing acquisition organizations to deliver promised software capabilities, large software-intensive systems frequently do not meet the cost, schedule, and performance objectives of the acquiring organization and end user. As many as 75% of all large-scale, custom software-intensive systems fail [1]. The primary reason is immature management practices [2].

Given this poor record, and an increasing demand on software capabilities, how do software acquirers reduce this risk? One approach is to choose a contractor mature enough to manage the software project; a technique to aid in this choice is using a contractor capability evaluation during source selection. Such evaluations provide acquirers with information to discriminate among contractors by assessing their ability to deliver such systems within cost, schedule, and performance objectives. The evaluation results can reduce the risks by helping the acquirer identify weaknesses in a developer's software development process before hiring them.

Purpose of Contractor Capability Evaluations

Contractor capability evaluations are formal, systematic methods that employ defined models for assessing a contractor's software development process. These

The Software Capability Evaluation and Software Capability Maturity Model (SW-CMM) are service marks of Carnegie Mellon University.

methods are used to identify strengths, weaknesses, and risks related to a contractor's defined or proposed software process. They also can be used to compare a contractor's defined or proposed software process with its actual software process in use on a given program.

Formal evaluations are performed by an external organization and the acquisition organization receives feedback on the evaluation. When an evaluation is done during source selection, evaluation results are a key discriminator to decision makers and increase the likelihood of selecting a contractor capable of developing the required software within program constraints. An acquirer also can use formal evaluations to identify risks inherent in the contractor's approach and to facilitate managing these risks beginning at contract award.

Success of contractor capability evaluations during a source selection assumes an evaluation of the contractor's process, commitment to using the process on the proposed project as evidenced in contractually binding documents like the Integrated Master Plan (IMP), and incentives for following and improving the proposed process during contract execution. Evaluation and commitment are determined during the source selection, and the incentive is generated by encouragement from the acquirer's leadership during contract execution and the appropriate use of award fee.

Capability Evaluations and Acquisition Reform

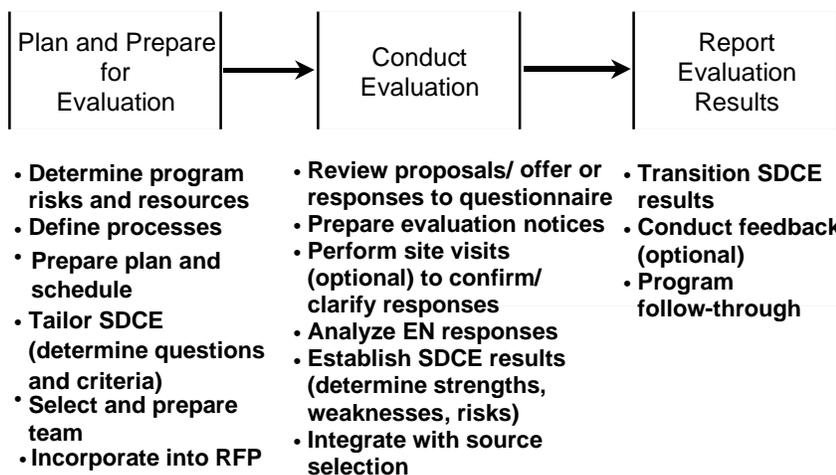
With the advent of the acquisition reform

"lightning bolts" [3], the Department of Defense (DoD) significantly increased its emphasis on risk management, early communication with project bidders, and use of past performance in source selection, while substantially reducing the scope of contractual requirements and oversight on new programs. This reduction in oversight heightens the importance of selecting a contractor capable of reaching cost, schedule, and performance objectives and following a well-defined process while meeting these objectives. As evidenced in DoD 5000.2-R, paragraph 4.3.5.5 [4], selection of a contractor with "a demonstrable, mature software development capability and process" is mandatory on all major defense acquisition programs.

Capability evaluations are consistent with the objectives of acquisition reform. Evaluation results provide early software development risk identification, thereby supporting the role of risk management in acquisition decisions. The evaluation process establishes a beneficial communication with the contractors' software organizations, beginning with the early phases of the acquisition, and supports the use of contractor-defined or commercial processes in lieu of contractually dictated standards and processes. Finally, capability evaluations request evidence of past use of proposed processes, supporting assessment of past performance.

Primary Capability Evaluation Techniques

The Software Development Capability Evaluation (SDCE) and the Software

Figure 1. *The SDCE process.*

Capability EvaluationSM (SCE) are two alternative, formal methods that the SMC and NRO predominantly use to evaluate the software development capability of contractors. The SDCE is the primary evaluation method in use at SMC; the SCE is used by the NRO and other government organizations.

The Software Development Capability Evaluation

Developed solely for use during source selections, the SDCE was created in 1993 by an Air Force Materiel Command (AFMC) Process Action Team (PAT), which included participants from government, industry, and Federally Funded Research and Development Centers (FFRDCs). The SDCE is based on the Software Development Capability/ Capacity Review (SDC/CR), developed by Aeronautical Systems Center, and on early versions of the SCE. The method is documented in AFMC pamphlet 63-103, "Software Development Capability Evaluation" [5] and discussed in "Software Development Capability Evaluation: An Integrated Systems and Software Approach" [6]. Guidelines for application of the SDCE, based on lessons learned and technology updates, are documented in Aerospace Technical Report TR-98(8550)-1 [7].

Acquisition philosophy recommends using capability evaluations to reduce software development risk; the SDCE supports this philosophy by enabling acquirers to consistently evaluate software development contractors for proven plans, processes, methods, and tools.

First, the proposed development approach is assessed by evaluating a contractor's written SDCE responses against the documented SDCE criteria. These responses are cross-checked with other portions of the proposal to determine consistency. The contractor's commitment to follow the proposed approach also is assessed by comparing material in the SDCE responses with the wording contained in any contractually binding documents. Examples of these documents include the Software Development Plan, IMP, and Work Breakdown Structure. Lastly, using experiential evidence supplied by the contractor, the evaluation weighs a contractor's past experience against its proposed approach; if the contractor proffers a new approach, an analysis of the rationale supplied for that approach is performed.

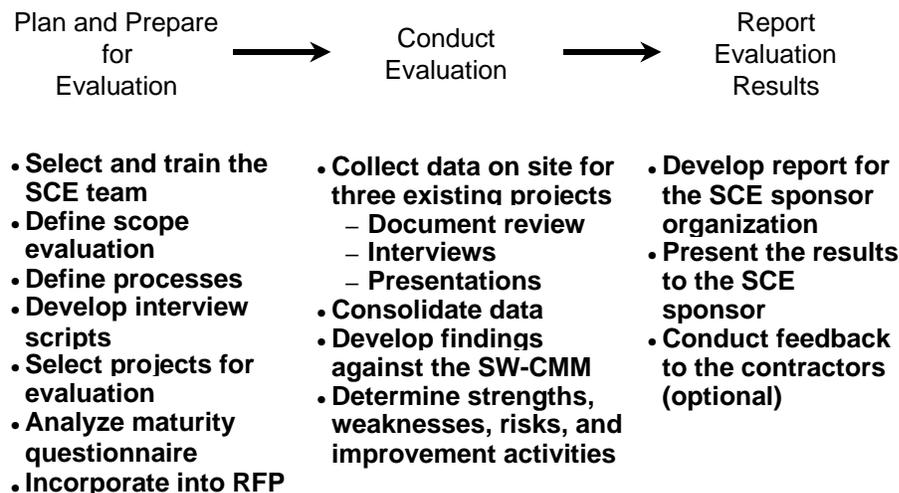
Figure 1 illustrates the SDCE

process. Planning for the evaluation is critical and includes tailoring the SDCE model and process for an individual acquisition. The tailored set of questions and criteria, and instructions for completing the SDCE, are developed and incorporated into the request for proposal (RFP). The contractors provide their responses and evidence of past performance for the acquisition team to review. Deficiencies are documented in evaluation notices (ENs) and provided to the contractor. Strengths, weaknesses, and risks are established and integrated into the source selection. The SDCE results for the selected contractor can be used as a basis for starting a risk management effort after the contract is awarded.

The Software Capability Evaluation

The Software Engineering Institute (SEI) developed the SCE to support source selections in major government acquisitions of software-intensive systems. The method was originally documented in *A Method for Assessing the Software Engineering Capability of Contractors* [8]. It was publicly baselined in *SCE Version 1.5 Method Description* [9] and the current version is documented in *SCE Version 3.0 Method Description* [10].

The SCE's purpose is to provide results that support senior management decision making. These results can be used as a discriminator to select contractors during a formal source selection, and to help assess process growth during contract monitoring. The SCE process supports a disciplined process improvement

Figure 2. *The SCE process.*

	SDCE	SCE
Origin	AFMC PAT, including Air Force, industry, and FFRDCs (SEI, Aerospace, MITRE)	SEI with DoD sponsorship and government and industry review
Focus	Specific software-intensive project	Organizational software process capabilities
Intended use	Source selection	Source selection and contract monitoring
Primary users	SMC, NRO, and ASC	Government, commercial, international

Figure 3. Differences: Origin, focus, and use.

program within the development organization. Results are obtained relative to the Capability Maturity Model for Software (SW-CMMSM) [11]. The SW-CMM is a public reference model used by software development organizations worldwide to improve their software processes. The SCE process uses the SW-CMM to identify the strengths, weaknesses, and risks of an organization's existing software development process.

Figure 2 depicts the SCE process. As with the SDCE process, planning is critical to the success of the evaluation. The pre-evaluation activities include tailoring the SCE to meet the needs of the acquisition and defining the scope of the evaluation and on-site activities. On-site visits are mandatory for the SCE process and the information acquired during these visits is analyzed to determine the findings that are incorporated into the source selection.

Comparing SDCE and SCE Techniques

The SDCE and SCE methods have been used to evaluate a contractor's ability to develop software-intensive systems. They gather information using a defined model and use evidence from existing projects to establish capability. Results are developed in terms of strengths, weaknesses, and risks, and both have a defined process for integrating these results into a source selection.

The differences in origin, focus, and use are shown in Figure 3. The key difference in this table is that the SDCE is focused on assessing the proposed process for a specific software-intensive project —

the one under bid. The SCE focuses on assessing the processes used by the organization bidding on the contract, on similar projects under way or recently completed.

Differences in preparation and implementation are shown in Figure 4. The chief distinctions are the use of site visits and the basis for establishing findings. The SDCE primarily evaluates the contractor's written answers to a tailored questionnaire and the documentation supporting the answers. Site visits are optional and performed, as necessary, to clarify contractor responses. To date, site visits have been done on less than half of the SDCEs for SMC. The SCE requires doing the evaluation at the contractor's site. At the site visit, documentation from the projects selected for evaluation, organizational process documentation, and interviews of project personnel are used to establish findings.

Another marked difference between

the SDCE and SCE, as shown in Figure 4, is in the basis for assessing a contractor's software development process. Although both the SDCE and SCE evaluate the existence and use of processes, the SDCE requires evaluators to determine of the quality of the processes as well. The SCE uses the premise that a well-defined and measured process is self-correcting and that as long as there is sufficient insight into the results of the process, the contractor is able to determine quality and correct for deficiencies.

Differences in Model Coverage

The SDCE model includes questions on the contractor's use of a system/software engineering environment (S/SEE) and in technology areas such as artificial intelligence, distributed processing, and object-oriented techniques. When using these areas of the model, the evaluation team must include members with expertise in the selected technologies to determine the quality of the proposed process. The SCE uses the SW-CMM as its model, which does not include specific technology areas.

The SDCE model contains an area focused on systems engineering process; the SCE's model includes inter-group coordination with systems engineering and software engineering, but does not have process areas specifically for the systems engineering process.

As shown in Figure 5, both methods encourage tailoring their respective models and evaluation processes for a specific acquisition. The SCE method includes

Figure 4. Differences: Preparation and implementation.

	SDCE	SCE
Questionnaire size	700+ questions; usually tailored to <100	100 questions; some may be "NA"
Questionnaire responses	Essay with supporting data	Yes / No; comment required for "Yes"
Site visit	Optional; no defined process	Mandatory; well-defined process
Results established by	Questionnaire responses and optional site visits	Site visits; not from questionnaire responses
Assessment basis	Process existence, use, and quality	Process existence and use only

	SDCE	SCE
Guidelines for tailoring	Minimal set	Extensive
Tailoring risk	Not acknowledged	"Appraisal Risk" identified, documented and accepted
Tailoring decisions	No experience required	Restricted to trained evaluators
Model tailoring	Extensive tailoring required	Depth-oriented and/or breadth-oriented
Process tailoring	Permitted	Constrain number of projects and site visits

Figure 5. Differences: Tailoring approach.

extensive guidelines for tailoring out (removing) model components and modifying the evaluation process, which requires documenting risks the evaluation team assumes with each tailoring decision. Although the SDCE does not have specific tailoring guidelines, the planning process necessitates tailoring the questionnaire to focus on the project risks.

While the SDCE and SCE methods differ in several key areas, they are both powerful tools that have been used successfully to discriminate between contractors based upon their software development capabilities.

Summary

In the source selection environment, contractor capability evaluations can assist in identifying the contractor with the best software capability and experience. They facilitate software risk identification early in a program's life cycle and provide an in-depth look at potential high-risk areas. The acquirer's use of these methods highlights to the developers the importance of using mature software development processes on the projects they bid and encourages them to develop good processes early in the program.

Contractor capability evaluation is an important activity at SMC and NRO. The SDCE is the primary evaluation method used at SMC, while the SCE is popular with industry and is the primary method used by other government organizations.

While contractor capability evaluations are recognized as an acquisition best

practice and are of immense value when acquiring a software-intensive system, a high-quality evaluation is resource- and time-intensive for the government and the contractor, and should be used with discretion. ♦

About the Authors



Bonnie R. Troup is an engineering specialist with The Aerospace Corp. She has more than 12 years of experience in software engineering and software acquisition management; she is the corporate focal point for the SDCE. During her employment at TRW, Troup was a project engineer and software engineering lead, where she participated in an SCE. Her work at Aerospace includes acquisition support for EELV, DMSP, and Milstar, and coordinating SDCE efforts for several program offices. She received her bachelor of science degree in applied mathematics from U.C. Riverside, and a master of science degree in computer science from Loyola Marymount University.

The Aerospace Corp.
Post Office Box 92957, M1-113
Los Angeles, Calif. 90009
Voice: 310-336-1334
Fax: 310-336-8266
E-Mail: bonnie.r.troup@aero.org

Brian P. Gallagher is a member of the technical staff with the SEI. He has more than 14 years of experience in software engineering and software acquisition management; he is working within the Software Engineering and Process Management



group at the SEI. Gallagher was previously employed with The Aerospace Corp., where he worked as a software acquisition and engineering advisor for several SMC and NRO

projects. He is a retired Air Force officer. He received his bachelor of science degree in management information systems from Peru State College, and a master of science degree in computer science/software engineering from Florida Institute of Technology.

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, Pa. 15213-3890
Voice: 412-268-7157
Fax: 412-268-5758
E-Mail: bg@sei.cmu.edu

References

1. Gibbs, W. Wayt, "Software's Chronic Crisis," *Scientific American*, Sept. 1994.
2. Software Technology Support Center, "Guidelines for Successful Acquisition and Management of Software-Intensive Systems: Weapons Systems, Command and Control Systems," Management Information Systems, Version 2.0, Hill AFB, Utah, Department of the Air Force, 1996.
3. Druyan, Darleen, "Lightning Bolt Initiatives," SAF/AQ Home Page (http://www.safaq.hq.af.mil/acq_ref/), May 31, 1995.
4. Department of Defense, DoD Regulation 5000.2-R Mandatory Procedures for Major Defense Acquisition Programs (MDAPs) and Major Automated Information Systems (MAIS) Acquisition Programs, Washington, D.C., Department of Defense, March 1996.
5. Air Force Materiel Command, AFMC Pamphlet 63-103 "Software Development Capability Evaluation," Vol. 1 and 2, June 1994.
6. Babel, P., "Software Development Capability Evaluation: An Integrated Systems and Software Approach," *CROSS TALK*, April 1997.
7. Haddad, R., Revised and Augmented SDCE Model, TR-98(8550)-1, El Segundo, Calif., The Aerospace Corp., June 1998.
8. Humphrey, W.S., and W.L. Sweet. *A Method for Assessing the Software*

- Engineering Capability of Contractors* (CMU/SEI-87-TR-23), Pittsburgh, Pa., Software Engineering Institute, Carnegie Mellon University, 1987.
9. The Software Engineering Institute, *Software Capability Evaluation Version 1.5 Method Description* (CMU/SEI-93-TR-17), Pittsburgh, Pa., The Software Engineering Institute, Carnegie Mellon University, 1993.
10. The Software Engineering Institute, *Software Capability Evaluation Version 3.0 Method Description* (CMU/SEI-96-TR-02), Pittsburgh, Pa., The Software Engineering Institute, Carnegie Mellon University, 1996.
11. Paulk, Mark C., et al, *Capability Maturity Model for Software* (CMU/SEI-91-TR-24), Pittsburgh, Pa., Software Engineering Institute, Carnegie Mellon University, 1991.
- Contains AFMC SDCE Pamphlet Vol. 1 and 2
- SCE Resources:*
Software Engineering Institute Web site <http://www.sei.cmu.edu/>
Contains SEI documents on SCE and CMM

Resources

SDCE Resources:

Los Angeles Air Force Base Web site
<http://ax.laafb.af.mil/~shulmaner/sdec.html>

Coming Events

CERT (Computer Emergency Response Team) Conference '99

Dates: Aug. 30-Sept. 2, 1999

Location: Omaha, Neb.

Topic: Information Protection Expo and Conference.

Provides a forum for discussion on issues concerning computer security and asset protection in our open environment. Topics include computer viruses, intrusions on privacy, data corruption, data theft, Web site vandalism, and hackers.

Hosts: Omaha SPIN (Software Process Improvement Network); Greater Omaha Chapter of Association for Communications, Electronics, Intelligence and Information Systems Professionals; the Software Engineering Institute; Applied Management Institute; and the Sarpy County Chamber of Commerce.

Contact: Debbie Jacobs/Susan Stewart

Voice: 402-293-2929/402-294-3697

Fax: 402-293-2907

E-mail: cert@omaha.com or diszone@aol.com

Internet: <http://www.omaha.com/cert>

ESEC/FSE '99 Joint 7th European Software Engineering Conference (ESEC) and 7th ACM SIGSOFT International Symposium on the Foundations of Software Engineering (FSE-7)

Dates: Sept. 6-10, 1999

Location: Toulouse, France

Topic: ESEC/FSE '99 will bring together researchers and practitioners of modern software engineering techniques to exchange new research results and reports related to traditional and emerging fields of software engineering. ESEC/FSE '99 also will include a program of tutorials and workshops on current topics in software engineering.

Sponsors: ACM, SIGSOFT, CEPIS, SUPAERO, and ONERA

Internet: <http://www.iam.unibe.ch/~esec99/>

2nd USENIX Conference on Domain Specific Languages (DSL '99)

Dates: Oct. 3-6, 1999

Location: Austin, Texas

Sponsors: USENIX, the Advanced Computing Systems Association, in cooperation with ACM SIGPLAN and SIGSOFT

Internet: <http://www.usenix.org/events/dsl99/>

The Second International Conference on The Unified Modeling Language – UML '99

Dates: Oct. 28-30, 1999

Location: Fort Collins, Colo.

Objective: UML '99 will bring together researchers in academia and industry who are developing processes, methods, techniques, and semantic foundations for the UML. The conference will provide a forum for discussing and evaluating promising approaches that will enhance the application of UML.

Sponsors: IEEE Computer Society Technical Committee on Complexity in Computing in cooperation with ACM SIGSOFT.

Contact: Robert France, conference chairman

Voice: 970-491-6356

Fax: 970-491-2466

E-mail: France@cs.colostate.edu

13th annual Conference on Software Engineering Education and Training (CSEE&T) 2000

Dates: March 6-8, 2000

Location: Austin, Texas

Theme: Software Engineering Coming of Age

Submission Date: Sept. 17, 1999

Contact: Susan A. Mengel

Voice: 806-742-3527

Fax: 806-742-3519

E-mail: mengel@ttu.edu

Internet: <http://www.se.cs.ttu.edu/mengel>