Risk Based Diagnostics

Ray C. Williams
Kate Ambrose
Laura Bentrem
Tom Merendino

September 2004

Acquisition Support Program

Technical Note
CMU/SEI-2004-TN-013

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

The Software Engineering Institute is a federally funded research and development center sponsored by the U.S. Department of Defense.

This work was created in the performance of Federal Government Contract Number F19628-00-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 252.227-7013.

For information about purchasing paper copies of SEI reports, please visit the publications portion of our Web site (http://www.sei.cmu.edu/publications/pubweb.html).

Contents

List of Figures

List of Tables

Acknowledgements

The funding for the work described in this report was provided by the Air Force through the Software Technology Support Center and the Computer Resources Support Improvement Program (CRSIP). The authors would like to thank Bruce Allgood, director of CRSIP, for funding the initial research for this project. We would also like to acknowledge the support of our colleagues at the Software Engineering Institute, especially John Waclo, John Foreman, Brian Gallagher, and Len Estrin

Executive Summary

Program Managers (PMs) and consultants responsible for developing or acquiring software intensive systems identify risks in different ways. Some PMs and consultants rely on free-form brainstorming or volunteered statements. Others select methods based on convenience and familiarity. Often, there is no attempt to compare alternative risk identification methods or to match the methods to program needs.

To address this situation, researchers at the Carnegie Mellon® Software Engineering Institute (SEI) have developed an initial risk diagnostics "roadmap" and populated it with a set of risk diagnostic tools. The roadmap will help PMs, consultants, and other personnel to compare risk diagnostic methods and choose the best tools for their particular situation.

This technical note describes the evolution of the roadmap and describes the attributes of risk diagnostic tools that qualify them as appropriate for the roadmap. The authors use the attributes to identify and select three candidate risk diagnostic methodologies developed by the SEI. Section 1 presents the background, related work at the SEI, and need for a roadmap of risk diagnostic methods. Section 2 outlines the qualifications that candidate risk diagnostic methods must have to be considered for the roadmap. Section 3 presents three initial candidates: the SEI Software Risk Evaluation, Architectural Tradeoff Analysis Method®, and SEI Commercial Off-The-Shelf (COTS) Usage Risk Evaluation. The section also explains why these candidates fit the requirements, while other candidates did not. Section 4 describes follow on work that could be performed if funding were available.

® Carnegie Mellon and the Architectural Tradeoff Analysis Method are registered in the U.S. Patent and Trademark Office.

Abstract

The Risk Focus Team at the Carnegie Mellon® Software Engineering Institute (SEI) has identified a means of characterizing risk-based diagnostic methods and techniques. The Risk Focus Team has constructed a tentative “roadmap” for consultants, program managers, and other personnel involved in the systems and software acquisition community. The roadmap will help them to identify the appropriate risk diagnostic techniques for assessing threats to program success.

This technical note describes the characteristics that determine whether a risk diagnostic method qualifies for the roadmap. The technical note identifies three methods, the SEI Software Risk Evaluation, Architectural Tradeoff Analysis Method® and the SEI Commercial Off-The-Shelf (COTS) Usage Risk Evaluation that fit the characteristics described. The technical note also describes the characteristics of diagnostic methods that do not qualify for the roadmap.

1 Background

Risk is the possibility of suffering loss. In an acquisition project, loss refers to the diminished quality, increased cost, delayed completion, or failure of the acquired product.

Currently, many consultants in program risk management have no structured approach for identifying risks, relying instead on free-form brainstorming or volunteered statements. We believe this constitutes a serious disservice to the project or program, especially since there are many diagnostic tools and methods available. These include the following tools developed by the Carnegie Mellon® Software Engineering Institute: Software Risk Evaluation (SRE), the Architectural Tradeoff Analysis Method® (ATAM®), the Commercial Off-The-Shelf (COTS) Usage Risk Evaluation (CURE℠) [Carney 03], the Independent Technical Assessment (ITA), Software Quality Assessment Exercise (SQAE[1]), and the Operationally Critical Threat, Asset, and Vulnerability Evaluation (OCTAVE ®). Using any one of the risk-based diagnostics mentioned in this technical note would be a major improvement over simple brainstorming.

However, choosing among diagnostic methods can be a daunting task. Each method has its own scope, focus, and outputs. Some methods identify risks explicitly while others identify them implicitly. In addition, different diagnostic methods identify and represent risks in different ways, making it hard to contrast and compare methods.

As a result, those consultants and program managers (PMs) who *are* aware of diagnostic tools often select methods based on convenience and familiarity rather than applicability to program needs. In effect, they commit resources without really knowing whether they have chosen the most appropriate method or combination of methods. As a result, forecasting the total time, cost, and effort involved; the chances of success; and the nature of ongoing benefits becomes difficult, to say the least.

A similar scenario occurs when consultants and PMs try to compare risk data from one program with the risk data with another, since the leave-behind data provided by one method often is not comparable to the leave-behind data of another. For all these reasons, there is a need for a means to organize and compare risk diagnostic methods along with a standard way of representing risk.

® Carnegie Mellon, the Architectural Tradeoff Analysis Method, ATAM, Capability Maturity Model, CMM, CMMI and OCTAVE are registered in the U.S. Patent and Trademark Office.

℠ CURE and Capability Maturity Model Integration are service mark of Carnegie Mellon University.

1 The SQAE methodology and Framework was developed by and can be licensed through the Mitre Corporation.

1.1 Related SEI Work

This work has been brought about by the authors' responsibilities in the Acquisition Support Program (ASP) at the SEI. The ASP helps the Department of Defense (DoD) acquisition programs to identify and mitigate risk. Because of this "outward" perspective, we in ASP have become "tool selectors and appliers," rather than "tool developers." We are sensitive to the danger that "if the only tool you have is a hammer, everything looks like a nail," and we want to find new tools for our work.

The technical note *A Roadmap of Risk Diagnostic Methods*,[2] represents the first step to defining a new set of tools. It outlines a vision of the future by comparing acquisition support to the diagnostic and therapeutic approaches found in health care. This technical note builds on that work by presenting a potential means for organizing and comparing tools. Called the Risk Diagnostics Roadmap (RDR), it will allow practitioners to select the most appropriate diagnostic tools for their situation. The RDR incorporates the following SEI diagnostic methods:

Software Risk Evaluation (SRE). SRE engagements provide an experience-based framework for discovering, documenting, and analyzing fundamental expressions of risk (called "risk statements"). SREs create a safe, collegial environment for program personnel to provide risk information through interviews.

Architecture Tradeoff Analysis Method (ATAM). ATAM engagements evaluate the software architecture chosen to implement its design. ATAM will be described later in this document.

COTS Usage Risk Evaluation (CURE). This method explores the risks inherent in using COTS components. CURE will be described later in this document.

Operationally Critical Threat, Asset, and Vulnerability Evaluation (OCTAVE). The OCTAVE method explores risks to information technology assets in great depth. OCTAVE is more concerned about risks to the entire enterprise than it is about risks to a specific project or program. We believe that OCTAVE belongs in the roadmap, but we have not yet explored OCTAVE enough to be certain.

Capability Maturity Model Integration SM *(CMMI*®). The CMMI model contains a Risk Management (RSKM) process area that requires depositing project threats in risk repositories. CMMI is not in the roadmap, but it motivates its creation. Specifically, it calls for a risk repository to implement the RSKM process area and in so doing, drives our call for a standard way of articulating risks.

[2] Williams, R.; Ambrose, K.; & Bentrem, L. *A Roadmap of Risk Diagnostic Methods*, (CMU/SEI-2004-TN-002) Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2004.

1.2 DoD Need

Program Managers in the Department of Defense need to fully understand the tools that can help them manage their programs. The roadmap outlined in this document represents a coherent, logical framework that will allow PMs to apply all these disparate pieces singly or in combination. As a side benefit, we hope that the roadmap will help researchers to identify and address the diagnostic needs of PMs that are currently not being fulfilled.

2 The Risk Based Diagnostic (Rock)

To be included in the roadmap, a diagnostic must have a set of fundamental characteristics. We have come to speak of a generic, risk based diagnostic that qualifies for the roadmap as a rock.[3] We intend to formulate a set of attributes that define a rock, and identify which diagnostics qualify as rocks (or could qualify, with only minor adjustments) and which diagnostics do not qualify (or would require too much redefinition to qualify) as rocks.

To qualify as a rock, a risk based diagnostic must meet three requirements:

- It must include an identification phase in which risks are explicitly or implicitly listed.
- It must have an analysis phase in which a useful view of the risks is created to help the project decision-maker address key risks or risk areas.
- It must be capable of turning implicit risks into explicit objects that can populate a risk repository associated with the Risk Management process area of the CMMI. (We refer to this final requirement as the risk statement leave-behind.)

A risk based diagnostic can have additional phases and still be a rock. The three diagnostics that are the focus of this paper (SRE, ATAM, and CURE) have at least one phase subsequent to analysis. These phases either provide recommendations for mitigating the risks identified and analyzed, or else actively engage PMs in planning to mitigate those risks. However, we anticipate that most candidate diagnostics for the roadmap will also include phases beyond analysis. The following sections will detail the key requirements of a rock.

2.1 Risk Identification Phase

We believe that all candidate diagnostics will have a particular look and feel. Scanning various methods for the look and feel can help us identify candidate rocks. Accordingly, we feel that a rock will have the following look and feel:

The diagnostic will be conducted by a team of at least two people, and at least the team leader will be “outside” of the project or organization. At this point, we anticipate no upper limit on the team size.

[3] From the characterization often heard in technical and managerial circles, “Rock Management.” A “rock manager” orders subordinates to “Bring me a rock.” When they bring him one, he invariably says, “Not that rock, you fool!” This continues until the right rock is brought—assuming such a thing exists or can be created. In this paper, we are trying to describe the rock before we look for more of them.

If there are project "insiders" on the team, they will be there for one or both of the following purposes: (1) to provide information about organizational roles and relationships, the meaning of acronyms, and the location of key information ("insight") and (2) to learn the process used by the team so that in-house capabilities can be developed by the organization ("transition").

The process will depend primarily on one of the following for identifying risks:

- the knowledge and expertise of the members of the project itself ("The People in the Program")
- the expertise of the team members ("Outside Experts")
- broader community expertise embodied in a process or tool ("The Process or Tool Itself")

Figure 1 illustrates this triad of risk information sources. This figure also includes the three risk-based diagnostics we will examine in this paper, the SRE, ATAM, and CURE, as well as our initial assessment of two candidate diagnostics, OCTAVE and SQAE. (Even though the ITA is not a rock because it does not specify a method identifying and analyzing risks, we have shown it in the figure since most ITAs use outside experts in the "red team" mode.)

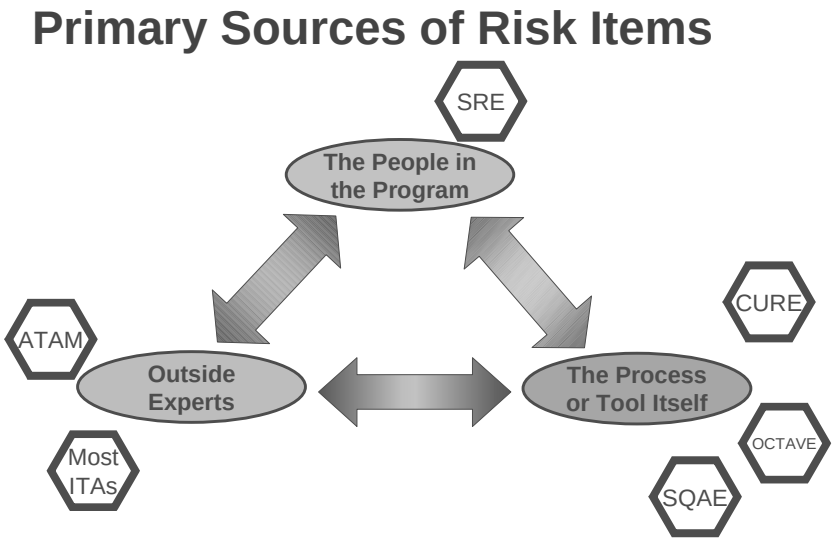

Figure 1: Primary Risk Identification Sources for Risk Based Diagnostics

To qualify as a rock, the Risk Identification phase of the candidate diagnostic method will focus on the future success of the program. When present conditions and past occurrences are identified, they will be used to anticipate the effects of those conditions or occurrences—not to assign blame.

The interactions between the team and the project personnel can either be many-to-many or many-to-one. For example, the team or sub-team conducting the diagnostic will either meet with a group or one person from the project.

The process can be conducted on-site at the project's location. Teleconferencing or video conferencing can be used when it is too impractical to meet face-to-face. Telephone conversations or video conferences should not be the only means of interaction.

2.2 Analysis Phase

To qualify as a rock, the analysis phase of a risk-based diagnostic will typically exhibit the following look and feel:

- The analysis processes will be group-based, rather than relying on one person's judgment.
- The analysis processes will isolate and prioritize significant risks or risk areas.
- The processes will have written descriptions that are clear enough for another team to follow, if someone on that second team has experience with the process. We have not determined whether defined qualifications or processes exist for being or becoming a team leader.

When criteria are necessary to make an analysis judgment, these criteria will be written in the analysis process description. Alternatively, a process for defining and documenting those criteria will be provided.

2.3 Risk Statements (Leave-Behind)

A key attribute of a rock is that it provides information that can be placed directly into a project's risk repository. Our model for the risk repository (and the risk management processes that use it) is the Risk Management (RSKM) process area in the CMMI [SEI 01, SEI 02]. CMMI does not characterize the type of a risk that goes into the risk repository. There is no guidance about form, length, or specificity. However, the underlying model for a risk is a discrete object that can be captured and will look like other objects in the repository.

The easiest way to define the characteristics of such risk objects is to look at the objects *already* in risk repositories, characterize them, and use the same attributes. This approach, however, is not practical since the risk objects that are already in repositories vary widely. We cannot claim to have seen all risk repositories, of course, but those we have seen show variety along the following dimensions:

Form. A risk object may describe the following:

- an undesirable future state (e.g., "We could miss milestone B.")
- a current issue or concern (e.g., "We don't have enough programmers on staff for this job.)
- a compound "if-then" construct (e.g., "If we don't hire enough programmers by May, we could miss milestone B.")
- a "condition-consequence" construct (e.g., "We don't have enough programmers to do this job, so we could miss milestone B and be late on the whole project.)

Specificity. Risk objects can be articulated at two levels:

- a high level (e.g., "We are applying several technologies that are new to us; and we may not have estimated the required learning curve correctly.")
- a low level (e.g., "We have no experienced Java programmers and object-oriented design methods are new to us.")

Clearly, if risk articulations are specific, there will be more of them than if they are not. Whether this is an advantage or a disadvantage will depend on the robustness of the project's risk management process—its ability to handle (analyze, plan, and track) a large number of individual risk objects.

Focus/orientation. Risk articulations can focus on two areas:

- the consequences of the risk materializing (e.g., "There is concern that we may be late in delivering the X module, which would affect the Y module.")
- the source of the risk ("The physics of the processes that are the basis of the X module are not completely understood, and understanding them may take longer than we planned.")

Vagueness. Items in the risk database can simply be vague (e.g., labor conflict). Without context, it is difficult for outsiders and those new to the project to understand how to analyze such a risk objectively or how a reasonable mitigation plan can be created to deal with it.

There may be more dimensions but these are enough to show that existing risk repositories cannot define the form of a risk object. If the risk objects in existing repositories were internally consistent, it would help; but this has not been the case.

In the absence of any standard risk object form, we propose to set a standard for the leave-behind created by risk-based diagnostics included in our diagnostics roadmap. To help us identify leave-behinds, we will look at those diagnostics that deal with risk in an *explicit*, *derivable*, and *actionable* manner, as defined below.

2.3.1 Explicit

The diagnostic needs to emphasize risks as a central theme, and to indicate that risks must be collected, written down, or otherwise specified, even if the form of the risk is not articulated.

2.3.2 Derivable

We must be able to derive the standardized risk object from the risks articulated (in any form) during the diagnostic. Most likely, we will need to look at past uses of the diagnostic and satisfy ourselves (and the diagnostic's process owner) that the standardized risk object conveys the same information as the original explicit form (and that the standardized risk object does not *distort* the risk from its original meaning).

2.3.3 Actionable

We must be able to address or mitigate risks through any or, preferably, all of the strategies:

- Reduce the impact of the risk, should it materialize (turn into a problem).
- Reduce the probability that the risk will actually turn into a problem.
- Shift the time frame for beginning any mitigation activity further into the future.
- Eliminate or ameliorate the present circumstances that give rise to the risk.

Our current model for the risk object is the *risk statement* generated by the Software Risk Evaluation process, one of the risk-based diagnostics that we will examine in this paper.

2.3.4 Graphic of a Rock

The following graphic summarizes the above discussion and presents a rock:

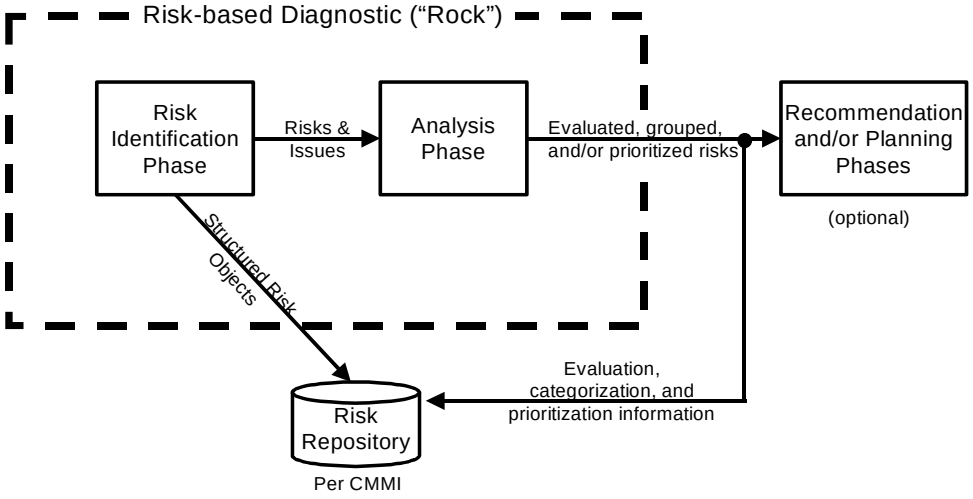

Figure 2: The Risk Based Diagnostic and its Context

2.4 Non-Rocks—Exclusionary Items

At this point, we have pretty good idea of what a rock looks like. But we need to say what a rock *doesn't* look like. What critical attributes will allow us say that a diagnostic does *not* qualify as a rock? We found that the most straightforward way to define this was by example. We looked for rock-like diagnostics that were familiar to us, and tested our understanding to see if these diagnostics would qualify; and if not, why not. The diagnostics were the CMM-Based Appraisal for Internal Process Improvement (CBA-IPI) [Dunaway 01], the Standard CMMI Appraisal Method for Process Improvement (SCAMPI) [Hays 93], and the Independent Technical Assessment (ITA) [Marz 01]. All three methods determine the current state of a project or organization, which make them rock-like. The following sections describe these methods and present our rationale for why they do not qualify as rocks.

2.4.1 CBA-IPI

The CMM-Based Appraisal for Internal Process Improvement (and its predecessor, the Software Process Assessment) is perhaps the most familiar of the model-based diagnostics used by the SEI and its transition partners. Like a rock, it has on-site data gathering and data analysis phases, and it has a particularly attractive attribute as a diagnostic: it uses a standard scale (the Software Capability Maturity Model—SW-CMM).

What makes the CBA-IPI process not a rock is its focus on the model rather than on risks. This makes it difficult to conceive how we could extract the standardized risk objects a rock should generate. At the same time, it is true that, according to the CBA-IPI process, an organization that is assessed at a lower maturity level is more at risk than one at a higher maturity level. Also, risks could possibly be framed in terms of specific Key Process Areas (e.g., "Our organization is weak in the ____ KPA, and as a result we might _____."). Therefore, it is possible that this method could be extended so that it could become a rock.

2.4.2 SCAMPI

The Standard CMMI Appraisal Method for Process Improvement is, in part, the successor to the CBA-IPI process. Yet it is also much more than that. First of all, it is not restricted to a single model. The SCAMPI technique can be performed against the SW-CMM, the Software Acquisition Capability Maturity Model (SA-CMM), the People Capability Maturity Model (P-CMM), and probably others. Furthermore, it is not restricted to being done *for* an organization (as a foundation or way-point for process improvement), but it can also be done *to* an organization, for auditing or source selection purposes. In this latter role, the SCAMPI technique also is the successor to the Software Capability Evaluation (SCE) method.

Another area of relevance is the SCAMPI technique's primary application with the Capability Maturity Model Integrated (CMMI) model. This includes a process area for risk management. It is precisely this process area's implicit risk repository that our concept of rock is intended to feed and support. Measurement against the CMMI model is not restricted

to a one-to-five scale like the SW-CMM model. In the context of the CMMI model's Continuous Representation, any one of the 25 process areas can be measured against a capability level of zero to five. This provides the potential for evaluating exactly how the organization falls short of the model. This, in turn, could be used by a particular project or program to articulate the risks it faces.

In spite of the SCAMPI model's broader reach and its connection to risk management, it has the same two deficiencies as the CBA-IPI process: it identifies *organizational* shortfalls, not project or program shortfalls; and does it not explicitly identify such shortfalls as risks.

So the SCAMPI model is not a rock, though it could be the basis for a new rock. We see real potential here for an optional follow-up process after a SCAMPI engagement has been performed to translate the model shortfalls into risk items that could be added to risk repositories.

2.4.3 ITA

The SEI uses the Independent Technical Assessment to gather, evaluate, and report issues facing a project or program. It can be performed *by* the project, or it can be done *to* the project. It always includes an on-site data-gathering phase. Generally speaking, the ITA generates a set of findings and a set of recommendations. Usually, these recommendations and findings are presented at the end of the on-site period or shortly thereafter. The presentation may be supplemented by a written report.

Outwardly, the ITA looks very much like a rock. It involves data gathering and analysis. The data gathered are often (but not always) characterized as risks to the project or program, and the recommendations are often presented as approaches for lowering these risks. However, this risk-oriented approach is not inherent in the ITA process. Moreover, the ITA leader determines the process to be used. Usually, the team leader selects whatever approach he or she has the most experience in using. Sometimes, however, the project or program's current circumstances (point in the life cycle, area of technical emphasis, domain, or pre-conceived major risk areas) can determine the approach. The approach can even be allowed to evolve during the data-gathering and analysis phase (i.e., it may have no predetermined process at all, with trust given to the team to come up with the best approach later, as they learn more about the issues).

In fact, the ITA can use a rock as its core process. For example, the SRE method has been used twice to date as the approach for ITAs. In a way, the ITA is a meta-process for risk determination. It defines processes for deploying the ITA team and reporting results, not necessarily identifying and analyzing those results. It also does not necessarily focus on risks, so the opportunity to generate standardized risk objects does not appear to be there. For these reasons, the ITA itself is not a rock.

2.5 Characteristics of Diagnostic Technologies

All the diagnostics that are ultimately included in the roadmap will feature similar characteristics or attributes that can be used to differentiate one from one another. In looking at the initial three diagnostics in the roadmap, we have attempted to identify the relevant characteristics. These have been divided into the following: nine thumbnail characteristics (Section 3, below) and an exhaustive list that includes every additional characteristic that we think could be relevant. We present the nine thumbnail characteristics in Table 1. The exhaustive list of characteristics can be found in Appendix A.

Table 1:Nine Thumbnail Characteristics

Characteristic	Description
1 Applicability	What characteristics must the project, program, or organization have for the diagnostic to be appropriate?
2 Purpose	What is the ultimate goal of the diagnostic? Can it be used for only one purpose at one time, or can it serve multiple purposes?
3 What the process looks for	Does the process look for risks in a particular project or program area? If so, what kinds of risks is it specifically seeking?
4 Current maturity	Where does this diagnostic stand in its evolution from "concept" to "community-owned?" If qualification to conduct or lead the diagnostic is required, approximately how many people are currently qualified?
5 Structure	What are the steps and durations of the steps of this diagnostic? How many on-site periods? How long do they last? How many team members are on a typical engagement? How long after the final on-site session are all deliverables provided and the diagnostic considered to be complete?
6 Qualifications required	Does a person have to be qualified or certified through some sequence of steps to participate on a team conducting the diagnostic? Are there additional requirements to lead such a team? If so, what are the steps for each? Who or what organization determines whether an individual is qualified?

Characteristic	Description
7 Source of risk information	Overall, who or what determines whether an issue, concern, or other item of discussion is to be captured as a risk? The primary sources identified to this point are the expertise of the diagnostic team, the project members themselves, and/or a predefined tool (the posited Team - Project - Tool triad).
8 Deliverables of the process	What should the project, program, organization, and higher level sponsors expect to receive as outputs (briefings, reports, other)? When should they expect to receive these items?
9 Effort and intrusiveness	How many team members are involved, for how many days? How many and what kind of project members are needed for meetings with the team conducting the diagnostic? What is the expected level of disruption of normal project work during the on-site period?

3 Comparison of Three Rocks

The following sections give more information about the first three risk diagnostics that we have identified as rocks.

3.1 Software Risk Evaluation (SRE)

The Software Risk Evaluation process relies on interviews for eliciting, capturing, and analyzing known issues and concerns. These elements are captured openly (on a flip chart or projected computer image) in a specific format (risk statement). The SRE process uses group-on-group interviews that follow rules of structure (peer groups only—no reporting relationships in the room) and coverage. The intent of the SRE is to create a critical mass of risk objects to be entered directly into a project or program's risk repository.

3.1.1 Thumbnail Characteristics

Applicability. The SRE technique can be applied to any project or program. It was originally developed for software development efforts, but it has proven to be effective for logistics and hardware-intensive efforts, as well. SRE technique variations have explored process-specific risks using a process improvement framework (e.g., the CMMI and SA-CMM processes) in place of the Taxonomy of Software Development Risks that originally supported the risk discovery interview.[4] The SRE technique also demonstrates effective methods for identifying appropriate issues, concerns, and risks. It is best used when there is no existing risk repository. While it *can* be used to add risk objects to an existing risk repository, this approach has not been formalized.

Purpose. The SRE technique is used to jump-start the risk management effort, or to bypass hidden communications barriers in the work group. Rules of confidentiality and non-attribution are usually imposed on the interviews to assure candor.

What the process looks for. Any item that anyone in the project or program believes may undermine the overall "Picture of Success." The "Picture of Success" (or, sometimes, "Picture of Failure") is a high level description of the desired outcome (or outcome to be avoided), expressed in terms of a specific date in the future.

[4] The Taxonomy was first published in *Taxonomy-Based Risk Identification* [Carr 93]. It has subsequently been republished in a different format, but with no change in content, in the *Continuous Risk Management Guidebook* [Dorofee 96] and in *SRE Method Description (Version 2.0) & SRE Team Members Notebook (Version 2.0)* [Williams 04].

Current maturity level. The SRE technique has been used in various forms for 11 years. It is mature and has been made available to the public without restriction.

Structure. A full SRE effort usually requires two on-site visits. The first visit (four or five days) is used to conduct interviews and analyze the results, culminating in a data confirmation briefing. The second visit (typically three days) helps the project or program decision makers develop a risk mitigation strategy to address the most important/highest priority risk areas determined by the SRE team (with project manager concurrence). This second visit also is capped by a briefing to everyone who was interviewed.

Qualifications. SRE team leads and team members can be qualified by the local organization using the process. To successfully use the process, team members and leaders should have experience and self-confidence; however, some people may be temperamentally unsuited for team leader and interviewer roles.

Source of risk information. The risk information comes from the interviewees. If they don't believe there is risk in a particular aspect of the project or program, no risk statement is recorded. The risks statements are recorded using the interviewees' words, and are confirmed by them as accurate. (NOTE: a variant of the process allows the interviewing team to add risk statements that they feel the interviewees have overlooked or denied.)

Deliverables of process. The deliverables of the SRE technique are a briefing at the end of the first period, an interim report, a briefing at the end of the second period, and a final report.

Effort and intrusiveness. An SRE team generally requires three outside members and one inside member full time during the first period, and two outside members full time in the second period. As many as 20 project people may be interviewed, each for three hours in the first period, and the same people will be expected to attend two one-hour briefings. The second period typically requires four or five decision makers (which must include the chief decision maker, presumed to be the project or program manager) for three days.

3.1.2 Why the SRE Technique Qualifies as a Rock

Relevant characteristics. The SRE technique explicitly identifies risks (risk statements), using a pre-defined format, and with the intention of directly loading them into a project risk repository. A risk is whatever an interviewee believes could threaten program success. The risks are analyzed by evaluating impact and probability, collecting them into risk areas (classifications), and prioritizing the risk areas using the Interrelationship Digraph method.

Risk ID phase. Risks are identified during the first visit.

Analysis phase. Risks are also analyzed during the first visit. The second visit ("Mitigation Strategy Planning") is beyond the definition of a rock.

Risk statements (leave-behind). As noted previously, the risk statement structure of the SRE technique is the model of what we would like to see all rocks produce.

3.2 Architecture Tradeoff Analysis Method (ATAM)

The Architecture Tradeoff Analysis Method is a scenario based, on-site process for evaluating a system's software architecture in light of its business goals and quality attributes. Risks are identified in terms of the gaps between the capabilities inherent within the evaluated architecture and the business goals defined by the organization as well as the tradeoffs the architecture has made or should make to meet its business goals and prioritized quality attributes.

3.2.1 Thumbnail Characteristics

Applicability. The ATAM process currently addresses risks of current or candidate *software* architectures. Because it targets the architecture, the diagnostic must be performed when architectural changes are possible, either at the beginning of the project or program, or at the point of a major redesign. However, it cannot be applied too early. The design team must be able to present a coherent architecture description. (All software designs have an architecture, but sometimes the architecture is not explicit, or even clear to the designers). The ATAM process is performed on the program's existing software architecture. If appropriate software documentation does not exist, then the ATAM effort does not proceed until the situation is corrected. Additionally, the on-site analysis phases (described later) cannot proceed without the participation of architecture decision maker(s), business goal manager(s) and the software's primary stakeholders. The ATAM process is not domain specific—it can be used for the software architecture of any end product.

Purpose. The ATAM is used to assure that the system architects and relevant stakeholders have considered the prioritized quality attributes and business goals in their design, and that they are aware of all the risks, non-risks, sensitivities, and tradeoffs that remain.

What the process looks for. The ATAM process first ensures that sufficient software architecture documentation exists and that stakeholders agree on what will be evaluated. Following this, facilitated sessions with the principal designers of a software development project and the major stakeholders in that project are scheduled. These two sessions elicit and document scenarios that are used to evaluate the software architecture in terms of quality attributes and business goals. The scenarios describe how the finished software design should behave, in terms of six quality attributes: availability, modifiability, performance, security, testability, and usability. Additional quality attributes can also be considered. After developing the scenarios, the system designers present their current architectures, which are tested against the scenarios. This generates risks, non-risks, sensitivity points, and tradeoff points for the candidate architectures.

Current maturity level. The ATAM process has been piloted and modified for seven years and is considered "mature" by SEI standards. At this point, it is being transitioned to the SEI Acquisition Support Program. Currently, there are seven qualified ATAM facilitators in the Product Line Systems group of the SEI. An additional number of ATAM facilitators (fewer than five) work in industry.

Structure. ATAM process consists of four phases, Phase 0 through Phase 3. Phase 0 lasts about a half day, and focuses on Partnership and Preparation. It involves the Lead Evaluator and perhaps one other team member. Phase 1, Initial Evaluation, is architecture-centric and focuses on eliciting detailed architecture information and on performing a top down analysis of the architecture. Phase 1 typically requires the full ATAM team (nominally four people including the Lead Evaluator) and the target organization's core team of technical decision makers for about two days on-site. Phase 2, Complete Evaluation, is stakeholder-centric, and elicits diverse stakeholder points of view. Phase 2 typically requires the full ATAM team and an assembly of the software product's stakeholders, for about two days on-site. Phase 2 is scheduled about one week after Phase 1 has been completed. In Phase 3, a final report is produced, typically within two weeks after Phase 2.

Qualifications. Participating on an ATAM team requires taking two two-day courses. Becoming a team Lead Evaluator requires an additional three two-day courses, participating in two ATAM efforts as a team member, and then being observed and approved by a qualified team Lead Evaluator while acting as team Lead Evaluator on a third ATAM engagement.

Source of risk information. The ATAM team determines the risks, sensitivity points, and tradeoff points for the evaluated architectures. It is crucial for the core team members of the customer's organization to agree that the risks determined apply to the program. These judgments depend on their personal expertise, training, program context, knowledge, and background.

Deliverables of process. The ATAM process results in a final report that documents all findings. There are no other deliverables.

Effort and intrusiveness. The ATAM process requires less team time and on-site participant time than the SRE. It may, in most cases, be less intrusive to the project as well, because it does not involve as large a cross section of project personnel as does the SRE. The Phase 2 meeting with stakeholders may be burdensome, depending on how difficult it is to assemble the major product stakeholders. However, key stakeholder participation helps to ensure that no critical stakeholder requirements are missed—an important benefit.

3.2.2 Why the ATAM Method Qualifies as a Rock

Relevant characteristics. The ATAM process explicitly identifies risks, though not in a pre-defined format, nor with the intention of directly loading them into a project risk repository. A

risk in the ATAM process is an architectural decision that is problematic in light of the quality attributes that it affects. The risks are analyzed by collecting them into risk themes (classifications), and by presenting them in the context of the architecture's capability to achieve the organization's desired goals.

Risk ID phase. Risks are identified during Phases 1 and 2 of the ATAM effort. The prior phase set up the criteria for identifying and analyzing the risks.

Analysis phase. Scenarios are generated and used to analyze risks during Phases 1 and 2 of the ATAM also. The identification and analysis steps in the ATAM process are not totally separate, but that is not a critical attribute for being considered a rock. The ATAM process is not intended to determine whether any of the risks identified are the most important risks to the project as the SRE technique does—that is left for the project to determine for itself later, in light of *all* the risks in the project's risk repository. The analysis phase includes categorizing risks into risk themes that highlight higher level risk trends.

Risk statements (leave-behind). As noted previously, the risks of the candidate architectures are explicitly collected and documented. It would not be difficult to capture risk statements in a Condition-Consequence form as the SRE does, nor to capture context to accompany those risk statements—it is simply not a priority of the current ATAM process. Rather, it is left to the project to do after the ATAM engagement is over. Sufficient risk "raw material" exists in the Phase 2 results to begin to address the needs of the organization's risk repository.

3.3 COTS Usage Risk Evaluation (CURE)

The COTS Usage Risk Evaluation examines the ways that an organization uses COTS software within a software-intensive system acquisition and/or development project. The CURE method, as developed by the SEI, elicits *COTS software usage issues* that have surfaced in prior programs. Thus, CURE materials, as provided by the SEI, are not intended to be general risk diagnostic tools nor are they intended for use as a COTS software product analyzer. However, the CURE approach to risk evaluation is not limited to COTS software usage risks. The scope of the CURE method was limited by its SEI developers who wanted to capture lessons learned from programs that used COTS software.

3.3.1 Thumbnail Characteristics

Applicability. The CURE approach elicits and analyzes the COTS software usage risks in a program or project. The method is most effective when applied early in a project's lifecycle, ideally at the time of contractor selection. CURE can be performed on either/both program acquisition offices and system development contractors. To succeed, the CURE team must have access to a project's key COTS software decision makers (e.g., program manager, chief architect, etc.).

Purpose. CURE finds risks early in the program relating to the use of COTS software products and reports those risks back to the organization's managers. This allows managers to map out a strategy to address the risks uncovered by the evaluation and to monitor their mitigation.

What the process looks for. The CURE process seeks to uncover programmatic evidence of COTS software risks that have plagued past programs. The process captures and leverages lessons learned. To suit program needs, the evaluation team tailors the focus of a CURE-provided agenda. The team uses the agenda to drive an in-depth data gathering discussion interview with key COTS software project decision makers. The data gathering effort consists of 15 discussion topics that are grouped into four main investigation areas: *General, Business, Infrastructure,* and *Lifecycle*.

Current maturity level. The CURE process has been in use since 2000. SEI evaluation teams have performed the process in organizations within the DoD and other government agencies as well as within industrial organizations. At this point (September 2004), the CURE process is being transitioned to the SEI Acquisition Support Program. Currently, there are seven qualified CURE evaluators within the SEI.

Structure. The CURE process consists of four primary activities: Initial Questionnaire, On-site Discussion Interview, Data Analysis, and Results Presentation. The Initial Questionnaire is sent to the client organization about four weeks in advance of the on-site Discussion Interview and is used to acquaint the evaluation team with the project. The On-site Discussion Interview is led by a CURE team, numbering three external evaluators. During the Discussion Interview, the team spends one full day with the target organization's key COTS software decision makers to gather program data. Following the Discussion Interview, the team performs a three-day Data Analysis process to formulate risks and strengths (typically 8-to-12 of each) that are significant to the program. The results of the team's analysis (including risk mitigation recommendations) are provided in a confidential Results Presentation Out-briefing within four days after the Discussion Interview.

Qualifications. Qualifying for a CURE evaluation team currently involves participating in an evaluation while being mentored by an experienced CURE evaluator.

Source of risk information. The CURE evaluation team alone formulates the list of risks and strengths that pertain to the evaluated program. The evaluation team utilizes the data gathered, their prior experience, and the guidance provided by the CURE materials to arrive at the list of risks and strengths. The team, leveraging their prior experience, also proposes one or more mitigations to each risk.

Deliverables of process. The CURE effort ends with a Results Presentation, as previously described. There are no other deliverables.

Effort and intrusiveness. One full day of dedicated participation that the CURE process requires from a project's key personnel could be burdensome. Understanding the importance of the CURE process and applying it early in the program lifecycle can justify the time and effort invested.

3.3.2 Why the CURE Process Qualifies as a Rock

Relevant characteristics. The CURE process explicitly identifies COTS software usage risks, though not in a pre-defined format, nor with the intention of loading them into a project risk repository. It does provide an agenda that structures the data gathering effort, as well as a database containing template risk factors and template risk conditions. The CURE database embodies significant knowledge resulting from previous studies and analyses of COTS software- intensive programs. CURE evaluators then use the database together with program-specific data to perform a structured data analysis. The evaluation team formulates a resultant set of COTS software usage risks and strengths that are ranked according to criticality, severity, or imminence.

Risk ID phase. Risks are identified during the Data Analysis activities of the CURE process.

Analysis phase. Risks are analyzed during the Data Analysis activities of the CURE process. The identification and analysis steps are not totally separate in the CURE process, but that is not a critical attribute for being considered a rock. As noted previously, the CURE evaluation team does rank the risks it has identified in order of criticality, severity, or imminence.

Risk statements (leave-behind). As noted previously, the COTS software usage-specific risks that have been identified by the evaluation team are explicitly collected and documented in the Results Presentation. It would not be difficult to translate risks identified in the CURE Results Presentation into a Condition-Consequence form. The CURE team generally reports results of their evaluation as follows:

- a perceived or anticipated negative consequence or a perceived positive strength
- supporting evidence (specific risk conditions/risk factors that lead to the negative consequence)
- recommended mitigation actions
- mitigation action owner (i.e., government/customer, contractor, or both)

4 Next Steps—Where Do We Go from Here?

This is the end of the effort that has been funded by CRSIP. At this point, no additional funding is available for further inquiry and cataloging.

If additional funding were available, we would do the following:

1. Complete the data table of Section 2 for CURE, ATAM, and SRE processes, and make this information available to the public.
2. Evaluate the SEI's OCTAVE and MITRE's SQAE diagnostic methods and include them, if possible, in this roadmap.
3. Find additional risk based diagnostics from all sources that meet our definition of a rock. Enlist the help of the International Council on Systems Engineering (INCOSE), Project Management Institute (PMI), IEEE, other FFRDCs, and UARCs in seeking out these diagnostics.
4. Create a complete roadmap for all known diagnostics with specific selection guidance to the program consultant.

That is as far as we can envision this work going at this time. To go further, the work would have to be embraced by the systems and software communities, and they would have to nurture this body of knowledge to help it grow. This work will never achieve its true potential if it is perceived as only a catalog of SEI diagnostics.

Perhaps one of the most useful aspects of continuing this work would be identifying holes in the roadmap of risk diagnostics—areas that should be developed to address risks in specific technical, process, and programmatic issues. With community leadership, various organizations could develop, pilot, and transition the needed diagnostics following a common framework and understanding of the new diagnostics' place in the roadmap.

A more immediate side effect of this work could be the development of process improvement based risk diagnostics as an adjunct to the CMMI method. This could be a separate process, or an extension of the SCAMPI technique that would explicitly state risks and identify process improvement gaps. These risks could then be fed into the project or program's risk repository to be managed by decision makers.

Appendix A

The following table provides additional characteristics of a risk-based diagnostic that may be needed to fully describe the method—in a catalog of methods, for example. These characteristics supplement the nine characteristics used to thumbnail the CURE, ATAM, and SRE processes in Section 3. The listing attempts to be as exhaustive as possible, but it needs to be used on all rock candidates. This would verify that all characteristics are valid and that they capture everything necessary to decide whether or not to use the diagnostic in a particular situation.

At this point, the four general groupings of characteristics, *Application Environment*, *Features and Restrictions*, *On-site Intervention*, and *Issues*, are only preliminary; we expect the arrangement of features to change as more diagnostics are added to the roadmap and more characteristics become apparent.

Characteristic	Description
A. Application Environment	This general category includes characteristics that define what must be in place before the diagnostic can be successfully executed. It also includes qualities that will lead to the intended long-term organizational improvement.
1. Support	Characteristics of the key people in the organization, and the extent to which they want the diagnostic to yield accurate results or lead to changes in the way they do business.
1.a) Higher management sponsorship	To what extent is direct, public support of the diagnostic needed? How many levels above the location of the diagnostic is it targeted?
1.b) Internal champions and change agents	To what extent do support people (usually recognized technical leaders) need to be present in the project, program, or organization for it to succeed or have the intended effect?
1.c) EPG or other process improvement organization	Does a staff (i.e., not project or program oriented) group focused on process improvement need to be in place to succeed or have the intended effect?

Characteristic	Description
2. Focus	What characteristics must the project, program, or organization have for the diagnostic to be appropriate?
2.a) Technical area	Is the diagnostic specific to a particular area of the technical problem (e.g., COTS product, software architecture)
2.b) Point(s) in the lifecycle	Is the diagnostic only useful at particular points of a project or program's lifecycle (e.g., at source selection, during initial design, or at milestone events)?
2.c) Domain	Is the diagnostic only useful in particular technical domains (e.g., embedded real-time software, databases, distributed communications)?
3. Intent	What is the ultimate goal of the diagnostic? Can it be used for only one purpose at one time, or can it serve multiple purposes?
3.a) Risk mitigation	If the diagnostic is focused on mitigating risk, rather than on the risks themselves, this should be stated.
3.b) Process improvement	If the diagnostic can help the project, program, or organization improve the way it works, do adjustments need to be made in the diagnostic (e.g., openness and communication of all diagnostic steps, special briefings)?
3.c) Audit	If the diagnostic can be used to judge the project, program or organization, perhaps leading to punitive action, do adjustments need to be made in the diagnostic (e.g., confidentiality and non-attribution of diagnostic steps, special protection of data)?
3.d) Other	Other purposes might include source selection, protection of corporate assets, and/or quality assurance. If so, what adjustments would be needed for these?

B. Features and restrictions	Key general characteristics.
1. Basis/foundation/theory	Where did the diagnostic originate (particular technology domain or professional discipline)? What other concepts or diagnostics was it based on? Does it have academic credentials?
2. Rigidity/rigor of the process	How much approach flexibility is there for the intervention team to determine on-site? Are there tailoring guidelines?
3. Variations for different conditions	Does the diagnostic include specific adjustments or alterations for circumstances such as point in the life cycle, application domain, or physical distribution of target project, program, or organization (e.g., co-located vs. widely separated)?
C. On-site intervention	Characteristics of the period when the diagnostic team interacts with the target project, program, or organization.
1.) Schedule	How long will it last? How much will the work of the target project, program, or organization be disrupted? How soon will the outcome be known, and what form will it take?
1.a) Duration	The typical period from a "go" decision until the diagnostic execution team has completed its work.
1.b) Preparation period	How long does it typically take to get everything in place (personnel trained, facilities arranged, meetings with target groups firmly scheduled) before the on-site intervention can begin?
1.c) On-site period	How much time altogether? Are there multiple phases? What is the schedule for a typical day?
2. Interaction structure	What's going to happen? What are the mechanics when the on-site activities are happening?
2.a) Intervention team size	How big can the team doing the work be? How small? Can it include people from the project?

2.b) Target group makeup and size	How many members of the project, program, or organization will be targeted over the entire on-site period? How many can be engaged in a single session?
2.c) Engagement rules	What rules apply during the sessions where data gathering is taking place?
2.d) Time period of interaction	How long does a data gathering session last? Is it variable (i.e., are there minimum and maximum limits)?
2.e) Who is invited	Are there selection rules for the people who meet with the diagnostic team? Rules for who can be in any given session (e.g., no reporting relationships allowed, people need to be from the same work area, people need to be randomly selected)?
2.f) Interaction configuration (many on many, many on one, etc.)	When the session takes place, how many team members and how many target group people can be in the room? Is one person in control of the discussions, or can anyone on the team ask any question at any time? Are scripts used?
3. Identify Phase	What is the process used to elicit risks? Is it only one approach, or does the diagnostic use several approaches and integrate the information?
3.a) Interviews	Can many interviews occur simultaneously? Do all team members have to hear all information firsthand to analyze it later? Is information gathered openly (e.g., written on flip charts)?
3.b) Surveys	Is an instrument distributed and results collected? Does this have to be done during the on-site period, or can it be before and/or after?
3.c) Checklists	Does the diagnostic use specific checklists to elicit data? If so, are they “yes/no” checklists or discussion guiding checklists?
3.d) Document review	Does the on-site period include reviewing working documents and artifacts created by the project, program, or organization?
3.e) Other	Other possibilities include expert systems, process models, and metrics from similar projects, programs, and organizations.

4. Analysis phase	Once the data have been gathered, how does the diagnostic team determine what items are the most important? Do the members of the target project, program, or organization participate in analysis steps?
4.a) Expert opinion	Is the risk data simply evaluated and prioritized by the native expertise of the members of the team conducting the diagnostic?
4.b) Comparison to standard	Is the data evaluated against industry or physical norms (e.g., medical tests)?
4.c) Prioritization	Are specific methods or tools used to determine the most important risks to the project, program, or organization? Examples might be Analytical Hierarchy Process, Comparison Risk Ranking, and multi-voting.
4.d) Interrelationships	Are the relationships between identified risks analyzed? Example techniques include Interrelationship Digraphs, Cause-Effect ("fishbone") diagramming and affinity grouping.
4.e) Categorization	Are risks put into pre-defined "buckets"? Categories could be defined by taxonomies, sources of risks, or area where the consequences would show up, for example. Does the diagnostic use or allow multiple categorizations?
4.f) Other	Are there any innovative analysis approaches that make this diagnostic unique?
D. Issues	Anything that might steer a decision maker away from applying this diagnostic approach.
1. Can process be modified easily to generate standardized statements of risk?	Does the diagnostic directly generate items that can go into a risk repository per CMMI, or can it be readily modified to do so?
2. Cost/burden to client	What pain will this diagnostic impose on the target project, program, or organization?
2.a) Out of pocket costs	How much direct funding must the target project, program, or organization provide to the organization that conducts the diagnostic?

2.b) Staffs of client staff	How much time will the target project, program, or organization lose from its normal business while the diagnostic is being conducted?
2.c) Facilities, equipment and other support	What further hidden costs are inherent in the conduct of this diagnostic?

Bibliography

URLs are valid as of the publication date of this document.

[Alberts 01] Alberts, C.; Dorofee, A.; & Allen, J. *OCTAVE Catalog of Practices, Version 2.0* (CMU/SEI-2001-TR-020, ADA396654). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2001. http://www.sei.cmu.edu/publications/documents/01.reports/01tr020.html

[Barbacci 02] Barbacci, M. *SEI Architecture Analysis Techniques and When to Use Them* (CMU/SEI-2002-TN-005, ADA413696). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2002. http://www.sei.cmu.edu/publications/documents/02.reports/02tn005.html

[Byrnes 96] Byrnes, P. & Phillips, M. *Software Capability Evaluation Version 3.0 Method Description* (CMU/SEI-96-TR-002, ADA309160). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1996. http://www.sei.cmu.edu/publications/documents/96.reports/96.tr.002.html

[Carney 03] Carney, D.; Morris, E. & Place, P. *Identifying Commercial Off-the-Shelf (COTS) Product Risks: The COTS Usage Risk Evaluation* (CMU/SEI-2003-TR-023, ADA418382). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2003. http://www.sei.cmu.edu/publications/documents/03.reports/03tr023.html

[Carr 93] Carr, M.; Kondra, S.; Monarch, I.; Ulrich, F.; & Walker, C. *Taxonomy-Based Risk Identification* (CMU/SEI-93-TR-006, ADA266992). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1993. http://www.sei.cmu.edu/publications/documents/93.reports/93.tr.006.html

[Dorofee 96] Dorofee, A. et al, *Continuous Risk Management Guidebook*. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1996.

[Dunaway 01] Dunaway, D.; & Masters, S. *CMM®-Based Appraisal for Internal Process Improvement (CBA IPI) Version 1.2 Method Description* (CMU/SEI-2001-TR-033, ADA3399227). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2001 http://www.sei.cmu.edu/publications/documents /01.reports/01tr033.html

[Hays 93] Hays, W.; Miluk, G.; & Kitson, D. *A Family of SCAMPI Appraisal Methods* (Presentation). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1993. http://www.sei.cmu.edu/cmmi/presentations/scampi-family.pdf

[Kazman 00] Kazman, R.; Klein, M.; & Clements, P. *ATAM: Method for Architecture Evaluation* (CMU/SEI-2000-TR-004, ADA382629). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2000. http://www.sei.cmu.edu/publications /documents/00.reports/00tr004.html

[Marz 01] Marz, T.; & Plakosh, D. *Real-Time Systems Engineering: Lessons Learned from Independent Technical Assessments (ITAs)* (CMU/SEI-2001-TN-004, ADA388771). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2001. http://www.sei.cmu.edu/publications/documents/01.reports /01tn004.html

[SEI 01] Software Engineering Institute, Assessment Method Integrated Team. *Standard CMMI Appraisal Method for Process Improvement Version 1.1: Method Definition Document* (CMU/SEI-2001-HB-001, ADA3399204). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2001. http://www.sei.cmu.edu/publications/documents/01.reports /01hb001.html

[SEI 02] Software Engineering Institute, CMMI Team. *CMMI® for Systems Engineering/Software Engineering/Integrated Product and Process Development/Supplier Sourcing, Version 1.1,* (CMU/SEI-2002-TR-011, ADA382587). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2002. http://www.sei.cmu.edu /publications/documents/02.reports/02tr011.html

[Williams 99] Williams, R.; Pandelios, G.; & Behrens, S. *SRE Method Description (Version 2.0) & SRE Team Members Notebook (Version 2.0*) (CMU/SEI-99-TR-029 ADA001008). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1999. http://www.sei.cmu.edu/publications/documents/99.reports/99tr029/99tr029abstract.html

[Williams 04] Williams, R.; Ambrose, K.; & Bentrem, L. *A Roadmap of Risk Diagnostic Methods*, (CMU/SEI-2004-TN-002) Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2004.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
(Leave Blank)	September 2004	Final

4. TITLE AND SUBTITLE	5. FUNDING NUMBERS
Risk Based Diagnostics	F19628-00-C-0003

6. AUTHOR(S)

Ray C. Williams, Kate Ambrose, Laura Bentrem, Tom Merendino

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213	CMU/SEI-2004-TN-013

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

11. SUPPLEMENTARY NOTES

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

13. ABSTRACT (MAXIMUM 200 WORDS)

The Risk Focus Team at the Carnegie Mellon® Software Engineering Institute (SEI) has identified a means of characterizing risk-based diagnostic methods and techniques. The Risk Focus Team has constructed a tentative "roadmap" for consultants, program managers, and other personnel involved in the systems and software acquisition community. The roadmap will help them to identify the appropriate risk diagnostic techniques for assessing threats to program success.

This technical note describes the characteristics that determine whether a risk diagnostic method qualifies for the roadmap. The technical note identifies three methods, the SEI Software Risk Evaluation, Architectural Tradeoff Analysis Method®, and the SEI Commercial off-the-shelf (COTS) Usage Risk Evaluation that fit the characteristics described. The technical note also describes the characteristics of diagnostic methods that do not qualify for the roadmap.

14. SUBJECT TERMS	15. NUMBER OF PAGES
software risk evaluation, software process improvement, risk diagnostic tools, software acquisition, roadmap	45

16. PRICE CODE

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