

7-2009

A Proactive Means for Incorporating a Software Architecture Evaluation in a DoD System Acquisition

John K. Bergey
Carnegie Mellon University, jkb@sei.cmu.edu

Follow this and additional works at: <http://repository.cmu.edu/sei>

This Technical Report is brought to you for free and open access by Research Showcase @ CMU. It has been accepted for inclusion in Software Engineering Institute by an authorized administrator of Research Showcase @ CMU. For more information, please contact research-showcase@andrew.cmu.edu.

A Proactive Means for Incorporating a Software Architecture Evaluation in a DoD System Acquisition

John K. Bergey

July 2009

TECHNICAL NOTE
CMU/SEI-2009-TN-004

Research, Technology, and System Solutions Program
Architecture-Centric Engineering Initiative

Unlimited distribution subject to the copyright.

<http://www.sei.cmu.edu>



This report was prepared for the

SEI Administrative Agent
ESC/XPK
5 Eglin Street
Hanscom AFB, MA 01731-2100

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.

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.

Copyright 2009 Carnegie Mellon University.

NO WARRANTY

THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

RFP/CONTRACT LANGUAGE AND ACQUISITION ARTIFACTS ARE PROVIDED AS EXAMPLES ONLY. THE SEI DOES NOT WARRANT THE EFFECTIVENESS OF ANY SUCH LANGUAGE OR THE ARCHITECTURE EVALUATION PLAN FOR USE IN DOD OR GOVERNMENT ACQUISITIONS. IT IS THE RESPONSIBILITY OF THE PROGRAM MANAGER AND CONTRACTING OFFICER HAVING COGNIZANCE OVER THE ACQUISITION TO DETERMINE THE APPROPRIATENESS AND/OR SUITABILITY TO A SPECIFIC ACQUISITION PROGRAM. MOREOVER, THIS REPORT DOES NOT ADDRESS OR TOUCH ON THE LEGAL TERMS AND CONDITIONS THAT ARE RELEVANT TO A SYSTEM OR SOFTWARE ACQUISITION. SUCH LEGAL TERMS AND CONDITIONS WILL HAVE TO BE APPROPRIATELY INCLUDED IN THE RFP/CONTRACT BY THE ACQUISITION ORGANIZATION IN CONCERT WITH ITS CONTRACTING OFFICER AND LEGAL STAFF.

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

Internal use. Permission to reproduce this document and to prepare derivative works from this document for internal use is granted, provided the copyright and "No Warranty" statements are included with all reproductions and derivative works.

External use. This document may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other external and/or commercial use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

This work was created in the performance of Federal Government Contract Number FA8721-05-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.

Table of Contents

Acknowledgments	vii
Executive Summary	ix
Abstract	xi
1 Introduction	1
2 The Importance and Benefits of Conducting an Architecture Evaluation	3
3 Acquisition Considerations	5
4 An Overview of the Approach for Incorporating an Architecture Evaluation in an RFP/Contract	7
5 Sample RFP/Contract Language	9
5.1 Section C: Statement of Work (SOW)	9
5.2 Section L: Instructions to Offerors	10
5.3 Section M: Technical Evaluation Criteria	10
5.4 Section J: Contract Data Requirements List (CDRL)	11
6 A Sample Software Architecture Evaluation Plan	13
6.1 A Two-Part Plan	13
6.2 First Part of the Software Architecture Evaluation Plan	14
6.3 Second Part of the Software Architecture Evaluation Plan	17
7 Summary	21
References/Bibliography	23
Appendix A Contract Data Requirements List (CDRL) and Data Item Description (DID) for a Software Architecture Description (SWARD) Document	A-1
Appendix B Acronym List	B-1

List of Figures

Figure 1: Proactive Approach for Incorporating a Software Architecture Evaluation in an RFP/Contract

7

List of Tables

Table 1:	Type of Information to Be Included in Related Acquisition Documents	12
Table 2:	Participation in ATAM Phases	18

Acknowledgments

The author would like to thank Stephen Blanchette, Tim Morrow, Michael Gagliardi, and Mark Klein for their careful review and comments. A special thank you is owed to Tim Morrow and Mike Gagliardi for their help in creating the RFP language that was used for conducting software architecture evaluations in DoD acquisitions.

Executive Summary

Software plays a critical role in almost every Department of Defense (DoD) acquisition and is often cited as the reason for frequent cost overruns, schedule slippages, and quality problems. As a result, there is increased emphasis on finding effective ways for an acquisition organization to reduce risk when acquiring *software-reliant*¹ systems. While there is no “silver bullet,” an architecture-centric acquisition approach has proven to be an effective way of reducing acquisition risk [Nord 2009]. At the heart of such an approach is enabling an acquisition organization to conduct a *software architecture evaluation* in collaboration with the development contractor early in the system development cycle.

A system’s software architecture should be evaluated as early as possible in the development cycle because the architecture

- is the earliest design artifact that represents the indispensable first step towards a software solution
- instantiates the design decisions that are the most profound, the hardest to change downstream, and the most critical to get right
- largely determines a system’s quality attributes (e.g., performance, interoperability, security, openness, safety, and so forth)
- plays a major role in the ability of a system to satisfy its key performance parameters (KPPs) and other stakeholder-specific acquisition and mission drivers
- is amenable to evaluation and enables design risks to be discovered early so they can be mitigated in a cost-effective and timely manner, thus avoiding costly rework downstream
- is the highest level abstraction of the software’s design—making it ideally suited to an acquisition organization’s technical oversight and contract monitoring responsibilities in light of its limited resources
- provides the knowledge base that paves the way for analyzing design tradeoffs and predicting the impact (i.e., quality, cost, and schedule) of proposed design changes and future plans to further evolve the system

While architecture evaluation is becoming increasingly more routine in the commercial workplace, it is still far from being common practice in the DoD.² This situation may be due, in part, to a lack of understanding of what is involved in conducting an architecture evaluation, the benefits it affords, and what it takes to include it in a Request for Proposal (RFP)/contract for a system

¹ A *software-reliant* system is one whose behavior (e.g., functionality, performance, safety, security, interoperability, and so forth) is dependent on software in some significant way.

² One notable exception is the Army, which is spearheading an effort to have its program managers apply architecture-centric acquisition practices (and architecture evaluations in particular) in their system acquisitions. That effort is the result of Army leadership and sponsorship of the Army’s Strategic Software Improvement Program (ASSIP) that has been providing training, funding, and guidance to its acquisition organizations and conducting workshops to encourage Army programs to adopt and pilot architecture-centric practices and share lessons learned.

acquisition. Another reason why architecture evaluation may not be routinely applied in the DoD is the mistaken notion that acquisition reform—and performance-based contracting, in particular—precludes it. This is not the case; software architecture evaluations have now been successfully (and proactively) conducted on U.S. Army, U.S. Navy, and U.S. Air Force programs using the SEI Architecture Tradeoff and Analysis Method[®] (ATAM[®]). One purpose of this technical note is to increase awareness throughout the DoD of the benefits of conducting an architecture evaluation. However, the primary purpose is to provide guidance on how to contractually incorporate architecture evaluations in an acquisition. The central idea is to provide a sample Software Architecture Evaluation Plan that can be easily customized by a DoD program office for use in its own RFP and contracts.

The sample plan described in this report is proven and practical and has been successfully used in DoD acquisitions. The plan covers all aspects—that is, the “who, why, when, where, and how”—of the government approach to conducting a software architecture evaluation during the contract performance phase of a DoD system acquisition. These aspects include describing the prerequisites for conducting the evaluation, the specific architecture evaluation method, how the results will be used, and the roles and responsibilities of the acquisition organization, including the architecture evaluation team, the system development contractor, and other designated stakeholders who will participate in the evaluation. Moreover, the plan is designed to be easily customizable to facilitate compatibility with the acquisition organization’s terminology, acquisition practices, and planned acquisition events that would impact the timing of the event-driven software architecture evaluation. In short, the sample Software Architecture Evaluation Plan is sufficiently comprehensive to safeguard the interests of both the acquisition organization and the ultimate development contractor. An important aspect of the plan is that it provides prospective offerors (i.e., bidders) with all the information they need to “cost out” their participation in the architecture evaluation and appropriately incorporate it in their technical and cost proposals.

In summary, the sample plan provides an acquisition organization with a proactive means for incorporating an architecture evaluation in an RFP/contract to reduce software acquisition risk. And it provides potential offerors the insight they need to understand the impact of, and government’s expectations for, conducting an architecture evaluation in an acquisition context.

[®] Architecture Tradeoff Analysis Method and ATAM are registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

Abstract

Department of Defense (DoD) acquisition programs routinely acquire systems that are highly software reliant. With the increasing functionality and complexity of these systems, software problems often contribute to schedule slippages, cost overruns, and system deficiencies. As a result, DoD acquisition organizations need to take proactive measures to reduce software acquisition risk. They cannot afford to just perform perfunctory reviews during software development and wait until after system delivery to determine whether key performance parameters (KPPs) and other acquisition/mission drivers that are important to stakeholders will be achieved.

Since the architectural design of a system and its software has a major influence on whether a system achieves its KPPs (and other acquisition/mission drivers), conducting an architecture evaluation is an effective means for reducing software acquisition risk. The evaluation involves the active participation of key stakeholders and focuses on identifying risks (and overarching risk themes) that can affect the architecture's ability to accommodate the system's quality attribute requirements (e.g., performance, safety, and security). Satisfying these quality attribute requirements is key to satisfying KPPs and other stakeholder-specific acquisition/mission drivers.

This technical note describes a proactive means for incorporating such a software architecture evaluation (in collaboration with the development contractor) early in the contract performance phase of a DoD system acquisition. The proven means that is described revolves around a sample Software Architecture Evaluation Plan that a DoD program office can easily customize and use in its own Request for Proposal (RFP)/contract. The sample plan covers all aspects—that is, the “who, why, when, where, and how”—of the government's approach to conducting a software architecture evaluation during an acquisition. Moreover, this sample plan provides acquisition organizations and potential offerors with the insight needed to understand the impact of, and government's expectations for, proactively conducting a software architecture evaluation in an acquisition context.

1 Introduction

Department of Defense (DoD) acquisition programs routinely acquire systems that are highly *software reliant*.³ Despite the increasing size, functionality, and complexity of these systems, software is often not given the visibility and management attention it deserves. As a result, software problems are often a major contributor to schedule slippages, cost overruns, and system deficiencies. To counter this trend, DoD acquisition organizations must take proactive measures, as early as practical, to reduce software acquisition risk. They cannot afford to just perform perfunctory reviews (such as a Preliminary Design Review [PDR] or Critical Design Review [CDR]) during software development and wait until after system delivery to determine whether key performance parameters⁴ (KPPs) and other acquisition/mission drivers that are important to stakeholders will be achieved. Fortunately, by focusing on the software architecture and quality attribute requirements,⁵ an acquisition organization can conduct such an evaluation early in the development cycle when any needed corrective measures can still be implemented in a cost-effective and timely manner.

This technical note describes a proactive means for incorporating such a software architecture evaluation in a DoD system acquisition in collaboration with the development contractor.

The technical note consists of these sections:

- Section 2 explains the importance of conducting a software architecture evaluation and its benefits.
- Section 3 discusses some key acquisition considerations that pertain to conducting a software architecture evaluation in an acquisition context.
- Section 4 provides an overview of the approach for proactively incorporating a software architecture evaluation in a Request for Proposal (RFP)/contract.
- Section 5 describes sample RFP/contract language that must be included in the main body of the Statement of Work (SOW) and other affected sections of the RFP/contract (e.g., Sections L and M) to accommodate including an architecture evaluation. The SOW⁶ language, in turn, references a comprehensive Software Architecture Evaluation Plan (described in Section 6) that will govern the actual conduct of the architecture evaluation and the follow-on activities.

³ A *software-reliant* system is one whose behavior (e.g., functionality, performance, safety, security, interoperability, and so forth) is dependent on software in some significant way.

⁴ KPPs are intended to capture the minimum operational effectiveness and suitability attributes needed to achieve the overall desired capabilities for a system being acquired by the DoD.

⁵ Quality attribute requirements are synonymous with the system's nonfunctional requirements and are described in detail by Barbacci and colleagues [Barbacci 2003].

⁶ If a Statement of Objectives (SOO) is used, a statement can be included giving notice of the government's intent to conduct a software architecture evaluation, and the sample contract language can subsequently be included in the SOW.

- Section 6 describes the sample Software Architecture Evaluation Plan and corresponding guidance to enable an acquisition organization to customize the plan for use in its own system acquisition.
- Section 7 provides a summary.
- Two appendices include (1) information on other contractual artifacts identified in this technical note that play a role in the software architecture evaluation and (2) a list of acronyms, respectively.

2 The Importance and Benefits of Conducting an Architecture Evaluation

Software plays a critical role in almost every DoD acquisition and is often cited as the reason for cost overruns, schedule slippages, and quality problems. As a result, there is increased emphasis on finding effective ways for an acquisition organization to reduce risk when acquiring software-reliant systems. While there is no “silver bullet,” an architecture-centric acquisition approach has proven to be an effective way of reducing acquisition risk [Nord 2009]. At the heart of such an approach is enabling the acquisition organization to conduct a software architecture evaluation. A prerequisite for conducting such an evaluation is acquiring a software architecture description document that appropriately describes the architecture.

The definition of a software architecture is

The software architecture of a program or computing system is the structure or structures of the system, which comprise software elements, the externally visible properties of those elements, and the relationships among them [Bass 2003].

Since a system’s software architecture conveys the software design decisions that are the most critical and the hardest to change downstream, the importance of evaluating the software architecture cannot be overstated. It is a proven means for identifying risks [Bass 2006]. The early identification of architectural risks plays a crucial role in uncovering potential system problems and avoiding costly rework downstream in the latter stages of software development or, worse, after the system has been delivered and deployed. Software problems resulting from poor architectural design decisions are very difficult and costly to fix if they are discovered late in the integration and test phase or after the system has been deployed.

The right software architecture paves the way for successful system development. The wrong architecture will result in a system that fails to meet critical requirements, suffers from budget and schedule overruns, and incurs high maintenance costs. As a result, evaluating the software architecture of a system as early as possible in the development cycle has proven to be an effective means of reducing software acquisition risk.

This report is based on the architecture evaluation method called the SEI Architecture Tradeoff and Analysis Method[®] (ATAM[®]) [Kazman 2000]. Two SEI case studies describe the results of using the ATAM to conduct architecture evaluations on a sophisticated DoD warfare information communications network and a wargame simulation system [Clements 2005, Jones 2001].

[®] Architecture Tradeoff Analysis Method and ATAM are registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

3 Acquisition Considerations

Despite the compelling reasons to perform a software architecture evaluation, it is not yet common practice among contractors and the DoD. However, architecture evaluation is now gaining momentum in the DoD—especially in the U.S. Army—as an effective means for reducing software acquisition risk [Blanchette 2008].

A major impediment to conducting an architecture evaluation in the DoD acquisition environment is that there is no incentive for an offeror to propose conducting an architecture evaluation on its own. If an architecture evaluation is not a requirement that applies to all offerors, an offeror could penalize itself from a cost and schedule standpoint by proposing to conduct one in its technical proposal. If an acquisition organization wants to promote such evaluations, it needs to create “a level playing field” by being proactive and including the requirement for an architecture evaluation up front in its RFP/contract.

Experience has shown that attempting to conduct an architecture evaluation reactively (i.e., opportunistically after a contract has already been awarded) is often viewed as being intrusive and disruptive from a cost, schedule, and resource perspective. And, a suitable software architecture description document with multiple views (which is a prerequisite for conducting an evaluation) is often overlooked and not a planned developmental artifact. While a major barrier has been the lack of software architecture documentation, this barrier is easily overcome by requiring that a documented software architecture be a contractual deliverable. This requirement can be very beneficial to programs, because having a documented software architecture is key to making design tradeoffs and wise decisions when it comes to understanding, changing, or upgrading a system’s software and hardware.

Incorporating an architecture evaluation in a system acquisition is also dependent on having an evaluation method that is compatible with the needs of a DoD acquisition organization and having an effective means for incorporating it in an RFP/contract. We explored these strategies for incorporating an architecture evaluation in an acquisition:

- Let each offeror propose its own evaluation method in its technical proposal.
- Let the acquisition organization and the winning contractor collaboratively choose an evaluation method after the contract is awarded.
- Let the government specify a common evaluation method up front in the RFP/contract—and use it across programs.

The primary problem with the first strategy is that the offerors are likely to propose their own unique or proprietary evaluation method that is used within their organization. That would require the acquisition organization to be prepared to analyze the pros and cons of each proposed method (a time-consuming task), decide how to determine the acceptability of methods, and deal with results that could differ widely from one offeror to another. In addition, the acquisition organization would need to develop or retain personnel with experience using the method. Since the second strategy is dependent on who wins the contract, the schedule, cost, and resource requirements would not be known until after the contract is signed, which is problematic from a contracting

standpoint. Moreover, there are no guarantees that the selected methods are effective, and suitable documentation and training may not be available.

The benefit of the third strategy—the government being proactive and specifying the evaluation method—is that the “who, why, when, where, and how” of an architecture evaluation can be completely described up front in the RFP/contract, which ensures a level playing field for all offerors. We have used the ATAM as the prescribed evaluation method with the end result that

- All parties will have a common understanding of what is required, and the roles and responsibilities of the acquisition organization and system developer will be delineated in the contract and will not need to be negotiated downstream.
- The cost and schedule for conducting the architecture evaluation can be determined and included in the offerors’ technical and cost proposals and will be evaluated during source selection when the government still has maximum leverage.
- A software architecture description document, which is a prerequisite for conducting an architecture evaluation, will be a required contractual deliverable and will not have to be developed “after the fact” at an additional (and potentially prohibitive) cost.
- The system developer will be responsible for integrating the architecture evaluation with its Project Management Plan (PMP), Integrated Master Schedule (IMS), Software Development Plan (SDP), and Risk Management Plan (RMP) from the outset.
- The contract will specify how the results of the architecture evaluation are to be used and that the system developer will be responsible for creating a risk mitigation plan (following the architecture evaluation) that must be submitted to the acquisition organization for review and approval in accordance with the program’s standard design review process.
- Use of a common evaluation method (i.e., the ATAM) will make training transferable across programs and produce predictable results that will enable lessons learned to be shared among programs and used as the basis for implementing incremental improvements that can benefit all participating programs and acquisition organizations.

The real significance of the factors listed above is that they all play a contributing role in ensuring that an acquisition organization will realize the “maximum payoff” when conducting an architecture evaluation:

- The acquisition organization will be able to identify architectural risks early in the development cycle when they can be mitigated more easily.
- The affected supplier will have all the requisite information needed to appropriately integrate the evaluation into its technical and cost proposals.
- These considerations will tend to motivate the supplier to judiciously develop the architecture from the outset.

The strategy that was chosen is ideally suited to a proactive approach. In contrast, in a reactive approach, when an architecture evaluation is conducted opportunistically, all the factors governing the evaluation have to be painstakingly negotiated. As a result, the outcome is less predictable and less likely to meet stakeholders’ expectations.

4 An Overview of the Approach for Incorporating an Architecture Evaluation in an RFP/Contract

The proactive approach (shown in Figure 1) for incorporating an architecture evaluation (which is described in this technical note) revolves around a sample Software Architecture Evaluation Plan that a program office or acquisition organization can customize for its own use. This sample plan, which is fully described in Section 6, is a practical and proven plan for conducting a software architecture evaluation in collaboration with the system development contractor during the contract performance phase of a DoD system acquisition. It has been successfully used to incorporate a software architecture evaluation into U.S. Army, U.S. Navy, and U.S. Air Force acquisitions. The plan has intentionally been written so a DoD program office or acquisition organization can easily customize it and include it in an RFP/contract. The plan covers all the details—the “who, why, when, where, and how”—of how the architecture evaluation is to be conducted to ensure that all offerors have a common understanding of what is required and what the government’s expectations are.

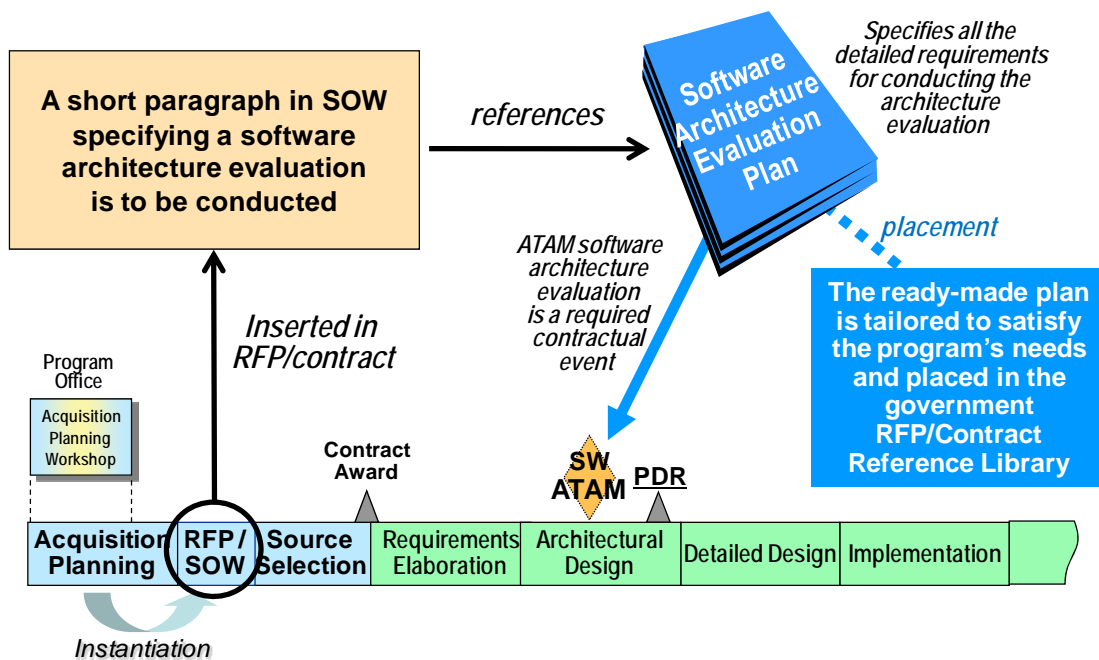


Figure 1: Proactive Approach for Incorporating a Software Architecture Evaluation in an RFP/Contract

Figure 1 shows the main elements of the approach in relation to a traditional DoD contractual timeline that serves as an acquisition point of reference. The timeline depicts typical acquisition events, such as source selection, contract award, and a PDR, superimposed on a set of representative development activities ranging from requirements elaboration through implementation. These events and activities set the contractual context for scheduling and integrating an architecture evaluation in a DoD acquisition. The development activities that are shown are just representative and are not intended to imply, or necessitate, a waterfall approach. The evaluation approach being described is compatible with any software development methodology, because it is event driven.

Ideally, the architecture evaluation should take place prior to the PDR, which is a traditional DoD acquisition event prescribed by DoD 5000 acquisition policy [DoD 2008]. This recommended timing enables the architecture evaluation results to be available as input to the PDR, thus making the PDR discussions less perfunctory and more meaningful: the system development contractor can present its analysis of architectural risks (discovered during the architecture evaluation) and discuss its plans for mitigating them. The evaluation is conducted by a team that has been commissioned by the acquisition organization and trained⁷ to conduct the evaluation. An acquisition organization should consider enabling a representative of the system development contractor to also serve on the evaluation team, if the representative meets the requirements to become an evaluator. The architecture evaluation plan described in this report provides such an option.

A Software Architecture Integrated Product Team (SA-IPT),⁸ which is appointed by the program office, is responsible for overseeing and managing the results of the software architecture evaluation and determining what follow-on action is required by the acquisition organization and the system development contractor.

Incorporating an architecture evaluation into an RFP/contract using this approach involves four basic actions, all of which are described in detail in Section 6:

1. Customize the Software Architecture Evaluation Plan in accordance with the instructions and guidance prescribed in Section 7, so it is compatible with the acquisition organization's terminology, acquisition practices, and contractually planned events.
2. Include the plan in the acquisition organization's government reference library (or its equivalent) that is the designated repository for all the documents referenced in the RFP/contract.
3. Include a short paragraph in the SOW (as prescribed in Section 6.1) to specify that an architecture evaluation is to be conducted in accordance with the plan included in the government's reference library.
4. Include the appropriate language (recommended in Section 6) in the following RFP sections to ensure that the system development contractor includes the software architecture evaluation as an integral part of its software development approach:
 - Section C (Instructions to Offerors)
 - Section M (Technical Evaluation Criteria)
 - Section J (Contract Deliverables)

The importance of placing the plan in the acquisition organization's government reference library is that offerors will then be able to access it and have time to analyze it. As a result, they can integrate it appropriately into their technical and cost proposals.

⁷ In evaluations using the ATAM, the team would consist of an SEI-certified ATAM Leader and two or three SEI-authorized ATAM Evaluators. The SEI's publicly available ATAM certificate and certification programs, which are described on the SEI's website (<http://www.sei.cmu.edu/>), qualify individuals to participate in or lead SEI-authorized ATAM evaluations. These programs allow representatives of the acquisition organization and system development contractor to qualify as ATAM Evaluators and even ATAM Leaders in some cases.

⁸ Alternatively, this could be the IPT responsible for software development or an ad hoc Software Architecture Working Group (SAWG). Assigning responsibility to a different team or group only requires changing the name of the SA-IPT, accordingly, in the Software Architecture Evaluation Plan described in Section 6.

5 Sample RFP/Contract Language

The sample language that needs to be included in the RFP/contract to fully integrate the prescribed architecture evaluation (and follow-on activities) in a system or software acquisition is described in the following sections, which correspond to the major sections of an RFP/contract. The sample RFP/contract language that is provided has been used in actual DoD acquisitions but may need to be customized to comply with local contracting requirements and policies as well as program-specific requirements.⁹ The purpose of this contract language is to

- specify the contractual requirements needed to ensure that the software architecture evaluation is applied properly in the DoD/government acquisition environment
- provide a common and equitable basis to enable all potential offerors to appropriately respond and cost out their involvement in the software architecture evaluation

The same language can be used for a competitive or sole-source acquisition, as long as the architecture evaluation is conducted after contract award rather than during source selection. If a program office wants to explore conducting an architecture evaluation during source selection or make significant changes to the plan, an acquisition planning workshop¹⁰ should be held first to ensure that the ramifications of doing so are fully understood.

5.1 Section C: Statement of Work (SOW)

The following language (which appears in a **shaded box**) is the primary text that an acquisition organization needs to include in the SOW.

Software Architecture Evaluation

As a software acquisition risk reduction measure, the contractor shall participate in and actively support a collaborative evaluation of the <System_Name> software architecture that is to be led by an evaluation team commissioned by the <Program_Name> acquisition office. The architecture evaluation shall be conducted prior to the Preliminary Design Review¹¹ (PDR) in accordance with the <System_Name> *Software Architecture Evaluation Plan* (<document_identifier>).

⁹ RFP/contract language and acquisition artifacts are provided as examples only. The SEI does not warrant the effectiveness of any such language or the architecture evaluation plan for use in DoD or government acquisitions. It is the responsibility of the program manager and contracting officer having cognizance over the acquisition to determine the appropriateness and/or suitability to a specific acquisition program. Moreover, this report does not address or touch on the legal terms and conditions that are relevant to a system or software acquisition. Such legal terms and conditions will have to be appropriately included in the RFP/contract by the acquisition organization in concert with its contracting officer and legal staff.

¹⁰ An acquisition planning workshop is a one-to-two day engagement that is facilitated by the SEI to assist a DoD organization with its acquisition challenges and explore ways to adopt an architecture-centric approach in order to reduce risk.

¹¹ Or a comparable event-driven activity that occurs *before* the software architectural design is approved and detailed design takes place.

An acquisition organization only needs to fill in the <text.delimited.by.arrows> appropriately. The specific factors governing how, where, why, and when the evaluation is to be conducted are fully described in the government-provided Software Architecture Evaluation Plan that will reside in the acquisition organization's RFP reference library/repository. The plan also specifies the activities the acquisition organization, evaluation team, and development contractor are responsible for after the evaluation.

The only other contract language (shaded box) that needs to be included in the SOW is the following statement:

The contractor shall produce, update, and maintain a <System_Name> *Software Architecture Description* (SWARD) document using the contractor's configuration management control system and deliver the SWARD document in accordance with <SWARD_CDRL_Identifier>.

The language above (or its equivalent) is required, because a documented software architecture is a prerequisite for conducting an architecture evaluation and needs to be a contractual deliverable. The Contract Data Requirements List (CDRL) for the SWARD document should also specify that a preliminary version of the software architecture description document is to be delivered, so a determination can be made as to whether the architecture is being suitably¹² documented. Otherwise, if changes were needed and they were not discovered until Phase 0 of the ATAM, the schedule could be adversely impacted. A sample CDRL for the software architecture description document is provided in Appendix A.

5.2 Section L: Instructions to Offerors

The following language (shaded box) should be included in Section L of the RFP, which provides instructions to an offeror with regard to what is to be included in its technical proposal.

The offeror is to provide a summary description of its involvement in the software architecture evaluation and describe how the evaluation will be integrated into the offeror's management and development practices. Particular emphasis should be given to how the architecture evaluation results (i.e., discovered risks and risk themes) will be integrated with the offeror's risk management (and mitigation) process. The offeror shall also include, as appropriate, any activities or considerations related to the software architecture evaluation in its Project Management Plan (PMP), Integrated Master Schedule (IMS), Risk Management Plan (RMP), and Software Development Plan (SDP) or their equivalent.

5.3 Section M: Technical Evaluation Criteria

Since technical proposals are evaluated based on the technical evaluation factors (and subfactors) specified in Section M, there should be some linkage between what the SOW and Section L re-

¹² Section 6, which describes the software architecture evaluation method, includes an initial evaluation activity (i.e., Phase 0) to ensure that the software architecture has been suitably documented prior to conducting Phase 1 of the ATAM.

quire and the factors in Section M in order to evaluate what an offeror proposes in its technical proposal with respect to the software architecture evaluation.

The existing set of evaluation factors and subfactors (and corresponding evaluation criteria) that the acquisition organization intends to use has to first be disclosed and understood before a determination can be made as to whether (1) another evaluation subfactor needs to be added, (2) an existing one needs to be modified, or (3) the current set is sufficient.

An example of an appropriate evaluation subfactor that would cover an architecture evaluation is “Risk Management.” And an example of the corresponding evaluation criteria would be “adequacy of response,” which is defined as the extent to which the proposed approach is *complete* and demonstrates an *understanding* of the requirements. In turn, *completeness* is defined as the extent to which the proposal describes approaches that address all requirements and associated risks; describes means for resolution of the risks; and includes sufficient, substantive information to convey to the evaluator a clear and accurate description of the approaches and how the requirements are to be satisfied. *Understanding* of requirements is defined as the extent to which the approach demonstrates an accurate comprehension of the specified requirements, the intended mission environment, and program goals.

The objective in Section M is to insure that the set of evaluation subfactors (and corresponding criteria) is sufficient to evaluate whether the offeror understands the software architecture evaluation approach and has appropriately integrated it with the offeror’s management and development practices. In other words, to evaluate whether an offeror understands what the architecture evaluation entails and appropriately integrates it with its system and software development practices. In particular, the source selection team should evaluate how each offeror plans to manage and mitigate any discovered risks, handle risk themes, and manage the impact on KPPs and business (i.e., acquisition) and mission drivers.

A Technical Interchange Meeting (TIM) is usually held with key acquisition office stakeholders and decision makers, so appropriate language can be drafted for Section M commensurate with the program’s specific needs and elaboration of its high-level evaluation factors and subfactors.

5.4 Section J: Contract Data Requirements List (CDRL)

Since a software architecture description document is a prerequisite for conducting an architecture evaluation, it must be a contractual deliverable included in the CDRL. The CDRL also needs to include an appropriate Data Item Description (DID) to specify the requirements for the software architecture description document that the system developer will be responsible for developing and delivering to the government. Appendix A includes a sample CDRL and DID for a SWARD document.

Some traditional acquisition documents are potentially affected when a program office decides to conduct an architecture evaluation. For example, if the acquisition office is going to require a System Engineering Management Plan (SEMP), Risk Management Plan (RMP), Software Development Plan (SDP), or a Software Test Plan (STP), or their equivalent, as deliverables, the acquisition office should consider including appropriate language in the CDRL/DID to specify what additional information (related to the software architecture evaluation) should be included in those deliverables. On the other hand, if the acquisition organization is going to develop some of the

governing documents itself, such as a System Engineering Plan (SEP) or a Test and Evaluation Master Plan (TEMP), the program office should consider adding text that appropriately describes the intended role of the software architecture evaluation. Such additions to these documents are not major considerations but are recommended, so a coherent and consistent approach to architecture evaluation is articulated and reinforced by the program office.

Table 1 identifies the type of information or direction (related to conducting an architecture evaluation) that should be included in these acquisition documents.

Table 1: Type of Information to Be Included in Related Acquisition Documents

Document	Type of Information to Be Included (Relative to Conducting an Architecture Evaluation)
SEMP	Describe: (1) how the architecture evaluation is integrated into the System Engineering Management Plan in relation to the program milestones, (2) how the system's quality attribute requirements (i.e., nonfunctional requirements) that drive the architectural design will be specified and managed, and (3) how the software architecture will be documented.
TEMP	Describe the role of architecture evaluation in the Test and Evaluation Master Plan and when the evaluation will be scheduled in relation to the program's planned milestones.
SEP	Describe: (1) how the architecture evaluation is integrated into the System Engineering Plan in relation to the system engineering milestones, (2) how the system's quality attribute requirements (i.e., nonfunctional requirements) that drive the architectural design will be specified and managed, and (3) how the software architecture will be documented.
SDP	Describe how the software architecture evaluation fits into the overall software development approach including how identified risks (and risk themes) will be analyzed and mitigated.
STP	Describe the role of architecture evaluation in the Software Test Plan and when the evaluation will be scheduled in relation to software testing milestones.
RMP	Describe how risks (and risk themes) emanating from the architecture evaluation will be integrated with the program's risk management system and subsequently managed (i.e., identified, tracked, and mitigated).

All other aspects of the architecture evaluation are encapsulated in the Software Architecture Evaluation Plan that is referenced in the SOW and described in the next section.

6 A Sample Software Architecture Evaluation Plan

The Software Architecture Evaluation Plan is intended to be used as a government-provided document that is part of, and accessible through, the acquisition organization's government reference library. This sample Software Architecture Evaluation Plan has intentionally been written so a DoD program office or acquisition organization can easily customize it and include it in an RFP/contract. It is comprehensive in that it covers all the factors that govern the "who, why, when, where, and how" of conducting an architecture evaluation. These factors include such items as prerequisites, schedule considerations, the evaluation method and evaluation report, the roles and responsibilities of the participants, and the important follow-on activities that the acquisition organization and development contractor will be responsible for once the evaluation is completed. The Software Architecture Evaluation Plan is comprehensive and contains all the information the development contractor needs to execute the plan—following contract award—in conjunction with the acquisition organization's responsibilities, which are also specified in the Software Architecture Evaluation Plan.

The complete Software Architecture Evaluation Plan is described below (shaded box). The plan is intended to be used as written, with customization and tailoring limited to the designated elements, except for those areas where the local contracting officer requires other changes to be made to become an approved document. An acquisition organization can easily customize the plan for its use by filling in the appropriate "names" in the *<name_fields>* that are italicized. Other *<text_fields>* that are bold and italicized should be reviewed and tailored carefully by the acquisition organization to ensure its suitability and compatibility with the specifics of the planned system acquisition. Other text that is shaded and delimited [in bold brackets] is optional and should be retained or deleted, as the acquisition organization deems appropriate.

6.1 A Two-Part Plan

The Software Architecture Evaluation Plan consists of two parts. The first part of the plan describes the purpose and key factors governing the execution of the architecture evaluation such as oversight considerations, the scheduling and location, the evaluation method, participants, and the architecture documentation. The second part, which is actually an addendum to the plan, describes the specific architecture evaluation method (i.e., the ATAM) and how it is to be applied in an acquisition context—information that prospective offerors need to know for planning and estimating purposes. The reason for partitioning the Software Architecture Evaluation Plan this way is that the first part of the plan is intended to be tailored in prescribed places, while the second part (the addendum)—which is specific to the evaluation method—is not to be changed in any way to ensure that the architecture evaluation is conducted in a consistent and coherent manner across DoD programs and that the integrity of the ATAM is preserved. The two parts of the sample architecture evaluation plan are described in the following sections.

6.2 First Part of the Software Architecture Evaluation Plan

Software Architecture Evaluation Plan

1. Purpose

The purpose of this Software Architecture Evaluation Plan (SAEP) is to provide a common understanding of how an evaluation of the <System_Name> software architecture is to be conducted in collaboration with the system development contractor by an evaluation team commissioned by the acquisition organization. Implementation of the SAEP will provide the <Program_Name> Program Office¹³ with an effective means of reducing software acquisition risk, commensurate with its technical oversight and contract-monitoring responsibilities.

2. Oversight

The architecture evaluation is to be conducted under the general oversight of the <System_Name> Software Architecture Integrated Product Team (SA-IPT). The SA-IPT Leader is responsible for overseeing and managing the results of the software architecture evaluation and determining what follow-on action is required in accordance with established procedures governing the <System_Name> contract and this plan.

The SA-IPT will consist of designated government stakeholders representing the <Program_Name> Program Office and may include (at the discretion of the program office) other stakeholders representing the <System_Name> contractor and organizations that will be using or supporting the <System_Name> system.

Once the architecture evaluation report¹⁴ is delivered to the SA-IPT Leader, the SA-IPT will be responsible for reviewing the report and enumerating the specific items (e.g., risks, clarifications, and/or issues) that the contractor is to address. The contractor will be responsible for developing mitigation strategies (in accordance with the contractor's standard format and risk management process) that address the items that were enumerated by the SA-IPT. Upon government approval of the contractor's proposed mitigation strategies, the contractor shall make any necessary revisions to the <System_Name> software architecture and update the corresponding Software Architecture Description (SWARD) document (refer to Paragraph 6 of this plan) accordingly.

The SA-IPT Leader, or other designated <Program_Name> Program Office representative, shall be responsible for performing any official contract administration function (e.g., notifying, coordinating, scheduling, and arranging) related to conducting the architecture evaluation and shall coordinate these functions with the contractor and team leader of the architecture

¹³ Alternatively, this could be the <Proper_Name> Acquisition Office.

¹⁴ The leader of the ATAM evaluation team—not the system development contractor—is responsible for producing and delivering the evaluation report.

evaluation, as needed.

3. Scheduling and Location

The <System_Name> software architecture evaluation shall be conducted before any significant level of detailed design or software coding has taken place but not until the developmental-baseline software architecture has been placed under configuration management control and documented in the Software Architecture Description document. The evaluation shall be completed prior to the <System_Name> Preliminary Design Review (PDR). The evaluation results, and any needed corrective actions, will be discussed during the PDR.

The architecture evaluation shall be conducted at the software development contractor's site and scheduled as a sequence of Technical Interchange Meetings (TIMs). Accordingly, the contractor shall identify, in conjunction with its other system and software development milestones, when and where the architecture evaluation TIMs are to be conducted in its Project Management Plan (PMP) and Integrated Master Schedule (IMS). In scheduling the architecture evaluation TIMs, the <System_Name> contractor shall make allowance to be prepared to discuss technical alternatives during the <System_Name> PDR for mitigating risks uncovered during the evaluation. Any risks will have been identified in the Architecture Evaluation Report (and outbrief presentation of the architecture evaluation results) described in Addendum A to this plan.

4. Evaluation Method

The method that is to be used for conducting the software architecture evaluation is the SEI Architecture Tradeoff Analysis Method (ATAM[®]) developed by the Carnegie Mellon[®] Software Engineering Institute. The procedure and rules for conducting the architecture evaluation in an acquisition context are described in Addendum A to this plan.

5. Participants

Since the architecture evaluation is a collaborative effort, the active participation of representatives from the program office, contractor organization, and stakeholder community is essential to conducting a meaningful architecture evaluation. The participants in the architecture evaluation can be broadly broken down into three categories: (1) project decision makers, (2) architecture stakeholders, and (3) the evaluation team itself.

The project decision makers are to include both government representatives (maximum of 8) and <System_Name> contractor representatives (maximum of 10), exclusive of the evaluation team. The government representatives typically include program office representatives, key technical representatives, contracting office personnel, and other affected government stakeholders who play a key decision-making role. The SA-IPT Leader must be included as one of these government representatives (for the reasons stated in Paragraph 2 above). The contractor

[®] Architecture Tradeoff Analysis Method, ATAM, and Carnegie Mellon are registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

representatives include the chief software architect (always) and other key contractor personnel who play an important decision-making role, such as the project manager, system architect, software developers, domain experts, and so forth. The participation of the chief software architect is **mandatory** throughout the architecture evaluation—no exceptions.

The architecture stakeholders (maximum of 20) constitute a broader and more diverse group of stakeholders who have a vested interest in the software architecture but may not be key decision makers. These stakeholders include program office representatives, contractor personnel (e.g., testers, maintainers, domain experts), future users of the system, and support personnel (e.g., logistics, operations, security experts).

The evaluation team is responsible for facilitating and conducting the software architecture evaluation. The team composition is specified in Addendum A of this plan. [A representative of the system development contractor may optionally participate as a member of (i.e., augment) the evaluation team. The contractor representative, though, must complete the mandatory training described in the addendum to this plan to qualify as a member of the evaluation team. The contractor representative must also be *external to the project* whose software architecture is being evaluated to ensure impartiality and objectivity. If the system development contractor elects to have a representative participate on the evaluation team, that participation must be appropriately incorporated in its technical and cost proposals.]¹⁵

6. Software Architecture Documentation

A major prerequisite for conducting the architecture evaluation is the availability of suitable software architecture documentation. Accordingly, the contractor shall deliver a Software Architecture Description (SWARD) document to the government in accordance with the Contract Data Requirements List (CRDL) no later than **10 days** prior to the scheduled date for conducting the architecture evaluation. Thereafter, the SWARD document is to be updated, using the contractor's configuration management control system, in accordance with the SWARD CRDL.

¹⁵ If the acquisition organization does not want to make this option available to the system development contractor, the shaded text within the bold brackets should be deleted from the plan.

6.3 Second Part of the Software Architecture Evaluation Plan

Addendum A: <System_Name> Software Architecture Evaluation Plan

1. Application

This addendum describes the rules for conducting a software architecture evaluation in an acquisition context using the SEI Architecture Analysis and Tradeoff Method (ATAM).

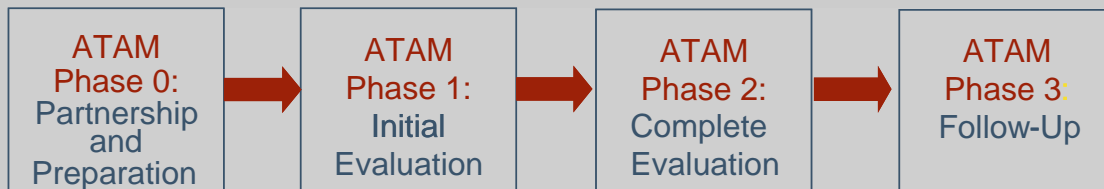
2. Architecture Evaluation Method

The software architecture evaluation shall be conducted by an SEI-certified ATAM Leader in accordance with the SEI Architecture Tradeoff Analysis Method (ATAM). The ATAM is described in *Evaluating Software Architectures: Methods and Case Studies*, Addison-Wesley, 2002; ISBN 0-201-70482-X. The book includes costing information (Chapter 2), describes the method's steps and phases (Chapter 3), and provides a case study in applying the ATAM (Chapter 6).

Additional architecture evaluation case studies are provided in *Software Architecture in Practice, Second Edition*, Addison-Wesley, 2003; ISBN 0-321-15495-9.

2.1 ATAM Phases

An ATAM evaluation encompasses four distinct phases, referred to as Phase 0 through Phase 3:



Phase 0 is a partnership and preparation phase that lays the groundwork for conducting the architecture evaluation. During this phase, the focus is on reviewing roles and responsibilities, evaluating the suitability of the software architecture documentation, and determining whether the evaluation participants and contractor are fully prepared to begin Phase 1 of the ATAM. Phase 1 and Phase 2 are the primary evaluation phases where the analyses take place. Phase 1, which involves the project decision makers, is architecture-centric and concentrates on eliciting architectural information and systematically analyzing the information. Phase 2 is stakeholder-centric and concentrates on eliciting stakeholders' points of view and verifying the results of Phase 1. Phase 3 is the follow-up phase that involves writing the final report.

Phase 0, which is the setup phase for the ATAM evaluation, will be tailored (i.e., reduced in scope) by the ATAM Leader to serve as the equivalent of an architecture evaluation readiness review. The tailoring is limited to eliminating those steps that have already been addressed in the governing contract itself and this plan. No additional tailoring of the method is permitted except that required to conform to the SAEP or approved by the government.

2.2 Scheduling of Phases and Evaluation Results

Phases 0, 1, and 2 of the ATAM are to be scheduled as a sequence of Technical Interchange Meetings (TIMs).

The architecture evaluation is to begin with Phase 0 in accordance with the contractor's Integrated Master Schedule (IMS). If the ATAM Leader determines that everything is in a state of readiness in Phase 0, the Phase 1 initial evaluation is to be conducted next in accordance with the IMS. Within one to three weeks of completing Phase 1, Phase 2 is to be initiated. The outbrief presentation of the ATAM results will be available at the conclusion of Phase 2 and given to the SA-IPT Leader. Within six weeks of completing Phase 2, the ATAM Leader will deliver a draft ATAM Architecture Evaluation Report to the SA-IPT Leader. Following a one-week period for the SA-IPT's review of the draft report, the ATAM Leader will update the draft report and deliver a final Architecture Evaluation Report to the SA-IPT Leader within one additional week, signaling the completion of Phase 3. Follow-on activities involving the analysis and mitigation of risks (and risk themes) can begin once the SA-IPT Leader provides direction to the system development contractor through the Contracting Officer's Technical Representative (COTR) or other appropriate channel. The evaluation results will be presented during the outbrief presentation at the conclusion of Phase 2 and formally communicated in the final report that is to be delivered at the conclusion of Phase 3.

3. ATAM Participants

The participants in the architecture evaluation can be broadly broken down into three categories: (1) project decision makers, (2) architecture stakeholders, and (3) the evaluation team itself. Table 2 identifies when these participants will need to participate in the various phases of the ATAM evaluation and the approximate number of days that their participation will be required.

Table 2: Participation in ATAM Phases

Participants	Typical Involvement in ATAM Phases (Nominal Effort)			
	Phase 0	Phase 1	Phase 2	Phase 3
Project decision makers	SA-IPT Leader and contractor's software manager and chief software architect (1 day)	All (1-1/2 days)	All (2 days)	SA-IPT Leader (1 day)
Architecture stakeholders	N/A	N/A	All (2 days)	N/A
Evaluation team	ATAM Leader (1 day)	All (1-1/2 days)	All (2 days)	All (5 days)

3.1 Evaluation Team

The Evaluation Team is responsible for conducting the software architecture evaluation in accordance with the ATAM. The evaluation team will consist of an SEI-certified ATAM Leader and two or three SEI-authorized ATAM Evaluators. To become an SEI-authorized ATAM Evaluator, individuals must first complete the SEI's ATAM Evaluator Certificate Program described below.

All team members must further commit to participating in every phase of the ATAM and contributing to the writing of the final report during Phase 3. Selective participation is not permitted,

because it is counterproductive.

4. ATAM Training Requirements

All members of the ATAM Evaluation Team need to have successfully completed the SEI's ATAM Evaluator Certificate Program.

4.1 ATAM Evaluator Certificate Program

The ATAM Evaluator Certificate Program is described on the SEI's website (<http://www.sei.cmu.edu/>).

4.2 Availability of ATAM Training

The courses identified above are part of the SEI's standard course offerings given at the SEI's facility in Pittsburgh, Pennsylvania. More information about these courses is posted on the SEI's website.

5. Roles and Responsibilities

The specific roles and responsibilities of the project decision makers, architecture stakeholders, and ATAM Evaluation Team are described below.

5.1 Project Decision Makers

The project decision makers are expected to actively and collaboratively participate in the ATAM evaluation commensurate with the prescribed steps for all four phases of the ATAM. The following clarification is provided to ensure that the specific roles and responsibilities for the indicated steps¹⁶ are well understood by all parties:

- Phase 0
 - The contractor is responsible for describing the candidate system (Step 2) and identifying architectural approaches (Step 4).
 - Any contractor personnel on the ATAM Evaluation Team must participate in the kickoff meeting (Step 6).
- Phase 1
 - The contractor is responsible for presenting the mission/business drivers (Step 2) with the assistance of a program office representative.
 - The contractor's chief software architect is responsible for presenting the software architecture (Step 3) and identifying architectural approaches (Step 4).
 - The contractor's chief software architect has the primary role in analyzing architectural approaches (Step 6).

¹⁶ These steps are described in the book *Evaluating Software Architectures: Methods and Case Studies*, Addison-Wesley, 2002; ISBN 0-201-70482-X.

- Phase 2
 - The ATAM Leader presents a summary of the business drivers with the assistance of the chief software architect (Step 2).
 - The ATAM Leader presents a summary of the architecture with the assistance of the chief software architect (Step 3).
 - The contractor’s chief software architect has the primary role in analyzing architectural approaches (Step 8).
- Phase 3
 - Step 2 (Holding the Post-Mortem Meeting) and Step 3 (Building Portfolio and Updating Artifact Repositories) will be performed at the discretion of the ATAM Leader as time allows.

5.2 Architecture Stakeholders

The software architecture stakeholders are only required to participate in Phase 2 of the ATAM evaluation. The specific roles and responsibilities of these stakeholders include actively and collaboratively participating in (1) reviewing the Phase 1 results (Steps 1 through 6 of Phase 2) and (2) suggesting scenarios and asking questions from their individual perspective (Steps 7 and 8 of Phase 2) commensurate with the prescribed activities of each of these steps.

5.3 Evaluation Team

The ATAM Leader will be responsible for leading and facilitating the evaluation and performing the roles and responsibilities of the team/evaluation leader in accordance with the ATAM and as described herein. The ATAM Leader will also be responsible for overseeing the writing of the Architecture Evaluation Report and delivering it to the SA-IPT Leader. In all the ATAM evaluations, the ATAM Leader *shall* be responsible for

1. facilitating, or appropriately delegating, all phases of the architecture evaluation
2. presenting the ATAM (Step 1 of Phase 0 and Phase 1)
3. presenting the evaluation results (Step 9 of Phase 1 and Phase 2)
4. leading the entire ATAM Evaluation Team in writing the Architecture Evaluation Report (Step 1 of Phase 3)

6. Architecture Evaluation Report

The ATAM Architecture Evaluation Report *shall* include the evaluation results identified in Chapter 3 of *Evaluating Software Architectures: Methods and Case Studies* along with any other findings that may be unique to the evaluation. A template that describes the required content for the ATAM Architecture Evaluation Report is included in Chapter 6 of that book.

7 Summary

Conducting a software architecture evaluation is an effective way for an acquisition organization to reduce software acquisition risk. An architecture evaluation is a foundational activity in adopting an architecture-centric acquisition approach and a proven means for identifying risks—early in the system development cycle—that can potentially affect the ability of a system to satisfy its KPPs and other acquisition and business drivers that are important to stakeholders.

This report describes a practical means for proactively conducting a software architecture evaluation during the contract performance phase of a system or software acquisition. The evaluation is performed by a trained team that is commissioned by the acquisition organization and done in collaboration with the system development contractor and other stakeholders who have a vested interest in how the system will perform. To facilitate incorporating a software architecture evaluation in a software or system acquisition, this technical note includes sample RFP/contract language and a sample architecture evaluation plan that can be suitably customized.

The sample plan is designed to cover the factors that will govern the “who, why, when, where, and how” of conducting an architecture evaluation. It has been used in multiple DoD acquisitions. Only minor customization is needed to adapt it to a specific acquisition organization’s policies, practices, and terminology and to appropriately synchronize the evaluation with other acquisition events that are part of the RFP/contract. Contract language and guidance are also provided for those sections of the RFP/contract that are impacted or affected by incorporating an architecture evaluation.

NOTE: If you have found this document to be useful or you have used it in an acquisition and are willing to share your experience, we would like to hear from you. Please send all feedback to info@sei.cmu.edu.

References/Bibliography

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

[Bachmann 2001]

Bachmann, Felix; Bass, Len; Clements, Paul; Garlan, David; Ivers, James; Little, Reed; Nord, Robert; & Stafford, Judy. *Documenting Software Architectures: Organization of Documentation Package* (CMU/SEI-2001-TN-010, ADA396052). Software Engineering Institute, Carnegie Mellon University, 2001. <http://www.sei.cmu.edu/publications/documents/01.reports/01tn010.html>.

[Bachmann 2002]

Bachmann, Felix; Bass, Len; Clements, Paul; Garlan, David; Ivers, James; Little, Reed; Nord, Robert; & Judith Stafford. *Documenting Software Architecture: Documenting Interfaces* (CMU/SEI-2002-TN-015, ADA403788). Software Engineering Institute, Carnegie Mellon University, 2002. <http://www.sei.cmu.edu/publications/documents/02.reports/02tn015.html>.

[Barbacci 2003]

Barbacci, Mario R.; Ellison, Robert J.; Lattanze, Anthony J.; Stafford, Judith A.; Weinstock, Charles B. & Wood, William G. *Quality Attribute Workshops (QAWs), Third Edition* (CMU/SEI-2003-TR-016, ADA418428). Software Engineering Institute, Carnegie Mellon University, 2003. <http://www.sei.cmu.edu/publications/documents/03.reports/03tr016.html>.

[Bass 2003]

Bass, L.; Clements, P.; & Kazman, R., *Software Architecture in Practice*, Second Edition. Addison Wesley, 2003.

[Bass 2006]

Bass, Len; Nord, Robert L.; Wood, William G.; & Zubrow, David. *Risk Themes Discovered Through Architecture Evaluations* (CMU/SEI-2006-TR-012, ADA45688). Software Engineering Institute, Carnegie Mellon University, 2006. <http://www.sei.cmu.edu/publications/documents/06.reports/06tr012.html>.

[Bergey 2001]

Bergey, J. & Fisher, M. *Use of the ATAM in the Acquisition of Software-Intensive Systems* (CMU/SEI-2001-TN-009, ADA396096). Software Engineering Institute, Carnegie Mellon University, 2001. <http://www.sei.cmu.edu/publications/documents/01.reports/01tn009.html>.

[Bergey 2005]

Bergey, John K. & Clements, Paul C. *Software Architecture in DoD Acquisition: An Approach and Language for a Software Development Plan* (CMU/SEI-2005-TN-019, ADA443494). Software Engineering Institute, Carnegie Mellon University, 2005. <http://www.sei.cmu.edu/publications/documents/05.reports/05tn019.html>

[Blanchette 2008]

Blanchette, Stephen & Bergey, John. "Training Architecture Practices in Army Acquisition: An approach to Training Software Architecture Practices in U.S. Army Acquisition." *Defense Acquisition Review Journal* 15, 3 (December 2008).

[Clements 1996]

Clements, Paul & Northrop, Linda. *Software Architecture: An Executive Overview* (CMU/SEI-96-TR-003, ADA305470). Software Engineering Institute, Carnegie Mellon University, 1996. <http://www.sei.cmu.edu/pub/documents/96.reports/pdf/tr003.96.pdf>.

[Clements 2001]

Clements, Paul; Kazman, Rick; & Klein, Mark. *Evaluating Software Architectures: Methods and Case Studies*. Addison-Wesley, 2002 (ISBN: 020170482X).

[Clements 2002]

Clements, Paul. *Documenting Software Architectures: Views and Beyond*. Addison-Wesley, 2003 (ISBN: 0201703726).

[Clements 2005]

Clements, Paul C.; Bergey, John; & Mason, Dave. *Using the SEI Architecture Tradeoff Analysis Method to Evaluate WIN-T: A Case Study* (CMU/SEI-2005-TN-027, ADA447001). Software Engineering Institute, Carnegie Mellon University, 2005. <http://www.sei.cmu.edu/publications/documents/05.reports/05tn027.html>.

[DoD 2008]

Department of Defense. *Department of Defense INSTRUCTION, 5000.02*. <http://www.dtic.mil/whs/directives/corres/pdf/500002p.pdf> (December 2, 2008).

[Jones 2001]

Jones, Lawrence & Lattanze, Anthony. *Using the Architecture Tradeoff Analysis Method to Evaluate a Wargame Simulation System: A Case Study* (CMU/SEI-2001-TN-022, ADA399795). Software Engineering Institute, Carnegie Mellon University, 2001. <http://www.sei.cmu.edu/publications/documents/01.reports/01tn022.html>.

[Kazman 1995]

Kazman, Rick; Abowd, Gregory; Bass, Len; & Clements, Paul. *Scenario-Based Analysis of Software Architecture*. University of Waterloo (CS-95-45). http://www.sei.cmu.edu/architecture/scenario_paper/.

[Kazman 2000]

Kazman, Rick; Klein, Mark; & Clements, Paul. *ATAM: Method for Architecture Evaluation*, CMU/SEI-2000-TR-004, ADA382629. Software Engineering Institute, Carnegie Mellon University, 2000. <http://www.sei.cmu.edu/publications/documents/00.reports/00tr004.html>.

[Nord 2009]

Nord, Robert; Bergey, John; Blanchette, Stephen; & Klein, Mark. Impact of Army Architecture Evaluations (CMU/SEI-2009-SR-007). Software Engineering Institute, Carnegie Mellon University, 2009.<http://www.sei.cmu.edu/publications/documents/09.reports/09sr007.html>.

Appendix A Contract Data Requirements List (CDRL) and Data Item Description (DID) for a Software Architecture Description (SWARD) Document

This appendix contains a sample CDRL and DID for a SWARD document that is a required contractual deliverable. Additional information and guidance on documenting software architectures are provided by Bachmann and colleagues [Bachmann 2001, Bachmann 2002, and Clements 2002].

The SWARD DID described in this appendix is a DOD-STD-2167A DID for a Software Design Description (SDD) that has been modified to specify the required content of the SWARD document using IEEE/EIA 12207.1-1997 as the template for the document.

Use of this CDRL and DID requires that the fields in **boldface text** be appropriately filled in.

CONTRACT DATA REQUIREMENTS LIST (1 Data Item)						Form Approved OMB No. 0704-0188									
The public reporting burden for this collection of information is estimated to average 110 hours per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0701-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. Please DO NOT RETURN your form to the above address. Send completed form to the Government Issuing Contracting Officer for the Contract/PR No. listed in block E.															
A. CONTRACT LINE ITEM NO. <Line Item Number>			B. EXHIBIT A		C. CATEGORY: TDP <u>IPSC</u> TM _____ OTHER _____										
D. SYSTEM/ITEM <System Name>			E. CONTRACT/PR NO. <Contract Number>		F. CONTRACTOR										
1. DATA ITEM NO. A01	2. TITLE OF DATA ITEM Software Design Description (SDD)				3. SUBTITLE Software Architecture Description (SWARD)										
4. AUTHORITY (Data Acquisition Document No.) DI-IPSC-81435A/T			5. CONTRACT REFERENCE <SOW Paragraph Number>			6. REQUIRING OFFICE <Program Office Name>									
7. DD 250 REQ LT	9. DIST STATEMENT REQUIRED D	10. FREQUENCY ONE/R		12. DATE OF FIRST SUBMISSION See Block 16		14. DISTRIBUTION									
8. APP CODE A	See Block 16	11. AS OF DATE N/A	13. DATE OF SUBSEQUENT SUBMISSION See Block 16			a. ADDRESSEE	b. COPIES								
					Draft		Final								
							Reg	Repro							
16. REMARKS <p>Block 9: Distribution authorized to the Department of Defense and U.S. DoD Contractors only in support of US DoD efforts. Other requests shall be referred to <name of cognizant organization>.</p> <p>Block 4: Utilize MS Office 3000 or later for all documents (drafts and final) with the exception of Unified Modeling Language (UML) artifacts that shall be delivered electronically in both the format native to the CASE tool used for capturing the information and in the XML Metadata Interchange (XMI) format. Additionally, provide final copy as Adobe PDF file, with digital signature for authentication.</p> <p>Contractor format acceptable.</p> <p>The Contractor shall deliver and update this data item in accordance with the Program Management Plan (PMP). Delivery schedule requirements identified herein should be construed as Government recommendations.</p> <p>Use paragraph 6.12 of IEEE/EIA 12207.1-1997 as the template for the document. The contents in this data item should be traceable back to the Software Requirements Specification. Change IEEE/EIA 12207.1-1997 section 6.12.3.c.3 to instead be "Identification of software requirements allocation to the software architecture." IEEE/EIA 12207.1-1997 sections 6.12.3.c and 6.12.3.d are to be elaborated using the following information: Software Architecture Description</p> <p>The primary products from software architectural design shall be a set of views created by the architect that describe the software system. Documenting a software architecture is a matter of documenting the relevant views, and then adding information that applies to more than one view. These views shall be analyzable and provide the information needed to conclusively show that the software architecture can, in fact, achieve the system's business/mission drivers and specified system quality attributes (e.g., performance, security, availability, modifiability, and so forth). Moreover, the software architecture description shall be sufficiently robust to enable an analysis to be performed that shows how the system will respond to any particular scenario that is representative of one or more of the quality attributes that the system is contractually required to satisfy.</p> <p>When documenting the software architecture, at least three types of views shall be provided:</p> <ul style="list-style-type: none"> • Module views – show how the software system is structured as a set of code units or modules; i.e., it documents the principal units of implementation. • Component-and-Connector views – show how the software system is structured as a set of software elements that have runtime behavior and interactions; i.e., it documents the units of execution. 						Upload to Repository	1	1	1						
												For delivery of electronic media:			
						<Organization Code>						1	1	1	
						<Organization Code>						1	1	1	
						<Organization Code>						1	1	1	
						15. TOTAL						4	4	4	
G. PREPARED BY			H. DATE		I. APPROVED BY		J. DATE								

CONTRACT DATA REQUIREMENTS LIST (1 Data Item)						Form Approved OMB No. 0704-0188									
<small>The public reporting burden for this collection of information is estimated to average 110 hours per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0701-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. Please DO NOT RETURN your form to the above address. Send completed form to the Government Issuing Contracting Officer for the Contract/PR No. listed in block E.</small>															
A. CONTRACT LINE ITEM NO. <Line Item Number>			B. EXHIBIT A		C. CATEGORY: TDP <u>IPSC</u> TM _____ OTHER _____										
D. SYSTEM/ITEM <System Name>				E. CONTRACT/PR NO. <Contract Number>		F. CONTRACTOR									
1. DATA ITEM NO. A01	2. TITLE OF DATA ITEM Software Design Description (SDD)					3. SUBTITLE Software Architecture Description (SWARD)									
4. AUTHORITY (Data Acquisition Document No.) DI-IPSC-81435A/T			5. CONTRACT REFERENCE <SOW Paragraph Number>			6. REQUIRING OFFICE <Program Office Name>									
7. DD 250 REQ LT	9. DIST STATEMENT REQUIRED D	10. FREQUENCY ONE/R		12. DATE OF FIRST SUBMISSION See Block 16		14. DISTRIBUTION									
8. APP CODE A	See Block 16	11. AS OF DATE N/A	13. DATE OF SUBSEQUENT SUBMISSION See Block 16		a. ADDRESSEE		b. COPIES								
							Draft	Final							
							Reg	Repro							
16. REMARKS Block 4 (cont'd): Examples of design decisions that need to be described with supporting rationale are: <ul style="list-style-type: none"> • Structure/modularity • Distribution/Communication • Concurrency • Deployment • Security • Fault management • Data management • State/mode management • Safety • User interface • Event handling • Synchronization • Time management • Startup/shutdown Document open architecture standard requirements conflicts in an appendix to the SWARD, including resolution thereof. Include a record of when, where and how Government approval of the resolution was provided, as well as pending actions. Blocks 12 & 13: Document delivery and review cycle in the Integrated Master Schedule (IMS), subject to the constraint that Government approval is an exit criteria of the PDR. Revise as required. Block 14: Delivery is satisfied by upload of the document(s) to the <program name> repository, or separate delivery on electronic media to indicated addresses (only if size of files or security/sensitivity considerations dictate). Notify the Government of data delivery via e-mail (addressees to be provided), with receipt required. If delivery cannot be accomplished by upload to the <program name> repository, then deliver electronic media to the addressees shown in Block 14.						Upload to Repository	1	1	1						
												For delivery of electronic media:			
						<Organization Code>						1	1	1	
						<Organization Code>						1	1	1	
						<Organization Code>						1	1	1	
												15. TOTAL	4	4	4
G. PREPARED BY				H. DATE		I. APPROVED BY		J. DATE							

Appendix B Acronym List

APW	acquisition planning workshop
ASSIP	Army's Strategic Software Improvement Program
ATAM	Architecture Tradeoff Analysis Method
CDR	Critical Design Review
CDRL	Contract Data Requirements List
COTR	Contracting Officer's Technical Representative
DID	Data Item Description
DoD	Department of Defense
IMS	Integrated Master Schedule
IPT	Integrated Product Team
KPP	Key Performance Parameter
PDR	Preliminary Design Review
PMP	Project Management Plan
QAW	Quality Attribute Workshop
RFP	Request for Proposal
RMP	Risk Management Plan
SAEP	Software Architecture Evaluation Plan
SA-IPT	Software Architecture Integrated Product Team
SAWG	Software Architecture Working Group
SDD	Software Design Description
SDP	Software Development Plan
SEI	Software Engineering Institute
SEMP	System Engineering Management Plan

SEP	System Engineering Plan
SOO	Statement of Objectives
SOW	Statement of Work
STP	Software Test Plan
SWARD	software architecture description
TEMP	Test and Evaluation Master Plan
TIM	Technical Interchange Meeting

REPORT DOCUMENTATION PAGE			<i>Form Approved</i> <i>OMB No. 0704-0188</i>	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE July 2009	3. REPORT TYPE AND DATES COVERED Final		
4. TITLE AND SUBTITLE A Proactive Means for Incorporating a Software Architecture Evaluation in a DoD System Acquisition		5. FUNDING NUMBERS FA8721-05-C-0003		
6. AUTHOR(S) John K. Bergey				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA 15213			8. PERFORMING ORGANIZATION REPORT NUMBER CMU/SEI-2009-TN-004	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) HQ ESC/XPk, 5 Eglin Street, Hanscom AFB, MA 01731-2116			10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES				
12A DISTRIBUTION/AVAILABILITY STATEMENT Unclassified/Unlimited, DTIC, NTIS			12B DISTRIBUTION CODE	
13. ABSTRACT (MAXIMUM 200 WORDS) Department of Defense (DoD) acquisition programs routinely acquire systems that are highly software reliant. With the increasing functionality and complexity of these systems, software problems often contribute to schedule slippages, cost overruns, and system deficiencies. As a result, DoD acquisition organizations need to take proactive measures to reduce software acquisition risk. They cannot afford to just perform perfunctory reviews during software development and wait until after system delivery to determine whether key performance parameters (KPPs) and other acquisition/mission drivers that are important to stakeholders will be achieved. Since the architectural design of a system and its software has a major influence on whether a system achieves its KPPs (and other acquisition/mission drivers), conducting an architecture evaluation is an effective means for reducing software acquisition risk. The evaluation involves the active participation of key stakeholders and focuses on identifying risks (and overarching risk themes) that can affect the architecture's ability to accommodate the system's quality attribute requirements (e.g., performance, safety, and security). Satisfying these quality attribute requirements is key to satisfying KPPs and other stakeholder-specific acquisition/mission drivers. This technical note describes a proactive means for incorporating such a software architecture evaluation (in collaboration with the development contractor) early in the contract performance phase of a DoD system acquisition. The proven means that is described revolves around a sample Software Architecture Evaluation Plan that a DoD program office can easily customize and use in its own Request for Proposal (RFP)/contract. The sample plan covers all aspects—that is, the "who, why, when, where, and how"—of the government's approach to conducting a software architecture evaluation during an acquisition. Moreover, this sample plan provides acquisition organizations and potential offerors with the insight needed to understand the impact of, and government's expectations for, proactively conducting a software architecture evaluation in an acquisition context.				
14. SUBJECT TERMS architecture evaluation, software architecture evaluation, architecture-centric acquisition, ATAM evaluation, evaluating quality attributes, architecture evaluation plan, architecture evaluation RFP language			15. NUMBER OF PAGES 47	
16. PRICE CODE				
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL	