Software Engineering Institute

A Workshop on Architecture Competence

Len Bass
Paul Clements
Rick Kazman
John Klein
Mark Klein
Jeannine Siviy

April 2009

TECHNICAL NOTE
CMU/SEI-2009-TN-005

Research, Technology, and System Solutions Program
Unlimited distribution subject to the copyright.

Carnegie Mellon

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.

Table of Contents

List of Figures

List of Tables

Acknowledgments

The authors of this report gratefully acknowledge the participants of the workshop, whose contributions and energy made it possible.

We are also grateful to Carolyn Kernan of the SEI, who was responsible for the formidable task of handling all the workshop’s many logistical matters.

We would like to thank Rolf Siegers, Shankar Kambhampaty, and Linda Northrop for their helpful comments on an earlier draft.

Abstract

This report summarizes a workshop on architecture competence that was held at the Carnegie Mellon® Software Engineering Institute (SEI) in June of 2008. The SEI invited accomplished practitioners from government, academia, and industry to discuss key issues in assessing the competence of organizations that use architecture to produce software-reliant systems. After several opening talks by individuals who recounted their experience in competence improvement efforts, workshop participants divided into working groups. Each group was tasked with working on a specific set of issues and was asked to produce a set of questions that could appear in a competence assessment instrument.

1 A Workshop on Software Architecture Competence

1.1 BACKGROUND

In 2007, the Carnegie Mellon® Software Engineering Institute (SEI) launched a project titled "Assessing and Improving Architecture Competence." The purpose of this project is to build a theory of competence that will let us effectively measure the competence of architects, architect teams, and (especially) architecture-producing organizations and to prescribe effective ways to improve competence. A previous report outlines a set of models that we are using to inform both this theory and our assessment and improvement approaches [Bass 2008].

The initial scope of our effort focuses on software architecture, but we define architecture competence in more general terms, as follows:

> *The architecture competence of an organization is the ability of that organization to grow, use, and sustain the skills and knowledge necessary to effectively carry out architecture-centric practices at the individual, team, and organizational levels to produce architectures with acceptable cost that lead to systems aligned with the organization's business goals.*

A number of organizations are pursuing purposeful efforts to improve their architecture competence. Consulting organizations are gearing up to help them in this quest. In addition, some standards bodies are releasing competence frameworks or competence models. Architecture competence, as an area of study and professional practice, seems to have arrived.

On June 17-18, 2008, we held a small focused workshop on architecture competence at the SEI. We invited about a dozen experts from around the world who do one of the following:

- oversee architecture competence improvement programs in their respective companies
- consult with companies who wish to increase their architecture competence
- have academia-based efforts of their own through which they are trying to build a competence assessment instrument or investigate some aspect of architecture competence

The purpose of the workshop was to help us understand the current trends, methods, purposes, outcomes, and experiences that pertain to helping architects and architecture organizations measure and improve their architecture competence.

This technical note summarizes the results of that workshop.

1.2 ORGANIZATION OF THIS REPORT

This report is laid out as follows:

- Section 1.3 explains the format of the workshop and lists the attendees.
- Section 2 summarizes the invited presentations made by five of our participants.
- Section 3 summarizes the SEI Architecture Competence Framework (henceforth called the SEI Framework), which we presented to the attendees for comment and use in prosecuting

® Carnegie Mellon is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

some hypothetical engagement scenarios. That section also discusses how the Framework leads straightforwardly to assessment instrument questions, and how those questions are informed and amplified by the contributing models we use in our work.

- Section 4 turns from architecture competence assessment to architecture competence improvement, which is the ultimate goal of this work. We discuss various strategies for improvement identified by the participants and discuss how particular failure modes identified from an assessment lead to improvement ideas. Section 4 also discusses the kind of framing or scoping that will be necessary through up-front negotiation with a client before a competence engagement can begin.
- Section 5 contains conclusions and ideas for future work.

1.3 WORKSHOP NARRATIVE

Agenda and opening talks. Table 1 shows the agenda for the workshop. After a short welcome and workshop overview that established the purpose and expected outcomes of the workshop, five invited 30-minute presentations were given by invited participants. Then, SEI staff gave a presentation on the SEI's architecture competence project.

Table 1: Workshop Agenda

DAY 1: June 17, 2008	
0800-0830	Continental breakfast
0830-0900	Welcome, introductions, purpose and goals of workshop
0900-1115	Five 30-minute invited presentations: Experiences with architecture competence improvement efforts in practice. (Includes 15-minute break.)
1115-1200	Overview of the SEI's Assessing and Improving Architecture Competence project
1200-1300	Lunch
1300-1315	Working-group formation and mission statement. Each of three working groups will be tasked with fleshing out a scenario dealing with an organization that wishes to improve its architecture competence. Appointed scribes will record the work.
1315-1700	Working groups work
1900	Workshop dinner
DAY 2: June 18, 2008	
0800-0830	Continental breakfast
0830-0900	Plenary: Status check on the working groups
0900-1200	Working groups work: Produce questions.
1200-1300	Lunch
1300-1400	Working groups work: Apply questions with role-playing. Prepare final presentation to group.
1400-1500	Working groups report (20 minutes each)
1500-1530	Conclusions, next steps
1530	Adjourn

Working groups formed. After lunch, the workshop participants divided into working groups. The purpose of each group was to prosecute a competence assessment/improvement scenario and then describe how group participants would go about assessing/improving the organization in their scenario.

As a prelude to working-group formation, the participants brainstormed a number of distinct engagement scenarios that describe an organization that wishes to assess and/or improve its architecture competence:

- An organization has experienced failures due to poor architecture practices and wishes to remedy the situation.
- An organization wishes to know how it compares to similar organizations regarding architecture competence.
- Two organizations wish to compare their architecture competence when they are merging or when one organization is acquiring another.
- A new company that wishes to be demonstrably competent is starting up.
- An organization wishes to advance the state of the art and demonstrate thought leadership in architecture.
- An organization is about to be audited for competence.
- An organization is about to audit another organization.

After a brief discussion that combined some of the original scenarios based on similarity of purpose, two working groups were formed. Working Group A adopted the following scenarios:

- An organization has experienced failures due to poor architecture practices and wishes to remedy the situation.
- Two organizations wish to compare their architecture competence when they are merging or when one organization is acquiring another.

Working Group B adopted the following scenarios:

- An organization wishes to know how it compares to comparable organizations.
- A new company is starting up that wishes to be demonstrably competent.
- An organization is about to be audited for competence.

Each working group was asked to accomplish the following:

1. Produce a recommendation of one or more useful outcomes of a competence assessment (e.g., a number from 1 to 5 similar to that used in an SEI Capability Maturity Model® [CMM®], a vector of strengths and weaknesses, an organizational certification, etc.)
2. Think about the SEI Framework in relation to a hypothetical organization. Are the practice areas the right ones? Are any missing? What else would you do to assess the competence of an organization?

® Capability Maturity Model and CMM are registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

3. Produce a list of sample questions that you would most want to ask of the organization. For each, who would you ask? What supporting evidence for the answers would you want? How would you measure the quality of the responses?

Scribes were appointed to record the work.

1.4 HIGHLIGHTS OF THE WORKING GROUPS' RESULTS

Populating the Framework with questions. During the afternoon of the first day, the working groups carried out their tasks. Both groups produced a list of high-priority questions they would ask during an assessment as part of their respective engagement scenarios.

Overnight, the questions were assigned to specific sections in the SEI Framework. The purpose of this assignment was to validate it by looking for

- sections that had few or no questions: Are these sections unimportant? Are they duplicative?
- questions that did not readily find a home in the Framework: Do they represent conceptual holes in the Framework that would lead us to neglect important areas of competence?

On the second day, the working format was changed to reflect the progress made on the first day and the overnight assignment of questions to the Framework. The two groups merged and spent the morning of the second day discussing the Framework and how their work either found a place in it or identified gaps in it. Section 3 summarizes this outcome.

Failure outcomes. Next, the participants identified a number of organizational deficiencies that an assessment might identify. These deficiencies, or "failure outcomes," can be used to judge the efficacy of the Framework by asking which questions in the Framework would help us successfully diagnose each of the failure modes. The failure outcomes brainstormed by the group were

- The organization has outsourced much of the architecture function and is critically dependent on a labor pool outside of the organization's control.
- The organization has outsourced development, so there is no internal source for developing architects.
- The organization depends on a single hero architect.
- The organization is geographically dispersed, work is distributed across teams, and problems are not discovered until integration time and are therefore costly to resolve. The root cause is divergence in the employees' understanding of the architecture.
- The architecture team is detached from the reality of the project/business, and the architecture is likely to be late.
- The business people are detached from the reality of the architecture and technology, and the project is likely to fail.

In addition, the SEI has identified a number of recurring organization-related risk themes discovered during dozens of architecture evaluations [Bass 2006]. These themes (e.g., failure to control the scope or failure to coordinate with important stakeholders) could also suggest failure outcomes that will allow us to calibrate our diagnostic instrument.

Improvement strategies in response to failure outcomes. Participants spent the afternoon of the second day discussing improvement strategies in light of the failure modes listed above. The operative question was, "Given one of these failure modes, what would you tell an organization to do about it?" Improvement strategies are discussed in Section 4.

2 Presentation Summaries

We invited five participants to make 30-minute presentations to set the stage for the discussion that followed. These presentations represented a broad cross section of the spectrum of approaches to architecture competence. Two presentations were from large U.S. defense contractors, and one was from a leading company in the Indian IT sector; all of the presenters have major architecture competence improvement programs in place in their respective companies. One presentation was by one of the best known consultants in the field of architecture, one whose company focuses on architecture competence improvement. One presentation was by a university researcher whose work in organizational coordination and its relation to architecture is the basis for one of the SEI models of architecture competence described by Bass and colleagues [Bass 2008].

2.1 ROLF SIEGERS, RAYTHEON

Rolf Siegers is an Engineering Fellow at Raytheon, a major U.S. defense contractor with U.S. $21.3 billion in sales in 2007 and 72,000 employees worldwide (http://www.raytheon.com/ourcompany). Rolf leads Raytheon's company-wide effort for architecture institutionalization and sustainment. The Institutionalization and Sustainment initiatives span architecture process and its deployment, tools, cost estimation, training, certification, reference architecture evolution, and collaborations with government, industry, and academia. Rolf sits on Raytheon's corporate Architecture Review Board, providing governance of the Raytheon Enterprise Architecture Process (REAP), the Raytheon Certified Architect Program (RCAP), and Raytheon's reference architectures.

Four very specific drivers have motivated Raytheon's drive for architectural excellence. Among these are two trends in the kinds of systems that they build and two policy mandates.

The first trend is the Global Information Grid (GIG). According to the Defense Acquisition Guidebook

> *The GIG vision is to empower users through easy access to information any time and any place, under any conditions, with attendant security. Program managers and Sponsors/Domain Owners should use this vision to help guide their acquisition programs. This vision requires a comprehensive information capability that is global, robust, survivable, maintainable, interoperable, secure, reliable, and user-driven. The goal is to increase the net-centricity of warfighter, business, intelligence, DoD enterprise management, and enterprise information environment management operations by enabling increased reach among the GIG users, increased richness in the information and expertise that can be applied to supporting operational decisions, increased agility in rapidly adapting information and information technology to meet changing operational needs, and increased assurance that the right information and resources to do the task will be there when and where it is required* [DoD 2003].

According to Rolf, Raytheon sees the GIG as leading to "next generation, net-centric systems of systems [that will] require [a] formalized architecting approach."

The next trend is "mission complexity," which means that more and more systems will be fielded with more and more demanding requirements and operating in more and more challenging and unpredictable environments. A strong architecture capability is needed to successfully field these systems.

The two mandates come from the DoD and the U.S. Office of Management and Budget, respectively. The first requires use of the DoD Architecture Framework (DoDAF) for "…all architectures developed by and for the Office of the Secretary of Defense (OSD)." The mandate for DoDAF V1.0 was released in 2004; the corresponding policy directive for DoDAF V1.5 was released in August of 2007.[1] The second mandate requires that every federal agency have a "strategic plan [that] will shape the redesign of work processes and guide the development and maintenance of an Enterprise Architecture and a capital planning and investment control process… The agency's capital planning and investment control process must build from the agency's current Enterprise Architecture (EA) and its transition from current architecture to target architecture…. The Enterprise Architecture must be documented and provided to the OMB as significant changes are incorporated [OMB]." Both make it clear that architecture competence is a definite requirement for companies wishing to do business with the U.S. government.

Raytheon's response to this need was to launch a coordinated and enterprise-wide competence improvement program that consists of several distinct initiatives. The program can be viewed as comprising four areas of focus: people, process, governance, and resources. The individual initiatives in the overall program include

- the Architecture Review Board (ARB)—a company-wide governance body that provides oversight and guidance on architecture policy, process, and certification, and so forth. The ARB also performs independent reviews of architectures.
- the Raytheon Certified Architect Program (RCAP)—a formally defined architect certification program
- the Raytheon Enterprise Architecture Process (REAP)—a standards-based architecture process
- reference architectures—a collection of domain-specific, partially populated templates for architecture efforts
- an architecture repository—a secure, metadata-enabled data store
- architecture tools that employ various vendor alternatives
- a cost estimation tool that helps estimate the effort and cost of developing an architecture
- collaboration opportunities—various means for sharing knowledge across Raytheon's community of architects
- architecture training, with multiple levels of detail based on the students' level of experience
- architecture standards—government and industry resources that can support the items listed above

---

1 See https://dars1.army.mil >quick links>General Public Documents.

2.2 DANA BREDEMEYER, BREDEMEYER CONSULTING

Dana Bredemeyer is the founder and CEO of Bredemeyer Consulting, which provides a range of consulting and training services focused on enterprise, systems, and software architecture. The company provides training and mentoring for software architects, works with architecture teams, and aims to accelerate the creation or migration of an architecture. It also helps management develop architectural strategies for improving architecture competence.

Dana recounted his experience and overall approach in helping an organization assess and improve its architecture competence.

First, he provided a definition of architecture that was a recurring theme throughout his talk: Architecture, he said, is the translation from strategy to technology. Using the translation metaphor, he pointed out that translating from French to English requires a working knowledge of French. Hence, the architect must be fluent in the organization's strategy.

The first step, then, is to investigate whether architecture is a strategic concern in the organization. For example

- Is the strategy firmly footed so that the architect can be successful?
- Is the architect allowed to be a strategic contributor to the formation of the strategy?
- Who is the lowest level manager in the organization whose purview includes architecture? If it's the CEO, do architects and the CEO talk readily with each other?
- Are architects present at all meetings where they could make a significant contribution?
- How important are the economic decisions being made by architects?

Dana also brings to bear his Architect Competency Framework shown in Figure 1. This Framework establishes five areas in which an architect must excel and the duties, skills, and knowledge associated with each. The five areas are technology, consulting, strategy, organizational politics ("like gravity, the weak force that holds everything together"), and leadership.

Dana introduced an "umbrella model" of organizations based on the scope of the decisions made at various levels. As he explains on his website

> *Architects work at different levels of scope, or focus of decision responsibility. As with the management hierarchy, generally the broader the scope, the more senior the architect, for the responsibility and impact is strategic and demands greater acuity in the areas of strategy, leadership and organizational effectiveness…*
>
> *Titles vary by organization, but to create a frame of reference for this discussion, let's just work with the following: A technical lead may lead a small team developing a service or system component. A software architect is generally responsible for the architecture of a product or application, or the software in an embedded product. In embedded systems, a system architect is responsible for the overall system, including software and hardware components. A product line or product family architect is responsible for the architecture decisions that need to be made across the scope of the product family, in order to meet strategic product family goals (reuse, consistency and brand identity, integration, etc.). A solution architect is responsible for the architecture of solutions that comprise various applications, products or systems.*
>
> *The critical insight here, is that there is a significant shift from tech lead to architect, and again from architect to product line/family architect and again to solution/portfolio architect, and chief architect and enterprise architect. As this field matures, we have to become*

more self-conscious about the fact that we have quite distinct pools of competencies that we need to develop, and that from these pools, we need to draw the talent that seeds the next higher layer (in terms of breadth of scope and strategic contribution). It is useful to be explicit about nurturing the architect tree [Bredemeyer 2008].

	What you KNOW	What You DO	What You ARE
Technology	In-depth understanding of the domain and pertinent technologies Understand what technical issues are key to success Development methods and modeling techniques	Modeling Tradeoff analysis Prototype/experiment/simulate Prepare architectural documents and presentations Technology trend analysis/roadmaps Take a system viewpoint	Creative Investigative Practical/pragmatic Insightful Tolerant of ambiguity, willing to backtrack, seek multiple solutions Good at working at an abstract level
Consulting	Elicitation techniques Consulting frameworks	Build "trusted advisor" relationships Understand what the developers want and need from the architecture Help developers see the value of the architecture and understand how to use it successfully Mentor junior architects	Committed to others' success Empathetic, approachable An effective change agent, process savvy A good mentor, teacher
Strategy	Your organization's business strategy and rationale Your competition (products, strategies and processes) Your company's business practices	Influence business strategy Translate business strategy into technical vision and strategy Understand customer and market trends Capture customer, organizational and business requirements on the architecture	Visionary Entrepreneurial
Organizational Politics	Who the key players are in the organization What they want, both business and personal	Communicate, communicate, communicate! Listen, network, influence Sell the vision, keep the vision alive Take and retake the pulse of all critical influencers of the architecture project	Able to see from and sell to multiple viewpoints Confident and articulate Ambitious and driven Patient and not Resilient Sensitive to where the power is and how it flows in your organization
Leadership	Yourself	Set team context (vision) Make decisions (stick) Build teams Motivate	You and others see you as a leader Charismatic and credible You believe it can and should be done, and that you can lead the effort You are committed, dedicated, passionate You see the entire effort in a broader business and personal context

Figure 1: Bredemeyer's Architect Competency Framework (© 2002 Bredemeyer Consulting)

Figure 2 illustrates a simple “umbrella” showing three levels of decision scope—one overall level that is broken down into three sublevels, each of which is further broken down further into three sub sublevels.

Figure 2: Bredemeyer’s "Umbrella Model"

This model, once instantiated for an organization being assessed, can help chart how ingrained architects are in the various decision points. It can also be correlated to the Architect Competency Framework by letting us describe what skills and knowledge an architect needs at each decision level in an organization.

Finally, Dana counseled us not to be too hard on ourselves because we don’t do a good job of quantifying the benefit of architects and architectural practices. He reminded us that much larger decisions are often made on golf courses on the basis of personal relationships, trust, and the advice of experts.

2.3 SHANKAR KAMBHAMPATY, SATYAM COMPUTER SERVICES, LTD.

Shankar Kambhampaty has been involved for 20 years in architecture, design, development, and management for several software projects executed globally with Satyam Computing Services. Satyam, based in Hyderabad, India, was founded in 1987. It is one of India’s largest IT firms, employing over 51,000 employees in 63 countries.

Shankar began his presentation by discussing the role of IT in India and the estimated need for IT professionals and software architects. India currently has 1.6 million IT professionals, and that number is estimated to grow to 2.3 million in 2010. Estimating that 1% of the professionals are software architects yields an immediate need for 16,000 software architects, and the number will grow as the number of IT professionals grows. 1% is a conservative estimate. It means that each software architect will be working on systems that employ 100 IT professionals.

Shankar then turned to Satyam and its growth. Satyam has a revenue of multiple billions of U.S. dollars and is growing at the significant rate of 20-30% per year. The growth figures are hard to measure because of the appreciation of the Indian rupee against the U.S. dollar. In any case, Satyam has career paths for architects, has focused architecture groups, has initiatives for architecture competence, and has measures for the value created by architecture competence.

2.3.1 Career Path and Architecture Groups

Satyam’s organization has three types of personnel that are relevant to the architecture competence discussion. Some people manage relationships with customers (relationship managers), some manage the delivery of products and projects (project/program managers), and others create solutions (solution architects). The career path for solution architecture begins with being a software engineer and then progresses through designer and business analyst. The next step up the path is to be a business consultant or technical architect, followed by practice head and solution head. The final step on the solution architecture career track is to be a consulting head.

Satyam has a number of architecture groups. The most relevant to the subject of this workshop are the database group, the infrastructure group, the enterprise group, and the solution architecture group.

2.3.2 Competence Initiatives

For someone wishing to progress through the solution architect career path, Satyam has structured learning programs. One begins by entering the aspiring architect program, progresses to the enterprise architect program and then pursues various industry certifications. The aspiring architect program consists of the following courses: Software Architect Big Picture Thinking; Architect Modeling; Platform Specific Architecture J2EE and .Net; and J2EE Architecture and Design Patterns. The specific topics are shown in Figure 3. In addition to the courses, the program involves case studies, readings, and a minimum of two Enterprise Architecture Workshops.

The enterprise architect program consists of a course that covers the following topics:

- The Zachman Framework
- The Open Group Architecture Framework (TOGAF) – the Core aspects
- The TOGAF Framework – Life Cycle and Governance aspects
- The Reference Model of Open Distributed Processing (RMODP) Framework – Engineering, Technology and Other Components
- Business Architecture to Application Architecture
- Focus on Business Analysis to Application Abstraction
- Application Architecture to Solution Architecture
- Focus on Nonfunctional Requirements
- SOA : What and Why
- SOA : Web Services and Beyond
- J2EE : Features and Emerging Trends
- .NET : CLR, COM+, WPF, WCF, WF

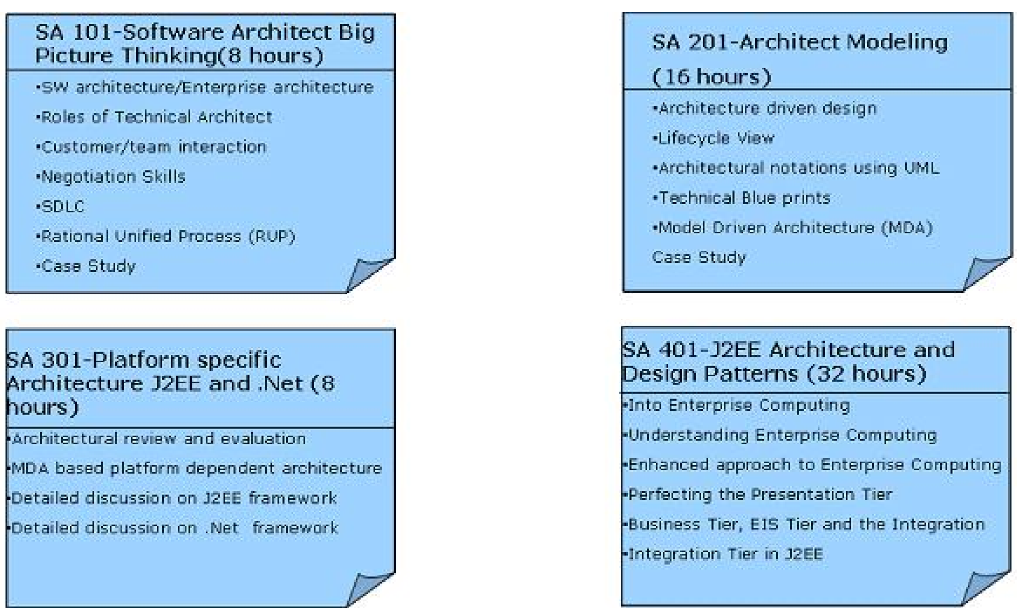

Figure 3: Satyam's Aspiring Architect Program (Satyam Computing Services, 2007)

In addition, Satyam has a standard set of tools for architecture representation that aspiring architects are taught to use. As a final step, architects seek industry certifications as Sun-Certified Enterprise Architects, IBM-Certified Solution Designers, or Microsoft-Certified Architects.

After an individual becomes an architect, additional resources are available for sharing experiences with others. The architects are encouraged to participate in a virtual architect forum and an internal technology review, to read a monthly architecture newsletter, and to write and speak about their work.

2.3.3 Value Measurement

Satyam also has a program to measure the value created by architecture competence. The company tracks these metrics:

- faster, measured by the number of architecture engagements completed successfully
- better, measured by the number of messages of appreciation received
- cheaper, measured by how much the final project comes in under budget
- larger, measured by the size and complexity of the completed engagements
- steadier, measured by the consistency across business units

2.4 DON O'CONNELL, BOEING

Don O'Connell has been a software/systems architect with The Boeing Company for 25 years. For the past nine years, he has worked in Boeing Phantom Works and is leading an effort to increase Boeing's architecture competence through the introduction of key practices such as architecture evaluation and architect certification.

Don's presentation focused on two aspects—the Boeing Software Architect Certificate and the Software Architect Conference.

2.4.1 Software Architect Certificate

At Boeing, software architects work on the software, computing, network, and storage architecture of computer systems, enterprise architectures, and mission systems. Their duties include

- software and system architecture requirements, analysis, and tradeoffs
- software and system architecture development and evaluation
- technical team leadership, project technical leadership, team building, and planning
- computing resources, networks, storage devices, busses, sensors, and communications hardware architecture and high-level system design
- high-level software design

Software architects are typically talented and have several years of experience in

- software design, program integration, and hardware and software integration
- computer, network, and storage design
- software architecting and system architecting
- management and leadership of technical projects
- requirements, prioritizations, and tradeoffs

Software architects

- typically serve in the role of chief architect or on the chief architect's staff for several years
- demonstrate an ability to lead a software team while balancing customer expectations and technical tradeoffs
- demonstrate an ability to mentor and teach others

A single software architect certificate for all the diverse domains within Boeing is difficult to achieve. The current process is intended to be refined to these specific domains of expertise:

- small real-time embedded systems—for example, flight controls inside a smart bomb, avionics flight controls, avionics software to which a pilot interfaces, or unmanned vehicle control systems
- missions systems—for example, airborne and ground-based Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance (C4ISR) missions systems or airplane cabin systems
- satellite systems including flight controls and payload systems
- enterprise systems—for example, airplane parts planning and distributions systems for dozens of airlines

The current Boeing Software Architect Certification process is initiated by self-nomination. The candidate's manager and functional manager must approve of the candidate if he or she meets eligibility qualifications. Once approval has been granted, the following occurs:

- The candidate completes the certificate material and packet and submits it to the nominating manager for approval.
- A board interview is scheduled, and the board reviews the packet. The board consists of selected managers and current certificate holders.
- The candidate is either awarded a certificate or given suggestions for strengthening his or her case.
- Software architects are expected to become recertified every five years.

Parts of this process are designed to evolve—specifically

- versions of certificate material instructions. For example, the DoDAF standard has recently changed, so the certificate material instructions must also change.
- the number of board members
- the granularity of the certificate. A finer grained certificate with domain indication will likely emerge over time.

The certificate program encompasses

1. architecture fundamentals and definitions
    - fundamental books or courses
    - definitions
2. architecture requirements
    - business case
    - quality attributes
    - product lines

- organization, people, and planning skills

3. architecture development
    - creation processes
    - documentation, views
    - frameworks
    - tools, commercial off-the-shelf (COTS) components
4. architecture modeling, evaluation, and measurement
    - modeling and simulation
    - evaluations based on the SEI Architecture Tradeoff Analysis Method® (ATAM®) and other methods
    - architecture analysis and measurement
5. architecture education and experience
    - experience and education
        - classroom and on-the-job training
        - work in architecture
        - must have at least 6,000 hours (three years) of valid software architecture work experience
    - types of domains
6. other architect certificates and classes

The primary sources for the certificate materials are

- Boeing training, courses, and software architecture certificate material
- SEI books and courses
- Object Management Group (OMG) books and courses (on topics such as TOGAF and Model-Driven Architecture [MDA])
- DoDAF material
- macro-scope material
- links to external sites
- other books, links, courses, and certificates

A software architect with a certificate has demonstrated sufficient knowledge and experience to

- lead a software architecture effort
- have decision making and approval authority for architecture products
- lead or participate in architectural reviews and evaluations

2.4.2 Boeing Software Architecture Conference

The purpose of this annual conference is to give software architects an opportunity to network with other software architects and discuss difficult software architecture problems and their solutions. Participants also provide suggestions about where Boeing research time and money should be spent within software architecture.

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

2.5 JIM HERBSLEB, CARNEGIE MELLON UNIVERSITY

James D. Herbsleb is a professor at the Institute for Software Research and director of the Software Industry Center in the School of Computer Science at Carnegie Mellon University. His research focuses on coordinating the intellectual work of individuals and teams. He is particularly interested in how software structure, interdependencies, and design instability give rise to the need for such coordination and how formal and informal organizational structures can support it.

Jim focused his presentation on coordination and collaboration, not just among architects but across project teams. He grounded his comments in Conway's Law, which is about the structure of organizations and the corresponding structure of systems that they design:

> *Any organization that designs a system will inevitably produce a design whose structure is a copy of the organization's communication structure* [Conway 1968].

Typically applied to software-based systems, Conway's Law indicates that the system being produced will reflect the organizational structure that produces it. According to this law, each team of an organization creates one or more singular components of the system (see Figure 4).

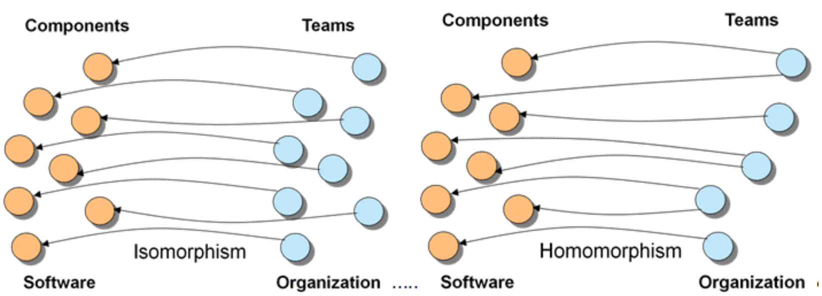

Figure 4: Representations of Conway's Law (Isomorphic and Homomorphic)

Jim then explored this ensuing question, which is depicted in Figure 5: What does the interaction between components imply about the needed interaction/coordination between the teams that produce them? He decomposed this overarching question into several dimensions:

- What are the coordination requirements, in terms of both complexity and uncertainty?
- What are the coordination mechanisms?
- What is the organizational coordination capability?

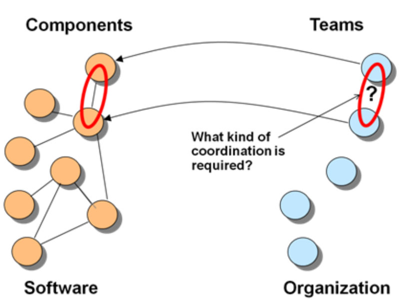

Figure 5: The Implications of Component Interaction on Team Coordination (Conway 1968)

Jim briefly elaborated on each of these dimensions of coordination and explained that the elements of complexity and uncertainty can levy the expectations or rigor for the degree of coordination that is required across teams within the organization. Complexity might involve component size, complex usage policies, or features that span multiple components. Uncertainty might relate to how functionality is allocated to components or how component interfaces are modified and maintained.

Coordination mechanisms span a wide range of communications mechanisms, documentation, and processes that Jim categorized as preparation, shared representation, and communication. Preparation includes planning processes and the plans themselves. Shared representation includes shared visibility into status, availability of test results, and so forth. And, communications include formal meetings as well as informal communications.

The capability of an organization to accomplish coordination relies on both organizational factors and human factors. Examples of organizational factors are geographical distribution as well as the consistencies and differences in processes and management practices across the teams and organizational units. Examples of human factors are experience level and particular skills and expertise—both technical (domain) and nontechnical (language).

Jim proceeded to connect the discussion more specifically to architecture, giving specific attention to architecture and organization congruence or the match between coordination requirements and coordination capabilities. He posited that research is needed on three topics:

- the “coordination view” of an architecture
- measuring the congruence between architecture and the organization creating it
- tactics for improving congruence

Jim concluded that architectural decisions create a “coordination landscape,” architecture and organizational structures are strongly related, and congruence is necessary for project success. He further explained that congruence is critical for developing the tactics necessary for improving congruence and capitalizing on (rather than being held captive by) structural relationships.

3 The SEI Architecture Competence Framework

3.1 INTRODUCTION AND OVERVIEW

This section outlines a baseline framework that will become the basis for assessment and improvement instruments for architecture competence at the individual, team, and organization levels.

This SEI Architecture Framework is composed of a number of competence areas. A competence area *describes* something that an architecturally competent organization must do; if the organization doesn't do it, we consider it less than competent. There may be many ways to carry out each competence area; a sample practice is one way to carry it out, and for each competence area below we list example practices. We would not penalize an organization for not performing one of our sample practices, as long as that organization effectively carried out the containing competence area some other way.

We cluster the practices into three competence area categories:

1. Engineering Competence Area category – These competence areas contain the practices necessary to create or evolve an architecture and to use that architecture to implement a software-reliant system. The Engineering Competence Areas are performed within the context of the Project Planning and Execution Competence Areas.
2. Project Planning and Execution Competence Area category – These competence areas contain the practices necessary to plan and manage the execution of a project that performs the Engineering Competence practices. A typical organization will have several projects running concurrently, with each project independently performing the activities of the Project Planning and Execution Areas and the Engineering Competence Areas. All of these activities are performed within the context of the Organizational Ecosystem Competence Areas.
3. Organizational Ecosystem Competence Area category – These competence areas include the practices necessary to create and sustain the environment in which the architecture is efficiently and effectively created and used. Most organizations have a single ecosystem that provides the context for all projects performed within the organization.

Each of the competence area categories is decomposed into a number of *activities*, and many activities are further decomposed into *competence areas.* This organization is summarized in Table 2. The sections following the table describe each competence area category.

Table 2: Overview of the SEI Architecture Competence Framework

Competence Area Category	Activities	Competence Areas
1. Engineering	Goals-to-requirements synthesis	Business Goal Elicitation
		System Technology Planning
		Architecturally Significant Requirements Creation
	Goals-to-requirements analysis	Technical Feasibility Modeling, Analysis, and Prototyping
	Requirements-to-architecture synthesis	Architecture Reconstruction
		Architecture Design
		Architecture Documentation
	Requirements-to-architecture analysis	Architecture Evaluation
	Architecture-to-system synthesis	Architecture Communication and Guidance
		Development and Test Oversight
	Architecture-to-system analysis	Conformance Assurance
		Architecture Reconstruction
2. Project Planning and Execution	Project management	Project Business/Mission Goal Definition and Communication
		Architecture-Based Estimation and Measurement
		Architecture-Centric Project Management
		Process Discipline
	Use of project-specific architecture tools	
	Training	
	Architect support	Architect/Stakeholder Collaboration
3. Organization Ecosystem	Enablement	"Establish"
		"Sustain"
	Orchestration	Business/Mission Goal Creation and Communication
		Governance
		Architecture-Centric Culture
		Coordination Support

3.2 COMPETENCE AREA CATEGORY #1: ENGINEERING

The Engineering Competence Area category is organized into *activities.* Each activity corresponds to one of the forward (synthesis) or backward (analysis) arrows in Figure 6:

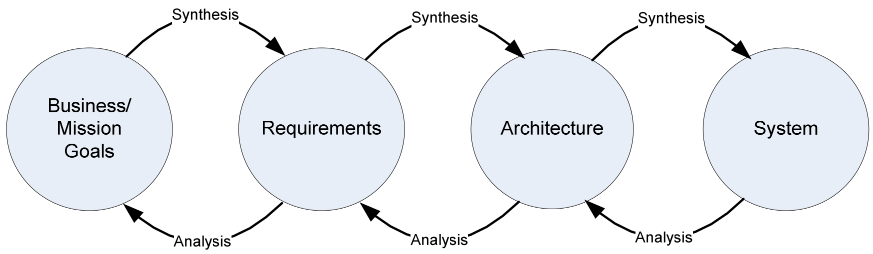

Figure 6: Activities in the Engineering Competence Area

This category focuses only on architecture-centric competence areas. It does not include all of the system engineering and software engineering practices necessary to successfully define, create, and deliver a software system.

3.2.1 Goals-to-Requirements Synthesis Activity

This activity transforms business and mission goals into software architecture requirements.

3.2.1.1 Business Goal Elicitation Competence Area

In this competence area, architects use stakeholder-centric practices to elicit business goals from business stakeholders in order to determine constraints and functional requirements and to determine and prioritize quality attribute requirements for the software architecture.

Sample Practices

- conduct stakeholder interviews
- conduct an SEI Quality Attribute Workshop (QAW)
- follow the standard (e.g., ISO-9126, the Functionality, Usability, Reliability, Performance, and Supportability [FURPS] model, Laprie) or a homegrown quality attribute taxonomy

3.2.1.2 System Technology Planning Competence Area

In this competence area, architects are involved in early technical decisions and tradeoffs.

Sample Practices

- system platform selection
- standards selection
- tools selection

3.2.1.3 Architecturally Significant Requirements Creation Competence Area

In this competence area, architectures engage stakeholders in order to understand, document, and validate quality attribute requirements, constraints, and architecturally significant functional requirements. These requirements should be—insofar as it is possible and practical—complete, concise, correct, and testable.

3.2.2 Goals-to-Requirements Analysis Activity

This activity validates that the software architecture constraints, functional requirements, and quality attribute requirements are aligned with the business and mission goals.

3.2.2.1 Technical Feasibility Modeling, Analysis, and Prototyping Competence Area

In this competence area, architects collaborate with others to assess the suitability and feasibility of the architecture constraints, functional requirements, and quality attribute requirements by developing prototypes, models, or other mechanisms to determine alignment with business and mission goals.

Sample Practices

- prototyping to assess the feasibility of meeting functional or quality attribute requirements
- demonstration development
- involvement with early-stage customer trials

3.2.3 Requirements-to-Architecture Synthesis Activity

This activity creates and documents the architecture.

3.2.3.1 Architecture Reconstruction Competence Area

The architects will use architecture reconstruction to document the existing architecture or to confirm that the implementation conforms to the existing documentation when either of these conditions occurs:

- the architecture is to be based on an existing system and the architecture of that system is not documented adequately
- the implementation is known or suspected to not conform to the documentation

Sample Practices

- reconstruct current baseline

3.2.3.2 Architecture Design Competence Area

In this competence area, the architects use the constraints, functional requirements, and quality attribute requirements to create the software architecture, either by modifying and extending an existing architecture or by creating a new architecture.

Sample Practices

- refactor existing architecture
- using the SEI Attribute-Driven Design (ADD) method
- using feature-driven design

3.2.3.3 Architecture Documentation Competence Area

In this competence area, the architects document the architecture for system stakeholders.

Sample Practices

- using the SEI Views and Beyond (V&B) method
- using the 4+1 method
- using the Siemens Five Views method

- following the IEEE-1471 standard

3.2.4 Requirements-to-Architecture Analysis Activity

This activity assesses how well the architecture satisfies the constraints, functional requirements, and quality attribute requirements.

3.2.4.1 Architecture Evaluation Competence Area

In this competence area, architects engage the system's stakeholder community to assess how well the architecture satisfies the stakeholders' concerns.

Sample Practices

- prototyping or modeling to mitigate risks
- implementing a Software Architecture Review Board (SARB)
- conducting an Architecture Analysis Tradeoff Method (ATAM) evaluation
- using checklists

3.2.5 Architecture-to-System Synthesis Activity

This activity transforms the architecture into a fully developed, running system.

3.2.5.1 Architecture Communication and Guidance Competence Area

In this competence area, architects ensure that the architecture is understood well enough for the system to be implemented, tested, and fielded. These practices augment the Architecture Documentation competence area and can be either reactive or proactive.

Sample Practices

- creating an architecture help desk
- creating an architecture podcast or vodcast
- making the architect a member of the development team

3.2.5.2 Development and Test Oversight Competence Area

In this competence area, the architects proactively work to ensure that the implementation conforms to the architecture.

Sample Practices

- having the architect develop critical elements
- making the architect a member of the development team
- conducting design/code reviews

3.2.6 Architecture-to-System Analysis Activity

This activity assesses whether the as-built system conforms to the as-designed architecture.

3.2.6.1 Conformance Assurance Competence Area

In this competence area, the architects or others use commercial or homegrown tools to do static or runtime checking to determine if the as-built system conforms to the as-designed architecture.

Sample Practices

- profiling to trace execution

- conducting static dependency analysis using Lattix or a similar tool

3.2.6.2 Architecture Reconstruction Competence Area

In this competence area, the architecture of the as-built system is created using reconstruction practices and compared to the as-designed architecture.

3.3 COMPETENCE AREA CATEGORY #2: PROJECT PLANNING AND EXECUTION

The Project Planning and Execution competence areas include those practices that manage and support the Engineering competence areas. These Project Planning and Execution competence areas deal with the people and processes used for a software system development project.

This competence area is organized around two types of activities—Project Management and Architecture Support.

This competence area category focuses only on architecture-centric competence areas and does not consider all of the project management, product/program management, and team management practices necessary to define, create, and deliver a software system.

3.3.1 Project Management Activity

This activity deals with architecture-centric project definition, planning, project management, and project execution.

3.3.1.1 Project Business/Mission Goal Definition and Communication Competence Area

The Engineering competence areas depend on clearly defined project business and mission goals. This competence area ensures that software architects are involved in the creation of these goals and that the goals are then communicated to all stakeholders.

This competence area is related to the Business/Mission Goal Definition and Communication competence area that's under the Organization Ecosystem category. This relationship shows that overall organizational business goals should flow down to specific project goals. This Framework separates the organization and project goals into distinct competence areas to reflect the differences in level of scope and specificity, because each may involve contributions from different stakeholders.

Sample Practices

- making architects members of the product/system planning team
- giving architects access to the organization's market and customer data and business case
- having architects participate in the creation of a project's vision statement

3.3.1.2 Architecture-Based Estimation and Measurement Competence Area

This competence area uses architecture as the basis of project estimation and measurement.

Sample Practices

- application of Paulish's methods for architecture-based estimation [Paulish 2001]
- use of the Constructive Cost Model (Cocomo) [Boehm 2000] for architecture-based estimation
- definition and collection of architecture-based metrics

3.3.1.3 Architecture-Centric Project Management Competence Area

This competence area leverages architecture as a consideration for project management and ensures that during resourcing on an individual project the architect's duties outside of that particular project are considered.

Sample Practices

- having projects follow organizational project management standards (See Section 3.4 on page 24.)
- having projects use an architecture-based work breakdown structure
- considering dependencies and location when making work assignments decisions
- ensuring that the Engineering competence areas are addressed
- providing resources to allow for stakeholder involvement
- including allowances for the architect's off-project workload in project plans

3.3.1.4 Process Discipline Competence Area

This competence area addresses how individual projects adhere to organizational architecture standards. The creation and maintenance of organization architecture standards are addressed in the Organization Ecosystem competence area category (see Section 3.4 on page 24).

Sample Practices

- tailoring projects and following organizational architecture standards
- updating projects and improving architecture standards

3.3.1.5 Project-Specific Architecture Tools Competence Area

This competence area addresses the need for tools at a project level. Tooling standards are considered in the Organization Ecosystem competence area category (see Section 3.4 on page 24).

Sample Practices

- making the appropriate and adequate tools available to architects (and others as needed)

3.3.1.6 Training Competence Area

This competence area addresses the need for tools and methods training at a project level. More general training needs are considered in the Organization Ecosystem competence area category (see Section 3.4 on page 24).

Sample Practices

- ensuring that training on project-specific tools and methods is resourced and scheduled

3.3.2 Architect Support Activity

This activity includes supporting the architect's role in the organization during project execution. This activity also considers how the architecture created by the Engineering competence areas is used by the project stakeholders.

3.3.2.1 Architect/Stakeholder Collaboration Competence Area

The key competence area in this category considers the need for architects and stakeholders to collaborate in order to create and use the architecture. Three particular collaborations should be considered: (1) architects and managers, (2) architects and the development/test teams, and (3) architects and other stakeholders.

Sample Practices

- having the architects advise managers on matters of organization structure, people, teams, and so forth
- having the architects support the managers and be sensitive to the managers' constraints in resource assignment, schedule, and other areas
- giving developers and testers channels for providing input about the architecture
- requiring developers and testers to understand architecture
- giving the architects access to stakeholders
- ensuring that the architects and stakeholders communicate "up, down, across, in, and out" throughout the project life cycle

3.4 COMPETENCE AREA CATEGORY #3: ORGANIZATION ECOSYSTEM

The Organization Ecosystem competence area category includes competence areas that affect all projects within an organization.

This competence area is organized around two activities: (1) those that enable the practice of architecture within the organization and (2) those that orchestrate both the Engineering and Project Planning and Execution competence areas to achieve the most benefit to the organization.

This competence area focuses only on architecture-centric issues and does not consider all the human resources, organizational development, business strategy, and general management practices necessary to operate a successful product development organization.

3.4.1 Enablement Activity

This activity establishes and sustains a pool of people with skills and knowledge to perform architecture activities within the organization.

3.4.1.1 "Establish" Competence Area

This competence area establishes a pool of qualified architects. The number of architects and their skills is based on the current and anticipated workload within the organization. The pool is populated through hiring from outside the organization and/or promoting from within. Pool quality is maintained by removing low-performing members or training underperforming or aspiring members.

Sample Practices

- creating technology roadmaps/projections to identify future needs for skills
- creating business plans to identify future workload needs
- defining job descriptions for architect positions
- creating a selection process to hire qualified architects
- training internal candidates to become qualified architects
- creating performance improvement plans for low-performing architects
- removing or reassigning low-performing architects

3.4.1.2 "Sustain" Competence Area

This competence area sustains the pool described above by ensuring that the architects are retained by the organization and that their skills and knowledge are refreshed and expanded.

Sample Practices

- defining a career track for architects, with job descriptions at multiple job levels
- providing ongoing training in architecture, management, business, software engineering, and/or other areas, not related to project-specific needs
- creating and sustaining an internal community of architects, across all projects/business units in the organization
- providing an internal architecture conference or workshop to promote the internal community and bring in external speakers to expand skills and knowledge
- having architects rotate across projects to prevent burnout and promote skill and knowledge sharing
- creating internal architecture certification programs
- supporting an architect's participation in external communities through attendance at conferences and workshops, preparation of experience or other papers, contribution to online communities, and/or participation in open source projects
- supporting and/or requiring architects to pursue external certifications

3.4.2 Orchestration Activity

This activity includes competence areas that enable architects to connect with and help determine the organizations business/mission goals.

3.4.2.1 Business/Mission Goal Creation and Communication Competence Area

In order for projects to have clear business/mission goals (see the Project Planning and Execution competence area category on page 22), the *organization* must have clear business/mission goals, and architects should contribute to the creation and revision of these goals.

Sample Practices

- having architects contribute to the planning process and advise business leaders on the organization's strengths, weaknesses, opportunities, and synergies and the threats and risks it faces
- creating and maintaining business/technology roadmaps to support planning

3.4.2.2 Governance Competence Area

This competence area constrains individual projects to increase organizational efficiency by allowing the reuse of skills and knowledge across projects and by not requiring every project to make decisions about tools and methods.

Sample Practices

- establishing and maintaining organizational architecture standards
- selecting and refreshing common tools for all projects to choose from
- establishing a chief architect role to oversee governance practices

- establishing an Architecture Steering Council to perform governance practices
- evaluating architectures using cross-project evaluation teams to share experience across projects

3.4.2.3 Architecture-Centric Culture Competence Area

This competence area considers how the organization has embraced architecture as a core competency.

Sample Practices

- considering architecture concerns in all decision making
- including an architecture role on all cross-functional teams
- assessing architecture competence at organization, team, and individual levels
- maintaining a chief architect role with authority over the competence areas outlined in this Framework

3.4.2.4 Coordination Support Competence Area

The performance of any of the competence areas outlined in this Framework requires coordination between people and teams. In many competence areas, the architecture itself will directly influence the coordination requirements.

Sample Practices

- creating and maintaining project management guidelines for work allocation that align work assignments with architecture portioning; for example, “Architecture elements are developed and tested by collocated teams.”
- using shared collaborative workspaces to support collaboration among local and distributed individuals and teams
- promoting architects’ travel to development sites to establish relationships, promote communication, and promote implementation conformance

4 Architecture Improvement Strategies

Improvement is the end game of competence assessment. It is ultimately unhelpful to tell organizations that they are incompetent in certain areas unless it leads to a strategy for improving their situation. The SEI Architecture Competence Framework should be used for baselining (or monitoring or reporting) current capability and for identifying an improvement plan. Other approaches to laying out an improvement strategy might involve Failure Modes and Effects Analysis (FMEA)[2] or scenario planning. These methods are not mutually exclusive.

The workshop participants compiled information related to these topics and the overall subject of improvement via a free-format, open brainstorming and discussion session:

- framing and scoping questions for an improvement effort (including an assessment event)
- organizational scenarios and failure modes that might serve as points of comparison for a specific improvement effort and/or as test cases to validate and verify frameworks, audit methods, and other improvement mechanisms
- improvement mechanisms

4.1 FRAMING AND SCOPING QUESTIONS

Following are the questions that the workshop participants listed for both understanding an organizational context and establishing the boundaries for an improvement endeavor within an organization:

1. What is the scope of the assessment? Is it a project? A whole organization? A business unit?
2. What is the nature of this organization's business, including the type of company it is, type of product it produces, the markets it serves, and so forth?
    a. sector: government/civil agency, DoD, defense industry, commercial (numerous sectors within commercial)
    b. type of product or service: system integrator, shrink-wrapped product, IT service
3. What are the organization's business objectives or mission?
4. What constraints are the organization operating under? These include legal and regulatory, such as those imposed by the U.S. Food and Drug Administration or the U.S. Federal Aviation Administration.
5. What is the expected/required lifetime of the systems being built? How large are they? What are their other characteristics?
6. What critical quality requirements are imposed by the business that are achievable now or expected to be achievable in the architecture?
7. What is the desired outcome of this assessment? What does the organization wish to accomplish by doing this assessment?

[2] FMEA is a risk identification and mitigation technique in which failure modes are classified by severity and impact. Risk priority numbers are calculated, and mitigation plans (which can imply improvement plans, depending on the situation) are developed for the highest priority risks.

8. What is the expected output format of the assessment? (e.g., executive summary, drill-down/detailed report, performance indicators)
9. What are the mechanics of the assessment?
   a. What is the duration of the assessment?
   b. How many views of the organization and information do we need?
   c. How many interviews do we wish to conduct? How many can we practically conduct, given other constraints?
   d. What are the predefined questions? In what areas might we dynamically generate questions and discussion scenarios (for the purpose of exploring unexpected or unique situations)?
   e. What is the desired weighting of the assessment categories (if any)?
10. What does the organization chart look like? Who can/should be involved in interviews or discussions during this assessment?
11. What is your development approach? (Consider dimensions that affect architecture.) How much outsourcing is done?

4.2 ORGANIZATIONAL SCENARIOS AND FAILURE MODES

Following are several organizational scenarios that might serve as diagnostic outcomes. Each scenario is considered a "deficient situation" and might be considered a failure mode itself. Or it would likely lead to one or more failure modes. These scenarios represent situations that the workshop participants would like to see detected with an assessment instrument or improvement effort.

1. The organization has a hero architect.
   a. This organization is critically dependent on this single individual, with no backup and no succession plan.
   b. There is no organizational structure or sustainment.
2. The organization is geographically dispersed, with work allocated to different teams.
   a. In this organization, project efforts typically fall apart, and there is a lot of rework.
   b. Yet, nobody detects the problems and divergences until it is too late.
   c. Typically, many problems are discovered during integration—with severity levels that often cause the project to fail. The root cause of the problems is divergent architecture.
3. The organization outsources its architecture function and is critically dependent on resources outside of its control.
4. The organization outsources development and therefore has no internal source for developing architects.
5. The architecture team is detached from the reality of the project/business, and the architecture is likely to be late.
6. The business people are detached from the reality of the architecture and technology and make unreasonable requests. The project is likely to fail as a result.

7. There is no architectural documentation. Work is aligned to projects, and there is no system-level documentation.

Specific failure modes were further discussed and characterized. Following is the list of failure modes that was compiled, as a continuation of the short list of organizational scenarios above. Participants agreed that ATAM risk themes, particularly as described in the SEI report *Risk Themes Discovered Through Architecture Evaluations* could be used to further enhance this list [Bass 2006].

Failure mode: hero architect

In this failure mode, creating visibility is the initial step needed. The organization may not culturally see a problem. Follow-on questions that should be asked include

- Why is there a dependence on a hero?
- Does the organization value architecture?
- Does the organization value this particular person?
- Is that person the root cause of the situation?

Failure mode: outsourced architecture

The situation of outsourced architecture frequently applies to government organizations. It was noted that if an organization outsources architecture design and definition, it doesn't have architectural competency. It was further noted that if outsourcing is done without understanding, there is no way to know if it is done correctly—unless the review function is also outsourced.

Workshop participants had a lengthy discussion about the motivations, decision-making processes and decision makers for outsourcing. It is possible to be successful with outsourcing, if appropriate mechanisms are enacted. For instance, one participant shared experiences about setting requirements and expectations, conducting joint organizational workshops, and so forth.

Failure mode: missing (or outdated) documentation

Numerous aspects of this failure mode were discussed, each of which would help frame investigative questions during an assessment or improvement effort:

- There may be project documentation but not system-level documentation.
- If there is no documentation, it is not clear who owns the architecture.
- If there are documents but they are out of date, it is because someone changed the system without the involvement of the architect (i.e., if the architect had been involved, he/she would have changed the documents).
- If the documents are perceived as missing or out of date, perhaps the stakeholders' needs were not completely understood in the first place or they have changed along the way (while the produced system itself remained the same).
- Alternatively, the original documentation may have been written at the wrong level of granularity. For instance, if data is included that is prone to change as the system is developed or when it is operational, then of course, it will change. While this type of information needs to be documented somewhere, including it in the architecture documentation becomes a maintenance issue.

- On a parallel note, if many people are changing the document—perhaps at a component level—then that's not architecture. The ensuing discussion on this point focused on when the architecture task is distributed and to whom.

One preventive/corrective action for this failure mode is to make sure that documenting the architecture is an estimated and budgeted task.

Failure mode: detached architecture team

The possible implications of this failure mode are that the team is not aligned with the business goals and deadlines and thus allow the architecture to be late. Workshop participants were interested in using the Framework to detect the likelihood of this situation.

Various root situations behind this failure mode were discussed. For instance, IT architects might respond to a business application request in a way that is congruent with their IT organizations' strategy and plans. While the team itself is not detached per se, they are serving multiple masters and may be "late" in the eyes of some. Another variant of this situation is cultural, where business teams make decisions that constrain the architecture (in an undesirable way) without the architect's being present.

One suggested mitigation for this was Coplien's process pattern called Architect Also Implements [Coplien 1995]. This paradigm requires the architects to tend to the consequences of their decisions and address such things as validating their assumptions, ensuring that their architectures are well-matched to the skills of those who implement them, and so forth. Another suggested mitigation was to examine whether architects are organized around function or projects (or a business unit or product line) and to switch between organizational arrangements.

For cultural situations, training and relationship building need to be part of the mitigation strategy. The objective is for business teams and others involved in the early stages of business analysis, concept selection, and so forth, to understand more about the parts of architecture they need to be concerned with and to know when to include architects in the discussion. Also, the overall team (business and technical, including architects) needs to have a shared understanding of strategy and tradeoff constraints. That understanding leads to the shared realization that some architecture decisions are not made by the architects and others may be made by architects but not without constraint by factors beyond their control. In the end, this training and relationship building lead to a coherent strategy and a responsive, aligned, coherent architecture.

4.3 IMPROVEMENT MECHANISMS

Workshop participants briefly shared their experiences with a few specific mechanisms for improving architecture competence:

- One approach that is useful and potentially transformational is a leadership development program where junior architects allocate a specific amount of time each month to meet with each other and senior leadership figures. They cover all topics: technical, leadership, communication, organizational politics, and so forth. The program that has been observed to work particularly well is done on a one-year basis (with defined beginning and defined conclusion), with groups of up to a dozen participants, and with full, committed participation of each group member.

- Online message boards, internal to an organization, on the subject of architecture have been observed to foster desirable levels of information exchange and networking.
- Structured analysis sessions have been observed to foster effective and focused improvement activities. The experience that was shared involved a routine weekly meeting, during which an architecture is evaluated, a case study examined, and so forth. The results of each discussion were archived appropriately for general access. And, where appropriate, improvement activities were launched.

5 Conclusions and Future Work

The goal of the workshop was twofold. The first goal was to understand what leading organizations are doing in the area of architecture competence. The second goal was to get feedback on the emerging SEI assessment instrument and suggestions for its improvement.

The presentations summarized above and the discussions gave a picture of the current state of the practice. The Boeing, Raytheon, and Satyam presentations took a perspective that was primarily focused on the individual architect. Organizations were important insofar as they recognized the importance of architecture and provided resources and encouragement for individuals to improve their skills. By contrast, the SEI Framework focuses on the organization and what the organization should do if it is serious about incorporating architecture practices. For example, one element in the SEI Framework is the incorporation of architects' input in future product definition and an understanding of the implications of these products on organizational frameworks and reusable assets. One realization of this incorporation is the existence of an architecture roadmap to complement the product roadmap. The architecture roadmap indicates the changes necessary to the architecture to enable the production of the products in the roadmap on schedule. None of the presentations reflected this level of architectural awareness in the presenter's organization.

The feedback on the SEI Framework was helpful and positive. The various elements in the Framework should be augmented with questions drawn from the Organizational Learning, the Organizational Coordination, and the Duties/Skills/Knowledge models, which underpin the Framework. Ideally, we should be able to state what value higher competence brings to an organization, but this is likely to be problematic in practice. Only one organization—Satyam—reported examining the value added from individual architecture competence, but no supporting data was presented.

From the SEI's perspective, the workshop resulted in a great deal of positive feedback about our Framework. It also pointed out to the Framework authors that the instrument still needs reworking and refinement before it can be used as a solid basis for assessment.

References/Bibliography

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

[Bass 2008]
Bass, Len; Clements, Paul; Kazman, Rick; & Klein, Mark. *Models for Evaluating and Improving Architecture Competence* (CMU/SEI-2008-TR-006, ADA482284). Software Engineering Institute, Carnegie Mellon University, 2008.
http://www.sei.cmu.edu/publications/documents/08.reports/08tr006.html

[Bass 2006]
Bass, Len; Nord, Robert; Wood, William; & Zubrow, David. *Risk Themes Discovered Through Architecture Evaluations* (CMU/SEI-2006-TR-012, ADA456884). Software Engineering Institute, Carnegie Mellon University, 2006.
http://www.sei.cmu.edu/publications/documents/06.reports/06tr012.html

[Boehm 2000]
Boehm, Barry W.; Abts, Chris; Brown, A. Winsor; Chulani, Sunita; Clark, Bradford K.; Horowitz, Ellis; Madachy, Ray; Reifer, Donald J.; & Steece, Bert. *Software Cost Estimation with Cocomo II*. Prentice Hall, 2000.

[Bredemeyer 2008]
Bredemeyer Consulting. *Care and Feeding of the Architect Tree*.
http://www.bredemeyer.com/Architect/ArchitectTree.htm (2008).

[Conway 1968]
Conway, M. E. "How Do Committees Invent?" *Datamation 14*, 4 (April 1968): 28–31.

[Coplien 1995]
Coplien, James O. *Architecture Also Implements.*
http://users.rcn.com/jcoplien/Patterns/Process/section16.html (1995).

[DoD 2003]
Department of Defense, Office of the Under Secretary of Defense Acquisition, Technology and Logistics. *Defense Acquisition Guidebook*.
https://akss.dau.mil/DAG/Guidebook/IG_c7.2.asp (2004).

[Paulish 2001]
Paulish, Daniel J. *Architecture-Centric Software Project Management: A Practical Guide*. Addison-Wesley, 2001.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE April 2009	3. REPORT TYPE AND DATES COVERED Final
4. TITLE AND SUBTITLE A Workshop on Architecture Competence		5. FUNDING NUMBERS FA8721-05-C-0003
6. AUTHOR(S) Len Bass, Paul Clements, Rick Kazman, John Klein, Mark Klein, Jeannine Siviy		
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-005
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)

This report summarizes a workshop on architecture competence that was held at the Carnegie Mellon® Software Engineering Institute (SEI) in June of 2008. The SEI invited accomplished practitioners from government, academia, and industry to discuss key issues in assessing the competence of organizations that use architecture to produce software-reliant systems. After several opening talks by individuals who recounted their experience in competence improvement efforts, workshop participants divided into working groups. Each group was tasked with working on a specific set of issues and was asked to produce a set of questions that could appear in a competence assessment instrument.

14. SUBJECT TERMS software architecture, architecture competence, software management, competence framework	15. NUMBER OF PAGES 42
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