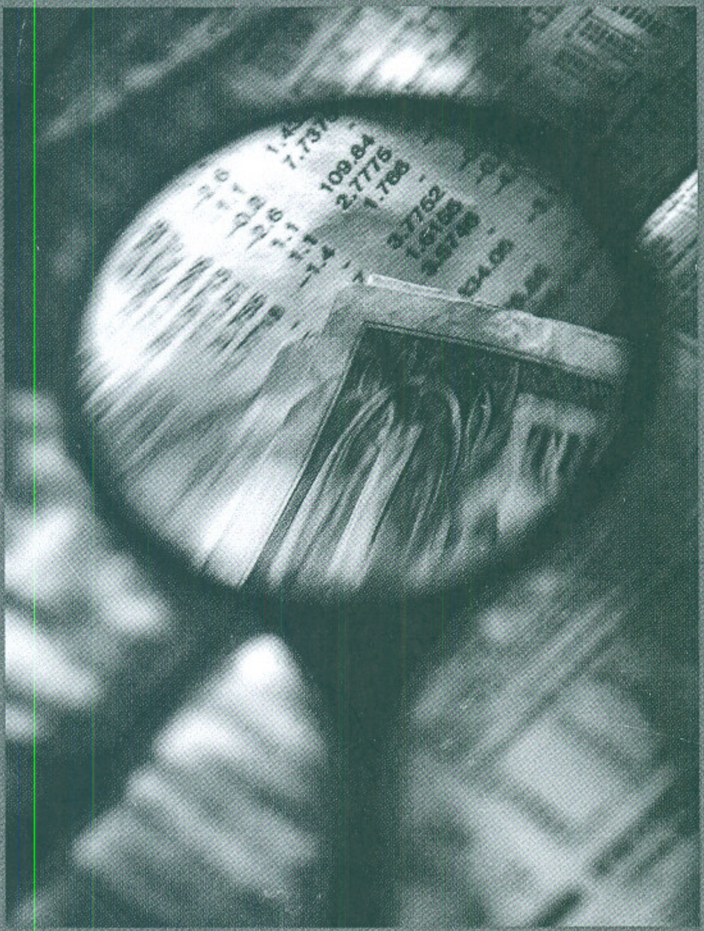
SEI SERIES IN SOFTWARE ENGINEERING

Evaluating Software Architectures



Methods and Case Studies

Chapter 6

Paul Clements
Rick Kazman
Mark Klein

6

A Case Study in Applying the ATAM

> In theory, there is no difference between theory and practice; In practice, there is.
>
> —attributed to Chuck Reid

We hope by now you are comfortable with the ATAM method and its underlying concepts and have digested the first simple case study, the BCS. In this chapter we recount a more complete case study and take considerable pains to describe not only the *what* and *why* of each step but also our personal experiences with the steps. This case study took place in the middle of 2000, and the client was the Goddard Space Flight Center, part of the National Aeronautics and Space Administration (NASA), located in Greenbelt, Maryland.

We chose to describe this evaluation for several reasons. First, the members of the architecture and development team at NASA were interested in not only evaluating the system at hand but also learning enough about the ATAM to be able to apply it themselves on systems in the future. As a result, everyone was painstakingly thorough in applying each step, and everyone was very conscious of the ATAM process itself. This provided a great deal of detail for us to relate. Second, while every engagement using the ATAM is unique in some way, this one on the whole was as typical as any in terms of the architectural insights it uncovered. And third, all three authors of this book participated.

After reading this chapter, you should have a feeling for what an evaluation using the ATAM is like in practice. At each step along the way, we share with you what happened during this evaluation plus any other advice or experience we have about that step in general. We take you through each step of each phase of the ATAM. Our goal for this chapter is to allow you to vicariously experience an actual evaluation.

We recount each step as follows:

- **Step summary:** a short prose recap of the step, followed by a checklist of the input artifacts necessary for the step, the activities carried out in the step, and the outputs produced by the step
- **Step description:** a description of the step's activities that reinforces the description of the steps given in Chapter 3 but with a focus on finer-grained and more practical details
- **How it went:** what happened at the ATAM evaluation that is the subject of this case study
- **Speaking from experience:** typical experiences with this step, things that might go wrong in practice, or simple words of advice that we can share

6.1 Background

The Earth Observing System is a constellation of NASA satellites with names like Terra, Aqua, and Landsat as well as other kinds of land- and air-based sensors whose collective mission is to gather data about planet Earth to fuel the U.S. Global Change Research Program and other scientific communities worldwide.

Satellites are tireless workers, sending their data 24 hours a day, 7 days a week, 365 days a year, year after year. Did you ever wonder what happens to all that data?

In the case of the Earth Observing System, the system that handles the data is called ECS. ECS stands for EOSDIS Core System, where EOSDIS in turn stands for Earth Observing System Data Information System. ECS takes the data from various downlink stations around the world. The data must first be stored, but that's only part of the story. The data must be processed into higher-form information and made available in searchable form to scientists around the world. And it never stops. Handling this data is the mission of ECS.

The numbers are eye-popping. Hundreds of gigabytes of raw data per day flow into the system—data about vegetation, lightning, surface images, atmospheric soundings, trace gas quantities, atmospheric dynamics and chemistry, ice topography, sea surface winds, ocean color, volcanic activity, evaporation amounts, and altimetry of all sorts. The data is processed and analyzed and refined. Hundreds of gigabytes come in, but the processing produces thousands of gigabytes that get warehoused and made available to the scientific community—every day. The overall mission is to understand earth processes and potential global climate changes and to provide information for planning and

The requirements for the ECS system are, roughly speaking, as follows:

- It must ingest, process, archive, and distribute data from 24 sensors on ten spacecraft, numbers destined to grow steadily over time.
- It must compute about 250 standard data products, which are predefined analyses that transform raw data into information packets about the physical world.
- It must archive the data and these data products in eight data centers (warehouses) across the United States.
- It must enable heterogeneous access and analysis operations at these data centers plus forty-some distributed scientific computing facilities.
- It must provide capacity for testing and incorporation of new processing algorithms and new data product definitions, and it must support data/information searches.
- It must do all of this while meeting the performance and reliability requirements to allow it to keep up with the never-ending flood of incoming data and never-ending stream of scientific analysis to be performed on it.

Before EOSDIS, satellite data was stored and formatted in ways specific to each satellite; accessing this data (let alone using it for analysis) was almost impossible for scientists not directly affiliated with that satellite's science project. An important feature of EOSDIS was to provide a common way to store (and hence, process) data and a public mechanism to introduce new data formats and processing algorithms, thus making the information widely available to the scientific community at large.

The first operational version of ECS was delivered in early 1999. Raytheon is the prime contractor. The system has been fielded incrementally, its growing capabilities coordinated with the launch of new spacecraft that will "feed" it.

In spring 2000, NASA wanted to know if the software architecture for this system would support the kinds of performance, scalability, modifiability, and operability that would be required to take the system into its mature years. The agency decided to use the ATAM to perform an evaluation of the software architecture for ECS.

6.2 Phase 0: Partnership and Preparation

Phase 0 establishes a formal agreement between the organization performing the ATAM and the client. How contact is made is beyond the scope of this summary, but many avenues are possible. If you are part of an evaluation unit within a larger organization, or if you are a consultant performing evaluations at large, somehow you and your clientele have to make contact with each other.

Phase 0 consists of the following steps:

1. Present the ATAM
2. Describe candidate system
3. Make a go/no-go decision
4. Negotiate the statement of work
5. Form the core evaluation team
6. Hold evaluation team kick-off meeting
7. Prepare for Phase 1
8. Review the architecture

In spring 1999, the project manager for ECS at NASA contacted our organization and inquired about our architecture evaluation capabilities. We pick up the story from there.

6.2.1 Phase 0, Step 1: Present the ATAM

Step Summary

The evaluation leader describes the evaluation method to the assembled participants, tries to set their expectations, and answers questions they may have.

Inputs

[ ] Viewgraphs for a presentation about the ATAM. Figure 6.6 on page 151 gives an outline for such a presentation.

[ ] A written description of the ATAM, to leave behind with the client for further reading.

[ ] Published ATAM papers. (Optional—some clients may find the papers too technical or overwhelming.)

[ ] ATAM effort data from previous exercises.

[ ] Anecdotal benefit data. (Optional—not all anecdotes of savings will apply to every client. Use judgment about sharing past experience if the information might put off the client because it doesn't apply to his or her situation.)

Activities

[ ] Evaluation organization representative makes sure the client understands the mechanics of the evaluation method to his or her satisfaction through one or more of the following activities:

[ ] Making a presentation about the method.

[ ] Giving the client a product/service description.

[ ] Giving the client a copy of this book and pointing out those sections listed in the Reader's Guide as pertaining to evaluation clients.

[ ] Referring the client to any of the published papers on the ATAM listed in the References section of this book. Mail them if requested.

[ ] Evaluation organization representative makes sure the client understands the costs and benefits of an architecture evaluation by providing the client with cost/benefit data.

[ ] Evaluation organization representative records questions asked about the method or the process for possible inclusion in a Frequently Asked Questions list. Duration and depth vary depending on the interest and background of the audience. This activity might take place any number of times with diverse audiences.

Output(s)

[ ] Names and addresses of contacts interested in pursuing the ATAM.

Step Description

This is usually the first time that a (prospective) client learns about what the ATAM is and what it can do for his or her organization. The object of the step is to explain the method clearly enough so that the client can decide whether or not to engage. The briefing that the evaluation team presents is the same one that is used in Phase 1 and Phase 2. However, in Phase 0, the client is also concerned about what an ATAM evaluation will cost in terms of time and other resources, so the standard ATAM briefing is augmented by cost data based on your own experience in conducting past evaluations using the ATAM. If you have no such data at this point, you can use the cost data in Chapter 2 as a starting point.

How It Went

Managers at NASA learned of the ATAM through public sources: our organization's Web site and published papers. They did enough homework to be convinced that architecture evaluation in general, and this technique in particular, would bring value to their organization. Two people from our organization went to visit NASA to make the presentation to the ECS project manager and two people on his technical staff. The part of the ATAM that produced a prioritized explicit articulation of the quality attributes resonated strongly with the project manager; he had been having thoughts about how to accomplish exactly that. He also saw the ATAM as a way to achieve consensus among a very disparate group of widely dispersed stakeholders: the consumers of the processed satellite data around the world, all of whom wanted the next increment of ECS to address their special needs.

Speaking from Experience

The presentation is informal, and typically only a few people are present: at a minimum, someone who can speak for the evaluation organization and someone who can speak for the client organization. Both need to be able to commit to the evaluation from their respective sides and to work out dates and logistics.

This presentation often results in interesting questions about the ATAM and its results because it is the point at which the client organization gains its first understanding of the method. We try to record questions asked during this presentation for inclusion in a Frequently Asked Questions list that we then turn around to give to our next potential client as part of ATAM read-ahead material.

At times the client organization sees this meeting as an opportunity for a "free lecture" on architecture and architecture analysis. We have had as many as 50 people attend the ATAM presentation. This is not a problem, as long as some "intimate" time with the clients is provided as well.

6.2.2 Phase 0, Step 2: Describe Candidate System

Step Summary

A spokesperson for the architecture to be evaluated describes the system to convey its essential functionality, the quality attributes, the status of the architecture, and the context.

Inputs

[ ] Candidate system viewgraphs (desirable but optional).

[ ] Other candidate system documents, especially architecture documents (desirable but optional).

[ ] Client organization nondisclosure forms, to be left for signature by evaluation team members when assigned. (Optional—use if required by client.)

Activities

[ ] Client representatives describe the candidate system with sufficient detail to convey the main architectural drivers (for example, business goals, requirements, constraints).

[ ] Client and evaluation organization representatives agree on necessary architecture documentation.

[ ] Work out issues of proprietary information. If necessary, have the client obtain nondisclosure forms for the evaluation team to sign.

[ ] Evaluation organization representative records business goals, architectural constraints, and a list of architecture documentation to be delivered to the evaluation team.

Outputs

[ ] List of business goals and architectural constraints.

[ ] List of architecture documentation that will be delivered to the evaluation team.

[ ] Client nondisclosure forms, to be signed by the evaluation team (optional—if required by client organization).

Step Description

A primary purpose of this step is to feed the go/no-go decision in the next step: Is there an architecture here that is far enough along to be evaluated? The client makes an overview presentation of some sort or perhaps just talks about the system informally.

The representative of the evaluation organization should listen for a few key things and ask about them if they are not forthcoming in the presentation.

- Listen for ways to define the scope of the system being evaluated. Does the evaluation include the software to generate or build the system? Does it include any software with which the primary system interacts?
- Listen for stakeholders or stakeholder roles to be mentioned.
- Listen for quality attributes of importance to be mentioned. This will give you hints about how to staff the evaluation team.
- Listen for any mention of commercial off-the-shelf software packages that have been chosen or mandated. These often have architectural and quality attribute impacts.
- Listen for the status of the architecture. Ideally, before this meeting, ask the client to bring a sample of the architecture documentation.

At the end of the client's presentation, summarize what you heard and see if he or she agrees. This information will be useful later.

How It Went

ECS turned out to be a case in which the architecture to be evaluated was already supporting a running, fielded system. We asked why the project manager wanted to evaluate an architecture already built. He told us he was concerned about the system's ability to handle much larger quantities of data, which it would have to do as more and more satellites joined the EOS fleet. If there was a flaw, he wanted to find it now rather than later. He also wanted exposure to the ATAM so that he could try to grow an architecture evaluation capability in his own organization. And last but hardly least, he viewed the consensus-building qualities of the ATAM, especially with regard to prioritizing stakeholder input, as an ideal means to achieve buy-in for the ECS project plan from a wide variety of ECS users—consumers of the data in the scientific community.

Speaking from Experience

Clients almost always have some sort of project briefing already packaged that they can present; the briefing was probably prepared for the project's sponsor somewhere along the way.

Unless your client is from an organization brimming over with architectural maturity, keep your expectations for the architecture documentation low. You want to establish what documentation exists and then decide if it is sufficient for proceeding with the evaluation.

Sometimes there is clearly an architecture, but documentation for it is poor or nonexistent. The ATAM can still proceed if the architect can speak authoritatively about the design decisions. An evaluation under these conditions is certainly likely—obligated, even—to report the lack of documentation as a serious risk, but as long as the client understands that the basis for the evaluation will be what the architect *says*, the exercise can continue. Better still, you can provide clients with documentation requirements and have them produce architectural representations that are suitable for analysis. We did this during the BCS case study, as we reported in Chapter 4: we provided a set of documentation requirements to the client, who responded with much improved architectural information and representations.

Frequently you will be asked to evaluate the architecture for a system that's already up and running. While this may seem akin to checking the barn door after the horse is on the track, it often makes sense. The system may be an early version with limited functionality, or the client may be concerned with its long-term viability or ability to grow, scale up, or accommodate change. In many ways using the ATAM to evaluate an already-running system is easier: the architect is unlikely to answer an analysis question by saying, "We haven't thought about that yet."

6.2.3 Phase 0, Step 3: Make a Go/No-Go Decision

Step Summary

Decide whether the state of the project warrants an ATAM evaluation

Inputs

[ ] ATAM go/no-go decision criteria specific to your organization.

[ ] Candidate system documents from Step 2.

Activities

[ ] Apply go/no-go decision criteria.

Output

[ ] If "no go," letter sent to client explaining the reasons for declining the work and suggesting remediation steps to enable future work.

Step Description

Apply your organization's criteria for continuing with the engagement or breaking it off. If there is not yet an architecture nor enough architectural decisions to be subjected to the scrutiny of an evaluation, then performing an evaluation will not be a good use of resources. If your decision is "no go," then explain to the client what is lacking and (depending on your relationship) help him or her address the project's shortcomings so that an evaluation will be fruitful.

How It Went

Not surprisingly, our decision was "go." The members of this client organization demonstrated architectural maturity through their clear understanding of the ATAM and why it would help them. Also, since an early version of the system was up and running, there was little doubt that they had an architecture to be evaluated.

Speaking from Experience

For some years now we've tried to codify our go/no-go criteria, and they always seem to elude us. You may be more structured than we are in this area, but in our organization we tend to judge each case on its own merits. Some customers are simply too important to our long-term interests to turn down. If their architecture is sorely lacking, then we often try to work with them to bring it to the state where it can be evaluated.

Use this step to set the client's expectations about what the ATAM will produce for the project. Even if there is no architecture in place, you may still wish to proceed with the joint understanding that the ATAM will produce a set of stakeholder-adopted scenarios that articulate the project's quality attribute priorities for the architecture. At that point, the clients may need to decide if they want to proceed under these conditions. They'll get better requirements, a set of stakeholders that have learned how to act in concert with each other, and a clear understanding of the architectural decisions that need to be made. Is that enough to warrant the investment? Perhaps. We don't think of this step as invoking a rigid set of criteria so much as achieving a mutual understanding and comfort level with what we expect the ATAM will produce in each case.

We have followed this advice and conducted evaluations when the architecture was substantially nonexistent or at least woefully underspecified. Even in these cases the ATAM was deemed beneficial by the participants. Scenarios always are useful, and many times new architectural insights emerge even for incipient architectures.

6.2.4 Phase 0, Step 4: Negotiate the Statement of Work

Step Summary

Agree on the contractual framework for the evaluation

Inputs

[ ] ATAM documents used in Step 1, with emphasis on cost, effort, and benefit data.

[ ] Candidate system documents from Step 2.

[ ] Sample statement of work from a previous exercise, or using a template specific to your organization.

Activities

These activities might take place over a span of time involving multiple meetings.

- [ ] Evaluation organization representatives negotiate a contract or statement of work with the client. Make sure that your statement of work resolves the following issues:
  - [ ] Period of performance.
  - [ ] Scope of work.
  - [ ] Scope of system to be evaluated.
  - [ ] Costs.
  - [ ] Deliverables.
  - [ ] Candidate schedule.
  - [ ] Responsibility for providing resources such as supplies, facilities, food, presence of stakeholders, and presence of architect and other project representatives.
  - [ ] The evaluation organization's availability (or nonavailability) for follow-up work.
  - [ ] Outline, contents, and disposition of the final report.
  - [ ] Client's agreement to participate in near-term and follow-up surveys.
- [ ] Record questions asked about the method or the process for possible inclusion in a Frequently Asked Questions list.

Outputs

- [ ] Statement of work as described above.
- [ ] Agreed dates for Phase 1 and Phase 2 steps.

Step Description

A statement of work establishes a contract between the evaluation group and the client's organization. This contract will help set expectations on both sides about what will be produced and its disposition. Make sure the contract specifies:

- Period of performance
- Scope of work
- Scope of system to be evaluated (for example, does it include software to generate or build the system? Does it include any software with which the primary system interacts?)
- Costs
- Deliverables, to whom they will be delivered, and when
- Candidate schedule
- Responsibility for providing resources such as supplies, facilities, food, presence of stakeholders, and presence of architect and other project representatives

- The evaluation organization's availability (or nonavailability) for follow-up work
- Outline, contents, and disposition of the final report
- Client's agreement to participate in near-term and follow-up surveys

The near-term and follow-up surveys will help you improve your application of the method by gauging results and incorporating improvements.

How It Went

Our organization uses a fairly standard statement of work. Our business development people used it to establish a contract with NASA, and there were no complications.

Speaking from Experience

What you can expect depends on how your organization is related to the client's organization and what the working relationship is between the two. You might expect to have to negotiate about schedule, cost, and delivery. The client will need to understand about participating in short-term and long-term surveys, and you should expect to have to explain that.

This agreement and the process surrounding it are often political. At times we have found that getting buy-in from a key stakeholder might require special rules. For example, we might need to agree on rules regarding the disposition of reports, such as, "We will show the report to you and give you a chance to formally respond before we show it to your funding organization."

6.2.5 Phase 0, Step 5: Form the Core Evaluation Team

Step Summary

Choose the evaluation team members who will participate in Phase 1.

Inputs

[ ] Candidate evaluation schedule (from Step 4 and client's desire).

[ ] Potential team members' availability (from internal schedules).

[ ] Candidate quality attributes of interest (for example, performance, modifiability, security, dependability) from client's description of the system in Step 2.

[ ] Team role definitions, given in Table 3.3.

Activities

These activities might take place over a span of time, including negotiation for services of domain/attribute experts.

[ ] Evaluation organization management (or designee):

    [ ] Form the evaluation team. Aim for an overall team size of four to six members. Participants discuss scheduling options and assign roles. Assign as many of the roles as practical at this time. Questioners may

be assigned based on initial understanding of what the driving quality attributes will be.[1]

[ ] Ascertain team members' availability during the evaluation period so that the evaluation can be scheduled with the client.

[ ] Make travel arrangements as necessary.

[ ] If team members' schedules do not support the schedule previously proposed to client, produce a revised schedule.

Outputs

[ ] Revised schedule to be proposed to client, if applicable.

[ ] List of team role assignments.

Step Description

This step consists of finding individuals who will serve on the evaluation team, scheduling their time, and assigning them team roles (see Table 3.3).

The evaluation leader should normally be the most experienced member of the team, unless a new evaluation leader is being groomed. In any case, the evaluation leader will be the primary "face" that the evaluation team shows to the project members and the stakeholders assembled to help with the evaluation, so it is important that this person show confidence and possess first-class facilitation and communication skills.

The duties of the questioners will depend upon the specific quality attributes of interest for the system being evaluated. Sometimes questioners are added to the team only after Phase 1 in which those quality attributes are discovered. Other teams have access to "jack-of-all-trades" questioners knowledgeable in many attribute areas, so the evaluation's team kick-off meeting determines their line of inquiry but their participation is already settled.

Figure 6.1 shows a small but typical evaluation team and how roles may be assigned to individuals.

All members of the team should be well grounded in architectural concepts and be perceived as objective and impartial by the development team whose architecture is being reviewed and by the client. This means that members of the development project itself, or members of a project that competes in some way, should not be chosen. Having individuals on the team who are knowledgeable about the relevant domain is a distinct advantage.

Team members should expect to devote approximately one or two person-weeks each to the evaluation: a day or two studying read-ahead material about the architecture and/or helping to set up the exercise, about three days for the exercise itself, and up to another week to help go over the analysis and write the final report. Travel time, if any, to and from the evaluation site is additional.

1. More questioners may be assigned later (see Phase 2, Step 0: Prepare for Phase 2) based on their quality attribute expertise.

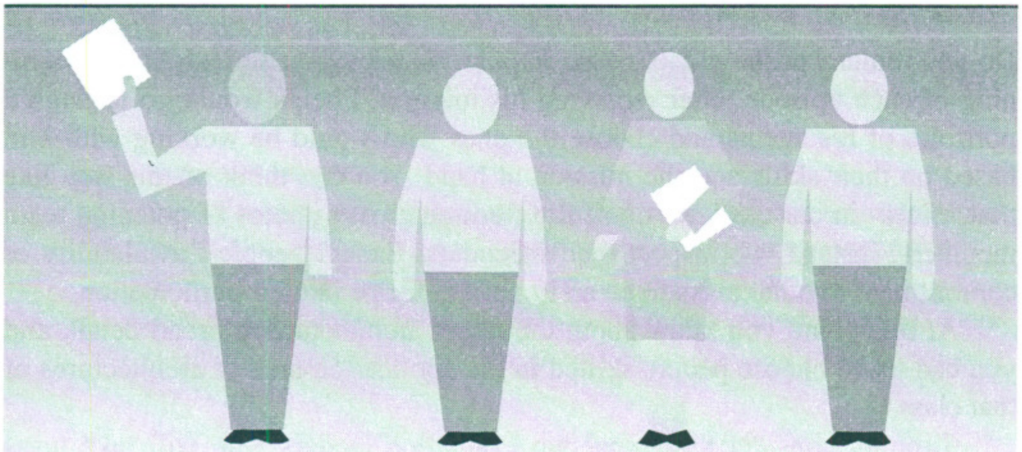


Figure 6.1 One Way to Assign Evaluation Team Roles to a Small Group of Individuals

How It Went

For this evaluation, the team consisted of the three authors plus a fourth member who was chosen because he had broad experience in large-system architectures, real-time systems, and government procurements. We assigned team roles as shown in Table 6.1.

Normally a single person would be the process observer, and another will be the process enforcer. For this evaluation, which featured a heightened awareness of the process steps, we decided to share these duties among all of us.

Table 6.1 ECS Evaluation Team Roles

Team Member	Role
#1	Team leader Evaluation leader
#2	Scenario scribe Questioner
#3	Proceedings scribe Questioner
#4	Timekeeper Questioner
All	Process observer Process enforcer

[ ] Client's nondisclosure forms, if procured during Step 2.

[ ] System documents received from client during Step 2.

Activities

[ ] Evaluation team leader:

- [ ] Send a message to the evaluation team announcing the time and place for the kick-off meeting and asking them to read the scenarios in this book (or in your own repository of scenarios) and to be ready to suggest those that may be applicable to the architecture being evaluated.
- [ ] Distribute any system and/or architecture documents already received from client.
- [ ] Announce roles and make sure that each team member is comfortable with the responsibilities of the role(s) he or she was assigned.
- [ ] Assign responsibility for writing various sections of the final report (if required), and for producing the viewgraphs for presenting results. The expectation is that the presentation and the report will be produced as the evaluation progresses, not all at once at the end.
- [ ] Make plans with the team to meet and confer throughout the ATAM exercise (for example, hold evening caucuses over dinner) to gauge progress, air concerns, and assess how the evaluation is proceeding (optional but strongly recommended).

[ ] Team members:

- [ ] Select appropriate example scenarios using your organization's own repository as candidates. Scenario scribe records the list.
- [ ] Sign nondisclosure forms (if provided); team leader returns them to client.

Outputs

[ ] List of selected example scenarios.

[ ] Signed nondisclosure forms (if provided by client) returned to client.

[ ] Responsibilities for final report and results presentation assigned.

Step Description

At the kick-off meeting, team members receive their role assignments. They also receive their assignments for specific sections of the results presentation and final report. Any information gleaned from the previous steps about the architecture to be evaluated is shared. The schedule for the evaluation is finalized. Travel logistics, if any, are worked out. Nondisclosure agreements, if any, are signed and returned to the client.

Depending on your team's experience, you may want to

- Go over the responsibilities and expectations associated with each team role
- Discuss possible analysis questions, based on your preliminary understanding of the architecture and its goals

Speaking from Experience

Do you remember the old *Mission: Impossible* television series? At the beginning of each episode, after receiving his mission, Phelps would go through a portfolio of his agents and choose the ones who would be working with him based on their skills and the mission at hand. You can think of this step like that, except in our case, we don't flip through glossy photos of potential team members. Instead, we inspect their calendars. Expect people's availability or commitment to other tasks to be an important factor in their participation.

At this point, you know about the target architecture in broad detail, and you can try to choose people skilled in the application area or architectures of that class.

Strongly consider assigning two evaluation leaders who will take turns facilitating the steps of the ATAM. Using two leaders helps prevent fatigue and keeps the proceedings fresh. Also, a fresh facilitator tends to pay more attention to the steps of the ATAM process, resulting in a higher-fidelity and more repeatable exercise.

Another point needs to be made regarding the formation of the team. There are times when a small team is possible and even desirable. For example, the project may be small and a big-budget evaluation is not warranted. In such a case it is perfectly reasonable to create a "mini-ATAM" team consisting of one to three people; each person wears multiple hats.

It is also acceptable to use people from the client's organization as members of the evaluation team. We know of several evaluations for which this worked particularly well. In one case the SEI leader of the evaluation decided that he wanted to engage the client organization a little more closely during the evaluation. He invited one of the architects to share the facilitation duties. This had a very positive energizing effect. The ECS case study is another successful example. In this case the goal of inviting NASA's participation was to start transitioning to NASA the capability of performing architecture evaluations. This also worked well for another organization whose members wanted to learn the ATAM.

6.2.6 Phase 0, Step 6: Hold Evaluation Team Kick-off Meeting

Step Summary

Assign roles and plan Phase 1.

Inputs

[ ] List of team role assignments from Step 5.

[ ] Sample scenarios from your own repository of scenarios from previous evaluations.

[ ] Candidate quality attributes of interest (derived business goals and architectural drivers) recorded during Step 2.

- Discuss letting a junior member of the team take on added responsibilities (such as making the ATAM presentation or carrying out the evaluation leader's role for a portion of the evaluation) as a way to gain experience

Besides assigning roles, responsibility for the final report and the results presentation of Step 9 in Phase 2 needs to be assigned. By assigning sections of the report and presentation, team members can be gathering the right information for those sections as the evaluation proceeds. Putting together the presentation and report then becomes a matter of assembly rather than creation, which dramatically increases the quality of the results and decreases the time it takes to produce them.

How It Went

We had set a date with the client for Phase 1 based on mutual availability. NASA had previously sent us some overview information about ECS and its architecture, and we went through that material as a group to start learning the relevant concepts and terminology. The meeting took about an hour and a half; one of the team members participated by telephone. We paid special attention to the data-intensive nature of ECS. This started us thinking about performance issues associated with high-workload systems.

Speaking from Experience

We often try to assign roles to team members based upon their expertise with specific subsystems or quality attributes. We look at the requirements that have been provided and find the key ones with respect to the quality attributes. We try to see if there are important quality attributes that are missing or underspecified, and we try to understand how the architecture (if it has been described) might satisfy, or fail to satisfy, the requirements.

6.2.7 Phase 0, Step 7: Prepare for Phase 1

Step Summary

The evaluation team leader communicates with the client to arrange the necessary logistics for the Phase 1 meeting.

Inputs

[ ] Sample Phase 1 agenda (see Figure 6.4).

[ ] Sample scenarios produced from Step 6.

[ ] Supply list, such as the one in Figure 6.2.

[ ] Template for presentation of architecture (see Figure 3.2).

[ ] Template for presentation of business drivers (see Figure 3.1).

Activities

[ ] Evaluation team leader communicates with client about Phase 1 to:

- [ ] Outline the purpose of the meeting.
- [ ] Confirm the time and place.
- [ ] Include an agenda.
- [ ] Ask the client to arrange a presentation of a system overview and context presentation, including business goals and constraints of the system. (Send Template for Presentation of Business Drivers.)
- [ ] Ask the architect (or ask the client to ask the architect) to present the architecture. (Send Template for Presentation of Architecture.)
- [ ] Include a list of applicable scenarios that may help stimulate thinking.
- [ ] Optional: Ask the client to bring an organizational chart showing the development team structure and the client's relationship to it.
- [ ] Assure the presence of the architect and any other project representative(s) whose presence is appropriate.

[ ] Arrange for food and necessary supplies (overhead projector, flipchart, markers, whiteboard, and so on) at the place of the meeting.

[ ] Assure the presence of the core evaluation team.

Output

[ ] Communication to client.

Step Description

Many logistical details must be handled before an ATAM Phase 1 meeting can take place. The most important issue is agreeing on a date for Phase 1 and making sure that the right people from both sides will be present and prepared to carry out their roles. In addition to the evaluation team, Phase 1 should be attended by people who can speak for the system whose architecture is being evaluated. This means people who can name the quality attribute goals that are driving the architecture and can articulate whatever future visions exist for the system. This also means people who can speak authoritatively about the architecture and the ways in which the architecture achieves the quality-specific attributes demanded of it. In short, the architect should be present.

Secondary details include making sure that necessary supplies will be present (such as an overhead projector, whiteboards, and flipcharts), making lunch arrangements, and finalizing the agenda. Figure 6.2 shows a sample supply list; you can use something similar to check off the things you need.

If the evaluation team has not yet received architecture documentation for the project, ask for it as part of this step; the team will need to look it over before Phase 1 begins. It's a good idea to brief the architect on what is expected of his or her presentation and to go over the kind of information he or she will convey. A good checklist is given in Figure 6.3. It's also a good idea to hold a similar conversation with the person who will be presenting the business drivers.

Facilities and Food

- Meeting room large enough to hold _______ people, preferably arranged with conference-style (U-shaped) seating, reserved for the duration of the evaluation
- Security badges for participants
- Food for meals and snacks if there are no nearby dining facilities
- Writing supplies
- Large flipcharts (two or three) for recording the brainstormed scenarios, issues, risks, etc.
- Markers for writing on the flipcharts
- Whiteboard for drawing diagrams, recording scenarios, tallying votes, etc.
- Markers for writing on the whiteboard
- Adhesive tape for hanging flipcharts around the room
- Blank viewgraphs for making new slides as needed
- Markers for writing on the viewgraphs
- Three pads of self-adhesive note papers, 2" x 2" or larger
- Name badges or "name tents" for participants

Electrical/Electronic Needs

- Laptop computer for use by the proceedings scribe
- Computer video projector for computer-based presentations
- Overhead projector for showing background slides, agenda, adopted scenarios, etc.
- Electrical extension cord

Read-Ahead or Other Written Material

- Overview of system being evaluated, including its context and goals, delivered by the client to the evaluation team leader ______ weeks before the start of the evaluation exercise
- Architecture documentation for the system being evaluated, as agreed upon during Phase 0, delivered by the client to the evaluation team leader ______ weeks before the start of the evaluation exercise
- Evaluation method overview, distributed to participants ______ weeks before the start of the evaluation exercise (optional)
- Copies of participants' end-of-exercise survey to hand out at conclusion of exercise
- Electronic copy of viewgraph templates for presentation of results

Figure 6.2 Supply Checklist for an ATAM Exercise

1. What are the driving architectural constraints, and where are they documented? Are they requirements or goals? Are they measurably quantitative or qualitative? In particular, what are the system's real-time constraints?
2. What component types are defined? For each, what are its:
   - Characteristics
   - Methods
   - Data members
   - Limitations
   - Composition rules
3. What component instances are defined by the architecture?
4. How do components communicate and synchronize? In particular:
   - Mechanisms used
   - Restrictions on use
   - Integration into component type definitions
5. What are the system partitions?
   - Composition
   - Restrictions on use and visibility
   - Functional allocations
6. What are the styles or architectural approaches?
7. What constitutes the system infrastructure?
   - Supplied functionality
   - Resource management
   - Uniform APIs
   - Restrictions
8. What are the system interfaces?
   - Identification
   - Participants
   - Identification of coordination mechanisms used
   - Typing of interfaces
9. What is the process/thread model of the architecture?
10. What is the deployment model of the system?
11. What are the system states and modes?
    - Control
    - Responsibilities
    - State knowledge dispersal
12. What variability mechanisms and variation points are included in the architecture (variability in terms of implementation changes and not data or scenario changes)?
13. How far along is the development? Were the block delivery dates met? Did the blocks meet their functionality requirements?
14. What documentation tree and human help do new employees get?

Figure 6.3 Checklist of Questions the Architect Should Plan to Answer

An organizational chart is useful to reveal who works for whom, and what people's areas of responsibilities are. Sometimes, during an evaluation, there may be tension between staff and management, and this will help you manage that. If you think this is useful, ask your client for it during this step.

How It Went

We carried out this step through a combination of e-mail and telephone conversations. We established a mutually agreeable date via e-mail. The team leader then telephoned the NASA project leader, and the two of them went over the agenda for Phase 1. They chatted about who would be present and what kind of presentations would be appropriate for communicating the business drivers (Phase 1, Step 2) and the architecture (Phase 1, Step 3). The team leader followed up by sending the NASA project leader templates for the presentations, a list of necessary supplies, and a written copy of the agenda (such as the one in Figure 6.4, which is based on the ATAM agenda given in Chapter 3).

Speaking from Experience

Speaking to the architect and the business drivers presenter in this step is the single most effective thing you can do to prevent unpleasant surprises during their presentations in Phase 1. Often, project representatives use presentations

Time	Activity
08:30–10:00	Introductions; Step 1: Present the ATAM
10:00–10:45	Step 2: Present Business Drivers
10:45–11:00	Break
11:00–12:00	Step 3: Present Architecture
12:00–12:30	Step 4: Identify Architectural Approaches
12:30–1:45	Lunch
1:45–2:45	Step 5: Generate Quality Attribute Utility Tree
2:45–3:45	Step 6: Analyze Architectural Approaches
3:45–4:00	Break
4:00–4:30	Step 6: Analyze Architectural Approaches
4:30–5:00	Action Items and Preparation for Phase 2

Figure 6.4 Sample Agenda for ATAM Phase 1

they have on hand rather than presentations that really convey the information desired for the ATAM. The more of their presentations you can see beforehand, the more of a chance you'll have to offer suggestions or midcourse corrections, and Phase 1 will go much smoother. One of the most frustrating experiences we've had as ATAM evaluation leaders is listening to a presentation that is not conveying the information the evaluation team needs. The result is an all-around bad day: the already-tight agenda is blown while the team digs for the right information, the project representatives feel awkward because they've misconnected, and the team leader feels guilty because he or she failed to communicate the team's needs clearly.

6.2.8 Phase 0, Step 8: Review the Architecture

Step Summary

The evaluation team walks through the architecture documentation.

Input

[ ] Architecture documentation identified in Step 2.

Activities

[ ] Evaluation leader facilitates a meeting with the evaluation team in which the architecture documentation is examined and team members attempt to explain it to each other and identify questions or areas of incompleteness.

[ ] Proceedings scribe captures a list of questions to present to architect either during or before Phase 1.

Output

[ ] List of questions for architect.

Step Description

During this step, the evaluation team meets to go through the architecture documentation provided by the client. In theory, this step could be combined with the evaluation team kick-off meeting; in practice, however, the architecture documentation may not have been transmitted by the time the kick-off meeting takes place.

For this meeting, all team members are expected to have read the documentation beforehand to become familiar with its organization and content. In addition, the team leader appoints one or more members of the team to lead a discussion about the architecture. The discussion leaders' task is to try to *explain* the architecture to the other members of the team, based on the information in the documentation. The other team members see if they agree with the explanation, and the whole team catalogs questions that arise during the meeting. These questions, if pertinent to the achievement of important quality attributes, are posed to the architect during the presentation of the architecture during Phase 1.

How It Went

One week before Phase 1 commenced, the evaluation team met for two hours to discuss the documentation. One team member participated by telephone. The team leader had appointed a discussion leader who led us through the documentation until we all had a good understanding of the role each document played in conveying the overall architectural picture. Each of us posed one or two functional scenarios—satellites sending down data, for instance, or a scientist making a query of the data warehouse—and we walked through the architecture until we were satisfied that we knew (or at least *believed* that we knew) how the scenarios would be handled by the architecture.

Speaking from Experience

As noted above, you could combine this meeting with the evaluation team kick-off meeting. However, people in our organization tend to prefer a couple of shorter meetings to one longer meeting; they are also easier to schedule. But there are other reasons to make this a separate, later meeting. First, documentation tends to arrive when it arrives. It might arrive very early, when the client first explains the system. It might arrive some time after that. But it is most likely to arrive very late, just before Phase 1 begins. More often than not, the architect is still working on it. Second, reviewing the documentation just before Phase 1 makes the architecture and its associated issues fresh in the evaluation team members' minds, which is just what you want. Early ATAM exercises did not include this activity as an explicit preparation step; rather, we left it up to each team member to look at the documentation on his or her own. The difference this step makes is noticeable. The team arrives at the client's site well-versed in the architecture and its documentation, with a *shared* understanding of what its status is. The result is a much more effective and cohesive evaluation team.

6.3 Phase 1: Initial Evaluation

Phase 1 of an evaluation using the ATAM is when the evaluation team is formally introduced to the architecture for the first time. Phase 1 comprises the *presentation* activities (Steps 1–3) and the *investigation and analysis* activities (Steps 4–6). Finally, by the end of Phase 1 a list of action items should be crafted that, upon completion, will lay the groundwork for Phase 2 and bring the evaluation to a successful conclusion.

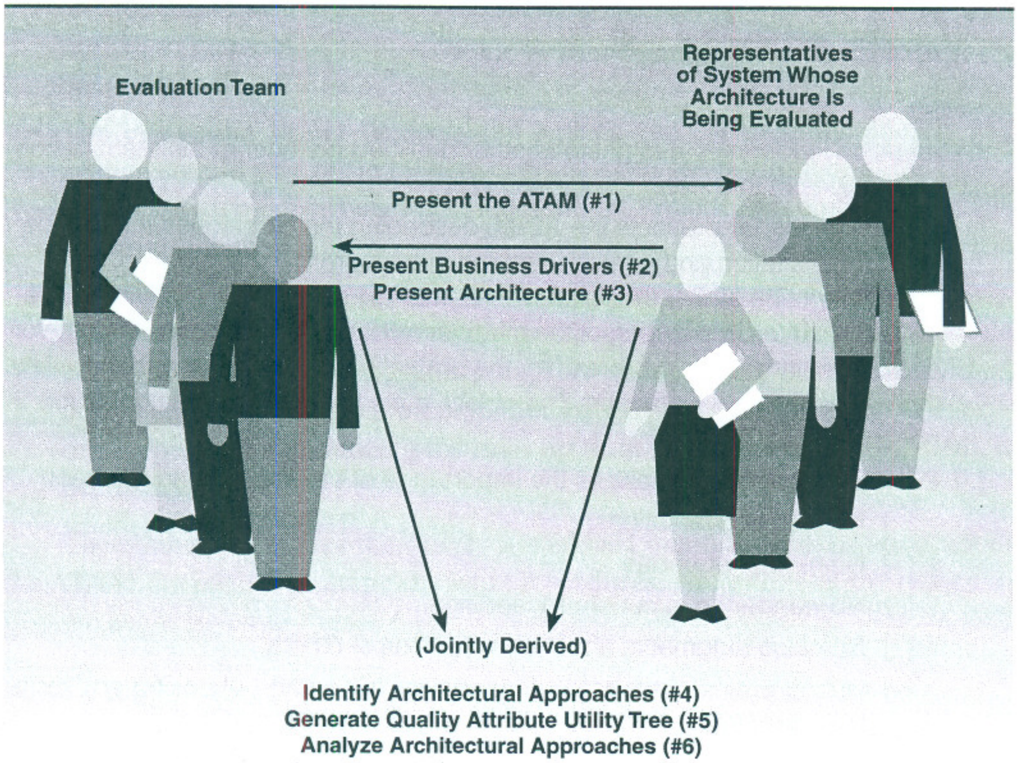


Figure 6.5 ATAM Phase 1 Information Exchange. (Numbers in parentheses refer to Phase 1 steps.)

The steps of Phase 1 are

1. Present the ATAM
2. Present business drivers
3. Present architecture
4. Identify architectural approaches
5. Generate quality attribute utility tree
6. Analyze architectural approaches

Figure 6.5 shows the flow of information in Phase 1.

In mid-April 2000, all four evaluation team members for the case study met with the core architecture and development team in the contractor's headquarters in Landover, Maryland. This was Phase 1 of the ATAM for ECS.

6.3.1 Phase 1, Step 1: Present the ATAM

Step Summary

The evaluation leader describes the evaluation method to the assembled participants, tries to set their expectations, and answers questions they may have.

Input

[ ] ATAM presentation viewgraphs.

Activities

The evaluation leader may designate another qualified member of the team to conduct the activities of this step.

[ ] Evaluation leader presents the ATAM, describing the techniques that will be used for elicitation and analysis and the outputs from the evaluation. The team entertains and answers questions, making sure all of the stakeholders have a clear understanding of the method and its goals. This can eliminate or avoid hostility among, for example, the project representatives who may view the evaluation as an intrusion. The object is discovery and communication, not criticism.

[ ] Evaluation leader emphasizes the importance of the rules of engagement:

  [ ] No side conversations.

  [ ] No hidden agendas.

  [ ] No wandering in/out of proceedings.

  [ ] No value judgments about contributions of others.

  [ ] During brainstorming, all scenarios are fair game—choosing and refining come later.

  [ ] During brainstorming, don't propose answers.

[ ] Evaluation leader asks the participants what their expectations are of the ATAM. Scenario scribe records them where all can see. Leader identifies those expectations that are *unlikely* to be met by the ATAM (perhaps because of the limitations of the method, the quality of prework, and so on) to avoid disappointment and misunderstanding later.

[ ] Evaluation leader clarifies the team members' roles and the roles of other participants.

Output

[ ] None.

Step Description

Step 1 kicks off the meeting, and hence there are some meeting kick-off details that need to be handled here.

- Have the client open the meeting, since he or she is the one at whose behest the stakeholders have assembled. The client should say a few words about the purpose of the meeting, ideally stressing the importance of the architecture evaluation. Then he or she introduces the evaluation leader.
- The evaluation leader should welcome all the participants and have them introduce themselves and their role with respect to the system being evaluated.
- Either the evaluation team can introduce themselves or the evaluation leader can introduce them one by one and explain the assignment each

team member has. This helps the participants understand why the evaluation team consists of four or five people—they can see that each member has a job to do.

- The evaluation leader presents the agenda for the meeting. Ideally the agenda should stay posted where everyone can see it throughout the meeting, or it can be distributed as a handout. (A typical agenda for Phase 1 is shown in Figure 6.4.)
- The evaluation leader explains the "rules of engagement" for the meeting. This makes it clear what behavior is expected of the participants: no side conversations, no coming and going, no hidden agendas, no individual dominates the discussion, participation by everyone, and respect for everyone's opinions. Announcing the rules up front will make enforcing them, if necessary, easier.

The evaluation leader then gives a standard one-hour presentation about the ATAM, laying out its steps and what it produces. An outline of the presentation we use is given in Figure 6.6.

How It Went

The project manager, who was sponsoring the evaluation, kicked off the meeting by stating the purpose and goals of the exercise and introducing the participants. Our evaluation leader then introduced the team members and stated the

Presentation	Explanation
Architectural analysis and the ATAM	Tells why architectural analysis is valuable and introduces the ATAM. Introduces concepts of risks, sensitivity points, and tradeoff points. Makes clear that purpose is not detailed analysis but finding trends and correlating architectural decisions to relevant impacts they have. Summarizes benefits of an ATAM evaluation.
Steps of the ATAM	Introduces the nine steps of Phase 1 and Phase 2 of the ATAM. Explains what information is sought in the business drivers and architecture presentations. Discusses architectural approaches. Shows small example of a utility tree. Introduces concept of scenarios, gives examples, and shows the stimulus–environment–response form. Shows examples of quality attribute questions used during analysis steps. Gives examples of risks, nonrisks, sensitivity points, and tradeoff points. Explains how the nine steps are distributed over two phases.
Example	A small ATAM exercise is exemplified. For each step, results are shown based on the BCS example given in Chapter 4.
Conclusions	Short summary and wrap-up. Reminder of the ATAM outputs.

Figure 6.6 Outline of Standard ATAM Presentation

team roles each one would be carrying out. He then gave the standard ATAM presentation. He emphasized the point that the purpose of the ATAM is to assess the consequences of architectural decisions in light of quality attribute requirements derived from business goals. He stressed that we were not attempting to precisely predict quality attribute behavior but rather were interested in identifying the key architectural decisions that affected the most important quality attribute requirements. As such our goal was to elicit and record risks, sensitivity points, and tradeoff points. We would also capture non-risks as they emerged during the exercise.

Questions from the audience centered around details of Step 5's utility tree. It turned out that this group had already done a great deal of thinking about iterative refinement of quality attribute goals and was anxious to codify those goals with the utility tree.

Speaking from Experience

In general, you can expect a few questions during the presentation, but most will come at the end. Some questions are about particular steps of the method, but most questions center on the results that the exercise will produce. You may be asked about the distribution of the final report; if so, you can relate the terms agreed to under the statement of work negotiated in Phase 0, or you can simply defer to the client and let him or her answer the question. Sometimes people ask about the agenda or about the difference between Steps 6 and 8 (the two analysis steps).

The evaluation leader normally makes the ATAM presentation, but that's not necessary. Having a junior team member lead this step provides excellent training.

It's a good idea to emphasize two aspects during this step. First, always make sure to express appreciation for the presence of the participants—especially the architect. The people you want to attend an evaluation are almost always the busiest, and they're taking time out from their normal routine to participate. Second, it is worth taking great pains to avoid saying anything that would introduce an "us versus them" flavor. There is no "them"—it's all "us." Everyone works as a single team to apply the method to the architecture.

6.3.2 Phase 1, Step 2: Present Business Drivers

Step Summary

A project spokesperson (ideally the project manager or system customer) describes what business goals are motivating the development effort and hence what will be the primary architectural drivers (for example, high availability, quick time to market, or high security).

Inputs

[ ] System overview presentation documents, if available from client.

Activities

[ ] Client or project manager presents the system's overview, including scope, business goals, constraints, and quality attributes of interest.

[ ] During the presentation, all evaluation team members will

- [ ] Look for ways to define the scope of the system being evaluated. Is software to generate or build the system included? What about software with which the primary system interacts?
- [ ] Listen for stakeholder roles to be mentioned.
- [ ] Listen for quality attributes of importance to be mentioned.
- [ ] Identify business goals and constraints.

[ ] After the presentation, evaluation leader recaps the list of business drivers. If desired, leader polls participants to ask for their lists of stakeholder roles, quality attributes, goals, and constraints. Proceedings scribe records the lists. Scenario scribe records the list of business drivers publicly.

Outputs

[ ] List of business goals.

[ ] List of quality attributes of interest.

[ ] Preliminary list of stakeholder roles that need to be represented during Phase 2.

[ ] Definition of scope of system. (Optional—this may not emerge until after the architecture is presented.)

Step Description

We want the architecture evaluation to focus on those aspects of the architecture that are important for achieving the business objectives of the system. So the ATAM includes a step that makes those objectives explicit. Here, a representative of the project makes a presentation, about an hour in length, that includes

- A description of the business environment, history, market differentiators, driving requirements, stakeholders, current need, and how the proposed system will meet those needs/requirements
- A description of business constraints (for example, time to market, customer demands, standards, cost, and so on)
- A description of the technical constraints (for example, COTS products that must be used, interoperation with other systems, required hardware or software platform, reuse of legacy code, and so on)
- Quality attribute requirements and from what business needs they are derived

During the presentation, evaluation team members listen for and list key information, including business goals, quality attributes, stakeholder roles, and any delimiting of the system's scope. At the end of the presentation, the evaluation leader summarizes the key business drivers. The proceedings scribe captures the summary for inclusion in the results presentation and the final report.

How It Went

The project manager described NASA's business objectives for ECS. The ECS processes, archives, and distributes earth science data from and about 15 scientific data collection instruments on seven major satellites. Examples of types of data are ocean color, sulfur dioxide from volcanoes, air temperature, moisture, sea surface winds, ice topography, and so on. The goal of the system is to make these data widely available, worldwide, 24 hours per day, in various forms in support of interdisciplinary earth science. There is a diverse user community with various interests in subsets of the collected data and their scientific processing.

The system is distributed at four data centers across the United States:

1. Earth Resources Observation System Data Center, Sioux Falls, SD
2. Goddard Space Flight Center, Greenbelt, MD
3. Langley Research Center, Hampton, VA
4. National Snow and Ice Data Center, Boulder, CO

Authority for various aspects of the system is decentralized. Data centers maintain control of their local facilities and interface with data users. Scientific teams maintain control over their science software and data products. The ECS project management team is responsible for maintaining and planning the evolution of the system. The intended operational life of the system is roughly from 1999 to 2013.

The size of the problem is unprecedented for NASA. The system is large, comprising about 1.2 million lines of custom code. It employs about 50 COTS products integrated with the custom code. The same version of the software runs on each of the distributed sites, deployed on about 20 Unix servers from multiple vendors at each site.

ECS manages large geospatial databases. It archives large amounts of data, adding about 1.5 terabytes per day. It distributes about 2 terabytes of data daily in several different formats. The system also executes complex science algorithms provided by the science community and interfaces to about 34 external systems. Figure 6.7 illustrates the system context for ECS in a viewgraph presented by the project manager. ECS is the portion of the system inside the bold rectangle.

The ECS team had done its homework about the ATAM and surprised us (pleasantly) by including an early version of a utility tree in their business drivers presentation. They had worked this out among themselves in advance of our Phase 1 visit. They felt this was the best way to explain the quality attributes of interest. There was an emphasis on maintainability and operability

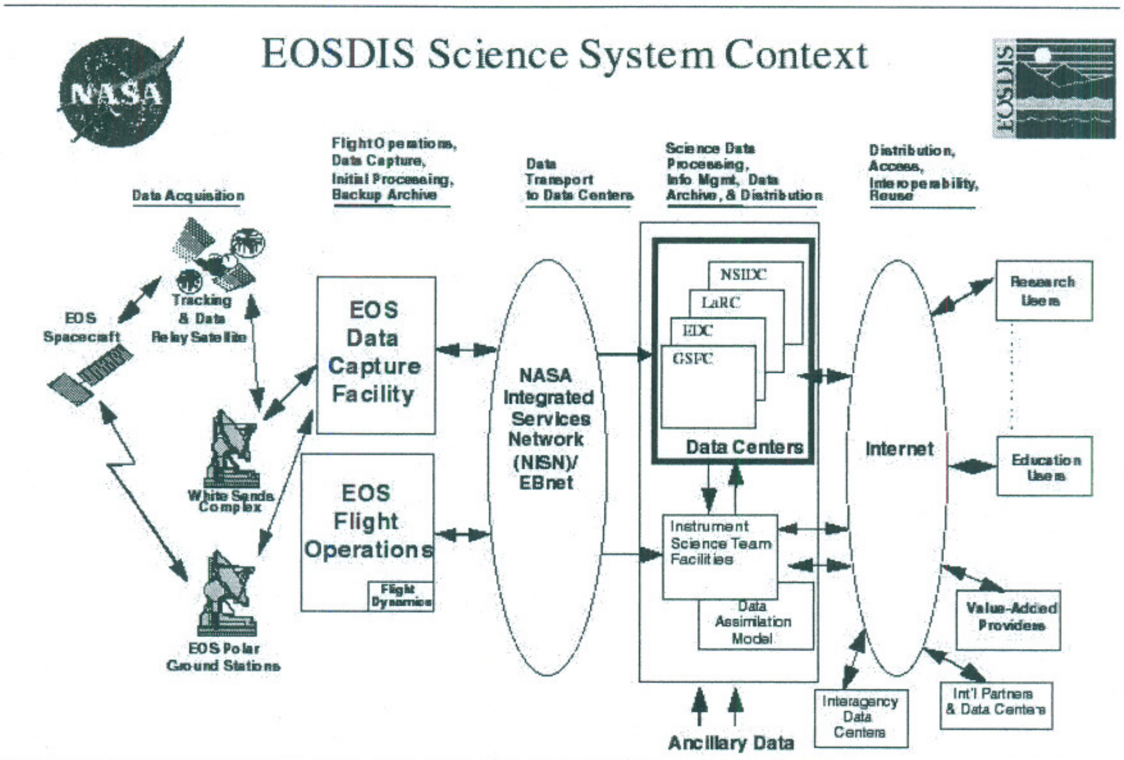


Figure 6.7 ECS in Context

concerns. The other quality attributes of concern were reliability, scalability, and performance. A subset of their preliminary utility tree is shown in Table 6.2.[2]

In Step 5, the scenarios are refined to be more specific (and hence, more analyzable) than some of these.

Quality attributes are shown on the left, and corresponding scenarios (with IDs for easy reference) are shown on the right.

At the end of the client's presentation, we captured the following essential business drivers for ECS.

- Primary business goals:
  - Support for interdisciplinary earth science
  - Concurrent ingest, production, and distribution of high data volumes
  - One-stop shopping (location transparency of system components and data)

2. Recall that in a utility tree, the first level is simply "Utility," the second level names a broad quality attribute, the third level is a refinement of the general quality, and the fourth level is a scenario expressing the third-level refinement. Table 6.2 shows the second and fourth levels, which the client labeled "Quality Attribute Goals" and "Attribute-Specific Requirements," respectively.

Table 6.2 ECS Quality Attributes of Interest

Quality Attribute Goals	ID	Attribute-Specific Requirements
Maintainability	M1	Changes to one subsystem require no changes to other subsystems.
	M2	Independently roll back subsystem deployments.
	M3	Reduce regression testing from 5 days to 1 day.
	M4	Reduce time to upgrade operating system, database, and archival management of COTS products by 50 percent or within 6 months of release, whichever is sooner.
Operability	O10	System should be able to reprioritize 1,000 orders in 20 minutes by user class, data types, media type, destination, or user.
	O14	System should be able to service 1,000 concurrent requests through V0 gateway or MTM gateway without operations intervention.
Reliability	R1	No system resources held by data inputs or outputs that are failed or suspended for more than 10 minutes.
	R2	Data errors with one part of request (input/output) should not prevent fulfillment of other parts.
	R6	No requests should be lost as a result of system overloads or failures.
	R8	Search, browse, and order submission are down for no more than 1 hour per week due to either failure or backup.
Scalability	S2	System can support 50 sites.
	S3	System can support ingest from 100 data sources.
	S4	System can support electronic distribution to 2,000 sites.
Performance	P1	Fivefold improvement for search response times for Landsat L-7 searches.

- Secondary business goals:
  - Support for externally developed science algorithms/applications
  - Science data reprocessing (that is, lineage, version management, processing, and reprocessing, including evolution of algorithms used)
- Other business goals:

- Assurance of data integrity for archive
- Automated operations to minimize operational costs
- Management of geospatial data

The proceedings scribe stored this list for later incorporation into the presentation of results.

Speaking from Experience

A project organization as well prepared for an ATAM evaluation as NASA is, in our experience, the exception rather than the rule. Not only had the NASA participants studied the method from the open literature before our arrival, but they also had proceeded to begin ATAM activities—drafting a utility tree and using it in their business drivers presentation—before the ATAM exercise even began. More often is the case where the project organization prepares a rudimentary presentation culled from existing viewgraphs for, say, their project sponsor. You should take steps to make sure that the business drivers presentation covers the information relevant to the evaluation. Talking to the presenter beforehand (during Step 7 of Phase 0) will help with this, but the evaluation leader should also be prepared to politely intervene during the presentation itself if the topic is wandering off course.

One thing to keep in mind as the business drivers are being presented is the importance of creating a distillation that can be used throughout the remainder of the evaluation. If the presenter has not done so, the evaluation team should elicit in bullet form five to ten key business drivers and their associated quality attribute. It is the ability of the system to serve these drivers that ultimately determines its success or failure. These drivers usually express important market differentiators and cost constraints that have strong implications for system quality attributes. One of the key aspects of the evaluation involves establishing the link between the risk themes that emerge during the evaluation and these business drivers. Keeping this in mind during the evaluation helps to focus the evaluation.

6.3.3 Phase 1, Step 3: Present the Architecture

Step Summary

The architect describes the architecture, focusing on how it addresses the business drivers.

Inputs

[ ] Description of important attribute-specific requirements from Phase 0, Step 2, and Phase 1, Step 2.

[ ] All available documentation for architecture being evaluated.

[ ] Description of any attribute-specific architectural approaches from Phase 0, Step 2.

[ ] Sample quality attribute characterizations, such as those shown in Chapter 5.

Activities

[ ] Architect describes technical constraints, other systems with which the system must interact, and attribute-specific architectural approaches.

[ ] Proceedings scribe records highlights of architect's explanations.

[ ] All evaluation team members note architectural approaches used/mentioned, potential risks in light of drivers, and additional stakeholder roles mentioned.

Outputs

[ ] Summary of architecture presentation, recorded by proceedings scribe.

[ ] Architecture presentation materials.

Step Description

In this step, the architect spends anywhere from one to three hours presenting the software architecture that is to be evaluated. As a result of the setup step for this phase, he or she will have received a suggested presentation template like the one shown in Figure 3.2. We ask the architect to cover the following information:

- The driving architectural requirements, the measurable quantities associated with these requirements, and any existing standards, models, and approaches for meeting these
- Major functions, key system abstractions, and domain elements along with their dependencies and data flow
- The subsystems, layers, and modules that describe the system's decomposition of functionality, along with the objects, procedures, and functions that populate these and the relations among them (for example, procedure call, method invocation, callback, containment)
- Processes, threads along with the synchronization, data flow, and events that connect them
- Hardware involved, including CPUs, storage, external devices, or sensors along with the networks and communication devices that connect them
- Architectural approaches, styles, patterns, or mechanisms employed, including what quality attributes they address and a description of how the approaches address those attributes
- Use of COTS products and how they are chosen and integrated
- A trace of one to three of the most important use case scenarios including, if possible, the run-time resources consumed for each scenario
- A trace of one to three of the most important change scenarios describing, if possible, the change impact (estimated size and difficulty of the change) in terms of the changed components, connectors, or interfaces
- Architectural issues and risks with respect to meeting the driving architectural requirements

During the presentation, the evaluation team members listen for architectural approaches; these will be catalogued in the next step. They also listen for quality attribute requirements that have not been previously captured.

How It Went

The project manager and one of the architects took turns presenting various aspects of the architecture for this large, complex data acquisition and management system. Figure 6.8 shows a viewgraph that NASA presented to give an overview of ECS.

The key subsystems of ECS are data management, data server, data ingestion, data processing and planning, management interoperability, and user interface. A portion of the system is concerned with the automatic "pushing" of raw data or synthesized data into the system. Another portion of the system is concerned with manual and automatic "pulling" of selected data products to the user community.

Data Management and Data Server Subsystems Managing data, at the heart of the system, is accomplished through two subsystems: the data management and data server subsystems. The data server subsystem is primarily responsible for data storage and retrieval, whereas the data management subsystem is

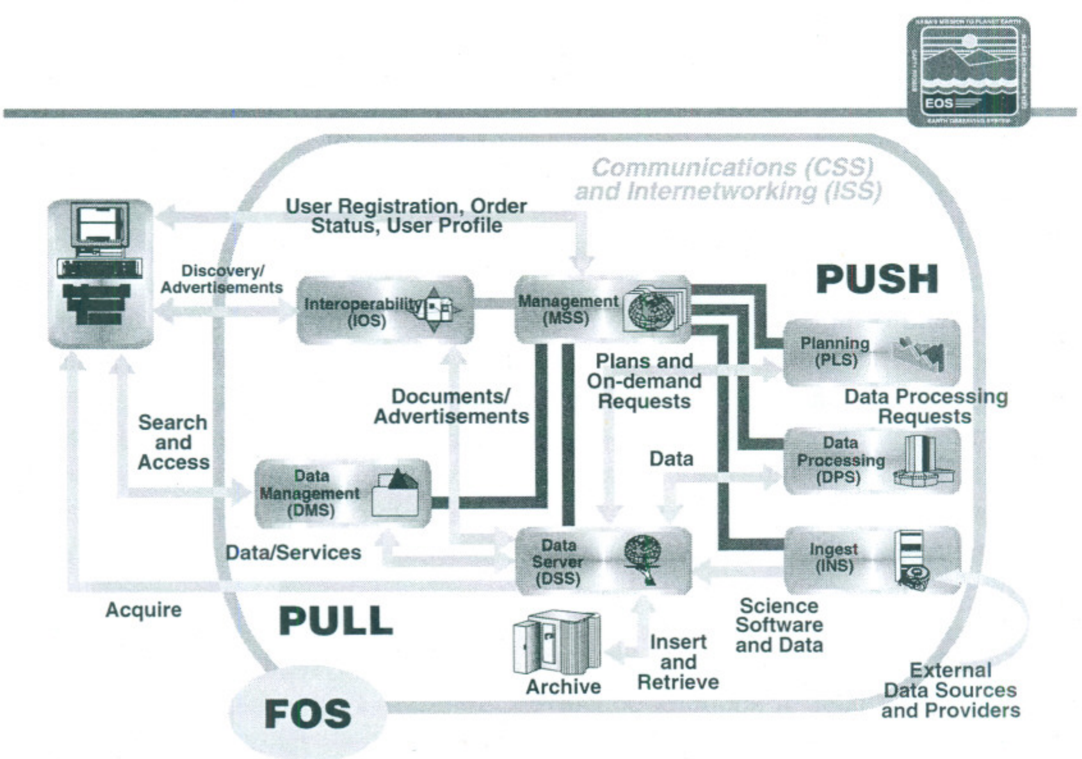


Figure 6.8 View of ECS Components, as Provided by the Client

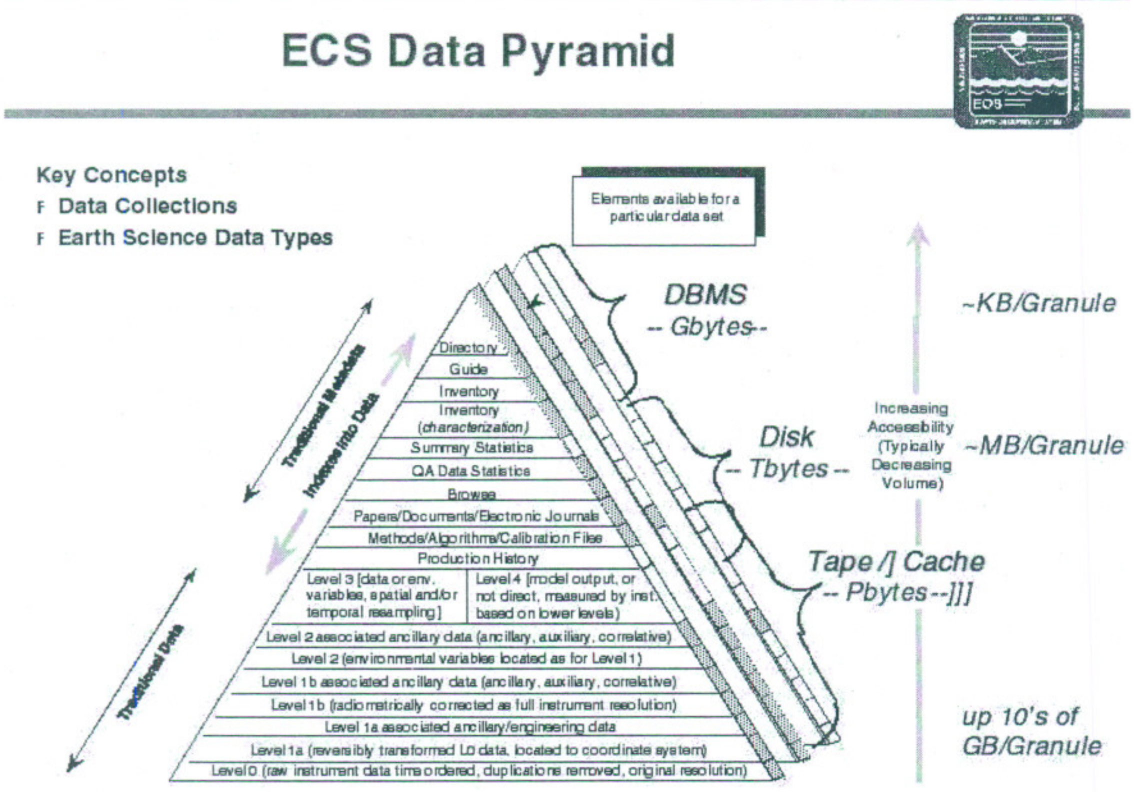


Figure 6.9 Common Metamodel for ECS Data

responsible for searching through the data. Data products are described by metadata (the model for which is shown in Figure 6.9) to enable browsing and searching through terabytes of data.

The metamodel defines all stored data types, their attributes, and their relationships. Each pyramid is known as an *earth science data type*. Raw sensor data is at the bottom of the pyramid. The next level is data synthesized from the raw data using science algorithms. Relevant metadata is stored further up the pyramid. Some attributes are defined across all data types; other attributes are specific to a data type. Storage of the product data associated with a data type can be configured on a type-by-type basis. Portions of a data collection may be allocated across multiple storage units (for example, silos) to improve performance—allocations may be changed over time to support new access patterns. Data collections may be replicated to protect against data loss. Support exists for on-line query of data.

Data Ingestion Subsystem The data ingestion subsystem is the entry point for instrument data. It supports concurrent ingestion from multiple data sources; accommodates diverse interface requirements; allows tailoring of data checking, conversion, and preprocessing; and performs data source authentication, data transmission checking, and metadata validation.

Data Processing and Planning Subsystems While the data ingestion subsystem is responsible for acquiring raw data, the data processing and planning subsystems are concerned with producing higher-level data products. Data processing supports the integration and execution of software algorithms developed by teams of earth scientists. The raw data collected by the ingestion subsystem is transformed and coalesced into higher-level data products using various earth science algorithms and then ultimately used by the earth science communities. Planning supports the automated scheduling of algorithm executions based on either data arrival or user requests such as:

- Routine production requests (always generate a product when input data arrives)
- On-demand requests (generate a product only when requested)
- Product requests tied to future events (a data acquisition request, for example)

Multiple algorithms may be chained together. Production rules determine which data are used for input. The data products managed by ECS include some generated by ECS using earth science software and products produced by other computation/science centers that are redistributed and reprocessed into higher-level products by ECS.

Management and Interoperability Subsystems The management subsystem supports manual and automatic ordering and delivery of data. Interoperability subscription mechanisms support automatic delivery of data when it becomes available, and notification mechanisms support notification based on ingestion of data of interest. Different media delivery options and data compression options are supported.

User Interface Subsystem ECS supports a Web-based interface that permits searches against metadata attributes and provides integrated, on-line access to browse data prior to ordering full data products.

Other Subsystems The flight operations segment (FOS) is outside the scope of ECS. ECS is supported by a communications infrastructure that is simply pictured as a ring around the ECS subsystems in Figure 6.8.

Other Views Deployment views were also presented but not heavily used during the evaluation. In fact, ECS comes with volumes of documentation, and we could have easily spent a month or more listening to presentations about the design and operation of the system. The point of the first four steps of Phase 1 is to acquire a broad overview of the architecture so that the evaluators can begin to formulate probing questions.

Speaking from Experience

At this point in an evaluation more detail is not necessarily better. Preparing one hour of the right high-level depiction of the architecture is a useful exercise in its own right and provides better information to the evaluation team than many hours of details. We have found it useful to review the architecture ahead of time. In fact, at this point in time the evaluation team should have already had an opportunity to see and react to at least some of the architecture documentation. Sending questions in advance to the architect can serve as a guide for what to present during this step. Let the utility tree direct you later to places where more detail is necessary, and let the subsequent analysis drive the need for detail.

Still, you can expect a flood of information to be poured on you at this point. The architects are in their glory and will talk for 20 hours if you let them. Often the information they provide is vague (and hence unanalyzable). Architecture teams often have canned presentations. Try to discourage architects from rattling through these. Keep the end goal in mind—to elicit enough architecture information to understand the architecture approaches that address the high-priority items from the utility tree.

We recommend here, just as we did for the business drivers presentation, that the evaluation team be watchful for and urge the architect to present key architectural approaches. One of the main goals of this presentation is to end up with a distillation of the approaches the architect has used to address what he or she considers to be the important quality attribute goals the architecture needs to satisfy. These will formally be listed in the next step of the ATAM, but being watchful for them here is important.

6.3.4 Phase 1, Step 4: Identify Architectural Approaches

Step Summary

Architectural approaches are identified by the architect but are not analyzed.

Inputs

- [ ] Description of important attribute-specific requirements from Phase 0, Step 2, and Phase 1, Step 2.
- [ ] Architecture presentation materials from Step 3.
- [ ] Description of any attribute-specific architectural approaches from Phase 0, Step 2.
- [ ] Architectural approaches identified by team members during presentation of the architecture.
- [ ] Sample quality attribute characterizations, such as those shown in Chapter 5.

Activities

- [ ] Evaluation team identifies approaches inherent in the architecture as presented. Options:
  - [ ] Evaluation leader asks architect to quickly identify the major approaches he or she thinks were used

[ ] Evaluation leader polls team to gather the approaches identified by each member.

[ ] Evaluation leader asks architect to validate the list gathered.

[ ] Scenario scribe records the list of approaches for all to see.

Outputs

[ ] List of approaches recorded by scenario and proceedings scribes.

Step Description

The main purpose of looking for approaches during the architecture presentation is to start formulating questions and drawing preliminary conclusions about how the architecture is realizing key quality attribute goals.

During this step, the evaluation leader asks the architect to recount the dominant architectural approaches used in the system, and the scenario scribe captures the list for all the participants to see. Evaluation team members augment the list with any other approaches they heard during the previous step or noted during their review of the architecture documentation in Step 8 of Phase 0.

How It Went

Several architectural approaches were identified during the ECS architecture presentation.

- The client-server approach is used heavily since this is a data-centric system.
- Distributed data repositories are used to accommodate the distribution of the user community, to enhance performance and for reliability.
- Distributed objects with location transparency are used to achieve modifiability in a distributed setting.
- A three-tiered layered approach separates the rules for automatic higher-level data generation and data subscription from data management and storage.
- Metadata can be thought of as an approach that supports usability, the ability to give meaning to terabytes of data.

Each approach suggests some issues to think about.

- Client-server approach: possibility of contention for databases and throughput issues
- Distributed data repository: issues of database consistency and possible modifiability concerns
- Distributed objects: a plus for modifiability but with potential performance consequences
- Three-tiered layers: again, a plus for modifiability but with potential performance consequences

- Metadata: sounds splendid for usability (but, to foreshadow a bit, could cause some modifiability problems)

Speaking from Experience

This is one of the most straightforward steps of the ATAM. We seldom devote more than 30 minutes to it. It is much more a matter of compiling than creating. The evaluation team already has a set of approaches in mind from working with the architect to craft the architecture presentation and from reviewing the architecture documentation in Step 8 of Phase 0. The architect, ideally, already explicitly named the approaches during the Step 3 presentation.

On the other hand, we have seen cases in which this step seems very mysterious, especially when the architect has not even thought of his or her architecture in terms of architectural approaches. Moreover, many stakeholders in the audience might not be familiar with what an architectural approach is, let alone know about specific ones. However, as the evaluation team extracts examples of approaches from the architecture, other people will begin to understand the essence of what is being captured. They will begin to see that an architectural approach is a collection of architectural decisions, which work in concert to contribute to the achievement of a quality attribute goal. They will also begin to understand the kinds of architectural decisions that the ATAM processes in its analysis mill. Participants who wondered how any method could claim to evaluate something as amorphous and monolithic as an architecture now begin to see that architectures emerge from many discrete decisions, and those can be analyzed and scrutinized.

Notice the pattern for Steps 2 through 4: present, distill, present, distill. In Step 2 business drivers are presented and then at the end of the presentation they are distilled into a list. In Step 3 the architecture is presented and then in Step 4 the architecture is distilled into a list of architectural approaches. These distillations should be in everyone's mind for the duration of the evaluation, and should be posted where everyone can see them. They serve to frame the whole evaluation. The rest of the evaluation determines whether the business drivers are supported by the architectural approaches.

6.3.5 Phase 1, Step 5: Generate Quality Attribute Utility Tree

Step Summary

The quality factors that comprise system "utility" (performance, availability, security, modifiability, and so on) are elicited, specified down to the level of scenarios, annotated with stimuli and responses, and prioritized.

Inputs

[ ] Business drivers and quality attributes from Step 2.

[ ] List of architectural approaches recorded during Step 4.

[ ] Template for the utility tree for the proceedings scribe to use when capturing the utility tree. The template can be a table such as Table 6.2 on page 156 with the entries blanked out.

Activities

[ ] Evaluation leader facilitates the identification, prioritization, and refinement (to scenarios) of the most important quality attributes. Address the following steps

  [ ] Assign "Utility" as root.

  [ ] Assign quality attributes identified as important to this system as children of root.

  [ ] Facilitate identification of third-level nodes as refinements of second-level nodes, for example, "latency" might be a refinement of "performance" or "resistance to spoofing" might be a refinement of "security." Use sample quality attribute characterizations to stimulate discussion.

  [ ] Facilitate identification of quality attribute scenarios as fourth-level nodes.

  [ ] Ask development organization participants to assign importance to the scenarios, using H/M/L scale.

  [ ] For those scenarios rated "H," ask the architect to rate them in terms of how difficult he or she believes they will be to achieve. Use H/M/L scale.

[ ] Questioners make sure that important quality attributes are represented in the tree or point out differences between what has been generated and what was presented as drivers or what appeared in requirements specification. Questioners also listen for additional stakeholder roles mentioned.

[ ] Scenario scribe records the utility tree publicly.

[ ] Proceedings scribe records the utility tree in the electronic record.

Output

[ ] Utility tree of specific quality attributes requirements prioritized by importance and difficulty.

Step Description

This step involves facilitated, directed brainstorming aimed at filling the leaves and internal nodes of a blank utility tree. The intermediate and final results should be made public for all participants to see with flipcharts, viewgraph slides, or a projected PC display.

The root (Level 1) of the tree is "Utility." Level 2 consists of broad quality attributes such as "performance." Begin filling in the second level by listing the quality attributes that were named in the Step 2 presentation of business drivers. Ask the participants for other quality attributes at this level. Evaluation team members (especially questioners, who are assigned to "oversee" certain quality attributes) are free to make suggestions, but all candidates should receive group consensus before being added to the tree.

After the group agrees that the quality attribute list at the second level is fairly complete, move to the third level and begin filling in quality attribute refinements. For each Level 2 quality attribute, ask what it means in more concrete terms or how it is measured. These will lead to the Level 3 quality attribute refinements.

Finally, proceed to Level 4, and solicit quality attribute scenarios that capture in analyzable detail what each quality attribute means. Aim for two or three scenarios for each quality attribute refinement on Level 3.

How It Went

For ECS, we had the prototype utility tree from Step 2 for a starting point, an unusual but happy circumstance that gave us a head start.

Working at the Top As the result of a previous independent (but ATAM-influenced) effort at articulating their goals for the architecture, the client team had a set of fifty-some "study goals" for the ECS architecture. These goals were each mapped to one or more "high-level goals" (to use the client's term):

- Operability (O)
- Maintainability (M)
- Scalability (Sc)
- Performance (P)
- Flexibility/extensibility (F)
- Reliability/availability (Ra)
- Security (S)
- Usability (U)

These became second-level quality attributes in the ECS utility tree. Quality attribute refinements on Level 3 quickly followed from these.

Crafting Scenarios To fill out the fourth level of the utility tree, we focused on refining the quality attribute refinements into analyzable quality attribute scenarios. For example:

- **Quality attribute refinement (Level 3):** Changes to one subsystem require no changes to other subsystems.
- **Quality attribute scenario (Level 4):** Deploy the next version (5B) of the science data server with an update to the earth science data types and latitude/longitude box support into the current (5A) baseline in less than eight hours with no impact on other subsystems or search, browse, and order availability.

The Level 3 quality attribute refinement describes the maintainability measure as the degree to which changes in one subsystem propagate to other subsystems. The quality attribute scenario refines this by specifying the subsystem, specifying a specific change, and specifying a bound on how long this change should take.

Another example, this one for reliability/availability:

- **Quality attribute refinement (Level 3):** Search, browse, and order submission downtime.
- **Quality attribute scenario (Level 4):** Search, browse, and order submission are down for no more than one hour per week due to either failure or backup.

The attribute-specific requirement describes the reliability measure as the amount of downtime for a specific system capability. The quality attribute scenario refines this by adding causes for downtime and a bound on the amount of downtime.

Finally, one more example, this time for performance:

- **Quality attribute refinement (Level 3):** Fivefold improvement for search response times for Landsat L-7 searches.
- **Quality attribute scenario (Level 4):** A Landsat L-7 search with 100 hits received under normal operations takes 30 seconds or less.

The attribute-specific requirement describes the performance measure of response time for searching for Landsat data. The quality attribute scenario refines this by specifying the size of the search in terms of the number of data base hits and an upper bound on the response time.

The NASA participants were quite conscientious at trying to pose scenarios in good stimulus–environment–response form, often reminding each other (before we had a chance to) to try to phrase the scenarios using the preferred structure.

Prioritizing Utility Tree Scenarios The ECS team was methodical in interpreting the difficulty; difficulty was defined in terms of an estimate of how long it would take to create a new architectural design and/or implementation to realize the scenario and the level of expertise needed to create the design. The scale they used for both difficulty and importance was numeric: 10 was low, 20 was medium, and 30 was high. They preferred a numeric scale because they could sum the two factors to produce an overall measure of importance. Table 6.3 contains a subset of the ECS utility tree (shown in the tabular template used by the evaluation team's proceedings scribe to capture it); the final priority associated with a scenario was calculated by summing the difficulty and importance.

Speaking from Experience

Quality Attribute Names Quality attribute names that appear at the second level of the utility tree (such as performance, reliability, operability, and so on) have different meanings to different communities. Sometimes they have special meaning within organizations. Sometimes they have only very broad,

Table 6.3 Subset of the ECS Utility Tree with Priorities[a]

Level 2: Quality Attribute	Level 3: Quality Attribute Refinement	Level 4: Quality Attribute Scenario	Importance	Difficulty	Sum
Maintainability	M1: Changes to one subsystem require no changes to other subsystems.	M1.1: Deploy 5B version of the science data server with an update to the earth science data types and latitude/longitude box support into 5A baseline in less than 8 hours with no impact on other subsystems or search, browse, and order availability.	30	30	60
	M2: Independently roll back subsystem deployments.	M2.1: Perform rollback of the science data server from M1.	20	20	40
	M3: Reduce regression testing from 5 days to 1 day.	M3.1: Regression test the science data server deployment from M1 in 1 day.	20	10	30
	M4: Reduce time to upgrade operating system, database, and archival management COTS by 50 percent or within 6 months of release, whichever is sooner.	M4.1: Upgrade from IRIX 6.2 to IRIX 6.5 and replace some hardware in 1 day.	30	30	60
		M4.2: Upgrade Sybase in 1 Day	30	20	50
		M4.3: Upgrade DCE in 1 Day	30	30	60
Operability	O10: System should be able to reprioritize 1,000 orders in 20 minutes by user class, data types, media type, destination, or user.	O10.1: Backlog management—after 24 hours of downtime, operations reprioritizes backlogged workload in 30 minutes to ensure tasks are worked off in priority order and that normal operations continue to be supported with no degraded throughput following resumption of normal operations.	30	20	50

(continued)

Table 6.3 Subset of the ECS Utility Tree with Priorities[a] (*continued*)

Level 2: Quality Attribute	Level 3: Quality Attribute Refinement	Level 4: Quality Attribute Scenario	Importance	Difficulty	Sum
Operability *(continued)*	O14: System should be able to service 1,000 concurrent requests through V0 Gateway or MTMGW without operations intervention.	O14.1: MODAPS down for 24 hours, recovers and requests 2 days of data; work off in priority order.	20	20	40
		O14.2: Receive 100 concurrent search requests, don't reject high-priority requests, and work-off without overloading system as capacity permits.	20	20	40
Reliability/ Availability	Ra1: No system resources should be held by data inputs or outputs that are failed or suspended for more than 10 minutes.	Ra1.1: L-7 fixed scene orders for electronic FTP push to a site whose FTP server is down, system suspends within 10 minutes of first failed request, and all resources are available while requests are suspended. Distribution to others not impacted.	30	10	40
	Ra2: Data errors with one part of request (input/ output) should not prevent fulfillment of other parts.	Ra2.1: Order for 100 granules, 3 on off-line tape/drive, system suspends these requests in 10 minutes of first failure and operator is able to resume remainder.	30	10	40
	Ra6: No requests should be lost as a result of system overloads or failures.	Ra6.1: DDIST must be cold-started due to hardware problem, pending orders identified and re-started in 5 minutes.	10	10	20
	Ra8: Search, browse and order submission downtime	Ra8.1: Search, browse and order submission are down for no more than 1 hour per week due to either failure or backup.	30	20	50

(*continued*)

Table 6.3 Subset of the ECS Utility Tree with Priorities[a] *(continued)*

Level 2: Quality Attribute	Level 3: Quality Attribute Refinement	Level 4: Quality Attribute Scenario	Importance	Difficulty	Sum
Scalability	Sc2: System can support 50 sites.	Sc2.1: Cross-site order tracking across 50 sites, status in 2 minutes for a 5-site order.	20	30	50
		Sc2.2: Cross-site user registration in 24 hours across 50 sites.	20	30	50
	Sc3: System can support ingest from 100 data sources.	Sc3.1: Receive ingest requests from 100 sites; work off in priority order and manage throughput to requirements.	20	20	40
	Sc4: System can support electronic distribution to 2,000 sites.	Sc4.1: Subscription fires for 2,000 users to send 1 GB of data to each; system works in priority order.	20	10	30
	Sc5: System able to scale to 10x requirements for ingest, distribution and processing without software changes.	Sc5.1: 10,000 data processing requests (DPRs) per day; an additional 6,000 DPRs will be planned and executed each day as part of normal operations with no additional staff or hardware.	20	30	50
Performance	P1: Fivefold improvement for search response times for Landsat L-7 searches.	P1.1: L-7 search with 100 hits under normal operations, result in 30 seconds.	30	20	50

a. Level 1 ("Utility") has been omitted to conserve space.

imprecise meaning. And yet one of the important goals of the ATAM is to elicit the precise quality attribute goals for the system being evaluated.

For the ECS system, operability was important. Operability is concerned with the level of operator intervention necessary to use the system. The stimulus for operability is a request for operator intervention. The response is the amount of time it takes for an operator to carry out the request. Operability directly impacts the number of operators needed to run the system.

Calibratability was an important attribute for another architecture evaluation. Calibratability is similar to operability. It is a measure of the amount of time it takes a calibration engineer to calibrate the physical models that are an integral part of the system.

Even an attribute as common as performance has different meanings for different people. To some people performance connotes functionality, to others availability, and to others it has to do with timing-related behavior.

The ATAM handles this ambiguity just fine since it has a built-in disambiguation mechanism. We let clients use whatever terms they want for the second level of the utility tree; the evaluation team requires clarity beginning at the third level of the utility tree. If the name of the quality attribute is ambiguous, its quality attribute refinement at Level 3 of the utility tree imbues it with meaning. This allows organizations to use familiar terminology while allowing the ATAM evaluation to proceed by creating unambiguous scenarios reflecting the organization's quality attribute requirements.

Missing Leaves It is often the case that people refine quality attributes on the second level into quality attribute refinements on the third level but then fail to provide any scenarios to instantiate the qualities any further. The result is a utility tree with a filled-out internal structure but some missing scenario leaves. That's OK. If nobody can think of a scenario, then it probably means that the quality attribute (or at least the third-level refinement of it) is not particularly critical to the system.

Well-Formed Scenarios Don't expect participants to propose scenarios in perfect stimulus–environment–response form. You should encourage people to think in those terms, but it is not necessary at this point to enforce well-formedness rules. That can come later, when you pick the ones for analysis. Remind people of the goal, though—it will help clarify their thinking. The ECS group, in fact, found the structured form of scenarios (stimulus–environment–response) helpful because it caused them to ask the right questions to produce clear, well-defined scenarios.

Scenarios versus Requirements You might expect a discussion about the relationship of scenarios to detailed requirements specifications. While it doesn't make sense to spend time crafting scenarios that duplicate functional requirements, we don't want to leave requirements-based scenarios out of the analysis mix, either. A good compromise is to try to identify certain key functional requirements that are indicative of the main work of the system, especially under environmental circumstances that were at the edge of the envelope of one or more system quality attributes.

Rank Assignment Some groups are not content to assign "H," "M," or "L" (or, as in the case of ECS, values of 10, 20, or 30) to scenarios without coming to a precise agreement beforehand as to what those terms mean. In most evaluations using the ATAM, we have simply let the stakeholders assign intuitive and informal meanings to the terms, but occasionally a group is determined to do better than that.

Here's one way. Ask the group to rate the scenario in terms of how important it is to the system's overall acceptability. Assuming all the scenarios are on the table, ask them to pretend that they aren't allowed to have them all. Which ones would they be the least willing to give up? That is, which ones would (if removed) leave an unacceptable product? Those are the "H" ones. For ranking in terms of difficulty, you can try to use a measure of time it would take to carry out the scenario.

- "H" could mean the scenario would require two or more person-years to implement, would require the services of a senior engineer, and had an unclear or unknown solution.
- "M" could mean a scenario with a known solution requiring two or more person-years to carry out by senior technical staff.
- "L" could refer to everything else.

We have observed, however, that even after a group exhaustively defines the terms, scenarios tend to be rated by simply comparing them to previously rated scenarios: "Do you think scenario 52 is as hard as scenario 38? No? Well, that makes it an 'M', then."

When assigning difficulty, try not to be too formal about projecting effort—we're after coarse-grained divisions here, not detailed analysis.

Finally, some people wonder why prioritization is done by consensus in Step 5 but by a detailed and carefully controlled voting procedure in Step 7. The primary reason is because this combination takes less time and it works. It's easier to form a consensus among the small group attending Phase 1, whereas a consensus informally wrung from a large group would be much less reliable. Nevertheless, if your group decides every scenario in the utility tree warrants a rating of (H,H), you may want to use a voting scheme similar to the one in Step 7. Let the architect and any designers present vote on the difficulty rating, but let everyone from the project team vote on the importance rating.

6.3.6 Phase 1, Step 6: Analyze the Architectural Approaches

Step Summary

Based on the high-priority factors identified in Step 5, the architectural approaches that address those factors are elicited and analyzed (for example, an architectural approach aimed at meeting performance goals will be subjected to a performance analysis). During this step architectural risks, nonrisks, sensitivity points, and tradeoff points are identified.

Inputs

[ ] Utility tree from Step 5.

[ ] List of architectural approaches from Step 4.

[ ] Analysis of architectural approach template (see Figure 3.5).

[ ] Sample quality attribute characterizations such as those shown in Chapter 5.

Activities

[ ] Architect identifies components, connectors, configuration, and constraints relevant to the highest-priority utility tree scenario nodes.

[ ] Evaluation team generates style-specific and quality-attribute-specific questions as starting points for discussion. Use the architectural mechanisms in the sample quality attribute characterizations as a guide.

[ ] Proceedings scribe records discussion and records risks, nonrisks, sensitivity points, and tradeoff points.

[ ] Scenario scribe records risks, nonrisks, sensitivities, tradeoffs, and open issues as they arise, as identified by evaluation leader.

Outputs

[ ] Analysis of architectural approach templates, filled out for analyzed scenario.

[ ] List of sensitivity points, tradeoff points, risks, and nonrisks.

Step Description

Analysis in this step does not entail detailed simulation or precise mathematical modeling. It is more of a qualitative analysis, perhaps similar to detective work in that we are looking for architectural clues that might reveal suspects. Precise analysis is impractical to carry out in real time in front of a large group. Moreover, it is not cost effective; it would take more time than it is worth to apply detailed analysis to many scenarios. On the other hand, our investigation is informed by knowledge of the mechanisms that are commonly used to realize quality attribute requirements.

To carry out this step, let the template for analysis of architectural approaches (see Figure 3.5) be your guide. Begin by making sure the scenario's stimulus, environment, and response are clearly stated; if not, take the time to put it in proper form. Then, with the architect's help, identify the architectural approaches that are relevant to carrying out this scenario. Using the style-based analysis questions, begin probing for known risks with the approach. Try to see how the approaches affect each quality attribute of interest, including those beyond the one with which the scenario is associated. Judge the answers you receive as problematic or not with respect to the quality attribute goals.

When a risk, sensitivity, tradeoff, or nonrisk is identified, make sure the scenario scribe captures it publicly.

At the conclusion of each scenario's analysis, each member of the team tries to formulate (in his or her mind) a set of risk themes based on the total set of risks identified so far. As more risks are uncovered, the team gathers evidence for or against the risk themes hypothesized. These risk themes play an important role in the presentation of results and the final report.

How It Went

In this section, we recount the analysis of three high-priority scenarios. These scenarios are M1.1, RA8.1, and P1.1, maintainability, reliability/availability, and performance scenarios, respectively. We discuss the analysis for scenario M1.1 in great detail and then offer a more cursory discussion for scenarios RA8.1 and P1.1.

For each scenario the stimulus, response, and attribute of concern are highlighted. The key architectural decisions impacting the scenario are then elicited and each is indicated as being a risk, nonrisk, sensitivity point, and/or tradeoff point. For each scenario that we analyzed we filled out an analysis template; our proceedings scribe kept a record by using a corresponding template in his laptop computer.

Analyzing Scenario M1.1 One of the highest-priority scenarios in the utility tree was maintainability scenario M1.1. Table 6.4 shows the results of its analysis in the tabular form captured by the proceedings scribe. The scenario was decomposed into its stimulus and response to ensure that each had been captured with sufficient precision. We then asked the ECS architect (and anyone else who could contribute) to enumerate the architectural decisions relevant to achieving this specific maintainability requirement. For each scenario a sequence of steps emerged. These steps provided grist for our group discussion, which led to the architectural decisions and associated risks, nonrisks, sensitivity points, and tradeoffs listed in Table 6.4.

We asked the architect to walk us through the scenario. He responded by giving the steps for realizing the scenario as follows:

1. Shut down the system in preparation for updating the science data server.
2. Perform a science data server database backup and check.
3. Install the new science data server code and GUI code.
4. Apply the changes to the configuration parameters that are related to the new capabilities.
5. Apply the science data server database patch.
6. With operations still down, perform a database check.
7. Reinitiate operations.

Enumerating steps is a useful way to consider a scenario, but this is not the end goal; the goal is to determine the impact the set of architectural decisions has on realizing the scenario.

The architect made the point that the upgrade was carefully controlled to keep the science data server interfaces, client library, and DLLs (Dynamic Link

Library) unchanged. This resulted in the recording of architectural decision AD1, which is about preserving the backward compatibility of the interface.

Table 6.4 Analyzing Scenario M1.1

Scenario #: M1.1	Scenario: (M1.1) Deploy 5B version of the science data server with an update to the earth science data types and latitude/longitude box support into 5A baseline in less than 8 hours with no impact on other subsystems or search, browse, and order availability.			
Attribute(s)	Maintainability			
Environment	During routine maintenance			
Stimulus	Deploy 5B version of the science data server with an update to the earth science data types and latitude/longitude box support into 5A baseline.			
Response	Less than 8 hours with no impact on other subsystems or search, browse, and order availability.			
Architectural Decisions	**Risk**	**Sensitivity**	**Tradeoff**	**Nonrisk**
AD 1 Backward compatibility of interface	R1			
AD 2 Static linkage of client stubs in servers (static binding of libraries)	R2			
AD 3 Single copy of key operational databases	R3	S1	T1, T2	
AD 4 Information about data types distributed throughout system	R4, R5, R6	S2		
AD 5 Name independence of subsystems			T3	
AD 6 Distributed objects with stable, simple API				NR1
AD 7 Uncontrolled dependencies among source files	R7			
Reasoning	For the qualitative reasoning and architecture diagram associated with these architectural decisions, see the discussion related to this table.			
Architecture diagram	None			

This was the mechanism chosen to prevent the propagation of this change beyond the science data server. However, further discussion of this mechanism revealed a risk. Backward compatibility only ensured backward syntactic compatibility, but there are cases in which the same interface can have different meanings (for example, a change in measurement units for some type of data). ECS has the capability of denoting such semantic variations in an interface, but this mechanism was bypassed. This discovery was recorded as risk R1:

R1. ECS is not using the infrastructure capability to "sign" an interface, thus ensuring only syntactic but not semantic compatibility. Consequence: interface may be syntactically compatible but semantically incompatible and system won't catch this. Could result in incorrect results or failure.

It was noted during the discussion of the above risk that if interfaces are required to change, then a new version of the entire system would have to be deployed. This is due to architectural decision AD2 about the static linkage of client stubs in servers. This concern was recorded as risk R2:

R2. Static linkage of client stubs requires that a new version of the system be deployed when an interface changes. Consequence: unintended changes may be included with the interface changes.

Another point made during the discussion of the scenario was the lack of a secondary set of databases that could remain operational while the primary databases were being upgraded. This was noted as architectural decision AD3. The consequence of this decision is risk R3:

R3. Single version of databases means that changes affecting the databases require significant testing. Consequence: changes require lots of testing and downtime.

This architectural decision was also denoted as sensitivity point S1:

S1. Increasing the amount of downtime associated with a software upgrade increases the risk of the upgrade because rollback is difficult.

Two tradeoff points were also associated with this decision:

T1. Availability may be negatively affected by having a single set of databases, but the single set is easier to maintain.

T2. While implementing with a single database might be less expensive (cost is an attribute as well), maintainability suffers since there might be a reluctance to upgrade—having database replicas reduces time and risk of software upgrades, hence you are more willing to do them.

Figure 6.10 highlights architectural decision AD4, which holds that information about data types is distributed throughout the system. This resulted in several risks:

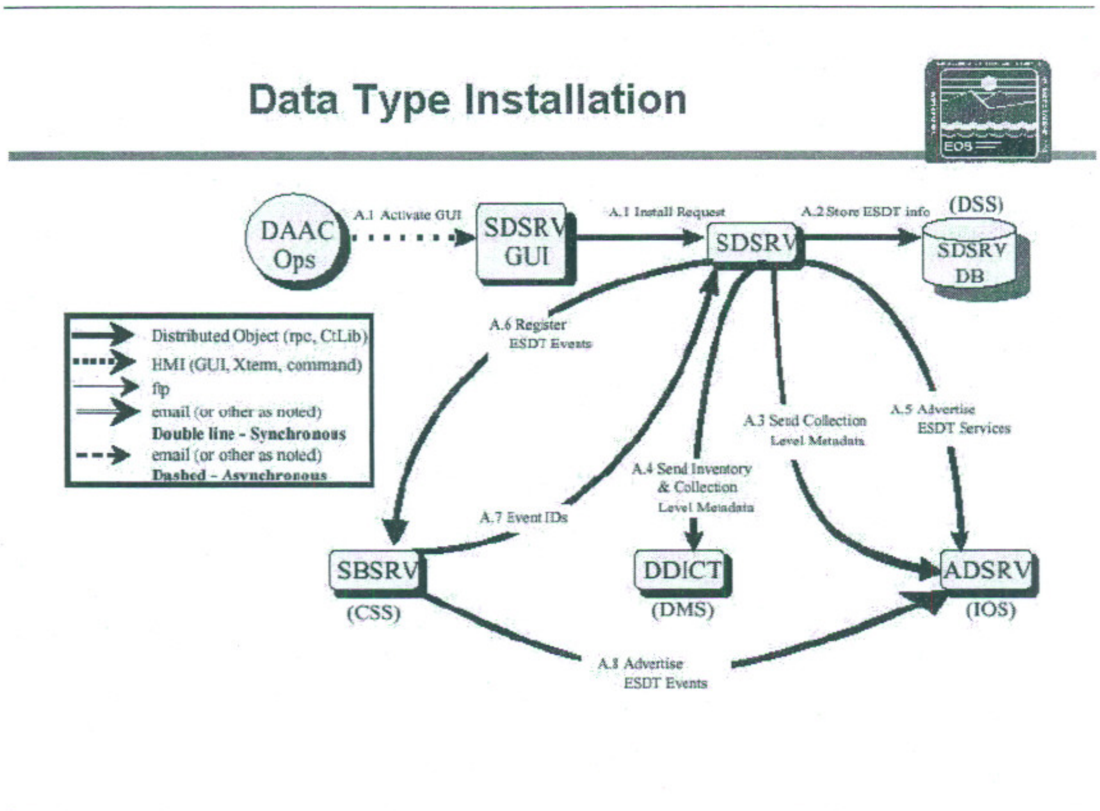


Figure 6.10 Data Type Installation

R4. Data type information is distributed throughout the system. Consequence: a change to a data type may "ripple" throughout the system.

R5. This decision makes it difficult to change data types. Consequence: increased modification costs and reluctance to make enhancements.

R6. All instances of data types may not be changed correctly. Consequence: database inconsistencies may result.

Architectural decision AD5 is one of the approaches that we identified in Step 4 of the method. A name server is used to provide location independence. This is a typical modifiability mechanism, namely a form of indirection. However, it is also very typical for indirection to have performance consequences and hence the tradeoff T3:

T3. Going through the name server enhances modifiability but imposes a performance cost.

Analyzing Scenario RA8.1 The ECS evaluation continued with scenario RA8.1, a reliability/availability scenario. Table 6.5 represents a portion of the architectural approach analysis table for this scenario; like Table 6.4, it reflects what was captured by our proceedings scribe.

Table 6.5 Analyzing Scenario RA8.1

Scenario #: RA8.1		Scenario (RA8.1): Search, browse, and order submission is down no more than 1 hour/week.			
Attribute	Availability				
Stimulus	Database goes down (for backup, therapeutic reboot, upgrade, fault).				
Response	Less than 1 hour of downtime/week.				
Architectural Decisions		**Risk**	**Sensitivity**	**Tradeoff**	**Nonrisk**
AD 8 Single on-line copy of Database		R8, R10	S3	T1, T4	
AD 9 Backup copy of Database on tape (not disk)		R9	S4		
AD 10 Database is not partitioned: single monolithic Database		R11	S5		
...		...	...	...	

There are two aspects of this scenario: routine database maintenance and recovery of a corrupted database. We focus on backup as the stimulus of interest for understanding this scenario. Again members of the ECS team enumerated a set of steps that elaborate the scenario.

1. Backing up is initiated by an operator. It entails running scripts that dump log files and/or the database, obtain index statistics, reorganize indices, and recompile stored procedures.
2. While the database backup is in progress, database searches experience increased response time.
3. If there is sufficient disk space to hold the backup, the database check is done against the backup, and upon completion of the check the database backup image is archived to tape.
4. If there is not sufficient disk space to hold the backup, the clients of the database are stopped, the science data server is taken down, the check is performed against the database itself, and then all system components are warm-started.

Once again these initial observations served as the starting point to identify key architectural decisions related to reliability, which in turn led to the identification of risks, nonrisks, sensitivity points and tradeoff points. But before we look at a couple of the architectural decisions let's step back for a second. The previous scenario was a maintainability scenario. The general concern of the scenario was localization of the effects of change to ensure that a change in one place did not propagate throughout the system. Therefore the architectural decisions that were highlighted revolved around mechanisms for preventing the propagation of change or finding places in which propagation was likely.

The current scenario is a reliability scenario. Reliability is concerned with the amount of time the system is available to perform the service for which it is intended. Therefore, the architectural decisions that are highlighted revolve around mechanisms that ensure the availability of services or decisions that cause services to be unavailable.

The focus of this scenario is on the effects on availability of backing up. Architectural decision AD8 highlights the fact that there is only a single on-line copy of the database. A risk results from this:

R8. There is only a single copy of the database on-line. Consequence: all activity on the database is halted for database consistency check and upgrade.

The fact that all database-related activity is halted when the database is checked for consistency has a clear impact on achieving availability goals. Likewise, architectural decision AD9, backing up to tape rather than disk, is the reason for the consequence in the above risk. That is, since the consistency checking cannot take place on tape, it must take place on disk, and this requires a quiescent database. We continued in the same vein, eliciting architectural decisions and determining whether they were risks, sensitivity points, and/or tradeoff points.

Analyzing Scenario P1.1 This scenario is a performance scenario. The ECS architect walked us through a simple performance analysis for this scenario. Table 6.6 has an extra column indicating the architect's estimates of the execution times associated with each architectural decision. (For this scenario the scenario steps simply became the architectural decisions.)

This simple analysis (that is, adding up the execution times associated with the architectural decisions) shows very clearly that the desired response time will be missed by a significant margin. Furthermore, it highlights potential bottlenecks in the metadata database and in the use of the object description language library. As in the other cases related risks were discovered.

Analyzing other scenarios This process continues until (1) time runs out, (2) the high-priority scenarios have all been analyzed, or (3) the client wishes to stop. The client may want to stop because of (1) or (2), or because he or she feels that the project organization can proceed to analyze the other high-priority scenarios on its own. In the case of ECS, the ECS project team tackled twenty additional scenarios without our assistance. This was extraordinary. In most cases, we aim for detailed analysis of just five to ten scenarios, and (even though we always emphasize that the project team is perfectly capable of continuing analysis on their own) the ATAM almost always stops when the evaluation team leaves.

Table 6.6 Analyzing Scenario P1.1

Scenario # P1.1	Scenario (P1.1): L-7 search with 100 hits under normal operations, result in 30 seconds				
Attribute	Performance				
Stimulus	Landsat L-7 search with 100 hits under normal operations				
Response	Search completed in 30 seconds or less				
Architectural Decisions	**Execution Time**	**Risk**	**Sensitivity**	**Tradeoff**	**Nonrisk**
AD 11 EDG as primary client	100 sec	R17	S8	T5, T8	
AD 12 Reused object description language library	65 sec	R18, R19	S9		
AD 13 Database not optimized for queries	12.5 sec	R20			
AD 14 Multiple data structure types used for query results	90 sec	R21	S10		
AD 15 Full materialization of metadata	30 sec	R22	S11	T6	
AD 16 All query results returned to gateway at one time	15 sec	R23, R24	S12, S13		
AD 17 Shared/multipurpose metadata database	100 sec	R25	S14	T7	

Significant findings emerged from just the three scenarios we recounted above.

- Maintainability is affected by the distribution of data type information throughout the system.
- Availability is impacted by having only a single copy of databases.
- Performance is impacted by the need to convert between various data types and the slowness of the object description language library.

These three findings formed the basis for some preliminary risk themes. As we analyzed other scenarios, these risk themes were refined to reflect commonalities among the risks that the scenarios as a whole revealed. Later, when we crafted the presentation of results in Step 9, the risk themes were mapped back to the business drivers that they could adversely affect, bringing closure to

Speaking from Experience

One of the main benefits of identifying architectural approaches in Step 4 is that there are attribute-specific questions associated with each approach. A first attempt at codifying approaches and associated questions is the ABAS work that we referred to in Chapter 5. You should create your own bank of architecture approaches and attribute-specific questions so that future evaluations using the ATAM can capitalize on this storehouse of design evaluation information.

You might be wondering how to find these architectural decisions, risks, sensitivity points, and tradeoff points. We would like to think that it is totally a by-product of executing the method. In actuality we know this is not the case. There is a certain amount of informed inquiry that relies on the experience of the evaluators. However, we feel confident that the ATAM will navigate you to the neighborhood of interesting and important architectural decisions.

One of the nice features about the ATAM is that it is "self-scaling." The utility tree tells you what to focus on, and the architectural approaches tell you where to focus. The scenario analysis performed in Step 6 is very constrained and compartmentalized. You can perform two days of analysis or two months of analysis, depending on how many scenarios you study and the level of detail at which you study them. However, in all cases it is possible to perform high-level analysis on the top five to ten scenarios and reveal important risks, sensitivity points, and tradeoff points.

It was a bit surprising to us to see how sensitive this analysis can be to social issues in addition to technical issues. At one evaluation, it looked like the success of the evaluation was in serious jeopardy at one point. The analysis of one scenario was taking particularly long and the evaluation participants were getting tired and frustrated. It appeared that the evaluation team was starting to lose control. Fortunately we were saved by the bell—the lunch bell—which gave us time to ponder the situation. We determined that three factors contributed to this state of affairs. (1) The scenario that we were analyzing actually had two scenarios embedded in it and the analysis was unwittingly oscillating between the two implicit alternatives. (2) Lunchtime was approaching and the participants needed a break. (3) The company was a foreign company. To optimize their discussions, the participants politely asked if they could discuss this scenario in their native language. Of course, we said, fine. This was a mistake. We no longer could guide the discussion and it proceeded down several blind avenues.

After lunch we refined the scenario and requested that the discussion take place in English. The scenario analysis proceeded smoothly and everyone returned to feeling positive about the evaluation.

The moral of this story is that even for the most technical aspects of the ATAM, social issues are just as influential as technical issues on the success of the evaluation.

Leading an Evaluation

We have participated in many evaluations of software and system architectures in many different application domains over the last ten years. In reporting on these methods, we have almost always concentrated on the technical aspects. But as we were developing and refining these methods we were exposed to a wide variety of systems, along with their organizations, organizational goals, management styles, and individual personalities. These experiences have convinced us of the need to explicitly manage the *nontechnical* aspects of running a technical review. There are aspects of the management, psychology, and sociology of performing architecture evaluations that need to be given some thought. Getting these factors wrong can doom the best technical effort.

This observation is not particularly an artifact of software development; it holds true of any complex engineering effort.

> *When Brunel and Robert Stephenson were building railways in the 1830s and 1840s, they were expected to involve themselves with raising capital, appearing before Parliamentary Committees, conciliating influential people who might oppose the necessary bill in Parliament, negotiating with landlords over whose land the tracks were to be laid, managing huge gangs of labourers, and dealing with subcontractors. Railways engineers had to be expert in finance, in politics, in real estate, in labour management, and in procurement. Why should we be surprised if software engineers may need to draw on expertise in mathematics, financial analysis, business, production quality control, sociology, and law, as well as in each application area they deal with? [Freedman 90]*

To lead an architecture evaluation you must be aware of all the issues, technical and nontechnical, that may arise. Therefore when you embark upon an architecture review you need to get buy-in at both managerial and technical levels. You need to get buy-in at the managerial level because stakeholders must be gathered together in sufficient numbers for the review to proceed. You need buy-in from the architect and key developers because these are the roles most able to speak to the design and potential changes to the design. Without buy-in from both managerial and technical participants, the process will not work.

Without understanding the "people" issues involved, a review may be chaotic and unsuccessful. For example, although the process depends on the presence of stakeholders, having too many participants causes problems in crowd control. In one of our reviews we had about forty stakeholders in the room, despite our pleas to keep the stakeholder group small and focused. This system was a large government acquisition with many participating government agencies and many contractors developing different parts of the hardware and software. Everyone wanted to establish a presence to ensure that their interests were properly represented and to impress their client. The result, however, was an overloaded review that was difficult to steer and focus.

Furthermore, it is important when setting the context for a review that you align the evaluees' expectations and goals with what the method can and cannot do. For example, in an ATAM evaluation we scrutinize the architecture's ability to achieve its quality goals. We don't do the organization's architectural design for them (although we can enumerate design possibilities and limitations). And we don't evaluate the project's functional completeness, domain modeling, or interface specifications (although we do offer these services as part of a larger engagement). Any review technique has limited aims and scope. Because of this, it is critical to set customers' expectations in advance so that they are not surprised or angry by what actually transpires.

In a SAAM exercise we led, one stakeholder complained, after it was finished, that we did not cover "domain scoping," or whether the component interfaces were adequately defined. This was completely true; that aspect is simply outside the scope of our standard SAAM exercise. Thus we apparently did not make it sufficiently clear at the start of the SAAM exercise just what we could and could not do within that structure.

The review leader is responsible not only for controlling the unruly participants but also for bringing out the shy persons. Their ideas may be equally important, but their shyness may prevent them from being heard.

As a leader you may have to decide what to do when the rules of the game are violated (for example, when people have side conversations, try to steal the agenda, or resist providing information). You may need to control the pace, sometimes forcing people to stay on topic so that the review time is efficiently used.

This sidebar cannot begin to list all of the possible situations in which you might find yourself. So what can you do? Be aware that problems *will* arise. The stakeholders won't follow the steps as you had intended. Information may not be provided, or it may be provided in an unsuitable form. Stay calm. Stay in control. Always keep the end in mind and return to the agenda. You do not need to slavishly follow the agenda (you might, at times, choose to allow discussion on an interesting side issue for a few minutes because it builds the confidence of the stakeholders to work together as a group), but neither should it be hijacked by the will of a single person. Keep in mind that the process has been proven to work, and it will be most beneficial to the majority if it is followed.

—*RK*

6.4 Hiatus between Phase 1 and Phase 2

In the period of time between Phase 1 and Phase 2 we worked with the client team to refine the presentations of both the business drivers and the architecture. In that way, the information would be much more crisp when the full community of stakeholders met in Phase 2. This is very much a standard part of Phase 1: obtaining information and then working with the architect to refine the presentation for Phase 2.

But the primary activity carried out during the hiatus is a continuation of Step 6. By continuing analysis during the break between Phase 1 and Phase 2, the hiatus provides a safety valve in case the agenda of Phase 1 did not allow enough time to explore scenarios in depth. The hiatus can allow the evaluation team to engage the architect in a more leisurely manner, through whatever means is convenient. A rule of thumb we use is that if our goal is to analyze ten scenarios in total, then we would like seven of them analyzed by the time Phase 2 begins. We use the hiatus to meet that goal.

6.5 Phase 2: Complete Evaluation

On a Monday in early May, we convened for Phase 2.

Phase 2 begins with the step

0. Prepare for Phase 2

This step occurs during the hiatus between the Phase 1 meeting and the meeting in which the other steps of Phase 2 are carried out.

After the preparation step, the team briefly repeats Steps 1–6 of Phase 1 in the presence of the larger set of stakeholders. Recall from Chapter 3 that the Phase 2 steps consist of validating the information we learned in Phase 1. The elicitation and analysis steps (architectural approaches, utility tree) are recapped and summarized for the larger audience, as opposed to being performed from scratch. These steps are not redefined in this section; refer to the Phase 1 section instead. When the summary of Phase 1 is complete, Phase 2 is supplemented by the following steps:

7. Brainstorm and prioritize scenarios

8. Analyze the architectural approaches

9. Present the results

6.5.1 Phase 2, Step 0: Prepare for Phase 2

Step Summary

Handle logistics for the Phase 2 meeting. Augment the evaluation team with additional questioners if necessary.

Inputs

[ ] All outputs generated during Phase 1.

[ ] Team role definitions (see Table 3.3).

Activities

[ ] Team leader addresses these points:

- [ ] Augment the evaluation team, if necessary, by adding questioners expert in quality attribute areas identified during Step 2 and Step 5 during Phase 1. Add team members to fill open roles, if any.
- [ ] Ascertain team members' availability during the evaluation period so that Phase 2 can be scheduled with the client.
- [ ] Make travel arrangements as necessary.

[ ] Evaluation leader communicates these points to the client:

- [ ] Reiterate the scope of the system being (and not being) evaluated.
- [ ] Send copy of utility tree generated in Phase 1.
- [ ] Make sure client invites to Phase 2 the stakeholders who will represent the stakeholder roles identified in Phase 1 and vouches for their attendance. Aim for roughly 10–12.
- [ ] Have client send stakeholders read-ahead material for Phase 2: business drivers presentation, architecture presentation, and scenarios from the utility tree from Phase 1 (optional).
- [ ] Send the client any read-ahead material you agreed to provide about the evaluation method.
- [ ] Ask for architecture documentation that was needed but missing during Phase 1.
- [ ] Send an agenda for Phase 2.
- [ ] Make sure person hosting Phase 2 has responsibility for arranging for site facilities, meals, and supplies.
- [ ] Have team members who are assigned to produce presentation of results (see Phase 0, Step 6: Hold Evaluation Team Kick-off Meeting) draft the sections on business drivers, architecture, utility tree, and architecture analysis using results of Phase 1 (optional but recommended).
- [ ] Have team members who are assigned to write final report draft the sections on business drivers, architecture, utility tree, and architecture analysis using results of Phase 1 (optional but recommended).

[ ] All agree on dates, time, and venue for Phase 2. Plan for the activities to take place on consecutive days if possible.

Outputs

[ ] Action item list and assignments, including

- [ ] Evaluation leader—summary of Phase 1.
- [ ] Client—list of stakeholders who will participate in the next phase.
- [ ] Architect—missing architecture documentation, if any.

[ ] All—dates, times, venue for next step(s).

[ ] Host of Phase 2— Arrange site facilities, meals, and supplies for next step(s).

[ ] Updated list of team role assignments.

[ ] First draft of business drivers, architecture, utility tree, and analysis portions of results presentation and final report (optional).

Step Description

This step is the analog to the last step of Phase 0, which laid the logistical groundwork for Phase 1. This step makes sure that Phase 2 will go smoothly by putting the wheels in motion to guarantee the attendance of the right people and the presence of the right supplies and other preparatory materials.

Knowing that the presentation of results in Step 9 has to be drafted in very short order, you can use this step to get a head start on that task. At this point, you have the business drivers, architecture presentation, architectural approaches, utility tree, analysis of several scenarios, and a list of preliminary results: risks, nonrisks, sensitivity points, and tradeoff points. While the steps in Phase 2 can (and usually do) temper or modify the results obtained so far, it will more than pay off to prepare the presentation ahead of time using Phase 1 outputs. It will be much easier to modify the presentation when preparing for Step 9 than it would be to build it from scratch.

Finally, you'll want to send an agenda to the people attending, such as the one in Figure 3.9. The time allotted to Steps 1–6 is for presenting and letting the participants react to the results of Phase 1.

How It Went

We confirmed the time and place for Phase 2 with the client via e-mail and telephone. We also confirmed the attendance of the stakeholder contingent. For Phase 2, stakeholders for the ECS evaluation included:

- The ECS software architect
- The ECS system architect
- An ECS designer knowledgeable about the COTS components used
- Representatives of the ECS operator community
- A representative from another government agency
- Several NASA representatives

We chose not to augment our evaluation team with any additional members. Finally, we spent some time drafting sections of the results presentation for Step 9, producing about 15 viewgraphs summarizing the results from Phase 1.

Speaking from Experience

The quality of an architecture evaluation is a direct function of the quality of the stakeholders who have assembled to help evaluate it. Sometimes the client is reluctant to involve certain stakeholders, for whatever reason. You must impress upon the client that if key stakeholders are not present, then the ATAM will not be able to evaluate the architecture from those key points of view. The results may look satisfactory, but in fact they will have an unseen hole in them. The reason the evaluation team listens for stakeholder roles throughout the client's overview presentation and the project's business drivers presentation is to help the client bring stakeholders to the table whom he or she might otherwise forget (or wish to forget).

6.5.2 Phase 2, Steps 1–6

In Phase 2, Steps 1–6 are summarized for the Phase 2 participants. They are allowed to ask questions, point out and help correct any mistakes or misunderstandings, and offer any suggestions for improvement. However, unless some catastrophic misunderstanding evidences itself during the summaries, wholesale overhaul of the Phase 1 outputs is discouraged.

The outputs of Phase 1 should be made available to the Phase 2 participants in summary form. Printed handouts are a good idea, or summarizing flipcharts hung in plain view also work. The idea is to instill in the participants' minds the principal outputs of Phase 1 so that they need not spend their energies on issues already surfaced and resolved.

During the ECS evaluation, we printed a summary of the business drivers, a list of the architectural approaches, and the utility tree on handouts for the participants. We put the current list of risks, nonrisks, sensitivity points, and tradeoff points on flipcharts hung along one side wall of the conference room. The stakeholders had a number of questions about the precise meaning of some of the scenarios in the utility tree and a few questions about the precise meaning of some of the risks we had recorded. We sensed a fair amount of interest in the business drivers presentation, which is not unusual: this is where the stakeholders see how (or if) their particular interests have made it into the consciousness of the system builders.

6.5.3 Phase 2, Step 7: Brainstorm and Prioritize Scenarios

Step Summary

A larger set of scenarios is elicited from the entire group of stakeholders. This set of scenarios is prioritized via a voting process involving the entire stakeholder group.

Input

[ ] Scenarios from the leaves of the utility tree developed during Step 5.

Activities

[ ] Evaluation leader facilitates brainstorming activity.

  [ ] Use the scenarios at the leaves of the utility tree to help facilitate this step by providing stakeholders with examples of relevant scenarios. Tell stakeholders that the leaves of the utility tree that were not analyzed are valid candidates for inclusion in this step.

  [ ] Put up the list of stakeholders to stimulate scenario brainstorming.

  [ ] Ask each stakeholder to contribute a scenario in a round-robin fashion.

  [ ] Tell participants not to be constrained by the utility tree structure, attributes in the utility tree, or scenarios at the leaves of the utility tree.

  [ ] Encourage participants to submit exploratory scenarios as well as use cases and growth scenarios.

  [ ] Open the floor for spontaneous contributions from any stakeholder. Don't let one or two people dominate. Don't let people propose solutions to the scenarios. Don't let people disparage or dismiss a particular scenario. Aim for around 30–60 scenarios.

  [ ] Questioners are responsible for brainstorming scenarios that address their assigned quality attributes. Make sure there are scenarios that represent each stakeholder.

  [ ] Scenario scribe records each scenario for all to see, being careful to use the exact wording proposed or adopted by consensus.

[ ] Evaluation leader facilitates prioritizing of scenarios.

  [ ] Allow 5 minutes for the consideration of voting.

  [ ] Allow people to walk around the posted flipcharts and propose consolidation (a person places an adhesive note next to a scenario with the number of another scenario that he or she believes is similar).

  [ ] After everyone sits down, the group adopts or rejects each consolidation proposal.

  [ ] Write the scenario numbers on the whiteboard (leaving the posted scenarios up where people can see them).

  [ ] Assign *n* votes to each participating member of the audience (including evaluation team members other than the team leader), where *n* is 30 percent of the number of scenarios. Each person may assign their *n* votes however they please: all for one scenario, one each to *n* scenarios, or any combination.

  [ ] Go around the room and have each person publicly assign *one half* of their votes. Then go around the room in the opposite direction and have each person publicly assign the other half of their votes. (This prevents anyone from having undue influence on the voting merely by accident of their seating location.)

  [ ] Tally votes in front of the users.

[ ] Use any naturally occurring break in the tallies to separate the high-priority scenarios from the lower ones. Only the high-priority ones are considered in future evaluation steps.

[ ] Allow anyone in the group to make impassioned pleas to include a scenario that would otherwise be excluded.

[ ] Exercise discretion to add scenarios that have not been voted "above the line," such as exploratory scenarios.

[ ] After prioritization, facilitate assignment of each high-priority scenario to a place in the utility tree. A scenario will already be present, will constitute a new leaf, or will constitute a new branch. If a whole new branch, have scribe record a possible risk that the relevant quality attribute was not considered by the architect.

[ ] Questioners make sure that scenarios represent the desired mix of quality attributes and/or stakeholder roles.

Outputs

[ ] List of high-priority scenarios.

[ ] List of remaining scenarios.

[ ] Augmented utility tree.

[ ] List of risks, if any, arising from mismatch between high-priority scenarios and utility tree.

Step Description

Utility trees provide a top-down mechanism for generating scenarios from business drivers and quality attributes. Scenario brainstorming, on the other hand, represents a bottom-up approach that lets stakeholders voice concerns connected to their roles outside the confines of any compartmentalized quality attributes. The two approaches in concert make it extremely likely that the relevant quality attribute requirements will be expressed to the evaluation team as a basis for analysis.

The scenario elicitation process allows stakeholders to contribute scenarios that reflect their concerns and understanding of how the architecture will accommodate their needs. Scenarios are collected by a round-robin brainstorming activity. In addition to proposing new scenarios, stakeholders can, if they wish, choose unanalyzed scenarios from the utility tree to put into the brainstorm pool. This can happen if a stakeholder feels that such a scenario is due more attention than it received during Phase 1. In fact, this step is the stakeholders' primary opportunity to recast the coverage and priority of quality attributes as embodied by the utility tree.

A particular scenario may, in fact, have implications for many stakeholders: for a modification, one stakeholder may be concerned with the difficulty of a change and its performance impact, while another may be interested in how the change will affect the integrability of the architecture.

Scenario Brainstorming The evaluation leader asks the stakeholders to propose scenarios. A good facilitation technique, to make sure everyone participates, is to go around the table (or the room) asking each stakeholder in turn to submit a scenario. After a round or two like this, the evaluation leader can then open up the floor to submissions from anyone in any order.

Members of the evaluation team should feel free to submit scenarios. In fact, this is part of their role duties, especially the questioners. The evaluation team by now should have some insights into the architectural strengths and weaknesses and the quality attribute requirements that are important to the stakeholder community. Posing scenarios to the stakeholders may be a way to jog their thinking, and it poses no harm: if the stakeholders do not think the scenario is relevant or important, the upcoming prioritization process will eliminate it.

The scenario scribe plays his or her most important role during scenario brainstorming. The scribe should write down each scenario as expressed by the person who proposed it or as expressed (after clarification and restatement) by the evaluation leader. The scenario scribe should not paraphrase or add anything to the scenario; this is an instance where creativity is not part of the job description. It is important to capture the scenario that the stakeholder(s) proposed, possibly after public clarification by the evaluation leader. The scenario scribe should not be afraid to hold up the proceedings until the wording has been captured exactly.

It is up to the evaluation leader whether the team members propose scenarios along with everyone else or wait until the dam-burst of stakeholder-posed scenarios has subsided. The former enforces the unity of the gathered participants; the latter defers to shy stakeholders who might appreciate being given priority consideration.

The evaluation leader lets the process continue until he or she feels the time has come to call a halt. This may be due to time constraints, but eventually the "momentum" of the crowd will begin to wane. When people have to think very hard in order to propose a scenario, then the scenario will probably be one of the more esoteric ones not likely to be adopted as a high priority anyway; thus, it's a good time to call a halt and move on.

At this point, the stakeholders have produced several dozen scenarios, all of which may potentially be "run" against the architecture. These scenarios are written on flipchart papers and hung on the walls around the exercise room as shown in Figure 6.11. However, the brainstormed scenarios will have the following undesirable properties:

- Many overlap, probing roughly the same issues.
- Some address issues that are unlikely to arise in the system's lifetime.
- Some address issues that are of low priority to the development effort.

Also the list probably contains too many scenarios for evaluation in the time allotted. Hence, the list must be winnowed down. We do this in two steps:

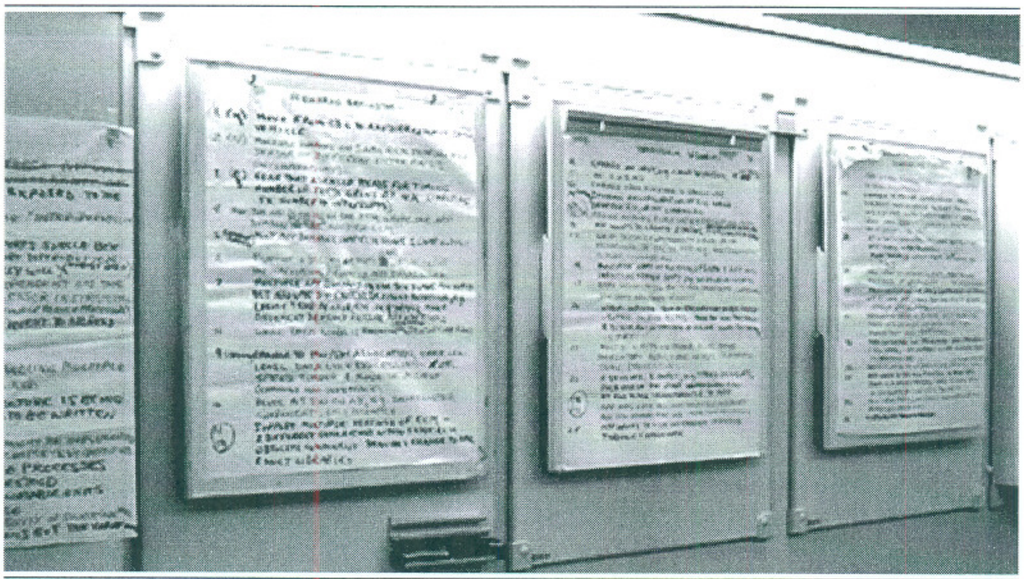


Figure 6.11 How the Room Might Look After Scenario Brainstorming

merging and voting. Participants first will merge almost-alike scenarios and then vote on the ones to adopt for the forthcoming analysis step.

Merging The motivation for merging scenarios, which should be made clear to the participants, is to make participants' votes go further. If people are concerned about an issue raised by two scenarios, they may split their votes among them, effectively eliminating both of them, whereas if the scenarios were merged the resulting single scenario would receive the sum of the votes and have a better chance of inclusion in the subsequent analysis.

To start the merging process, announce that the stakeholders have about 15 minutes to familiarize themselves with the scenarios. They should get up and walk around the room, browsing the flipchart pages. Their goal is to find pairs of scenarios that essentially address the same issue and hence can be merged.

After the time is up, have the participants sit down again, and then ask for proposals to merge pairs of scenarios. The evaluation leader should ask the group for a consensus opinion about each proposal. Barring objection, the proposal is accepted. Members of the evaluation team should also interject objections, if they feel that the two scenarios might actually reveal different aspects of the architecture. If a merger is accepted, then the scenario scribe should cross out one scenario, or modify the wording of one so that it reflects both of the original concerns. Merge conservatively: if anyone objects to a merger proposal, it's usually dropped. The process is rather like the climactic question at a wedding: "If anyone here knows any reason why these two should not be married. . . ."

A variation on this procedure is to have the participants make their proposals while they are walking around and browsing the scenarios. If they see a scenario that they think should be merged with another, have them write the second scenario's number on an adhesive note and post it next to the first scenario. That way, everyone can see all of the proposals as they are being made. The evaluation leader then asks for consensus about each proposal as explained above.

Voting After merging is complete, take the number of remaining scenarios and multiply it by 30 percent. Round that number up to the nearest even integer. This is the number of votes that each participant can cast for scenarios. For example, if there are 55 scenarios after merging, each participant receives 18 votes ($0.30 \times 55 = 16.5$, rounded up to 18).

Write the number of each remaining scenario on the whiteboard, a flipchart, or a blank viewgraph everyone can see. Go around the room and ask each participant to cast half of his or her votes. People can distribute their votes among the scenarios however they like: one vote for each of several scenarios, or all votes for a single scenario, or something in between. When everyone in the room has voted, repeat the process in the opposite direction, having each participant cast the second half of his or her votes. Everyone except the evaluation leader (including other members of the evaluation team) gets to vote.

During the voting process, record the votes on a flipchart or blank viewgraph. This can be done very informally; Figure 6.12 shows an example from a recent evaluation.

When voting is concluded, the highest-rated scenarios are the ones that will be used for the upcoming analysis step. The combined goal for Phase 1 and Phase 2 is to evaluate roughly 12–15 scenarios. For longer evaluations, choose a correspondingly higher number. There will almost always be a natural break point in the vote totals; choose the scenarios above the break.

At this point, we open the floor for what we call the "impassioned plea" round. Participants are allowed to make a compelling argument to add a scenario that was not included as a result of voting. The evaluation leader should be especially sympathetic to suggestions from the client, the architect, or the customer.

The proceedings scribe should record the selected scenarios. If possible, a handout that lists these scenarios should be made for the participants as soon as practical.

Although the voting process we have described may sound somewhat arcane, there are several reasons why we prefer it.

- Voting in two rounds means that people who vote last will not have undue influence on the outcome. Since the voting is public, if everyone cast all their votes at once, people at the end could largely determine which scenarios are included by (1) not wasting their votes on scenarios that already have a high score and (2) clustering their votes on otherwise-unpopular scenarios.
- Public voting contributes to the group dynamic.

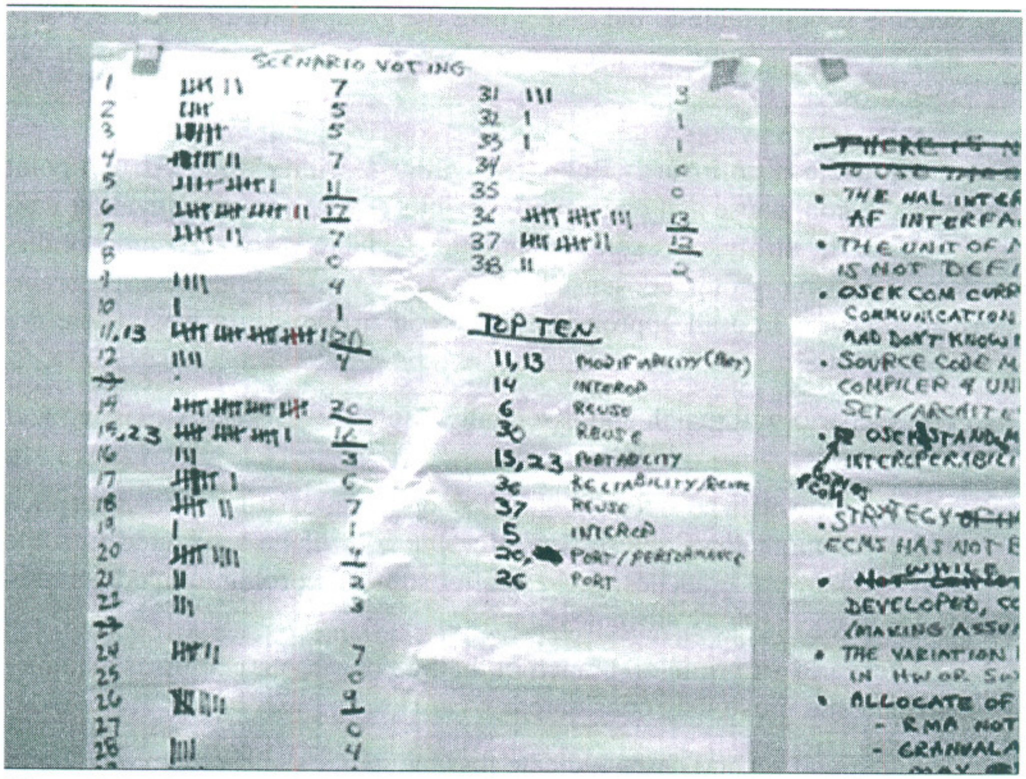


Figure 6.12 Recording Scenario Vote Totals

- It's fun. Participants almost always enjoy this step of the ATAM the most. They enjoy watching dark-horse scenarios emerge triumphant.

Here are some other voting schemes that we have either tried or heard suggested, and why we do not prefer them.

- Secret ballot. Under this scheme, the group is a collection of individuals and there is no sense of group identity.
- One-pass public voting, where everyone casts all their votes at once. As we said, this allows people at the end to influence the outcome. It's also necessary under this scheme to make sure the customer of the system votes last, so that other stakeholders don't vote for scenarios just to play to the customer's desires.
- Nonvoting consensus. The group chats for a while and announces a verdict (or the evaluation leader facilitates a consensus verdict). This is an unreliable way to extract the sense of a large group because it favors the loud and boisterous participants and puts the shy and quiet ones in the worst position possible.
- Show of hands. For each scenario, have the group vote by a show of hands whether that scenario is "high," "medium," or "low" in importance, and then proceed to analyze all of the scenarios that received a "high." This

combined input drives the evaluation from this point forward. Finally, voting and seeing which scenarios come out on top is just plain fun. There's an element of suspense to it, like watching a horse race.

This step is another excellent place for the evaluation leader to stand aside and let a junior member of the team take over to gain experience.

Often the scenarios contributed by the evaluation team end up being adopted, so team members shouldn't feel shy about contributing. Why is that? We're objective. We're focused on scenarios that are going to reveal problems. Also, when the emperor is wearing no clothes (the design is flawed), we're the ones who aren't afraid to point it out by proposing the scenarios that will do just that. We also have a sense by now of where the problems are.

6.5.4 Phase 2, Step 8: Analyze Architectural Approaches

Step Summary

This step reiterates the activities of Step 6 using the highly ranked scenarios from Step 7. Those scenarios are considered to be test cases for the analysis performed so far. The analysis may uncover additional architectural approaches, risks, sensitivity points, and tradeoff points which are then documented.

Input

[ ] List of high-priority scenarios from Step 7.

Activities

This step continues the analysis using the scenarios Step 7 as input. In addition to the activities listed for Phase 1, Step 6, the following activities take place.

[ ] Timekeeper divides available time by the number of scenarios to come up with a time budget for the analysis of each. This is a target time only; it is reasonable that the highest-priority scenarios receive more time and attention. Also, the earlier scenarios tend to take more time to explore.

[ ] Questioners note for each scenario places where architect failed to invoke a style or structure earlier revealed as an approach to achieving the quality attribute addressed by the scenario.

Outputs

[ ] List of architectural components involved in or affected by each scenario and the (potential) effects on system attribute-specific requirements.

[ ] List of scenarios annotated with their (potential) sensitivity points, issues, and attribute-specific tradeoffs. Issues include decisions not yet made.

[ ] List of risks (sensitivity points associated with important functions).

[ ] Summary of architect's explanation of each scenario's mapping, as recorded by proceedings scribe.

Step Description

The analysis performed in this step is exactly the same kind of analysis performed in Step 6 but applied to any new scenarios that resulted from brainstorming.

How It Went

In this section, we recount the analysis of scenario 13 (O15.1), an operability scenario. ("O15.1" is a designator as to where this scenario was placed in the utility tree.) Scenario 13 is one of the scenarios that resulted from scenario brainstorming in the previous step. It is not the highest priority scenario, however, the client chose to analyze this scenario since the group had not yet analyzed an operability scenario.

We asked the architect to walk us through the scenario. He responded by giving the steps for realizing the scenario as follows:

1. Operator creates a table of expected executions by looking at the planning and subscription database, and ingest schedules with historic rates including scanning and generated logs for failures.
2. Operator writes an SQL query against the parameter identification database to determine which algorithm executed last. This involves writing SQL queries to determine which algorithms started but did not complete. It also involves writing an SQL query against the data distribution database to get algorithm data (but operator can't get granule time ranges that are not classified by users).
3. Operator scans application logs to find failures (using the data distribution database) by writing SQL queries against the science data server database and the ingest database for algorithm data and scans application logs for failures.

As a consequence of analyzing this scenario we generated additional risks and sensitivity points, but no new tradeoff points or nonrisks (Table 6.8).

The new risks that were elicited appear below.

R26. The lack of system-wide request tracking leads to a very labor intensive manual process for finding the status of any functional thread. Consequence: status requests require considerable operator intervention.

R27. Inconsistent format and reporting makes it difficult to correlate information across subsystem logs. Consequence: inability to create a system-wide picture of resource requirements for executing scientific algorithms will eventually lead to performance degradation.

R28. Product distribution information is unavailable at the granule level. Consequence: results in the inability to create accurate forecasts of system usage.

Table 6.8 Analyzing Scenario 13 (O15.1)

Scenario #: 13 (O15.1)	Scenario 13 (O15.1): Create a report for the execution of the results of a specific run of a scientific algorithm.			
Attribute	Operability			
Environment	During routine operation			
Stimulus	Request to make a report documenting the results of the execution of a specific scientific algorithm.			
Response	Operator must create report within 2 hours of request.			
Architectural Decisions	**Risk**	**Sensitivity**	**Tradeoff**	**Nonrisk**
Distributed logs	R26, R27	S15, S16, S17		
No granule-level order tracking	R28	S17, S18		
Error and requests in logs are not cross-correlated	R27, R28	S16, S17		
Subsystems don't make available performance throughput information; no integrated throughput information kept	R27, R28, R29	S19, S20		
No system-wide request tracking	R26	S16, S17		
Error information for some subsystems (for example, science data server and subscription management) not in database	R27	S15, S17		
No automated roll up of the throughput data	R27, R28, R29			
Subsystems do not retain historical data on throughput that could be used for generating expectations	R30	S17, S20, S21		
Reasoning	The scenario walk-through follows this table. The walk-through reveals many reasons why it is very cumbersome for an operator to fulfill this scenario.			
Architecture diagram	None			

R29. Lack of system throughput information results in an inability to derive system performance information, especially in real time. Consequence: operators cannot tune system performance in real time.

R30. The lack of historical throughput data makes it very labor intensive to calculate expected performance. Consequence: performance predictions are rarely made and often inaccurate.

The new sensitivity points that were elicited are:

S15. Performance is sensitive to the level of detail captured in logs and to whether logs are stored in files or databases.

S16. Performance is sensitive to the degree to which the various logs are integrated and coordinated.

S17. The level of operator intervention, skill, and effort is directly correlated to the logging approach.

S18. Since products are ordered at the granule level, the ability to track user satisfaction is sensitive to the level of order tracking.

S19. The ability of the operators to manage system load in real time is sensitive to the availability of system throughput data.

S20. System performance is sensitive to the availability of throughput data (since such data is required by the operators to tune the system).

S21. Performance is sensitive to tracking the throughput associated with various types of scientific calculations.

Speaking from Experience

In another ATAM-based evaluation all of the high-priority scenarios turned out to be performance scenarios. However, when it came to choosing a scenario to analyze, almost all of the stakeholders wanted to look at lower priority reliability scenarios. They were already aware of the system performance blemishes; even though the scenarios were very important and very dificult to deal with, this was not new news. In such a case, use your judgment about how to proceed so that you bring the most value to your client.

6.5.5 Phase 2, Step 9: Present Results

Step Summary

Based on the information collected in the ATAM (approaches, scenarios, attribute-specific questions, the utility tree, risks, nonrisks, sensitivity points, tradeoff points), the ATAM team presents the findings to the assembled stakeholders.

Inputs

[ ] Participants' end-of-exercise survey as shown in Figure 10.1.

[ ] List of issues, sensitivity points, and tradeoff points from previous steps.

[ ] Template for presentation of results (Figure 6.13).

Activities

[ ] Data gatherer hands out the end-of-exercise survey.

[ ] Evaluation leader announces a break, during which time:

  [ ] Evaluation team meets to prepare presentation of results.

  [ ] Participants fill out surveys.

[ ] After break, evaluation leader summarizes findings based on publicly recorded lists of risks, sensitivity points, tradeoff points, and other issues. If a final report is part of the statement of work, the evaluation leader announces when it will be submitted.

[ ] Team leader thanks participants for their help and announces the schedule for delivery of the final report.

  [ ] Team leader makes sure evaluation sponsor receives electronic or printed copy of presentation viewgraphs before evaluation team leaves.

[ ] Data gatherer collects participants' end-of-exercise surveys. These are turned over to the team leader.

Outputs

[ ] ATAM participants' survey, filled out by all non-evaluation-team participants.

[ ] ATAM results presentation, in electronic or printed form.

Step Description

In this step, the evaluation team summarizes the findings of the ATAM exercise. After Step 8 is concluded, the evaluation leader announces a break of about 45 minutes to an hour. During this break, the evaluation team meets to prepare the viewgraphs for the final presentation. Figure 6.13 shows the outline of the presentation that we use. The first slide recaps the steps of the method. Subsequent slides summarize the findings.

The presentation, with questions, typically takes between one and two hours.

How It Went

We made a one-hour presentation that summarized the ATAM exercise, the business drivers, the architectural approaches, the utility tree, and the analysis we performed (during Phase 1, the hiatus, and Phase 2). We recapped the risks, nonrisks, sensitivity points, and tradeoff points we discovered. Finally, we presented the five risk themes that we saw running through the individual risks we captured:

1. There was no consistent strategy for controlling and referencing data types and interfaces.
2. There was a single on-line copy of the database, and this resulted in performance and availability compromises.
3. The COTS database imposed performance restrictions.

Outline of Results Presentation	Explanation
ATAM Steps	A viewgraph listing the nine steps of Phase 1 and Phase 2
Business Drivers	A summary of the business drivers given in Step 2 of Phase 1
Architectural approaches	A summary of the architectural approaches indentified in Step 4 of Phase 1
Utility tree	The utility tree built in Step 5 of Phase 1
Scenarios	A summary of the scenarios brainstormed and prioritized during Step 7 of Phase 2
Scenario discussions	A summary of the analysis performed in Step 6 of Phase 1 and Step 8 of Phase 2 (typically 1–2 viewgraphs per scenario subjected to analysis, presented using the Analysis of an Architectural Approach template)
Risks, nonrisks, sensitivity points, tradeoff points	A compilation of the risks, nonrisks, sensitivity points, and tradeoff points found during the exercise
Other issues	A compilation of any other issues uncovered during the evaluation, each listed along with a short explanation of its cause and impact (typically 1 issue per viewgraph)
Summary	Summary in terms of the risk themes identified by the evaluation team after examining the risks as a group; explanation of how each risk theme, if not addressed, will threaten achievement of specific business drivers

Figure 6.13 Outline of Results Presentation

4. The object description language library was slow and could not be (easily) multi-threaded.
5. The system had little support for quiescing.

Then, for each risk theme, we identified the business driver or drivers to which it posed a threat. For example, the risk themes 2, 3, and 4 affected the achievement of one of the primary business drivers: concurrent ingest, production, and distribution of high data volumes.

Speaking from Experience

The fact that the evaluation team can prepare a comprehensive one-hour summary of an in-depth architecture evaluation during a 45-minute break is a testament to the rigor of the ATAM steps. It is also one of the aspects of the ATAM that makes it popular with clients: results are available immediately at the end of the exercise; no waiting is necessary.

This step is the best illustration of the maxim that preparation pays off. Look at how much up-front preparation has come home to roost:

- When the evaluation team was formed in Phase 0, segments of the presentation were assigned to individual team members.
- As much of the presentation as possible was drafted in Step 0 of Phase 2.
- The business drivers and architectural approaches were summarized and captured publicly.
- The analysis of each scenario was carried out publicly, and the results were recorded in a template.
- The risks, nonrisks, sensitivity points, and tradeoff points were publicly recorded and displayed throughout the exercise.

The fact that almost all of the information was publicly recorded before the results presentation yields two happy circumstances. First, the proceedings scribe can easily turn the electronic record into viewgraphs with a few deft cut-and-paste operations. Second, the information in the summary comes as a surprise to no one. We have never experienced an evaluation using the ATAM where someone was upset at the results, because the results (by the time of the presentation) are not new. The reiteration of previously derived results also reinforces the repeatable, methodical nature of the method, to the pleasure of the participants.

Finally, mapping the risk themes to the business drivers they affect produces two positive effects. First, it gives the impression of closing the circle. Although the business drivers help us to establish context for the evaluation in general and begin to nail down the utility tree in particular, participants sometimes cannot remember exactly why they were so important. Mapping the risks back to the business drivers instills the drivers with an importance they might not otherwise have. Second, it puts the ATAM results in terms that are unmistakably significant to the client because they are couched in terms that he or she is used to discussing.

6.6 Phase 3: Follow-Up

The steps of the follow-up phase are:

1. Produce the final report
2. Hold the postmortem meeting
3. Build the portfolio and update artifact repositories

6.6.1 Phase 3, Step 1: Produce the Final Report

Step Summary

Inputs

[ ] Template for final report (see Figure 6.14).

[ ] Outputs from Phase 1 and Phase 2.

Activities

Using a template for the final report:

[ ] Write what you did: method summary, who was present, scenarios recorded.

[ ] Write what you saw: architecture summary, analysis summary, results of applying scenarios.

[ ] Write what you concluded: draw analytical conclusions, identify risks, make recommendations.

[ ] Have entire team review report.

[ ] Send report to client for review, to correct factual mistakes.

[ ] Revise report based on client's feedback.

[ ] Transmit final report.

Output

[ ] Final report.

Step Description

After Phase 2 is over and the evaluation team has recovered from the intensity of the exercise, the team leader convenes a meeting (or sends e-mail) to remind the team members of their individual responsibilities in producing the final written report for the client. An outline of the final report template that we use is given in Figure 6.14.

We strive to produce the final report, have all team members review it, and transmit it to the client within ten working days of the end of Phase 2.

How It Went

The final report for the NASA ECS ATAM exercise was 48 pages in length and followed the template shown in Figure 6.14.

Speaking from Experience

Once again, preparation pays off. Sections 1, 2, and 3 of our final report template are fairly pro forma and do not change from exercise to exercise. Section 4, a brief narrative about the particular evaluation, contains blanks for the time and place of the exercise, the name and nature of the system whose architecture was evaluated, and the names and contact information of all the participants. Sections 5–10 are narrative forms of the information from the presentation of results produced at the end of Phase 2, and the production of these sections is

Executive Summary	
Section 1	Introduction
Section 2	Evaluating a Software Architecture
Section 3	ATAM Overview
Section 4	The ATAM for (name of system)
Section 5	Summary of Business Drivers
Section 6	Summary of Architecture Presentation
Section 7	Quality Attribute Utility Tree
Section 8	Scenario Generation, Consolidation, and Prioritization
Section 9	Analysis of Architectural Approaches
Section 10	Risks, Sensitivities, Tradeoffs, Nonrisks, and Other Issues
Section 11	Conclusions

Figure 6.14 Sample Outline for an ATAM Final Report

straightforward. Section 11 summarizes the risk themes and relates them to the business drivers they impact.

6.6.2 Phase 3, Step 2: Hold the Postmortem Meeting

Step Summary

Hold a meeting to gather surveys and data, look for improvements to be made to the method or its artifacts, and assess the overall quality of the exercise.

Inputs

- [ ] Evaluation team post-exercise improvement survey.
- [ ] Evaluation team post-exercise effort survey.
- [ ] Participants' end-of-exercise survey (see Figure 10.1).
- [ ] Sample agenda for postmortem meeting (see Figure 6.15).
- [ ] Process observer's report.

Activities

- [ ] Team members:
    - [ ] Fill out evaluation team post-exercise improvement survey.
    - [ ] Fill out evaluation team post-exercise effort survey.
- [ ] Team leader:
    - [ ] Collect process observer's report.

[ ] Collect non-evaluation-team participants' effort data. Most will have submitted this via the participants' end-of-exercise survey. Contact the client to receive effort data for anyone who participated but did not fill out that form.

[ ] Collect evaluation team post-exercise improvement surveys.

[ ] Collect evaluation team post-exercise effort surveys.

[ ] Schedule and hold a postmortem meeting with the evaluation group to discuss the lessons learned from the exercise and record ideas for improvement. Bring the surveys for circulation, or distribute them prior to the meeting.

[ ] Summarize changes necessary to reflect improvements.

[ ] Update the ATAM process model.

Outputs

[ ] Completed surveys and cost data.

[ ] Improvement ideas.

[ ] Updated ATAM process model.

Step Description

The ATAM prescribes the use of three after-action surveys (plus another that will be discussed in the next step):

- Participants' end-of-exercise survey (handed out at the end of Phase 2)
- Evaluation team post-exercise improvement survey
- Evaluation team post-exercise effort survey

The first three gauge the effort and impressions of all the participants, whether a member of the evaluation team or a representative of the client or project organization.

In addition, the process observer should submit his or her written notes, which should contain areas where the process deviated from what is prescribed.

The purpose of all of this data and information is to help the ATAM process become more effective and more repeatable. A deviation from the prescribed process can be handled in one of three ways:

1. Recognize that the circumstances were unique and the deviation was warranted.
2. Recognize that the deviation was undesirable and take steps to avoid it in the future.
3. Recognize that the deviation represented a situation likely to occur again, and change the ATAM steps to accommodate it.

A sample agenda for the postmortem meeting is given in Figure 6.15.

Duration	Activity
30 minutes	Evaluation team members' quick impressions of process, and estimation of value for client
30 minutes	Review of • Process observer's report • Evaluation team post-exercise improvement survey • Participants' end-of-exercise survey
15 minutes	Summary of total effort required for the exercise, using • Evaluation team post-exercise effort survey • Effort data from the participants' end-of-exercise survey
30 minutes	Discussion of specific suggested changes to ATAM process and materials; assignment of action items

Figure 6.15 Sample Agenda for ATAM Postmortem Meeting

How It Went

We held a postmortem meeting three days after we transmitted the final report to the client. The ECS evaluation required 128 person-hours on the part of the evaluation team and 154 person-hours on the part of the project organization.

For this evaluation, all members of the evaluation team served as process observers. We found no major deviations but spent our time discussing which steps we thought the client found most valuable and why.

Participants reported highest satisfaction with the utility tree part of the exercise. They felt that overall the exercise was a valuable use of their time; in fact, many expressed pleasure at having the quality attribute goals explicitly articulated and prioritized.

Some of the participants would have liked more read-ahead material about the method.

Speaking from Experience

The postmortem meeting is a hard meeting to hold because everyone is tired and the final report is finished. But it's the best way we know to bring home the lessons of each ATAM experience to our architecture evaluation group as a whole. Our entire group attends each postmortem meeting, not just the evaluation team that was involved. Attending a postmortem is an excellent way for junior members of the group, or people being trained for ATAM-based evaluations, to hear real war stories and begin to gain a sense for what an actual evaluation is like.

Resist the temptation to change the ATAM after every experience using it. We will be the first to admit that there is a big difference between writing and speaking about the method and actually carrying it out—you will find places to tailor and improve the method as you gain experience with it. However, when something does not work out as planned, the cure more often lies in more experience rather than reflexively making changes that may never converge.

6.6.3 Phase 3, Step 3: Build Portfolio and Update Artifact Repositories

Step Summary

Build up the portfolio of artifacts generated during the ATAM exercise. Record survey and effort data. Update the list of scenarios and analysis questions.

Inputs

- [ ] Utility tree, scenarios, and analysis questions used during the evaluation.
- [ ] Survey and cost outputs from the postmortem meeting.
- [ ] Presentation of results from Step 9 of Phase 2.
- [ ] Final report.
- [ ] Process observer's report.
- [ ] Survey of client long-term benefit.

Activities

There may be a designated role in the evaluation organization to build the exercise portfolio and update the artifact repositories. Otherwise, the responsibility falls to the team leader.

- [ ] Team leader establishes the evaluation portfolio for the evaluation just completed, including
  - [ ] Participant evaluations
  - [ ] Team evaluations
  - [ ] Process observer's report
  - [ ] Copy of presentation of results
  - [ ] Copy of final report
  - [ ] Long-term benefits survey
- [ ] Team leader adds scenarios to the scenario repository.
- [ ] Six months after evaluation, team leader sends to client a survey of long-term benefits.

Outputs

- [ ] Updated repositories.
- [ ] ATAM end-of-exercise portfolio.

Step Description

You should maintain repositories of the artifacts you used or produced during each previous evaluation. These will serve you during future evaluations.

In addition to recording the cost and benefit information, store the scenarios that you produced. If future systems that you evaluate are similar in nature, then you will probably find that the scenarios that express the architecture's requirements will converge into a uniform set. This gives you a powerful opportunity to streamline the evaluation method: you can dispense with the scenario brainstorming and prioritization steps of the ATAM altogether and simply use the standard scenario set for your domain. The scenarios have in some sense graduated to become a checklist, and checklists are extremely useful because each architect in the developing organization can keep a copy of the checklist in his or her desk drawer and make sure the architecture passes with respect to it. Then an evaluation becomes more of a confirmation exercise than an investigatory one. Stakeholders' involvement becomes minimal—as long as you have confidence in the applicability and completeness of the checklist with respect to the new system under evaluation—thus reducing the expense of the evaluation still further.

Besides the scenarios, add the analysis questions and the participants' comments to a repository as well. Future evaluation leaders can read through these and gain valuable insights into the details and idiosyncrasies of past evaluations. These exercise summaries provide excellent training material for new evaluation leaders.

Finally, keep a copy of the final report, sanitized if necessary to avoid identifying the system or spreading any incriminating remarks. Future evaluation teams will appreciate having a template to use for the reports they produce.

How It Went

Six months after the ATAM exercise, we sent the ECS program manager our standard long-term improvement survey, which asks what the long-term effects of the ATAM exercise were.[3] Here's what he had to say.

Before the ATAM:

- *Justifications for key architectural decisions were not well documented; knowledge limited to a few system architects.*
- *Little analysis on risks, sensitivities, and tradeoffs associated with architectural decisions, especially for ongoing maintenance changes.*
- *Stakeholders other than development staff viewed all requested changes as largely independent and equally plausible.*

As a result of our first ATAM experience:

- *Operations, customer, architect, and development stakeholders reached a common understanding of how the system addresses the 45 quality attribute requirements addressed by this review.*

3. A sample of such a survey is given in Chapter 10.

- *Stakeholder team identified and documented ~50 key architectural decisions and more than 100 associated risks, sensitivities, and tradeoffs.*
- *Based on these results and the prioritized quality attribute requirements, the stakeholder team was able to identify and compare the technical merits of more than 60 proposed architectural changes.*

We anticipate greater stakeholder satisfaction given buy-in to tradeoffs.

Speaking from Experience

In our organization, the task of building the exercise portfolio falls to a conscientious support person who is responsible for obtaining the artifacts from the team leader and for reminding the team leader when it is time to issue the long-term improvement survey.

6.7 For Further Reading

To learn more about the Earth Observing System, visit NASA's Web site at http://eospso.gsfc.nasa.gov/. Terra is the name of one of the major satellites in the EOS fleet, and http://terra.nasa.gov contains excellent information about and images from that satellite.

6.8 Discussion Questions

1. Try building a utility tree for a system that you are familiar with. Are any of the attributes discussed in this chapter relevant to your system? Do the attributes have the same meaning for your system as they do for the ECS system? How do you know whether they do or not?
2. Choose a couple of the quality attributes from the utility tree you created for the previous question. What are the architectural approaches used in your system to achieve those quality attribute requirements?
3. Pretend that you are leading the evaluation that we discussed in this chapter and are responsible for constructing the final presentation. In particular, you are thinking about the last few slides that show how the risk themes impact the business drivers. What would you say on these final slides of the presentation?

4. The NASA client for the ECS evaluation is planning to incorporate the ATAM as part of his office's software maintenance and evolution process. How would you incorporate the ATAM into your organization's development and maintenance process?

5. One of the benefits the client perceived from this ATAM exercise was the chance to achieve consensus among the stakeholders about which system enhancements to tackle first. Suppose that was your only goal. Propose a new method, called Stakeholder Consensus Realization Analysis Method (SCRAM), and define its steps, phases, participants, and artifacts produced.