ATAM

Architecture Trade-off Analysis Method with case study

Bart Venckeleer, inno.com
SEI – Software Architecture Tools and Methods

- Active Reviews for Intermediate Designs (ARID)
- Architecture-Based System Evolution
- Architecture Competence Assessment
- Architecture Expert (ArchE)
- Architecture Tradeoff Analysis Method (ATAM)
- Attribute-Driven Design (ADD) Method
- Cost Benefit Analysis Method (CBAM)
- Mission Thread Workshop
- Quality Attribute Workshop (QAW)
- Views and Beyond Approach to Architecture Documentation
ATAM in relationship with other SEI methods

- Quality Attribute Workshop (QAW)
- Active Reviews for Intermediate Designs (ARID)
- Cost Benefit Analysis Method (CBAM)
- Views and Beyond Approach to Architecture Documentation
- Architecture Competence Assessment

clarified quality attribute requirements
Why Architectural Analysis?

- The earlier you find a problem in a software project, the better off you are.
- An unsuitable architecture will bring disaster on a project.
- Architecture evaluation is a cheap way to avoid disaster.
The ATAM is based upon a set of attribute-specific measures of the system:
- Analytic (performance & availability)
- Qualitative (modifiability, safety, security)

The ATAM workshops typically take three days and involve 10-20 people:
- Evaluators
- Architects
- and other system stakeholders

Benefits:
- clarified quality attribute requirements
- improved architecture documentation
- documented basis for architectural decisions
- identified risks early in the life-cycle
- increased communication among stakeholders

An architecture evaluation doesn’t tell you “yes” or “no” or “6,75 out of 10”. It tells you where the risks are.
Risks, Nonrisks, Sensitivity Points and Trade-off Points

Risks are potentially problematic architectural decisions.

Nonrisks are good decisions that rely on assumptions that are frequently implicit in the architecture.

A sensitivity point is a property of one or more components (and/or component relationships) that is critical for achieving a particular quality attribute response.

A trade-off point is a property that affects more than one attribute and is a sensitivity point for more than one attribute.

The rules for writing business logic modules in the second tier of your three-tier client-server style are not clearly articulated. This could result in replication of functionality, thereby compromising modifiability of the third tier (a quality attribute response and its consequences).

Assuming message arrival rates of once per second, a processing time of less than 30 milliseconds, and the existence of one higher priority process, then a one-second soft deadline seems reasonable.

The latency for processing an important message might be sensitive to the priority of the lowest priority process involved in handling the message. The average number of person-days of effort it takes to maintain a system might be sensitive to the degree of encapsulation of its communication protocols and file formats.

If the processing of a confidential message has a hard real-time latency requirement then the level of encryption could be a trade-off point.
Architecture Trade-off Analysis Method

- Preparation (elapsed time 20d)
  - Business track
    - Interview project leader
    - Interview business representatives
    - Prepare quality attribute tree
    - Prepare scenario brainstorm
  - Architecture track
    - Interview architect
    - Interview lead developers
    - Prepare example architecture documentation
    - Identify approaches
- Analysis
What Result Does an Architecture Evaluation Produce?

- Answers to two kind of questions:
  - Is the architecture **suitable** for the system for which it was designed?
  - Which of two or more competing architectures is the most **suitable** one for the system at hand?

It meets its quality goals
- Predictable behaviour
- Performance
- Modifiable
- Security

The system can be built
- Staff
- Budget
- Legacy
- Time

*If the sponsor of a system cannot tell you what any of the quality goals are for the system, then the architecture will do.*
Summary of the ATAM Steps

- **Presentation**
  - Present the method
    - Evaluation leader
  - Present business drivers
    - 12 slides; 45 min; guidelines for content; project manager
  - Present architecture
    - 20 slides; 60 min; guidelines for content + checklist with questions; architect

- **Investigation and Analysis**
  - Identify the architectural approaches
    - architect
  - Generate the quality attribute utility tree
    - Workshop, templates for utility tree and scenarios
  - Analyse the architectural approaches
    - workshop, templates for risks, nonrisks, sensitivity points and trade-off points

- **Testing**
  - Brainstorm and prioritise scenarios
    - Voting workshop
  - Analyse the architectural scenario’s
    - See step 6

- **Reporting**
  - Present the results
    - Document template; evaluation team
CASE Study: Purchase2Pay
Step 2: Present Business Drivers

- **Customer presents**
  - System overview
  - Business goals
  - Constraints
  - Quality attributes of interest

- **Evaluation team**
  - Define scope
  - Interactions between systems
  - Capture stakeholder interests
  - Pinpoint important quality attributes
  - Identify business goals and constraints

- **Evaluation leader**
  - Public summary

1. List of business goals
2. List of quality attributes of interest
3. Preliminary list of stakeholder roles for phase 2
4. First iteration on scope definition
Purchase2Pay: Business Context and Drivers

- Business environment
  - Invoice documents that are received from business units worldwide are centralized in one geographical location: Bratislava.

  - Bratislava is responsible for
    - Labeling invoices
    - Scanning invoices
    - Posting the invoices to SAP

  - Invoice verification is performed by business workers in Europe
Purchase2Pay: Business Context and Drivers

➢ Driving requirements

- Every invoice must get a unique identification
- Every invoice must be stored electronically in a central repository
- Every invoice must be submitted to a verification process (review and approval)
- Posters must make use of the electronic version of the invoice
- Reviewers and approvers must make use of the electronic version of the invoice
- The lead time from invoice reception to invoice approval is less than one week
Purchase2Pay: Business Context and Drivers

- Major stakeholders and users
  - Stakeholders
    - Procurement
    - Finance
    - Legal
  - Users
    - Scanner
    - Poster
    - Reviewer
    - Approver
Purchase2Pay: Business Constraints

- Time to market
  - Very fast

- User demands
  - Easy to use
  - Fast

- Standards
  - SAP
  - Documentum

- Cost
  - To be minimized
Purchase2Pay: Technical Constraints

- **COTS**
  - SAP
  - Documentum

- **Integration requirements**
  - SAP – Documentum

- **Hardware requirements**
  - Run on current hardware and network infrastructure
Purchase2Pay: Quality Attribute Requirements

- **Usability**
  - E-documents can easily be read when displayed on screen

- **Availability**
  - Very High availability during Bratislava business hours
  - High availability 24/7 for other business workers

- **Performance**
  - **Throughput**
    - Infrastructure must allow to process at least 400 invoices per hour ➔ every 9 sec a request for a document upload is submitted; there will be a download every 3 sec!
  - **UI responsiveness**
    - average 5 s
    - worse case 10 s
Step 3: Present Architecture

- Architect describes
  - Technical constraints
  - Other systems involved
  - Attribute specific architectural approaches

- Evaluation team notes
  - Architectural approaches used/mentioned
  - Potential risks towards drivers
  - Additional stakeholders mentioned

1. Summary of architecture presentation
2. Updated list of stakeholder roles for phase 2
3. Second iteration on scope definition
Purchase2Pay: Driving Architectural Requirements

- COTS
  - SAP
  - Documentum

- Cost
  - SAP client license cost

- Load
  - Bratislava
    - Avg document upload and download: 1 doc every 9 sec each
  - Europe
    - Avg document download: 1 doc every 3 sec
Purchase2Pay: High-level Architectural View
Purchase2Pay: Trace Use case Scenario
Purchase2Pay: Trace Use case Scenario

: Actors::Invoice Poster

: SIWorkItem
  - display()
  - selectWorkItem()
  - openWorkItem()

: SIViewWorklist
  - display()

: IDocumentumHttp
  - get()

: IAcrobatViewer
  - display()

- update()
- close()
Step 4: Identify the Architectural Approaches

- **Evaluation team**
  - Interviews the architect to identify major approaches used
  - ‘Tour de table’ polling for approaches identified

- **The architect**
  - Validates the material gathered

1. List of approaches recorded by scenario
Architectural Approaches

- **Client Server**
  - Sap client – Sap server
  - Documentum client – Documentum Server

- **Multistep process integration**
  - Posting tool spawns acrobat viewer
  - Verification tool spawns acrobat viewer

- **Data consistency integration**
  - eConnector creates invoice objects in SAP based on documents posted to Documentum
Step 5: Generate the Quality Attribute Utility Tree

- Evaluation leader facilitates
  - Identification
  - Prioritization
  - Refinement to scenarios
  Of most important quality attributes.

- Questioners are responsible
  - To highlight important quality attributes
  - Point out difference between generated and presented drivers
  - Listen for additional stakeholders mentioned

<table>
<thead>
<tr>
<th>Utility</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Data latency</td>
</tr>
<tr>
<td></td>
<td>- (M, L) Minimize storage latency on customer DB to 200 ms</td>
</tr>
<tr>
<td></td>
<td>- (H,M) Deliver video in real time</td>
</tr>
<tr>
<td></td>
<td>Transaction throughput</td>
</tr>
<tr>
<td></td>
<td>- (M,M) Maximize average throughput to authentication server</td>
</tr>
</tbody>
</table>

- Modifiability
  - New Product Categories
  - Change COTS
    - (H,L) change web user interface in < 4 person weeks

- Availability
  - Hardware Failure
    - (L,M) power output at site 1 requires traffic redirect to site 3 in < 3 s
    - (H,M) network failure is detected and recovered in < 1,5 min

- Security
  - Data confidentiality
    - (L,H) customer database authorisation works 99,999% of time
## Purchase2Pay: Quality Attribute Tree

<table>
<thead>
<tr>
<th>QA-L1</th>
<th>QA-L2</th>
<th>BP</th>
<th>TP</th>
<th>Scenario</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance</td>
<td>Latency</td>
<td>H</td>
<td>H</td>
<td>Opening a e-invoice for reading takes less than 3 seconds from any site that is in scope of p2p</td>
</tr>
<tr>
<td>Performance</td>
<td>Throughput</td>
<td>H</td>
<td>H</td>
<td>Opening documents at a continuous rate of 2 documents per second has average response time better than 3 sec per doc for any of the sites in scope of p2p</td>
</tr>
<tr>
<td>Availability</td>
<td>Overall</td>
<td>H</td>
<td>H</td>
<td>A site that is disconnected due to network failure is re-connected with full bandwidth in less than 2 hours</td>
</tr>
<tr>
<td>Availability</td>
<td>Overall</td>
<td>H</td>
<td>M</td>
<td>Hardware failure of one CPU in the infrastructure components (SAP, Documentum) has no effect on realization of QA</td>
</tr>
<tr>
<td>Availability</td>
<td>Overall</td>
<td>H</td>
<td>H</td>
<td>There will be no more than 4 unavailability situations per year</td>
</tr>
</tbody>
</table>
Step 6: Analyse the Architectural Approaches

- Scenario: Detect and recover from HW failure of prim CPU
- Attribute(s): Availability
- Environment: Normal operations
- Stimulus: One CPU fails
- Response: 99,999% availability

Architectural decisions

<table>
<thead>
<tr>
<th>Arch. decision</th>
<th>Sensitivity</th>
<th>Trade-off</th>
<th>Risk</th>
<th>Nonrisk</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backup CPUs</td>
<td>S2</td>
<td></td>
<td>R8</td>
<td></td>
</tr>
<tr>
<td>No backup channel</td>
<td>S3</td>
<td>T3</td>
<td>R9</td>
<td></td>
</tr>
<tr>
<td>Watchdog</td>
<td>S4</td>
<td></td>
<td></td>
<td>N12</td>
</tr>
<tr>
<td>Heartbeat</td>
<td>S5</td>
<td></td>
<td></td>
<td>N13</td>
</tr>
<tr>
<td>Failover routing</td>
<td>S6</td>
<td></td>
<td></td>
<td>N14</td>
</tr>
</tbody>
</table>
Step 7: Brainstorm and Prioritise Scenarios

- Workshop to generate
  - Use case scenario’s
    - Expectations on usage of the system
  - Growth scenario’s
    - Expectations on growth of the system
  - Exploratory scenario’s
    - Expectations on huge change

- Workshop to merge prioritize
  - Merge similar scenarios
  - #Votes = 30% of scenarios rounded up to even number
  - 55 scenario’s → 18 votes

1. List of high priority scenario’s
2. List of remaining scenario's
3. Augmented utility tree
4. List of risks, arising from mismatch between high-priority scenario’s and utility tree
Step 8: Analyse the scenario’s

- See step 6

<table>
<thead>
<tr>
<th>Arch. decision</th>
<th>Sensitivity</th>
<th>trade-off</th>
<th>Risk</th>
<th>Nonrisk</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backup CPUs</td>
<td>S2</td>
<td></td>
<td>R8</td>
<td></td>
</tr>
<tr>
<td>No backup channel</td>
<td>S3</td>
<td>T3</td>
<td>R9</td>
<td></td>
</tr>
<tr>
<td>Watchdog</td>
<td>S4</td>
<td></td>
<td></td>
<td>N12</td>
</tr>
<tr>
<td>Heartbeat</td>
<td>S5</td>
<td></td>
<td></td>
<td>N13</td>
</tr>
<tr>
<td>Failover routing</td>
<td>S6</td>
<td></td>
<td></td>
<td>N14</td>
</tr>
<tr>
<td>Scenario:</td>
<td>E-connector looses connection with SAP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------</td>
<td>-------------------------------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Attribute:</td>
<td>Availability-Overall Availability</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stimulus:</td>
<td>Temporary network fault</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Response:</td>
<td>The system has an overall availability of 99.25% (max 2 hour down/ month)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Architectural decision</th>
<th>Sensitivity</th>
<th>Trade-off</th>
<th>Risk</th>
<th>Non-risk</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCTM content server runs on a clustered environment with 2 nodes</td>
<td>Common mode failure can not be handled</td>
<td></td>
<td></td>
<td>Probability of common mode failure is low</td>
</tr>
<tr>
<td>Integration relationship between SAP and Documentum is 'data consistency' and is not protected</td>
<td>Human user must report malfunction</td>
<td></td>
<td>From complaint to resolution &gt; 2 hours</td>
<td></td>
</tr>
</tbody>
</table>
### Scenario:
Invoice poster needs e-document for data entry in SAP/R3

### Attribute:
Performance-Latency

### Stimulus:
Document request to Documentum

### Response:
Document is available for processing in less than 10 s

<table>
<thead>
<tr>
<th>Architectural decision</th>
<th>Sensitivity</th>
<th>Trade-off</th>
<th>Risk</th>
<th>Non-risk</th>
</tr>
</thead>
<tbody>
<tr>
<td>E-documents are scanned in color at 200 dpi</td>
<td>Size of document is sensitive to quality of scanning</td>
<td>Usability vs Performance</td>
<td>Document too large for roundtrip in 10 s.</td>
<td></td>
</tr>
<tr>
<td>E-documents are not cached</td>
<td>Every document must be fetched from DMTM</td>
<td>Development cost vs bandwidth cost</td>
<td>Document roundtrip time exceeds 10 s.</td>
<td></td>
</tr>
</tbody>
</table>
Step 9: Present Results

1. Introduction
2. Evaluating a Software Architecture
3. ATAM overview
4. The ATAM for <system name>
5. Summary of Business Drivers
6. Summary of Architecture Presentation
7. Quality Attribute Utility Tree
8. Scenario Generation, Consolidation, and Prioritisation
9. Analysis of Architectural Approaches
10. Risks, Sensitivities, trade-offs, Nonrisks, and Other Issues
11. Conclusions
What Are the Benefits of Performing an Architectural Evaluation?

- Puts stakeholders in the same room
- Forces an articulation of specific quality goals
- Results in prioritization of conflicting goals
- Forces a clear explicitation of the architecture
- Improves the quality of architectural documentation
- Uncovers opportunities for cross project reuse
- Results in improved architectural practices
Top 3 problematic decisions

➢ Performance (latency)
  • One view
    ▪ Many small sql statements
    ▪ Each query optimized
    ▪ $400 \times 30 \text{ ms} = 12 \text{ sec}$

➢ Modifiability (evolutive maintenance)
  • Monolithic applications
  • Organization of components
  • No test infrastructure to accommodate major refactoring

➢ Security
  • Input validation
  • Functional credentials
  • User management
Inno.com experience

- ATAM assessments are too often executed when it becomes clear that the project cannot be delivered
  - Positive example seen at Alcatel
  - Financial sector started to adopt ATAM recently

- In general projects spend little effort during the requirements workflow to specify and analyze quality attribute requirements

- The quality of a software system depends largely on the skills and experience of individual developers

- The preparation of a quality attribute workshops is extremely important but also difficult and time consuming
  - Brainstorm relevant example quality attribute scenario’s
  - Be prepared to educate stakeholders while moving forward

- Architectural documentation is sparse and often inaccurate