

**Architecture Tradeoff Analysis Method**  
**ATAM (Example)**

Enterim LLC

Drumm, Paul M.

## Change History

Date	Summary	Person
4/16/07	Document created	P.Drumm
4/18/07	Started Phase 0	P.Drumm
4/18/07	Outlined steps further	P.Drumm
4/19/07	Added to Phase 0	P.Drumm
4/20/07	Added to Phase 1 step 2	P.Drumm
4/20/07	Added to Phase 0	P.Drumm
4/21/07	Added to Phase 0 and Phase 1 step 1	P.Drumm
4/22/07	Finished Phase 1 step 1 and added to phase 1 step 2	P.Drumm
4/22/07	Added Phase 1 Step 3 & Step 4	P.Drumm
4/22/07	Phase 0 preparation revised	P.Drumm
4/22/07	Added Phase 1 step 5 and 6, all contributed to step 5.	P.Drumm
4/22/07	Contributed to the Performance and Testability scenarios in Phase 1 Step 5	P.Drumm
4/22/07	Contributed to the Availability and Security Scenarios in Step 5	P.Drumm
4/28/07	Continued on Step 6 with 4 analysis scenarios	P.Drumm
4/29/07	Continued on Step 6 with 1 analysis scenario	P.Drumm
4/30/07	Modified layout, added references.	P.Drumm
4/30/07	Continued with step 7	P.Drumm
5/1/07	Added Phase 2 Step 8	P.Drumm
5/1/07	Part 2; Edits made to some sections.	P.Drumm
5/2/07	Added Phase 3 Steps 1-3	P.Drumm
5/4/07	Added Phase 2 Step 9	P.Drumm

5/5/07	Added Visio Diagrams to Phase 1 Step 3 and Part 2	P.Drumm
--------	---	---------

# Table of Contents

Part 1: ATAM of HW1 .....	1
Partnership and Preparation.....	1
Step 1: Present the ATAM.....	1
Step 2: Describe the candidate system.....	2
Step 3: Make a go/no-go decision.....	2
Step 4: Negotiate the statement of work.....	2
Step 5: Form the core evaluation team .....	3
Step 6: Hold Evaluation Team Kick-off Meeting.....	4
Step 7: Making necessary preparations for Phase 1.....	4
Step 8: Review the Architecture .....	5
Phase 1: Evaluation.....	6
Step 1: Present ATAM .....	6
Step 2: Present the Business Drivers .....	6
Step 3: Present the Architecture .....	8
Step 4: Identify and Catalog Architectural Approaches.....	11
Step 5: Generate Quality Attribute Tree.....	11
Step 6: Analyze Architectural Approaches .....	14
Phase 2: Evaluation (Continued) .....	22
Step 7: Brainstorm and Prioritize Scenarios.....	22
Step 8: Analyze Architectural Approaches .....	23
Availability: Hardware Errors.....	24
Modifiability: New Performance Measures.....	24
Step 9: Present Results .....	25
Phase 3: Follow-up .....	26
Step 1: Produce Final Report.....	26
Step 2: Post Mortem Meeting - Collect Data for Measurement and Process Improvement.....	27
Step 3: Build Portfolio and Update the Artifact Repositories .....	28
Part 2: ATAM findings compared to HW1 Architecture.....	28
Appendix A: References .....	1

## **Part 1: ATAM of HW1**

### ***Partnership and Preparation***

The Ubiquitous Education System (UES) is a comprehensive online educational system that will provide students, staff, faculty and others with access to the tools and functionalities required by CSUF, presented in detail in the software requirements specification document.

Towards the end of the project's architectural design the representative CSUF project manager raised some concerns pertaining to UES's robustness in regards to future enhancements to integrate existing legacy applications and the University's goal to double its online degree program enrollment in 3 years.

These concerns voiced by the client provided the basis for Global Inventory Inc. to conduct an evaluation of the system's architecture.

After finishing the architecture for the Ubiquitous Education System (UES), the architecture team called for external help to get their process artifacts evaluated. Several architectural evaluation methods were evaluated, including ATAM. ATAM was selected due to its current and natural execution.

None of the members of the Global Inventory Inc. team had chaired an ATAM evaluation before and thus the team decided to call for external help. The external consultant was the natural leader of the evaluation team and guided the evaluation team through the evaluation.

### **Step 1: Present the ATAM**

The evaluation leader imitated the process by describing the ATAM evaluation method to the evaluation team and other VIP stakeholders at Global Inventory Inc. During the presentation the evaluation leader presented and handed out data about ATAM, including examples and data gathered during previous evaluations. The following documents and artifacts were handed out or presented

- *PowerPoint Slides* for a presentation – The slides carried the below information
- *Architectural analysis and ATAM* – A description on why architectural analysis is valuable and introduction ATAM
- *Steps of ATAM* – Introduction to the 9 steps of ATAM
- *Example* – The evaluation leader presented a case study from an ATAM processed that he had guided in the past.

The meeting attendees where polled to ascertain that they had a good understanding of the process that was going to be carried out. Contact information for the presenter and

evaluation team was given out to the attendees in case any of the attendees had any questions.

## **Step 2: Describe the candidate system**

The project manager presented the business case and the functional requirements. Then the architect presented the architecture and handed out the architectural documentation. The idea was to give the members of the evaluation team insight into the functionality, quality attributes, the current status of the architecture, and the context.

During the presentation the architectural drivers were presented together with what tactics had been applied to the architecture to meet the non-functional requirements.

The main quality attributes were usability, modifiability, and performance.

The documentation package consisted of

- View documentation from decomposition, deployment, and client server.
- UML diagrams of the UES User Experience Module
- Interface Documentation of the UES User Experience Module Decomposition
- Software Requirement Specification
- Use cases and usage scenarios
- Vision and scope document

The evaluation team had at this time no comments on missing or incomplete architecture documentation. The evaluation team made sure to gather the business goals and any architectural constraints that they found.

## **Step 3: Make a go/no-go decision**

After the presentation of the candidate system, the evaluation leader made an evaluation on whether the current state warranted an ATAM evaluation. Using the input gained from the previous step, including the candidate documentation, the leader decided that an evaluation was not only possible but also recommended at this point.

## **Step 4: Negotiate the statement of work**

The Ubiquitous Education System (UES) is a very large system, it was thus decided that only a portion of the architecture would be evaluated at this time. At the time the UES User Experience Module was the most elaborated and documented module.

As one of the major modules of the system, the UES User Experience Module played a vital role in fulfilling the system's requirement of being Ubiquitous. The module was considered a high risk early in the inception phase and it was thus desirable to evaluate

whether or not the team had selected to most suitable architecture to meet the functional and non functional requirements.

The UES UEM module was responsible for scaling the data provided to the connected devices. Data, web pages and media streams returned to a handheld device with low resolution and low bandwidth internet connection would be scaled down, and only the most vital parts of the contents would be provided.

After agreeing on the ATAM scope, dates were set for Phase 1 and 2.

The negotiated statement of work agreed upon is summarized below.

- The evaluation will take place over a two-week period.
- The scope of work is that an ATAM evaluation will be performed according to guidelines laid out by SEI document CMU/SEI-2000-TR-004.
- The system to be evaluated is the UES User Experience Module, client devices, the web servers, and operating systems.
- All costs of the consultant will be paid for in their entirety for the two-week stay. Any further costs would be discussed on an ad hoc basis.
- At the end of the two-week period, the final ATAM evaluation will be delivered and presented to the stakeholders. The ATAM evaluation package will consist of
  - An architectural documentation
  - Presentation of Approved business goals
  - Quality requirements of the UES User Experience Module based on quality scenarios
  - Matrix of quality requirements to architectural choices made
  - Defined architectural trade-off and sensitivity points
  - The set of risks, nonrisks, and risk themes
- The project manager will take responsibility for allocating resources (time) of any needed stakeholder, such as the architect, to the evaluation team. The project manager will also be responsible for facility allocation via the office manager for their department

### **Step 5: Form the core evaluation team**

Global Inventory Inc. was a very small team and it was thus necessary to give each team member several roles during the ATAM evaluation. In addition to the external ATAM leader the team consisted of 3 Global Inventory Inc. team members.

As the team leader had thorough understanding of the ATAM process it was innate to assign him most of the lead and process roles.

<b>Member</b>	<b>Role</b>
1	Team leader, evaluation leader, process enforce, questioner

2	Timekeeper, questioner
3	Proceeding scribe, questioner
4	Scenario scribe, data gatherer, process observer

Table 1.1 Evaluation Team Role Assignments

The project manager for UES made sure that the relevant team members had sufficient time to allocate towards their team duties.

### **Step 6: Hold Evaluation Team Kick-off Meeting**

The evaluation team leader gathered all the team members selected in the previous step were for a kick-off meeting. During the meeting, the team leader distributed the documents and announced the roles. A final schedule was agreed upon for phase 1 which will be presented in Step 7.

Quality attribute scenarios were selected and the team agreed on quality attributes of interest from step 2. These were usability, modifiability, and performance.

Typical of any web application some example generic scenarios were chosen and presented.

- What happens to the page response time as users are added to the system
- What happens to the database response time as users are added to the system
- What happens to the user interface as different browsers are used as interpreters
- What happens to the user interface when a certain type of device/browser combination is attached to the system

It was decided that the evaluation leader would write the final document for the ATAM process.

### **Step 7: Making necessary preparations for Phase 1**

The evaluation team along with the project manager held a meeting to prepare for the Phase 1 analysis. Included in this meeting was the proposed schedule for Phase 1. It was also asked that the project manager make sure the architect was present. The Phase 1 schedule was communicated as spelled out below. This was the same schedule as developed in Step 6

Day 1 – Phase 1

Step 1: 9-10 AM – Present ATAM

Step 2: 10-11 AM – Present Business Drivers

Step 3: 11 AM – 1 PM - Present the Architecture

Step 4: 1-2 PM – Identify Architectural Approaches

Break

Step 5: 2-3:30 PM – Generate Quality Attribute Utility Tree

Step 6: 3:30 – 5:30 – Analyze Architectural Approaches

Wrap up – 5:30 – 6 PM – finalize items and prepare for phase 2

Attendee list and attendee roles (\*) that may need to be available

- ATAM evaluation team
- UES Project Manager
- Architect
- Lead developer
- Configuration manager \*
- SQA lead \*
- Test engineer \*
- Deployment engineer \*
- System engineer \*
- Global Inventory Customer \*

Partial Supply List presented to Project Manager

- Starbucks French Roast Coffee and sandwiches
- Room to hold 10 people at a conference table with white boards on every wall
- Six flip charts with stands
- Two LCD projectors
- Wireless network hookup
- Conference call capability
- Internet capability

## **Step 8: Review the Architecture**

All of the architecture documentation provided in Step 2 was evaluated by the team. The team scribe wrote down a formal list of questions that will be given to the architect. The evaluation leader is delegated with the responsibility of making sure the rest of the team members understand the current architecture.

A list of questions for the architect to answer during phase one was generated, a partial list is presented below.

- What documentation do you have?
- What types of views did you document?
- Do you have interface documentation?
- What are your architectural drivers and constraints?
- What were your perceived general and quality attribute based scenarios
- What is the driving hierarchy of your system based on your quality attributes?
  - From the architectural drivers what were your tactics?
  - From your tactics what architectural patterns did you realize?
  - From your architectural patterns, what design patterns were used?
- At what stage in the SDLC is the project at?

- How does the system handle change requests and how did you adapt your architectural design to handle those changes?

## Phase 1: Evaluation

### ***Step 1: Present ATAM***

The project manager for UES will introduce the evaluation team and turn over the meeting to the evaluation team leader. The team leader will present a detailed overview of how the ATAM process works. This presentation goes through some of the main points of ATAM such as risks, trade off point, sensitivity points. The 9 steps of the ATAM are walked through so that all the stakeholders understand the entire process. A short example of the entire process is walked through to give the participants a feel for what will happen. Meeting “ground rules” are covered so the participants know how to act and will not engage in activities that are counter-productive to the agenda such as having conversations while someone else is talking or proposing answers during brainstorming[Clements02]

Some basic ideas and outputs about the ATAM are also covered such as

- Methods for elicitation of responses
- How data is analyzed
- Utilization and generation of utility trees
- Scenario brainstorming and prioritization of those scenarios
- The main outputs of the evaluation
  - Scenarios elicited and prioritized
  - Questions used to understand the architecture
  - Finalized Utility Tree with prioritizations set
  - Architectural approaches identified
  - Risks and non-risks
  - Sensitivity and tradeoff points

### ***Step 2: Present the Business Drivers***

The UES project manager introduced the known limitations of CSUFs’ current Blackboard system and complaints voiced by students enrolled in online degree programs, several of the CSUF department chairpersons decided to initiate a move and present a case to the University president for development of a new Ubiquitous Education System (UES) which would ultimately meet the needs of current students and help the University attract more enrollment in its online degree programs in the future. This attempt was successful and at the end of an evaluation and bidding process Global Inventory Inc was awarded the project.

UES will be replacing the current blackboard system at CSUF and may also be marketed to other educational institutions offering online programs. The new system will provide much wider access and much needed functions lacking in the current system. The Ubiquitous Education System (UES) will provide the venue required to educate students online worldwide, at anytime and via any device.

Main business goals and functions:

- The UES system shall allow educating online students anywhere at anytime via any device
- Support several hundred users simultaneously
- Have the ability to stream audio and video to a multitude of devices using commercial receiving systems on the customer
- Good performance with any device and any browser
- Have the ability to recognize device capabilities when requests are made and tailor the user experience around that device whether it be a rich experience via a standard workstation or a text based only output of a lecture to a text only device
- File sharing
- Real Time Collaboration
- Email
- Chat
- Student ePortfolio™ system for profile and resume sharing
- Ability to add more features easily

After the presentation by the project manager the team leader identified the Major Quality Attributes Identified based on Project Mangers Business Drivers

- Performance
- Scalability
- Modifiability
- Usability

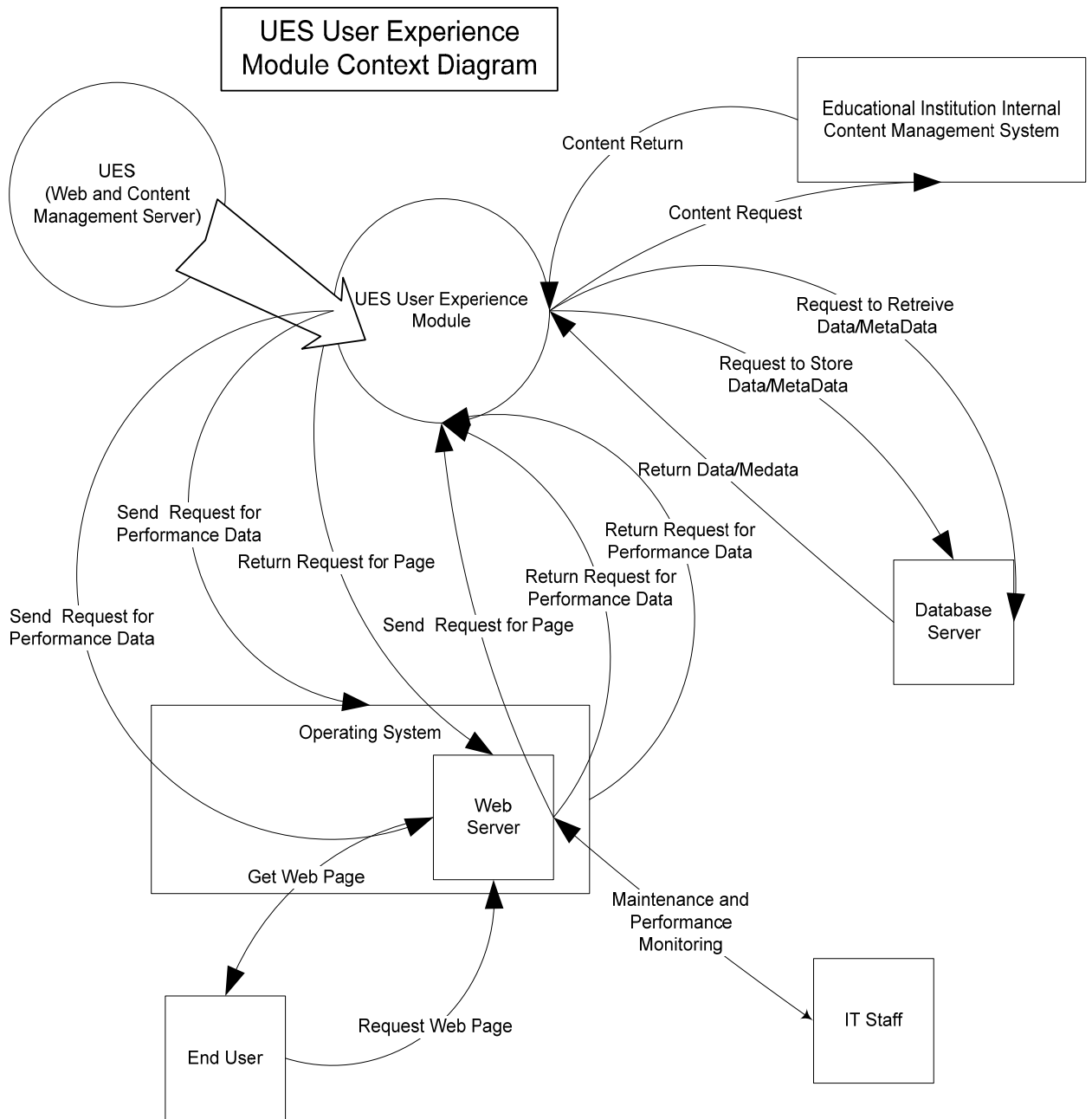
The major stakeholders were identified

- Project manager
- Architect
- Developer
- Global Inventory Customers, represented by the Marketing Manager

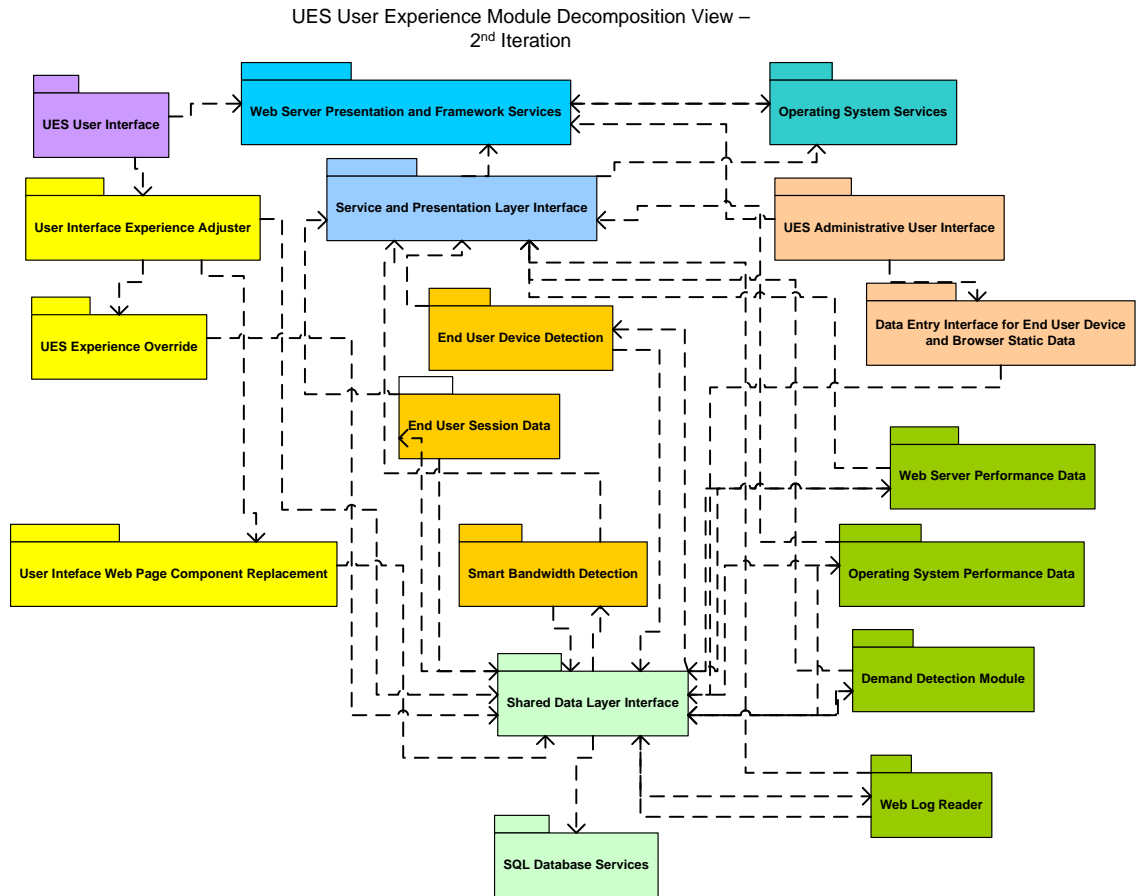
### Step 3: Present the Architecture

The architecture of the UES User Experience Model is thoroughly documented using multiple views. The modules main focus is on delivering device independent content and content tailored to a device's features and current bandwidth, and the architecture was derived from ADD and QAW sessions. The views and diagrams presented were:

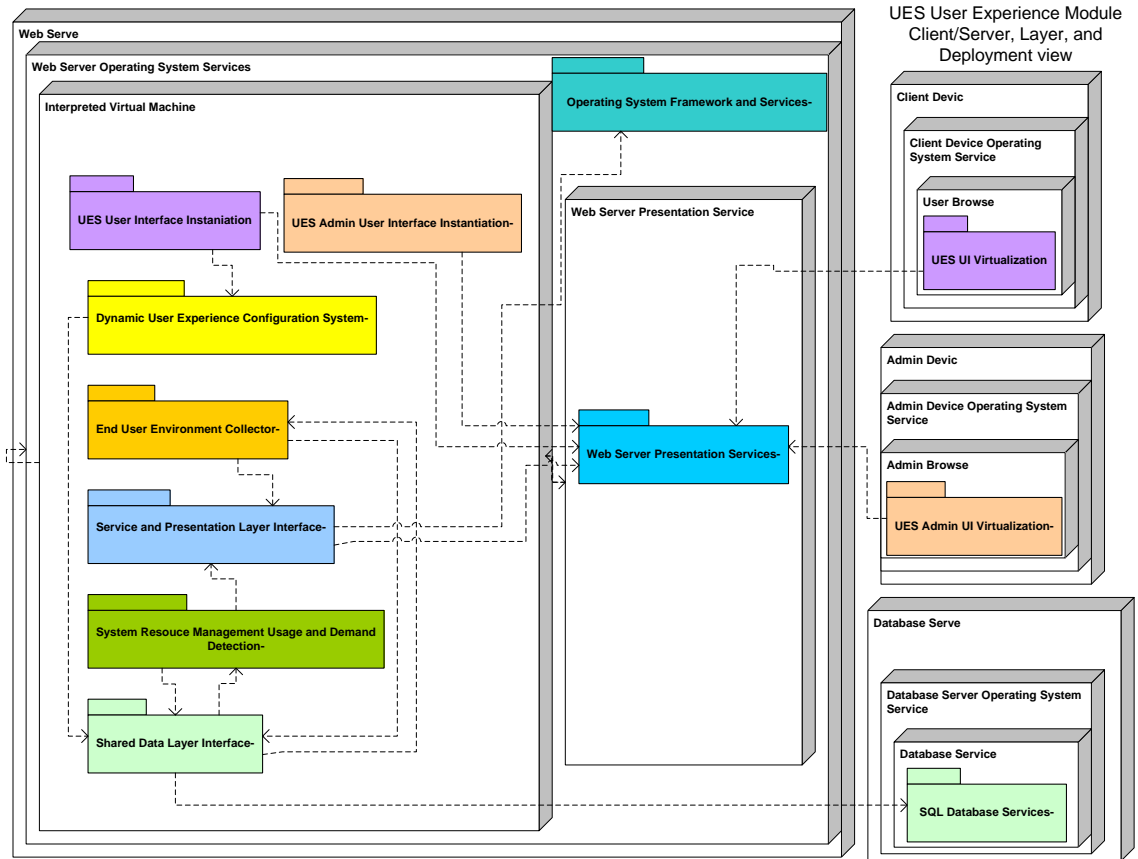
- Context diagram, showing the modules position in the complete systems



- Module decomposition view, showing which and how sub modules communicated.



- Deployment, Layered and Client / Server View, showing how processing was distributed, how layers were divided and how communication between modules were performed.



The highlights from this presentation could be summarized as follows:

- System designed C#, a modern object oriented language.
- The use of C# mandates use of .Net framework. The team decided to base the system on .Net3.5 as it's expected to be the standard when the product is ready for release.
- Major COTS library used for UI
- Commercial Relational Database Server back-end
- SMART Bandwidth Detection. Based on:
  - Network transmissions
  - Web server Performance data
  - Database server performance data
  - Operating system Performance data
- Layered Architecture for separation of UI, Business logic and data retrieval.
- The web servers are using Microsoft Network Load Balancing to balance user load

- The database server is clustered using Microsoft Clustering services

### **Step 4: Identify and Catalog Architectural Approaches**

Several tactics have been evaluated and patterns have been applied to ensure a sound architectural design.

Exceptions and exception handling has been implemented to handle error conditions.

Removal from service, process monitor and configuration has been implemented as error prevention mechanisms.

Blackboard pattern was selected to fulfill the requirements of device independent data and data tailored to a device's bandwidth. Several processes contribute and cooperate to provide the UES User Experience Module with necessary data to give the end user data necessary data.

Hierarchal Layers pattern was selected to solve the requirement of thousands of simultaneous users and low response time. By applying the hierarchal layers pattern the idea was to reduce the resource needed for database connection and connectivity. The same pattern was used to separate the user interface from the business and data connectivity logic.

The Component Configurator pattern and the Call and Return pattern have been applied to achieve modifiability on the User Experience module.

The RDBMS dictates a client-server pattern between the database and the connection layer.

The Network load balancing of web servers is a client server pattern to distribute load, and increase availability

Clustering of the database server is an availability tactic that uses the communicating processes pattern.

The .NET framework is an example of a Virtual machine/interpreter.

### **Step 5: Generate Quality Attribute Tree**

Level 2: Quality Attribute	ID	Level 3:Quality Attribute Refinement (General)	Level 4: Quality Attribute Scenario (Specific)	Importance	Difficulty
Modifiability – the cost of change	M1	Ability to add new features (modules, dlls) with out major	Deploy new executable in <5 seconds to all web servers	H	M

		system disruption at deployment time			
	M2	Change end user device/browser/OS descriptions dynamically	Administrative interface lets administrator add more device/browser/OS descriptions to system	H	L
	M3	Implement new major features without affecting other modules	A new feature is added to the system without any code being changed in any other functional modules	H	H
	M4	Change web page content to match device or browser	Content is dynamically changed to device/browser/network conditions without recompiling based on data from various subsystems	H	M
	M5	Move Database	DBA needs to change/move database to another system in under 10 seconds	M	L
Note: This was added by Step 6 reprioritization	M6	A new performance measurement is needed to be gathered from the operating system	Administrator at run time needs to add performance criteria to be evaluated without affecting the system.	M	H
Usability	U1	Minimize the impact of errors	A user makes an error in data entry and the system informs them of it so it can correct the error	M	L
	U2	User understands feature use	When a use holds her mouse over a object a window will explain how to use it	M	M
Performance	P1	Response Time	A student logs into the UES to download final version of HW assignment after collaborative work is completed. The file is downloaded in less than 3 seconds.	M	M
	P2	Response Time	Faculty logs into UES to upload course material at the start of the semester.	M	M

			She/he is able to upload the course syllabus in less than 2 seconds.		
	P3	Throughput	While 500 students are logged in, 100 attempts to download midterm exam; UES throttles bandwidth and maintains a minimum of 20 kb/s throughput. All downloads complete; no noticeable slow down is experienced by the other 400 students	M	H
Testability	T1		Load testing of a modified stored procedure implemented in the staging Environment; 500 simultaneous queries are submitted to execute the Stored procedure with varying input Parameters. The UES returns query results to all requestors in less than 4 seconds	M	L
	T2		Developer tests email function by taking the role of a faculty member and sends a test email to 100 students enrolled in a class; All student test accounts receive email successfully	M	L
Availability Note: This was reprioritized by Step 6 reprioritization	A1	Hardware Errors	If servers encounter hardware failures (power, hard drive) the system must still be available	<del>M</del> H	M
	A2	Module failures/ Exceptions	Error in a module must not create ripple effects but be isolated to the module where the error occurred.	H	M

	A3	Software maintenance	New version of database server, DBA must be able to upgrade / hot swap without shutting down the system	M	L
	A4		Administrator must be able to update / replace stream server without downtime	M	L
Security	S1	Authentication	User needs to be authenticated to have access to the system	H	L
	S2	Authorization	Users needs to be authorized to view grades	H	L
	S3		Users need to be authorized to view class contents	H	L

### **Step 6: Analyze Architectural Approaches**

<b>Scenario #</b>	M1 – Deploy new executable < 5 seconds to all web sites		
<b>Attribute</b>	Modifiability		
<b>Environment</b>	Normal Operations		
<b>Stimulus</b>	Need to deploy new executable		
<b>Response</b>	Deployed with no side affects		
<b>Architectural Decisions</b>	<b>Risk/NonRisk</b>	<b>Sensitivity</b>	<b>Tradeoff</b>
Multiple web servers using network load balancing	R1,R2,NR3	S1,S2	T1
.Net virtual machine	NR2		
<b>Reasoning:</b> Ensures common code base in minimal time frame. The .NET framework allows file copy deployment and immediate virtual machine recycling.			
<b>Architecture diagram:</b> N/A			

R1. Must deploy executables across multiple machines on the network which may cause latency and exceed the 5 second limit.

R2. A network interruption will leave a web server in transitioned state during deployment.

NR3. An increased number of copies of the binaries across multiple machines may have different users looking at different code versions for a short period.

S1. More web servers will make deployment take longer unless staged on machines first prior to deployment.

S2. Network changes will affect the ability to modify the executables on the web servers.

T1. Increasing the number of web servers increases availability and performance but at the cost of maintainability and modifiability.

NR2. The .Net Platform will automatically load the new binaries as soon as they are placed on the system.

<b>Scenario #</b>	M3 - A new feature is added to the system without any code being changed in any other functional modules		
<b>Attribute</b>	Modifiability		
<b>Environment</b>	Design Time		
<b>Stimulus</b>	Need to add new features		
<b>Response</b>	New features added		
<b>Architectural Decisions</b>	<b>Risk/NonRisk</b>	<b>Sensitivity</b>	<b>Tradeoff</b>
Layered Architecture	R1	S1	T1
Semantic Coherence	NR2	S2	T2
.NET Framework	NR3		T3
<p><b>Reasoning:</b> Semantic coherence and forming hierarchical layers lets the system be easily modified at design time without having major affects on other parts of the program functionality. It is known that this may cause more overhead and has some performance tradeoffs.</p>			
<p><b>Architecture diagram: See Architectural Document Listing</b></p>			

R1. The layered architecture can be considered as an abstraction of common services; however, it is not clear how the interface parameters between layers interoperate and if they are subject to change that may make interface behavior inconsistent.

S1. Having a small number of layers makes it more difficult modify functionality

T1. There are possible run-time performance issues that may occur with multiple layers.

NR2. Semantic coherence is an architectural tactic for modifiability

S2. Too little semantic coherence may cause overlap of functionalities that may make it more difficult to integrate new features with the appropriate interfaces.

T2. Having the right level of semantic coherence must also be balanced with performance as having a highly separated system may lead to performance problems.

NR3. This transparent to the .NET framework

T3. More classes have to be instantiated leading to more memory overhead which may lead to performance issues.

<b>Scenario #</b>	M4 – Content is dynamically changed based on browser/network/device conditions and subsystem data without recompiling system		
<b>Attribute</b>	Modifiability		
<b>Environment</b>	Run Time		
<b>Stimulus</b>	Need to modify content		
<b>Response</b>	End user sees content based on device and system.		
<b>Architectural Decisions</b>	<b>Risk/NonRisk</b>	<b>Sensitivity</b>	<b>Tradeoff</b>
Separation of UI from core logic and data layer	NR1	S1	T1
Shared Data Layer Interface/Blackboard	R1,R2,R3	S2	T2
Separation of Service and Presentation Layer access	R4	S3	T4
Main Data Repository on Separate Server	R5		T5
.NET Framework	NR2		
Dynamic User Configuration System	R6	S4	T6
<b>Reasoning:</b> Since the UI content was going to change dynamically based on a user device, browser, and network speed, a methodology or framework had to be put into place to modify web page content at run time. Separating the system into layers and giving various modules functional responsibilities facilitated easier retrieval of performance and allowed for a blackboard pattern to control various data entities back and forth between the various layers.			
<b>Architecture diagram:</b> See architectural document listing.			

NR1. This was a mandatory separation in order to accomplish the desired goal

S1. Decreasing the separation makes it harder to dynamically adjust content and to introduce new features into the system.

T1. Introduces a layer of complexity in interlayer code performance but not in true end-user perceived performance

R1. The system must traverse through a layer and a controller that has accessibility to 4 different data sources and if one of the data sources does not respond then the system may not respond and consequently may not show content.

R2. The system may suffer latency issues due to the data calls and responses being made from the system and may actually increase the end users page turn time that we are trying to improve

R3. The data source may not be able to retrieve the device characteristics if it is not recognized by the system leading to possible incorrect content being displayed.

S2. The more data sources involved in the blackboard system means a longer latency in deciding on the best content to provide and may cause the web page to time out.

T2. The resource usage of the data calls to the various data sources is improved by pooling the connections in a single shared data layer and among multiple data entities rather than having a multitude of threads being called from individual components which may cause a latency as compared to a direct call.

R4. This decouples the calling webpage from making its own calls directly to the OS and server for performance data which, like R1-R3, may increase latency and cause exceptions if no data is found or available.

S3. Separation and redirection of any calls interacting with the OS or web services means that all calls are funneled through one component. Too many calls piped through this interface may result in users not seeing the proper content if a bottle neck occurs in this component.

T4. This permits more modifiability having the system access a common interface to the service and presentation layer performance data at the expense of inte-component call performance.

R5. This separates the resource usage between data and presentation servers. If the connection between servers goes down then it is possible that no content will display on the end users system.

R6. Separation and redirection of any calls interacting with the OS or web services may increase latency and queuing if other calls are going through that same interface. It is unclear if there is a prioritization of requests or as to why that would not be needed.

T5. There is a fine trade off between an increase in performance by allocating the resources on a separate server and having the network latency between the web server and data server. This difference may decrease as more users access the system using more data pages in memory thereby serving content to the requesting caller sooner.

NR5. The .NET framework facilitates the dynamic content presentation by allowing page component substitution.

R6. This module determines what content is displayed to the user. There is no clear default content store in case there is an issue with this module or module it draws from.

S4. Any changes to this module or submodules this component accesses will have a performance and usability quality effect on the end user experience.

T6. This component could have been put within the individual page abstraction but the system would not be able to be as easily modified with changes to performance or web page content adjustments.

<b>Scenario #</b> Note: added by Step 7 as a priority scenario.	M6 - A new performance measurement is needed to be gathered from the operating system		
<b>Attribute</b>	Modifiability		
<b>Environment</b>	Run Time		
<b>Stimulus</b>	Need to add new operating system performance monitor criteria		
<b>Response</b>	New criteria evaluated added without affecting other systems.		
<b>Architectural Decisions</b>	<b>Risk/NonRisk</b>	<b>Sensitivity</b>	<b>Tradeoff</b>
Layered Architecture	R1	S1	T1
Semantic Coherence	NR2	S2	T2
.NET Framework	NR3		T3
<b>Reasoning:</b> The system may need to evaluate new operating system performance criteria at run time. Separating the Administrative User Interface into its own module and using semantic coherence help achieve these goals.			
<b>Architecture diagram:</b> See Architectural Document Listing			

R1. The layered architecture can be considered as an abstraction of common services; however, it is not clear how the interface parameters between layers interoperate and if they are subject to change that may make interface behavior inconsistent.

S1. Having a small number of layers makes it more difficult modify functionality

T1. There are possible run-time performance issues that may occur with multiple layers.

NR2. Semantic coherence is an architectural tactic for modifiability

S2. Too little semantic coherence may cause overlap of functionalities that may make it more difficult to integrate new features with the appropriate interfaces.

T2. Having the right level of semantic coherence must also be balanced with performance as having a highly separated system may lead to performance problems.

NR3. This transparent to the .NET framework

T3. More classes have to be instantiated leading to more memory overhead which may lead to performance issues.

<b>Scenario #</b>	P3 – While 500 students are logged in, 100 attempts to download midterm exam; UES throttles bandwidth and maintains a minimum of 20 kb/s throughput. All downloads complete; no noticeable slow down is experienced by the other 400 students		
<b>Attribute</b>	Performance		
<b>Environment</b>	Under stress		
<b>Stimulus</b>	User needs to download midterm		
<b>Response</b>	Midterm is downloaded successfully		
<b>Architectural Decisions</b>	<b>Risk/NonRisk</b>	<b>Sensitivity</b>	<b>Tradeoff</b>
System Resource Usage and Demand Detection	R1,R2	S1	T1
File Server Separated from Web Server	R3		T2
<b>Reasoning:</b> This performance feature keeps file system transfers from overwhelming the system. Having a semantic module to keep track of performance of session data, server data, and web server data allowed modification of web session performance. The system queues the request to maintain the proper client-server and server-client data transfer at 20 kb/s.			

**Architecture diagram: N/A**

R1. The module must gather data from a variety of sources to compare performance and hence latency and execution time may become an issue.

R2. There is no load performance data on this component and if overwhelmed or unavailable may have undesired affects on the entire system,

S1. Any changes to the computations of acceptable system performance will affect the session queuing and bandwidth adjustment on the system.

T1. This is a performance trade off with availability. The system must be perceived as being available but still be able to service the user requests.

R3. Having the file server separated from the web server may lead to performance issues.

T2. Separation of responsibilities allows more resources to be used for a given task.

Offloading the file handling to the file server increases available resources to the web server at the cost of additional client server calls and latency.

<b>Scenario #</b> Note: Step 7 readjusted the priority of the scenario.	A1 – If servers encounter hardware failures (power, hard drive) the system must still be available		
<b>Attribute</b>	Availability		
<b>Environment</b>	Degraded Operation		
<b>Stimulus</b>	Hardware failure		
<b>Response</b>	System remains functional		
<b>Architectural Decisions</b>	<b>Risk/NonRisk</b>	<b>Sensitivity</b>	<b>Tradeoff</b>
Client-Server	R1,R2	S1	T1
<b>Reasoning:</b>			
<b>Architecture diagram</b>			

R1. It is apparent that the system would not recover if the file server went down.

R2. Adding too many servers will increase the cost of the system.

S1. Adding more servers increases availability as the functions become more distributed.

T1. Increasing the number of client-server relations internal to the system will help offset performance issues but will increase the cost of the system.

<b>Scenario #</b>	A2 – Error in a module must not create ripple effects but be isolated to the module where the error occurred		
<b>Attribute</b>	Availability		
<b>Environment</b>	Degraded Operation		
<b>Stimulus</b>	Error in Module		
<b>Response</b>	Other modules remain unaffected		
<b>Architectural Decisions</b>	<b>Risk/NonRisk</b>	<b>Sensitivity</b>	<b>Tradeoff</b>
Semantic Coherence	R1	S1	T1
.NET Framework	NR1		T2
Blackboard Structure	NR2		
<b>Reasoning:</b> The decision to use the .NET Framework facilitated the use of the built-in exception handling capabilities of the system. This in conjunction with using semantic coherence prevents a ripple affect from occurring through out all modules.			
<b>Architecture diagram : See documentation package</b>			

R1. The exception handling was not demonstrated clearly enough to say that it will prevent module exception ripple affect from happening. The system exception handling must be made more visible in the system.

S1. A high degree of separation of responsibilities makes the system more apt to prevent ripple effects. A low degree of semantic coherence will result in ripple affect through out the system.

T1. Separation of responsibilities makes the code modular and easier to modify when adding new features.

NR1 – The .NET framework allows a fine grain control over how exceptions are handled

T2. Having a virtual machine may not perform as well as straight XML or HTML on a web page when preventing module errors.

NR2. Having data source requests and control in one area will actually help prevent ripple affects.

## Phase 2: Evaluation (Continued)

### Step 7: Brainstorm and Prioritize Scenarios

A list of scenarios where elicited from the gathered stakeholders and their prioritizations after voting is listed below. Each stakeholder was given 9 votes. Three scenarios were consolidated into one scenario and one scenario was removed. Based on the voting and time constraints, only the two highest voted scenarios (5 and 6) will be evaluated. These were the only two that had more than 10 votes attributed to them. Scenario 5 was added to the Utility Attribute Tree as scenario M6. Scenario 6 was determined to already have a similar scenario that was identified as A1. This scenario was increased in importance to H from M. Those scenarios were reanalyzed and added to the output of step 6. These scenarios were determined to be similar to what the architect had already planned for and the brainstorming session did bring out an alignment of the stakeholders concerns with the architectural design already in place.

Stakeholder	Role
1 – ATAM Team	Team leader, evaluation leader, process enforce, questioner
2 – ATAM Team	Timekeeper, questioner
3 – ATAM Team	Proceeding scribe, questioner
4 – ATAM Team	Scenario scribe, data gatherer, process observer
5	User Performance Engineer
6	Customer Advocate
7	Possible End user
8	Architect
9	System Engineer

Scenario#	Description	Votes
1	A new device and browser is introduced that is not in the known list of devices	3
2	The network switch loses a power supply	3
3	A fault in a module makes it unavailable	6
4	A new binary is deployed to all the web servers	
5	A new performance measurement is needed to be gathered from the operating system	12
6	One of the servers goes down.	12
7	The database server goes down	
8	The file server goes down	3
9	An unauthorized user logs on to the system and they need the audit trail	
10	System should respond to end user within 3 seconds in normal operation	3

11	System should respond to end user within 5 seconds in case of high load.	3
12	A new functionality needs to be added to the system without modifying the code in the other modules	3
13	A user tries to log into the system 20 times	6
14	A user tries to access content in a class they are not enrolled in.	3
15	Available bandwidth for all systems out to the internet goes down to 30 kb/s	
16	The operating system has a service pack applied to it	
17	The .NET framework has security hot fix.	
18	The WMI interface is not available on the operating systems	3
19	The default content is not set up for a web page	6
20	A user without thorough computer and web knowledge want to know what a button does if they click on it.	3
21	A user tries to upload a 100 GB file	
22	A user tries to download a 100GB file via a dial up connection	
23	An exception is thrown that affects the .NET application pool the application resides in.	6
24	The application has memory leak	
25	A raid array on the sql server goes out causing a hot spare to kick in and the array to rebuild causing slowdowns and timeouts in SQL queries	3
26	A piece of web page content is corrupted.	
27		
28	A server administrator changes permissions on the directory the web logs reside in and the system can no longer access them to calculate performance based content modification	
29	A Denial of Service attack is launched at the system	
30	500 users access the system that was designed for 200 users.	
31	200 Students attempt to download Final exam simultaneously.	6
32	200 students attempt to send emails to teammates simultaneously a day before HW due date.	

### **Step 8: Analyze Architectural Approaches**

During step 7 it was discovered that the architect had made the false assumption regarding the importance of scenario A1 in the utility tree. It was also discovered that the addition of new components for performance data gathering had not been evaluated. It was necessary to go back and add a new leaf in the Quality Attribute Tree. Because these two scenarios were given the most votes during Step 7 it was decided to further analyze these scenarios.

### **Availability: Hardware Errors**

Scenarios 6 from Step 7 could be mapped to scenario A1 in the utility tree. It was discovered that the architect had made a tradeoff between cost and availability in that it was accepted that if one of the servers went down the whole system would be affected. Depending on what server was affected, the whole system could become unavailable. For many customers this would seem like a valid tradeoff, but for many customers this would be unacceptable and the system would thus not satisfy their requirement of a system that should be available 24/7/365.

To fully satisfy this goal it would be necessary to provide a full redundancy for all servers with at least one backup server for each. The team found the choice of applying the client-server pattern would not provide any gain in respect of availability. They agreed that some tactics for fault detection and fault prevention needed to be implemented.

The need for redundancy could either be implemented through the tactics of active redundancy, passive redundancy or spare.

To detect errors a combination of ping / echo and heartbeat combined with a modified voting tactic to verify data integrity should be implemented in order to decide on availability and when to have a master challenge. The voting mechanism running as a service on a subset of the servers are responsible for deciding what server should be the master based on the accuracy of the responses received.

The ping / echo and heartbeat tactics were considered appropriate to detect a total unavailability of one or more server. They would provide sufficient confidence if a server went offline due to severe hardware failure such as power or network failures. These tactics were not considered sufficient to detect hardware errors that could lead to data errors. Such failures include hard drive failures and minor network errors.

Applying a tactic for passive redundancy seemed to be appropriate in this scenario. This was due to the nature of the application, as it's neither a life supporting application nor would a failure pose any danger to human life. This was still considered a tradeoff point between availability and cost. Several commercial systems for server redundancy are available. Implementing such a system from ground up was considered a major cost driver and thus it seemed appropriate to base a redundancy solution on COTS.

### **Modifiability: New Performance Measures**

During Step 7, it was discovered that nothing in particular was done to implement scenario 5. Although the scenario could be mapped to scenario M1 and M3 in the utility tree, it seemed that nothing was actually done to achieve this in architecture. In other words, it was discovered that very little had been done to achieve the modifiability desired and required from a modern system. The original design showed a predefined set of performance gathering points, without any flexibility. The lack of modifiability in this

point was considered a serious flaw in the design as it also violates the scenarios M1 and M3. The ATAM team thus recommended a total review of the tactics and patterns applied to the whole system to achieve the desired level of modifiability. Such a review was considered outside the scope of this report, but the team came up with a set of recommended changes that could be applied to the system as a whole.

The performance measures module was not detailed enough in that it should have been documented and detailed on its own. For the HW1 report, the module should have been presented as a sub module with a defined interface. Without such the implementation and the understanding of the module is not clearly communicated.

To satisfy M1, M2 and M3 for the particular module it's necessary to find tactics and apply patterns that are not documented in the current report. To be able to add new performance measures as seamlessly as possible it would be desirable to Defer Binding Time. Tactics such as Runtime Registration, Configuration Times and Polymorphism should be applied.

By letting the components self register at runtime, a dynamic deployment can be achieved. Self-registration would also remove the overhead of manual runtime registration

By using separate configuration files, the complete measurement module could be replaced. As the whole measurement module could be a cost driver it was considered a good idea to start with an empty performance monitor module and then evolve the system as customer requirements increases.

By using the development languages support for polymorphism the module and service abstraction can be achieved. By providing an abstract base class or interface for other developers the system can dynamically evolve. Through this mechanism, only the public interface to the modules needs to be identical while the contents can be totally independent of each other and the core of the system.

### ***Step 9: Present Results***

During ATAM the evaluation team documented the architecture and the results of the workshops. The outcome of each step is documented in this report.

A two-hour presentation was organized so that the evaluation team could present the process, the artifacts and the ambiguities that were unveiled. Process artifacts that were presented were:

- Architectural approaches documented
- Scenarios and prioritization from BS session
- Utility Tree
- Risks discovered
- Non-risks documented
- Sensitivity and tradeoff points found

In addition, the ATAM evaluation team grouped the risks found during the process, with a focus on those risks that were documented during step 6, into risk themes. The three themes were:

1. Availability
2. Performance gathering
3. System performance

Availability is a major concern in a system that requires an uptime of 24/7/365. Not every customer of the system would require an uptime of almost 100%, but the architecture of the system needs to be designed to provide such availability. Various degrees of availability can be achieved based on an installation's hardware and configuration. The ATAM processed showed that the architecture contained load balanced web servers using Microsoft Network Load Balancing and clustered database servers based on Microsoft Clustering Services. The ATAM evaluation unveiled that such a measure was not taken for file servers. It is thus recommended to base file servers on fault tolerant hardware from vendors such as Stratus Technologies or Marathon technologies.

The architecture showed a serious weakness in respect to its performance gathering modules. Important performance data was not collected and there were no documented ways of dynamically adding new performance data providers. An explicit need for a separate performance controlling module was discovered.

By adding several layers to the data and business logic modules, performance degradation must be expected. The need to use a layered architecture was founded in the need distribute the functionality to several servers and the need for logic separation to allow simpler maintenance and replacement. Even though the architectural approach was well documented and rooted in other quality attributes, the ATAM team considered the tradeoffs significant. Due to the nature of these tradeoffs, it is recommended to build a prototype to evaluate the performance of the system.

## **Phase 3: Follow-up**

### ***Step 1: Produce Final Report***

A final report was generated by the ATAM evaluation team at Global Inventory Software and led by the outside consultant. All evaluation team members contributed and were responsible for various parts of the report. The report covered what the evaluation team did, what the observations of the team were, and the team's conclusions. We prepared a draft and sent it over to the project manager for review. The project manager then made a few suggestions and we then turned in a final copy. A shortened version of it is below with references to sections already covered in this report. The report format was based off a template the consultant had previously used [Clements02a].

4. Introduction - this section was a brief over view of the vision and scope of the exercise. This introduced some of the goals of the

exercise, who should be involved, and what the scope of the evaluation encompassed.

5. Evaluating a Software Architecture
  - i. This covered the business goals and reasons for an organization to evaluate an architecture.
6. ATAM Overview
  - i. Covered a brief description of what ATAM is and what it can do for a company. This was covered in Phase 0 Step 1.
7. The ATAM for Global Inventory
  - i. Description of ATAM evaluation for Global Inventory, its overall goals, and how it was tailored as compared to other ATAM evaluations the team leader had done before.
8. Summary of Business Drivers
  - i. Covered in Step 2 of this paper. The business drivers from step 2 were incorporated in to his section.
9. Summary of Architecture Presentation
  - i. Covered in step 3 and 4 of this paper
10. Quality Attribute Utility Tree
  - i. Covered in step 5 of this paper
11. Analyze the Architectural Approaches
  - i. Covered in Step 6 of this paper
12. Scenario Generation, Consolidation, and Prioritization
  - i. Covered in step 7 of this paper
13. Analysis of Prioritized Architectural Approaches
  - i. Covered in Step 8 of this paper
14. Risks, Sensitivities, Tradeoffs, Nonrisks, and Other issues
  - i. Covered in step 9 of this paper
15. Conclusions
  - i. Covered in step 9 of this paper

## ***Step 2: Post Mortem Meeting - Collect Data for Measurement and Process Improvement***

A meeting was held after the final report was delivered. The discussion was centered on improvement of the ATAM process held for Global Inventory. Since the evaluation team had not done an ATAM before, they did not have much to compare it to but it was decided that it was too short of a process for such a large system and that they would have liked to spend more time on it. Everything else about the ATAM was well accepted by both the evaluation team and the stakeholders.

Surveys were sent out to team members and the client stakeholders to evaluate the team effectiveness and cost. The survey showed that the members were satisfied with the cost and effectiveness of the ATAM. The customer stakeholders were also satisfied with efforts and it showed in the survey. The stakeholders that participated also gave the cost

in terms of hours of their resources that they had participated in the ATAM. The team itself was found to have had spent 231 hours total working on the ATAM over the life of the project. The various company members surveyed came up to 102 hours total for all company members that had participated over the life of the evaluation.

Six months after the ATAM another evaluation was given to the stakeholders. No opinions had substantially changed in six months and the Global Inventory team was busy making changes to the architectural approaches to improve the risks that were found.

### ***Step 3: Build Portfolio and Update the Artifact Repositories***

The lead evaluator showed the team evaluations members how to form a repository for future evaluations. In this repository (SharePoint) the team stored all the artifacts of the ATAM process. Out of the brainstorming scenario list came out a generic list that the team could build on for future uses. The cost analysis survey, in terms of resources, was also stored as a guideline for future ATAM evaluations. Certain parts of the evaluation were broken down further by ATAM activity so that the next evaluation could be better scheduled and resources accommodated. The consultant that was serving as the team evaluator also was able to put the artifacts into a personal repository for future use except for the architectural documentation.

## **Part 2: ATAM findings compared to HW1 Architecture**

By conducting the ATAM we found several flaws and ambiguities in the architecture designed in HW1. The ATAM led to further abstraction and delegation of responsibilities and qualities to new modules.

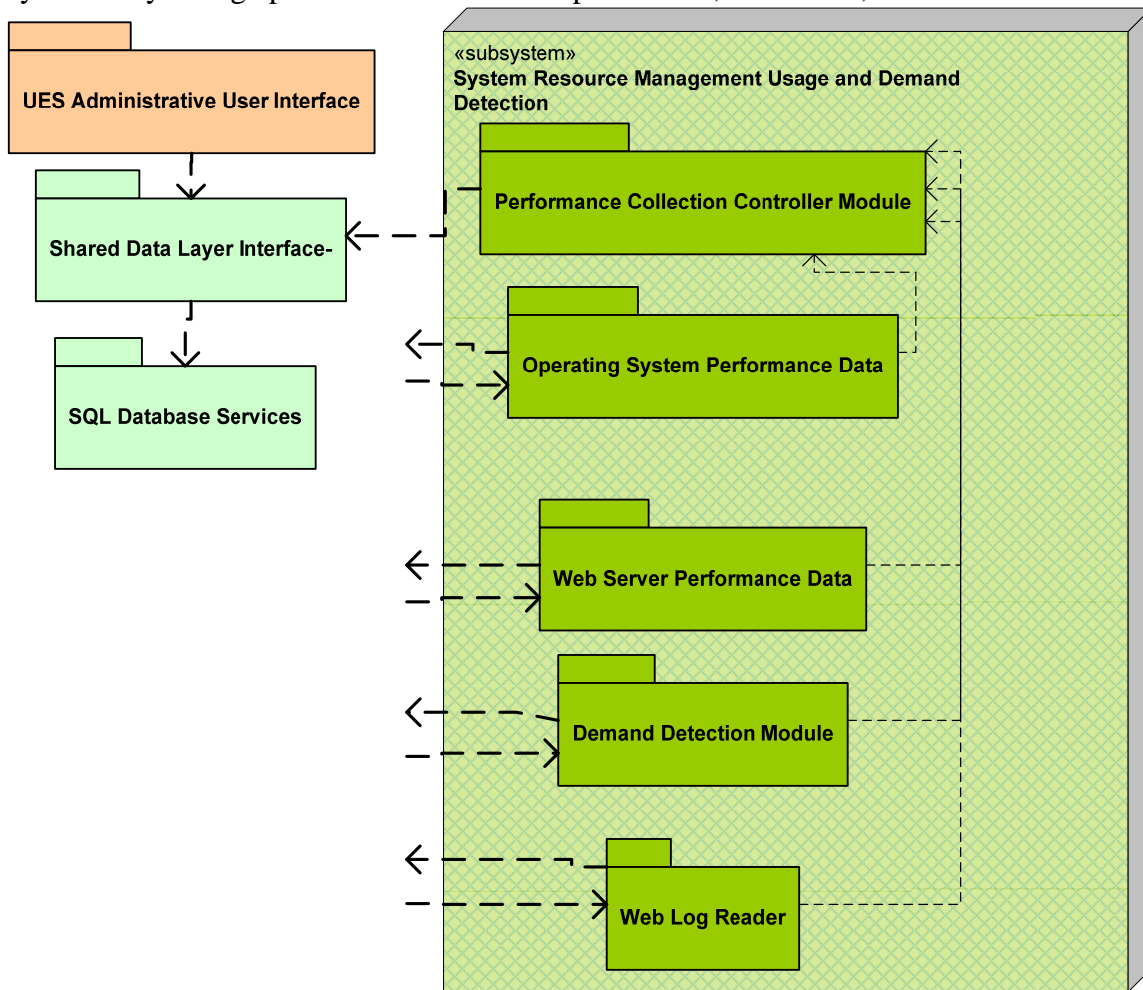
A comparison between the two architectures will provide a rationale for an architecture selection or in our case reconstruction by comparing the fitness of software architecture candidates to be used in constructing the UES. Comparing software architectures implies a set of criteria, a comparison without any criteria produces no sound reasoning about a selection.

For example, we have a goal to increase reusability among different product architectures; a specification of future commonly used artifacts is required. A prime candidate is a performance data module to be used in UES and our future products; therefore, the module needs to be build based on supporting tactics and with that goal in mind.

Our original architectural design was fundamentally sound, however after performing ATAM we concluded that certain reconstructs would be necessary to address the most important requirements pertaining to availability and performance qualities.

In the original design we had not accounted fully in an architectural sense for 24/7/365 availability. In the reconstruct, we decided to provide full redundancy for all servers with at least one back up server. The team also decided that tactics for fault detection and fault prevention had to be implemented to satisfy the client's need in this regard.

We also realized that our original design lacked flexibility in performance data gathering, which resulted in the decision to build a complete and separate performance data-gathering module. Such a module would show up as a single module in the module diagram of the UES documented in HW1 and in the module diagram in Phase 2 step 3. This module should be part of the System Resource Management Usage and Demand Detection group of sub modules that currently has four modules. This would add a fifth module to control collection activities where the UES administrative interface could dynamically change performance collection parameters, thresholds, and activities.



The effort and resources spent on the ATAM proved to save the team from embarking on a project that would not have satisfied the customers' expectations and would have been very hard to maintain and extend. In respect to the UES architecture presented in HW1 it's safe to say that the ATAM paid for it self.

## Appendix A: References

Bass, Clements, Kazman, Software Architecture in Practice Second Edition, Addison-Wesley, 2003

[Clements02a] Clements, P. Kazman, R. Klein, M. Evaluating Software Architectures: Methods and Case Studies, Addison Wesley, 2002

Clements, P. Bachmann, F. Bass, L. Garlan, D. Documenting Software Architectures: Views and Beyond, Addison Wesley, 2002

Kazman, R. Klein, K. Clements, P. ATAM: Method for Architecture Evaluation, SEI, 2000, <http://www.sei.cmu.edu/pub/documents/00.reports/pdf/00tr004.pdf>