The System Architecture Tradeoff Analysis Method® (SySATAM®)

Mike Gagliardi and Bill Wood

Purpose of the System ATAM – 1

The System ATAM is a method that helps stakeholders ask the right questions to discover potentially problematic architectural decisions (risks)

Discovered risks can then be made the focus of mitigation activities—for examples:

- changing architecture
- further analysis
- extending prototyping.

Tradeoffs can be explicitly identified and documented

- Tradeoffs made already
- Upcoming tradeoffs

Purpose of the System ATAM – 2

The purpose is **NOT** to provide precise analyses. . . the purpose **IS** to discover risks created by architectural decisions.

We want to find *trends:* correlations between architectural decisions and predictions of system properties.

Presentation Outline

What is an ATAM?

Similarities and Differences between ATAM and System ATAM

Highlights of Differences

Experiences and results

Phase 2 – Stakeholders

The following is a partial list of potential stakeholders:

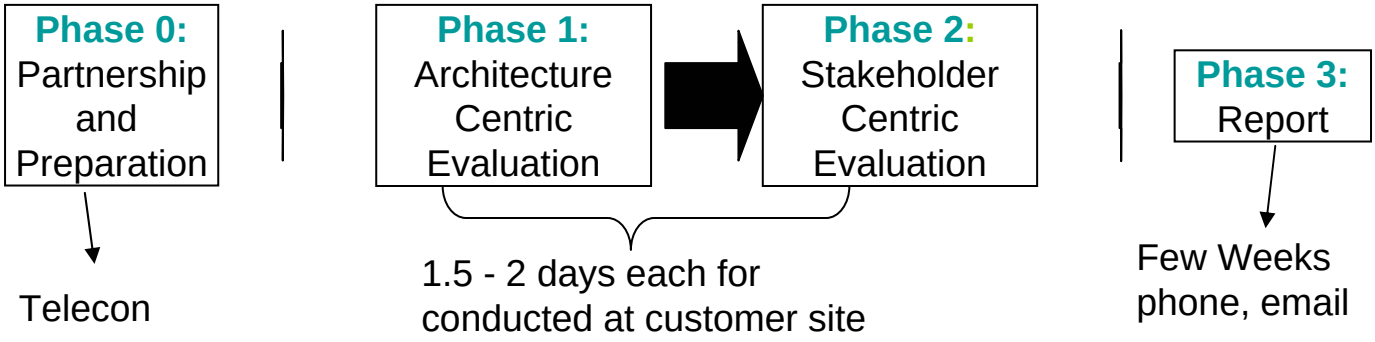
software architect
maintainer
tester
performance expert
security expert
project manager
customer (buyers, acquirers)
application builder
system administrator
service representative
system architect
developer
integrator
standards expert
reliability/availability expert
safety expert
product line manager
end user
mission specialist/planner
network administrator
domain representative
device H/W expert

What is an ATAM -1

Process

- Actors
  - sponsor (Program management) and architects (6)
  - Lead Evaluator – has lead evaluator training
  - Evaluation team (4)- all have taken ATAM training courses
  - Stakeholders (20)

Schedule



What is an ATAM -2

Technical Basis

- Business and Mission Drivers
  - New threats, capabilities, technology, automation, legacy
  - Scalability, schedules, budgets, joint, coalition, FMS
- There is a documented software architecture (SAD, UML Diagrams)
  - Multiple viewpoints, views, framework
- Quality attributes are the architecture drivers
  - Performance : avoid too slow, too late, bottlenecks
  - Availability : avoid fragility due to failures
  - Security : avoid spoofing, unauthorized access
  - Usability : avoid operator overload
  - Sustainability : avoid hard to update functions and new COTS
  - Interoperability, scalability, extensibility etc

What is an ATAM -3

Technical Basis (Continued)

- Scenarios represent the quality attributes
  - Stimulus, environment, response
  - " A tank commander's COP shows an identified threat, he has authorization to engage the threat, when it comes within his range he conducts a successful engagement and reports it via the COP".
  - Elicited in a meeting with stakeholders (or from previous QAW)
- Architectural approaches can be identified and analyzed

  Passive and active redundancy, publish/subscribe, client/server, reliable protocol
- Architectural Decisions
  - Provide a tool to assist with mapping spectrum allocation to force structure
  - Break down a system into components for transportation
  - Use a proxy-based pub/sub

What is an ATAM - 4

Technical Basis (Continued)

- Walking scenarios through the software architecture, and having the ATAM team and stakeholders probe the quality attributes exposes architectural risks and maps each risk to business drivers
- These risks can be “rolled up” into risk themes mapped to business drivers

Results- content

- A number of scenarios (10 to 15) are analyzed and documented
- Table of risks, trade-offs, programmatic issues, atta-boys
- Rollup of the risks into risk themes

Results- documents

- Summary Outbriefing after Stakeholder Phase (1 hour)
- Report (50, 60 pages) of findings with an Executive Summary (2 pages)

Commonalties and Differences -1

The System ATAM (including software) basically conforms to the ATAM process, technology, and results as follows

Process	**Actors**	**System and Software Architects** **Fast Tracking of subject matter experts (SME)** **SM designers**
	Phases	More careful scoping (what's in, what's out)
Technical	**Architecture**	**Need system (block diagrams) and software architecture views and white papers**
	Quality Attributes	A few additional QA (transportability, shake and bake, force modularity, spectrum management)
	Scenarios	Stress system aspects as well as software
	Analysis	Combination of system and software architects System Architectural Approaches
Results	No differences in either the outbriefing or the report	

Highlights of Experiences -1

ATAM

- Four 2 day courses providing the basic software architecture knowledge, including an ATAM team lead evaluator course
- Have conducted numerous ATAMS
- Have an ATAM Reference Guide for the team
- Have extensive set of templates to assist the team in all activities
- External organizations (commercial, DoD contractors) have qualified leads

SySATAM

- Have a process in-place for conducting SySATAMs
- Still in piloting Phase- have conducted 2 SySATAMs
- Have extensive set of templates to assist the team in all activities

Highlights of Experiences -2

SME Experiences

- On one system an Evaluation Team member was also an SME
- On the other the SME was a seasoned Mechanical Engineer and a domain expert
  - Took the SME training
  - Evaluation team had to initially prompt the SME for risks.

New Quality Attributes and associated risks

- Force Modularity, Mobility, Spectrum Management
- Logistics, installation, mechanical checks

New Considerations

- DoDAF operational views
- experimental simulation and analysis results
- white papers
- Manual versus automated activities are more prevalent

Highlights of Experiences -3

Architectural Representations

- System architecture documentation consists mainly of block diagrams and sequence diagrams and some DoDAF lower level views

Stakeholders

- System engineers tend to trump the software engineers
- Good exercise for system and software arch and eng to get on the same page

Surprises

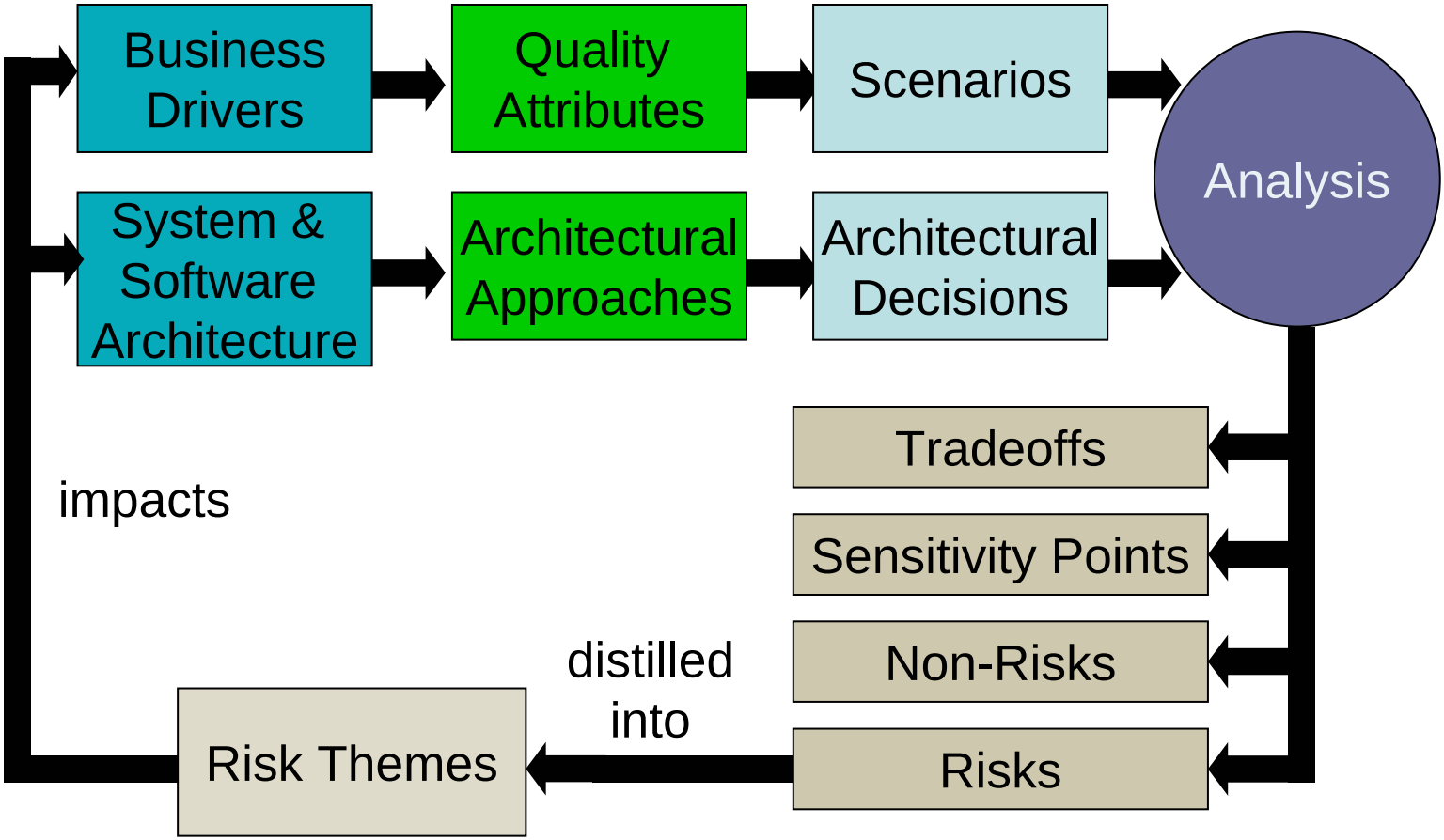
- Preparation phase was easier than expected, scoping was straightforward

Highlights of Experiences - 4

Typical Risk Themes

- There are a number of significant system engineering issues that require further analysis as a basis for architectural decision
- CONOPS for Using Programs has not been updated/supplemented to take this system into effect
- Architectural support for flexibility is powerful. However, without careful management of flexibility it could become overly complex and impose an unnecessary cognitive burden on users.
- Approach to automate and reduce test time not thought out
- Fault Tolerance approach needs to be developed

Conceptual Flow of ATAM



Conclusion

System ATAM is a natural extension to the ATAM

- Basic approach works just fine

SME is needed with functional/domain expertize

- Fast track training was effective

Risk Themes identified areas to help the programs choose what to explore to firm up the architecture

- Both software and system risks were revealed

Have been too busy “doing” to develop lessons learned

- But need to do more pilots first

For Additional Information

Jay Douglass
Business Development
Product Line Systems Program
Telephone: 412-268-6834
Email: jcd@sei.cmu.edu

Linda Northrop
Director
Product Line Systems Program
Telephone: 412-268-7638
Email: lmn@sei.cmu.edu

Technical Details:
Mike Gagliardi
Product Line Systems Program
Telephone: 412-268-7738
Email: mjg@sei.cmu.edu

U.S. Mail:
Software Engineering Institute
Carnegie Mellon University
4500 Fifth Avenue
Pittsburgh, PA 15213

World Wide Web:
http://www.sei.cmu.edu/architecture

SEI Fax: 412-268-5758