

Architecture Tradeoff Analysis Method

- Adding Agility to Architecture Tradeoff Analysis Method for Mapping on Crystal
 - Let's talk about the Intro and Agile Process Models in class
 - What is this???

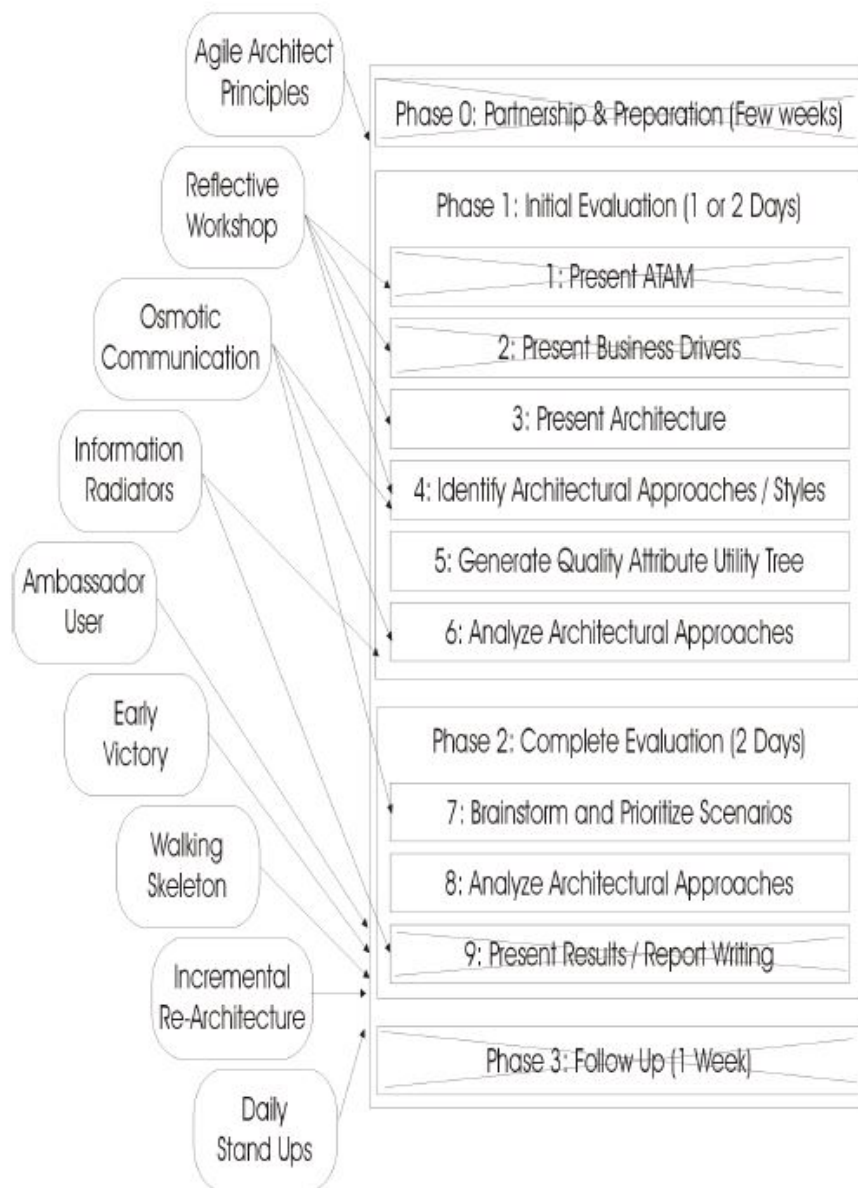
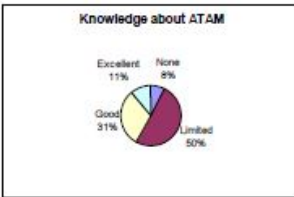
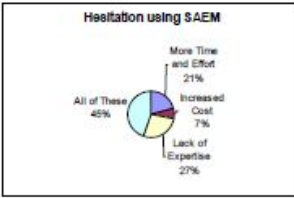
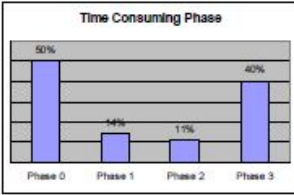


Table 1: Survey-1, Agile Process Models

SN	Question	Yes %	No %										
1	Do you know about agile process models for software development?	60	40										
2	Do you intend to use crystal or any other agile process in future?	56	44										
3	How would you rate your knowledge of crystal agile process model?	<div><p>Knowledge about Crystal</p><table><thead><tr><th>Knowledge Level</th><th>Percentage</th></tr></thead><tbody><tr><td>Excellent</td><td>0%</td></tr><tr><td>Good</td><td>26%</td></tr><tr><td>Limited</td><td>39%</td></tr><tr><td>None</td><td>35%</td></tr></tbody></table></div>		Knowledge Level	Percentage	Excellent	0%	Good	26%	Limited	39%	None	35%
Knowledge Level	Percentage												
Excellent	0%												
Good	26%												
Limited	39%												
None	35%												
4	What form of the agile processes are you most using at the time?	<div><p>Agile Model Used</p><table><thead><tr><th>Model</th><th>Percentage</th></tr></thead><tbody><tr><td>None</td><td>69%</td></tr><tr><td>XP</td><td>27%</td></tr><tr><td>Other</td><td>4%</td></tr><tr><td>Crystal</td><td>0%</td></tr></tbody></table></div>		Model	Percentage	None	69%	XP	27%	Other	4%	Crystal	0%
Model	Percentage												
None	69%												
XP	27%												
Other	4%												
Crystal	0%												
5	What feature of your agile model you like the most?	<div><p>Agile Feature</p><table><thead><tr><th>Feature</th><th>Percentage</th></tr></thead><tbody><tr><td>Respond to Change over Plan</td><td>10%</td></tr><tr><td>Close Communication with Customer</td><td>43%</td></tr><tr><td>Code over Documentation</td><td>29%</td></tr><tr><td>People over Process</td><td>28%</td></tr></tbody></table></div>		Feature	Percentage	Respond to Change over Plan	10%	Close Communication with Customer	43%	Code over Documentation	29%	People over Process	28%
Feature	Percentage												
Respond to Change over Plan	10%												
Close Communication with Customer	43%												
Code over Documentation	29%												
People over Process	28%												

– What do you see here that intrigues you?

**Table 2: Survey-2, Software Architecture
Evaluation Methods**

SN	Question	Yes %	No %
1	Is software architecture evaluation an important factor in delivering quality software product?	93	07
2	Do you know about SEI software architecture evaluation methods?	76	24
3	Are you using or intend to use ATAM in your organization in future?	69	31
4	Phase 0 and phase 3 of TAM could take more time if communication between customer and development / architecture evaluation team is delayed?	82	18
5	Effective communication channels can improve the time constraints?	89	11
6	Close involvement of customer with architecture evaluation team can discover customer likes/dislikes about architectural decisions easily?	82	18
7	A simple architecture evaluation method with minimum steps and less documentation can make the process fast and easy to use?	71	29
8	How would you rate your knowledge of ATAM?		
9	Why often teams hesitate to use software architecture evaluation methods such as ATAM, ARID, CBAM?		
10	Which phase you feel is more time consuming?		

- What do you see here that intrigues you?
- Conclusion: "We have applied agility to ATAM to make it suitable for crystal process model. We have proposed the complete elimination of phase 0 and 3, and steps 1, 2 and 9. Moreover, our analysis showed that there can be an overall light weight effect on ATAM."
 - * What exactly did they propose?
 - * How well did they justify that conclusion?

- Utility Trees

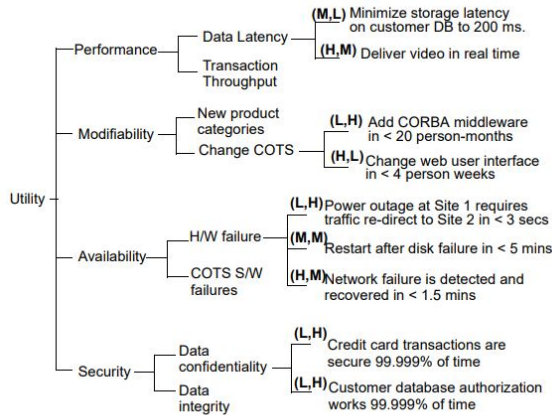


Figure 3: A Sample Utility Tree

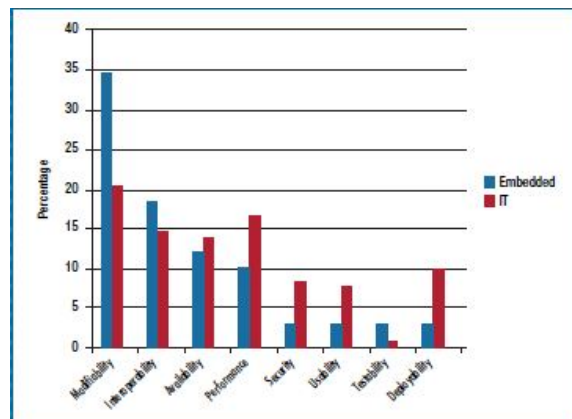
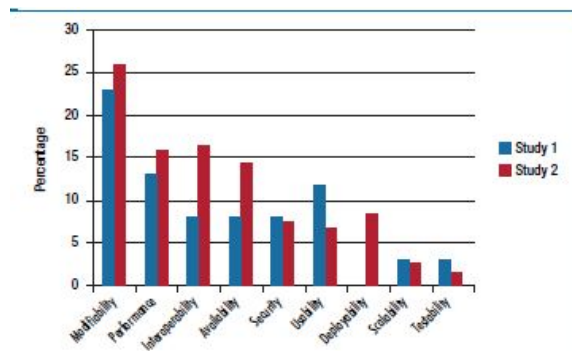
- Makes us add detail to quality attributes
- Levels are
 - * Broad, but related group (utility)
 - * Quality Attributes
 - * What we will measure
 - * Definition of success
 - Low, Medium, High for importance for success of the system and for perceived risk (how easy the architecture will make meeting this)
- Toward Agile Architecture
 - CONCERNS ABOUT LONG-TERM deterioration in large-scale agile projects' quality (called technical debt1) have renewed the agile community's interest in software architecture. (explain!)
 - Agile teams need to balance
 - * delivering near-term functional requirements (based on the agile philosophy of delivering user value early and often) and
 - * meeting near- and long-term quality attribute goals (without which the project can grind to a halt because system complexity makes efficient modifications impossible).
 - Claim: quality attribute prioritization can be difficult in early increments
 - Project goals
 - * obtain empirical evidence on the most common quality attributes
 - * and their relative importance
 - Data
 - * ATAM studies from 15 years of projects at SEI (31 total projects)

TABLE 1

Architecture Trade-Off Analysis Method project profiles.

Study	Time frame	Project type (%)			
		IT	Embedded	Government	Nongovernment
1	1999–2006	58	42	88	12
2	2006–2013	78	22	66	34

- * Data from utility trees
- * Study 1
 - Projects from 1999 - 2006
- * Study 2
 - Nine projects and 348 quality attribute scenarios (what is that??)
 - Projects from 2006 - 2013
- Methodology
 - * Collect the data into one dataset
 - * Code a sample of 100 records using a set of shared tags to assess and address variability across analysts
 - * Independently code the rest of the records and perform final reconciliation
 - * categorize, summarize and analyze the full data set
- Results



The top 20 quality attribute concerns from Study 2.

Rank (by no. of scenarios)	Quality attribute	Concern
1	Modifiability	Reducing coupling
2	Performance	Latency
3	Interoperability	Upgrading and integrating with other system components
4	Modifiability	Designing for portability
5	Usability	Ease of operation
6	Availability	Fault detection
7	Interoperability	Ease of interfacing with other systems
8	Modifiability	Designing for extensibility
9	Availability	Fault recovery
10	Performance	Resource management
11	Deployability	Minimizing build, test, and release duration
12	Modifiability	Reusability
13	Availability	Preventing faults
14	Scalability	Increased processing demands
15	Security	Authorization
16	Interoperability	Resource and data sharing
17	Security	Resisting attacks
18	Deployability	Configuration or dependency management
19	Modifiability	Configurability and composability
20	Deployability	Backward compatibility or a rollback strategy

- * Then they give a summary of modifiability, performance, availability and deployability concerns

- * Deployability is an "emerging" concern

– Using the Data

- * They propose these things:

- "Incorporate high-ranking quality attributes and concerns as checklists at designated points during the incremental life cycle"
- "Weaving quality attribute concerns into the life cycle" Maybe during retrospectives or planning

- * Basically, they are calling for ATAM to be done/revised during each iteration