IT Architect Regional Conference

Enterprise Innovation through Architecture Resilience

16 – 17 July 2009, The Legend Hotel

ITARC Malaysia 2009

Session Topic: Architecting with Right Quality Attributes in Mind

Presented By: Max Poliashenko

IASA Atlanta, IASA Fellow

The use, disclosure, reproduction, modification, transfer or transmittal of this work without the written permission of IASA is strictly prohibited. @ IASA 2006

Session Objectives And Takeaways

- Understand crucial role of Quality Attributes in solution architecture:
  - Overview of principal 15 Quality Attributes
  - Discussion of ways to improve each QuAt
  - Understand why and how QuAts can compete
  - Overview of ATAM and CBAM analysis methods
- Gather QuAt requirements proactively
- While improving one QuAt, consider impact on other Systemic Qualities

MSC MALAYSIA National ICT Initiative

IASA

The use, disclosure, reproduction, modification, transfer or transmittal of this work without the written permission of IASA is strictly prohibited. @ IASA 2006

A typical story of your first project

- You gathered all functional requirements
- You designed and developed a solution
- You fixed all major defects, and it passed QA
- You delivered!!!
- Deployment glitches? Slow start-up and performance?
- Concurrency locks? Security troubles?
- Scalability worries?
- What did go wrong?

MSC MALAYSIA National ICT Initiative

IASA

The use, disclosure, reproduction, modification, transfer or transmittal of this work without the written permission of IASA is strictly prohibited. @ IASA 2006

Non-functional requirements!

Examples:

- start-up time, memory footprint, how fast it can process data input
- how many database connections it uses at any given time, how many concurrent users it can serve before they start noticing significant slow-down in system response
- how it protects its critical and sensitive information, what authentication mechanisms it uses
- if it can work in a different country, how it handles updates, and so on…

- These are usually called Quality Attributes, Quality of Service or Systemic Qualities

The use, disclosure, reproduction, modification, transfer or transmittal of this work without the written permission of IASA is strictly prohibited. @ IASA 2006

Quality Attributes

- They are called by different names
- Their number and importance depend on the operational context
- Should define requirements (explicitly and preferably quantitatively) for all Quality Attributes important in a given context before the design
- They may become requirements for Framework or Platform (do you need business requirements to build an architecture?)
- These attributes sometimes compete, hence, need to define relative priority
- Design for the right QuAts rather having them shaped by a chance!

MSC MALAYSIA National ICT Initiative

IASA

The use, disclosure, reproduction, modification, transfer or transmittal of this work without the written permission of IASA is strictly prohibited. @ IASA 2006

Quality Attributes (QuAt)

- End-User Quality Attributes:
  - **Performance**
  - **Reliability/Availability/Resiliency/Recoverability**
  - **Security**
  - **Usability**
  - **Globalization**

- Other attributes mostly impact system development cost - we can call them Development Quality Attributes:
  - **Maintainability/Extensibility**
  - **Flexibility/Modularity**
  - **Testability**

- Qualities that impact operational cost while being less visible to the end-users. We can call them Operational Quality Attributes:
  - **Scalability**
  - **Supportability**
  - **Interoperability**
  - **Portability**

The use, disclosure, reproduction, modification, transfer or transmittal of this work without the written permission of IASA is strictly prohibited. @ IASA 2006

Are these same as ASRs?

- What are Architecturally Significant Requirements?
  - Quality Attributes - yes!
  - Architecturally significant functional requirements
  - Constraints (technology and policies)
    - Supporting software and hardware platforms
    - Implementation languages and frameworks
    - Third-party products and components
    - Existing internal and external systems
    - Resource, skills and operational limitations
    - Standard compliance
    - Enterprise policies, standards and guidelines

The use, disclosure, reproduction, modification, transfer or transmittal of this work without the written permission of IASA is strictly prohibited. @ IASA 2006

Performance

- Response time, latency, capacity, throughput
- Perceived performance (responsiveness)
- Start with qualitative requirements
- Try to make them quantitative. How?
  - Be as specific as you can about the hardware and the environment (data size)
  - Use typical environment easily re-creatable in your lab
  - Set your target numbers by
    - Measuring reference applications
    - Running prototypes
    - Asking customers or Prod Mgrs for their wishes

MSC MALAYSIA National ICT Initiative

IASA

The use, disclosure, reproduction, modification, transfer or transmittal of this work without the written permission of IASA is strictly prohibited. @ IASA 2006

Performance

- Use requirements to shape your architectural decisions: to validate or eliminate certain choices, etc.
- Measure performance early and often
  - Create benchmarks and run them often, monitoring trends
  - Use QA automation tools
  - Pay particular attention to start-up performance (both cold and warm)
- What to do if you are falling short of requirements?

The use, disclosure, reproduction, modification, transfer or transmittal of this work without the written permission of IASA is strictly prohibited. @ IASA 2006

Performance Tuning

- Get a good profiling tool – it can show you unexpected things
- Address performance problems in the order of their impact and importance
- Tune database schema, queries
- Minimize network traffic and chattiness
- Use caches and snapshots
- If nothing helps - improve responsiveness: use asynchronous calls, multithreading, on idle processing, batch jobs
- Address start-up performance

MSC MALAYSIA National ICT Initiative

IASA

The use, disclosure, reproduction, modification, transfer or transmittal of this work without the written permission of IASA is strictly prohibited. @ IASA 2006

Scalability

- **Scalability** is a desirable property of a system, a network or a process, which indicates its ability to either handle growing amounts of work in a graceful manner, ***and*** to be readily enlarged
  - Vertical (scaling up): the system can take advantage of more power on the same CPU, such as more RAM, faster hard disk, etc.
  - Horizontal (scaling out): the system can benefit from additional CPUs in its environments (farms or clusters)

The use, disclosure, reproduction, modification, transfer or transmittal of this work without the written permission of IASA is strictly prohibited. @ IASA 2006

Scalability

- Ironically, scalability often competes with performance
- Scalability must be architected:
  - Use distributed architectures
  - Use stateless interfaces
  - Use component pools
  - Use optimistic rather pessimistic locking
- Poor performance is more noticeable than poor scalability
- Build-in degradation of capabilities at the scalability limits

The use, disclosure, reproduction, modification, transfer or transmittal of this work without the written permission of IASA is strictly prohibited. @ IASA 2006

Reliability/Availability

- Most systems experience failures in their lifetime that can be analyzed and predicted statistically
- Reliability as a quality attribute that describes how long on average a software system can function as designed before it experiences failures in a given operational environment
- An alternative definition: how often a software system experiences failures in a given operational environment
- Common metrics used to describe Reliability is Mean Time Between Failures (MTBF)
- Availability: describes percentage of time that the system is fully functional in typical operational environment, e.g. 99.9%
- Resiliency, Recoverability, Accuracy

MSC MALAYSIA National ICT Initiative

IASA

The use, disclosure, reproduction, modification, transfer or transmittal of this work without the written permission of IASA is strictly prohibited. @ IASA 2006

Architect for Reliability/Availability

- Start with the process (SDLC, CMM)
  - Project Planning, Monitoring and Control. Chaotic projects are not known to produce great software
  - Measurement and Analysis: what can't be measured cannot be improved
  - Solid mechanism for gathering, documenting, validating and managing requirements. Without clear and consistent requirements, your system is unlikely to achieve high quality
  - Quality Assurance. This must be pervasive throughout the process. Otherwise, how do you know that you are meeting the requirements? Developers told you so? Trust but verify
  - Robust Source Control and Configuration Management. You must control what goes into your code
  - Risk Management. Every project has its risks, they must be managed, or Murphy's law rules
  - Organizational Training. Quality code is produced by people who know what they are doing

MSC MALAYSIA National ICT Initiative

IASA

The use, disclosure, reproduction, modification, transfer or transmittal of this work without the written permission of IASA is strictly prohibited. @ IASA 2006

Architect for Reliability/Availability

- Other processes:
  - PSP/TSP from SEI
  - Six Sigma
  - Test-driven development
  - Strong architectural involvement and governance
- Get reliability requirements
- Do resiliency analysis
- Do risk analysis
- Leverage redundancy
- Create fail-over mechanisms, plan for disasters
- Build monitoring facilities, instrumentation
- Fault Tree Analysis, Fault Seeding or Reliability Confidence Limits

MSC MALAYSIA National ICT Initiative

IASA

The use, disclosure, reproduction, modification, transfer or transmittal of this work without the written permission of IASA is strictly prohibited. @ IASA 2006

Security

- *Security Quality Attributes of an IT system describe its ability to protect the confidentiality of customer data as well as integrity and availability of both the data and the system itself*
- The cost of securing an IT system usually grows rapidly as the level of protection is increased, similarly to that of availability
- Depending on whether you are designing an IT system for a bank or a soccer club, the Quality Attributes requirements may be vastly different

MSC MALAYSIA National ICT Initiative

IASA

The use, disclosure, reproduction, modification, transfer or transmittal of this work without the written permission of IASA is strictly prohibited. @ IASA 2006

Discovering Security requirements

Sample questions to ask

- What is the environment that this system will operate in (office intranet, home, internet, etc.) and is it going to interoperate with other applications outside of this environment?
- Who are the typical users, what do they do and what kind of data and documents they care about protecting?
- Is access to the application restricted? If so, how is access granted, to anyone who knows the password or must user be authenticated first?
- Is all the system data accessible to all the users or is access granted selectively and if so, by what criteria?
- What happens if the protected information is compromised?
- Will the users of the application use it differently depending on their roles?
- Who will install the application and serve as its admin?
- Should certain application data be protected from some users?

MSC MALAYSIA National ICT Initiative

IASA

The use, disclosure, reproduction, modification, transfer or transmittal of this work without the written permission of IASA is strictly prohibited. @ IASA 2006

Security good practices

- Start modeling security threats. You can start by identifying assets that you want to protect, and any external interaction points that may be used to threaten their confidentiality, integrity or availability
- Identify possible attack vectors, assessing the risks of threats, prioritizing them, and planning the responses and mitigations
- When planning security measures, remember that you may always be at disadvantage compared to the attackers because of the following reasons:
  - You have to defend all the vulnerable points, while they can pick which one to attack. This means that while you are only prepared against the known attack vectors and threats, they can keep searching for new ones. So should you
  - You must always have your defense on, while they can attack whenever they please
  - You cannot break the law and are subject to various rules and regulations while they are not

MSC MALAYSIA National ICT Initiative

IASA

The use, disclosure, reproduction, modification, transfer or transmittal of this work without the written permission of IASA is strictly prohibited. @ IASA 2006

STRIDE approach to security

- Spoofing. Is there any way for an attacker to pose as a valid user or as a valid server or any other authorized actor? Holes in authentication mechanisms, DNS cache poisoning, or susceptibility to replay attacks can lead to such vulnerabilities
- Tampering. Unauthorized or malicious data modifications may happen to the files, data in the database or data transmitted over the wire
- Repudiation. This is not a very sophisticated threat. Basically, a dishonest user denies performing certain actions (like placing an order) exploiting the fact that your application cannot prove otherwise. Usually, weak authentication and lack of the right level of auditing may allow for this
- Information disclosure. Compromising information to other authorized or unauthorized users may happen due to weak authentication and poor authorization mechanisms or caused by spoofing
- Denial of Service (DoS). This is a very nasty attack, which overwhelms a system with a flood of requests affecting its availability
- Elevation of privilege. This very dangerous situation may happen both within your application or in the environment in which it operates. The attacker gains the status of a trusted user and the amount of damage can be proportional to a power that a system administrator has in your application

MSC MALAYSIA National ICT Initiative

IASA

The use, disclosure, reproduction, modification, transfer or transmittal of this work without the written permission of IASA is strictly prohibited. @ IASA 2006

Security good practices and counter-measures

- **Authentication**. In many cases you want to keep track of any user or external system that interacts with your application. This allows you to enable authorization and auditing (discussed later) which, in turn, help to improve your Security
- **Authorization**. Authenticating a user doesn't automatically make your application secure or its sensitive assets protected. However, it gives you an identity principal that you can use to enforce application access rights and privileges
- **Application level privileges** can be administered individually, or via user roles with user principals inheriting privileges from their roles
- **Running with least privilege**. Use separate built-in admin accounts to do admin tasks such as creating installing updates, new tables, columns or doing data backup

MSC MALAYSIA National ICT Initiative

IASA

The use, disclosure, reproduction, modification, transfer or transmittal of this work without the written permission of IASA is strictly prohibited. @ IASA 2006

Security good practices and counter-measures

- **Storing sensitive data**. The best way to do this is actually not storing it at all. Try to store instead some of its derivatives, such as hashes or digests, rather than the data itself
- **Encryption**. If hashing is not an option and you must be able to reverse the sensitive information, such as credit card numbers, you need to encrypt it
- **Use SSL** (Secure Sockets Layer) or TLS (Transport Layer Security) for communicating potentially sensitive data across the network. This applies both to internet protocols and to the TCP traffic on a LAN
- **Filtering and throttling**. These techniques offer protection against DoS attacks

MSC MALAYSIA National ICT Initiative

IASA

The use, disclosure, reproduction, modification, transfer or transmittal of this work without the written permission of IASA is strictly prohibited. @ IASA 2006

Security good practices and counter-measures

- **Auditing and logging**. Auditing is very important to most applications as a tool to keep control and accountability of the application data changes as well as a way to revert and repair unwanted changes, done mistakenly or on purpose
- **Use session tokens** when defending against a replay attack
- **Validate all input.** Security experts often advise: You have to assume that all input is evil until proven otherwise. Be mindful of code injection
- **Check your bounds**. If you use older languages make sure you check the size of the data before copying it
- **Do not hardcode any keys.** Once you have decided to use one of the encryption algorithms, MAC or salted hashes, the next question is: where do I store the secret key? Once the key becomes known to one person, it will get exposed to the entire world in no time
- Finally, **do not rely solely on security through obscurity** - once it is out, there is no easy way to fix it
- To use or not to use biometrics?

MSC MALAYSIA National ICT Initiative

IASA

The use, disclosure, reproduction, modification, transfer or transmittal of this work without the written permission of IASA is strictly prohibited. @ IASA 2006

Usability

- Is it architect's job?
- Usability sometimes may be more important then performance
- Your success lies in solving the right problem (sometimes, less is more)
- UI architecture should not be an afterthought
  - Accessibility and Help Facilities
  - Navigability and Feature Discovery
  - Consistency and Aesthetics
- Architects should be aware of usability workflow

The use, disclosure, reproduction, modification, transfer or transmittal of this work without the written permission of IASA is strictly prohibited. @ IASA 2006

Globalization/Localization

- A true globalization story
- What things do you need to take care of when localizing a product:
  - Different formats: date, time, numbers, money
  - Calendars, right-to-left layout, symbols
  - Language: text and graphics
    - Non-Latin based alphabets, other character sets
    - multi-byte code pages, collations
    - Encoding standards (UTF 8,16…)
    - String comparison:
      - Straße and Strasse; «мёд» and «мед»
      - “しんかんせん” and “シンカンセン”
    - Sorting order: “**Österrike**” and “**Österreich**”

MSC MALAYSIA National ICT Initiative

IASA

The use, disclosure, reproduction, modification, transfer or transmittal of this work without the written permission of IASA is strictly prohibited. @ IASA 2006

Effective translation practices

- Factor out of the code all the text and strings that the end-user may see. Sometimes, it is difficult to translate a word without knowing the context
- Treat two strings that appear in different places as separate resources even if they say the same. Same words in one language may be translated differently into another language depending on its context
- Avoid forming messages by concatenating strings. Use formatting functions instead
- Always have a native speaker proofread the translated text, best in the context as logically close to the real UI as you can
- When your application cannot find the right resource string that should have been translated, don't throw an error or ignore it. In production, have a fall-back string (like English phrase). In debug mode, have it output instead some highly noticeable text (red "XXXX", for example)
- Leave at least 30% more text space unused to accommodate more verbose languages

MSC MALAYSIA National ICT Initiative

IASA

The use, disclosure, reproduction, modification, transfer or transmittal of this work without the written permission of IASA is strictly prohibited. @ IASA 2006

Locales and Cultures

- Locales are cryptic and relationships are lost:
  - Italian in Switzerland LCID=2064
  - Italian in Italy LCID=1040
  - French in Switzerland LCID=4108
- Cultures (.NET): language-COUNTRY-Alphabet
  - **sr-SP-Cyrl** – culture: Serbian language, located in Serbia and using Cyrillic alphabet
  - DateTime.Now.ToLongDateString(), will produce: "**Samstag, 17. März 2007**" fot "de-CH" culture
  - Organized in hierarchies: "zh-CN" -> "zh-CHS" -> "zh"
  - .NET is culture aware and knows how to fall back
- How do you use non-standard cultures like Spanish in US?

MSC MALAYSIA National ICT Initiative

IASA

The use, disclosure, reproduction, modification, transfer or transmittal of this work without the written permission of IASA is strictly prohibited. @ IASA 2006

Maintainability/Extensibility

- IEEE more formally defines it as:

"…the ease with which a software system or component can be modified to correct faults, improve performance, or other attributes, or adapt to a changed environment"

- Nowadays, it is universally accepted that Maintainability is not just a good idea, it is a business requirement for many of the new applications and products

The use, disclosure, reproduction, modification, transfer or transmittal of this work without the written permission of IASA is strictly prohibited. @ IASA 2006

Principles of Maintainability

A lot of the principles and techniques of better software maintainability are the same as for better flexibility and agility:

- Modularity and componentization
- Weak coupling between components
- Multi-tier or layered structure
- Data encapsulation
- Object orientation
- Aspect oriented programming
- Investing in a good Architecture

MSC MALAYSIA National ICT Initiative

IASA

The use, disclosure, reproduction, modification, transfer or transmittal of this work without the written permission of IASA is strictly prohibited. @ IASA 2006

Best Practices for Maintainability

Best practices that are unique to Maintainability and emphasize the process or internal governance:

- Invest in SDLC and formal Architecture role.
- Have coding standards, guidelines and naming conventions..
- Design before you code
- Use known design patterns, make code intent clear
- Use a layered architecture and object encapsulation whenever possible – this isolates changes to contained areas
- Keep good documentation, including code comments, engineering design documents and architectural artifacts
- Leveraging or investing in Frameworks. Frameworks are great things
- Code Generation / Software Factories
- Invest in unit tests – they reduce risk during code modifications
- Simplicity (KISS principle)

MSC MALAYSIA National ICT Initiative

IASA

The use, disclosure, reproduction, modification, transfer or transmittal of this work without the written permission of IASA is strictly prohibited. @ IASA 2006

Flexibility/Modularity

- Business environment always evolves. New requirements arise, old requirements get obsolete
- *Flexibility: the ability of an IT system to undergo modifications and adaptation to a new environment or purpose different from which it was designed for*
- Maintainable system is a pre-requisite for flexibility. However, there are some differences between the two. For example, the simplicity principle is often compromised when designing for Flexibility
- There is also another a specific kind of Flexibility – Modularity: *ability of the system parts, modules, to function independently*

MSC MALAYSIA National ICT Initiative

IASA

The use, disclosure, reproduction, modification, transfer or transmittal of this work without the written permission of IASA is strictly prohibited. @ IASA 2006

Modularity/Reusability

- Another technique to achieve modularity is Service Oriented Architecture (SOA) that is based on several principles:
  - Service encapsulation
  - Service loose coupling
  - Service contract
  - Service documentation
  - Service discoverability
  - Service composability
  - Service independence and autonomy
- Dependency Injection pattern (CAB, Spring)

The use, disclosure, reproduction, modification, transfer or transmittal of this work without the written permission of IASA is strictly prohibited. @ IASA 2006

Portability

- *Degree of re-engineering and configuration effort required to install and run the same software application in different technological environments*
- The difference with Flexibility is that while flexibility usually involves ability to adapt to changed environment and ways of usage, portability is only concerned with a changed environment without changing the way the end-users interact with the application
- Like Flexibility, Portability may conflict with the simplicity principle of Maintainability. It also can negatively affect Performance
- Portability can also be applied to the source code, UI, documentation, or even design (example: MDA)

The use, disclosure, reproduction, modification, transfer or transmittal of this work without the written permission of IASA is strictly prohibited. @ IASA 2006

Portability

- Portability holds the promise of offering benefits to a variety of stakeholders: lower overall cost of development, maintenance, and support for an IT organization, as well as familiarity with UI and portability of skills for the end-user
- It can also increase some of the IT costs as it both may restrict technological options and make it difficult to satisfy requirements with respect of other Quality Attributes
- It is important to do a careful cost-benefit analysis of Portability requirements
- Sometimes, a separate solution taking advantage of re-use, Flexibility and Modularity may be a less costly option

The use, disclosure, reproduction, modification, transfer or transmittal of this work without the written permission of IASA is strictly prohibited. @ IASA 2006

Testability

- Contemporary software systems are fairly complex in nature: comprised of many components, may run in a distributed, real time or asynchronous environment, may include interactions with users or external systems through a variety of interfaces and APIs. There are many opportunities for the unexpected, when something may go wrong. At the same time, any defect in the software is a potential security vulnerability
- A single defect can impact the bottom line
- By the IEEE definition, Testability "…is the degree to which a system or component facilitates the establishment of test criteria and the performance of tests to determine whether those criteria have been met"

MSC MALAYSIA National ICT Initiative

IASA

The use, disclosure, reproduction, modification, transfer or transmittal of this work without the written permission of IASA is strictly prohibited. @ IASA 2006

Testability Best Practices

- Give QA visibility early on, invite to the meetings, provide read access to design artifacts
- Instrument applications, provide QA with harnesses and test interfaces
- Sprinkle code with Asserts
- Robust event and error logging facilities
- Some instrumentation may be turned off in production
- Build suites of unit tests
- Build suites of automation tests
  - they can also be used for BVT, regression tests, benchmarks

The use, disclosure, reproduction, modification, transfer or transmittal of this work without the written permission of IASA is strictly prohibited. @ IASA 2006

Supportability

- Total cost of ownership is an important attribute of any software
- *Effectiveness of operating and maintaining your application in its production environment*
- This is an overarching Quality Attribute
- Supportability may include:
  - *installability, configurability, updateability, upgradability*
  - *trainability, serviceability, auditability, ease of administration and of migration*
  - *error logging, ease of troubleshooting, self-healing capabilities*
  - *and, finally, uninstall*

The use, disclosure, reproduction, modification, transfer or transmittal of this work without the written permission of IASA is strictly prohibited. @ IASA 2006

Supportability Best Practices

- Start with installation. Even a small installation hick-ups could potentially flood the Customer Support center
- Flexible configurability may also make it easy to adapt to new environments and repair installation defects
- Good instrumentation and robust error logging facilities
- Problems prevention – not only resolution. Analyze instrumentation data
- Self-repairing and auto-healing tools
- Monitoring tools
- Quality and effectiveness of documentation
- Design for running side-by-side with older versions and in multi-tenant environment

MSC MALAYSIA National ICT Initiative

IASA

The use, disclosure, reproduction, modification, transfer or transmittal of this work without the written permission of IASA is strictly prohibited. @ IASA 2006

Interoperability

- Your application is not alone but rather coexists with a variety of other business applications that are also very important to the user
- Many companies tried to “own” the customer. Customers prefer Best of Breed instead
- Just like consumers grew to expect export/import level of interoperability between their productivity suite applications, they will soon expect true interoperability from them
- Your app must interoperate with other applications that the customer uses and to play with them nicely or else…

MSC MALAYSIA National ICT Initiative

IASA

The use, disclosure, reproduction, modification, transfer or transmittal of this work without the written permission of IASA is strictly prohibited. @ IASA 2006

Interoperability Best Practices

- Develop a full API that exposes most of the business functionality to other applications. This doesn't mean having all your internals widely available or opening all the powerful features to anonymous access. In fact, the API should be secured in the same fashion as your UI
- Provide adaptors to your API using all the most popular technologies such as COM, .Net, Java, Web Services, etc. This way you would allow other applications to talk to you even if you were not ready to talk to them
- Use standards for your API, as much as they are applicable. This would allow your customers to string various applications of their business suite together even without you necessarily knowing about the rest of the applications.
  - SOA is a powerful architecture style that promotes Interoperability

MSC MALAYSIA National ICT Initiative

IASA

The use, disclosure, reproduction, modification, transfer or transmittal of this work without the written permission of IASA is strictly prohibited. @ IASA 2006

When Quality Attributes Compete

- Quality Attributes of your system are mainly determined by its architecture which is very expensive to change, once implemented
- This is why it is important to know your non-functional requirements upfront, so you can incorporate them in your system's architecture
- These requirements can affect not only high-level architectural decisions, such as the choice of application platform and technologies, but also they should guide your low-level design and trade-off considerations as you implement a system's functional requirements
- As you start making architectural decisions, you will almost certainly run into situations when improving one Quality Attribute hurts the others

MSC MALAYSIA National ICT Initiative

IASA

The use, disclosure, reproduction, modification, transfer or transmittal of this work without the written permission of IASA is strictly prohibited. @ IASA 2006

Quality Attributes Trade-off and Sensitivity points

- Architectural decisions that may impact more than one attribute are called *trade-off points*. In contrast, decisions that "qualitatively" affect single attributes are called *sensitivity points*
- The discussion around trade-off and sensitivity points may uncover missing requirements and improve communication between stakeholders as well as identify new risks
- The impact of decisions promoting one Quality Attribute on the others is not always the same: improving Performance may improves Security or hurts, depending on what system parameters are being altered
- Trade-off relationships are not always reflexive
- Because of the trade-off points, it is sometimes difficult to architect all the Quality Attributes requirements in a linear fashion. This is why an iterative design can be useful here
- Trade-off and sensitivity points are also very important in ATAM methodology developed by SEI at Carnegie Mellon
- ATAM is complemented by Cost Benefit Analysis Method (CBAM)

MSC MALAYSIA National ICT Initiative

IASA

The use, disclosure, reproduction, modification, transfer or transmittal of this work without the written permission of IASA is strictly prohibited. @ IASA 2006

Quality Attributes Trade-off Matrix

	Performance	Scalability	Reliability	Security	Usability	Globalization	Maintainability	Flexibility	Portability	Testability	Supportability	Interoperability
Performance				+	+							
Scalability	-		+					+				
Reliability	-			+							+	
Security	-		+		-	-		-	-	-	-	-
Usability	-			-				-	-	-		
Globalization	-		+		+		+		-	-	-	
Maintainability	-		+	+						+		
Flexibility	-	+	-	-	-		+		+			
Portability	-		-	-	-		-	+			+	+
Testability	-		+	-			+		+		+	
Supportability	-		+	-			+					
Interoperability	-			-			-	+	+		-	

The use, disclosure, reproduction, modification, transfer or transmittal of this work without the written permission of IASA is strictly prohibited. @ IASA 2006

Sample Questions eliciting QuAt Requirements

- Security: Are all the users inside the intranet? Are different user roles required?
- Globalization: Which languages and locales are to be supported?
- Usability: Do you have existing systems which this system replaces? What is the level of sophistication of a typical user?
- Performance: Are there any requirements on start-up time? How about warm start-up?
- Reliability: Are you planning to have SLAs?

MSC MALAYSIA National ICT Initiative

IASA

The use, disclosure, reproduction, modification, transfer or transmittal of this work without the written permission of IASA is strictly prohibited. @ IASA 2006

Example of Utility Tree from ATAM

- Utility
  - Security
    - Authentication
      - (M,H) Integrate with customer SSO
      - (L,M) Enforce strong passwords
    - Authorization
      - (L,H) Use role authorization
      - (H,L) Allow user-defined roles
    - Auditing
      - (H,H) Comply with SOX
    - Data Encryption
      - (L,H) Use 128 bit encryption
      - (H,L) Recover data if customer looses key
      - (L,H) Only customer reference data needs encryption
  - Availability
    - End-user availability
      - (H,H) 99.9% availability
    - Partner channel availability
      - (L,M) 95% during work days
  - Performance
    - Start-up
      - (H,H) Cold start time < 5 second
      - (M,H) Warm home page load < 1 sec
    - Transaction Throughput
      - (L,H) < 1 sec for internal clients
    - Response Latency
      - (H,H) < 1 sec for internal clients
      - (L,L) < 5 sec for external clients

- Arrange QuAt requirements as scenarios grouped by QuAt in a tree and assign them Complexity and Importance weight
- Concentrate on scenarios with High complexity or importance
- Look for trade-off or sensitivity points

The use, disclosure, reproduction, modification, transfer or transmittal of this work without the written permission of IASA is strictly prohibited. @ IASA 2006

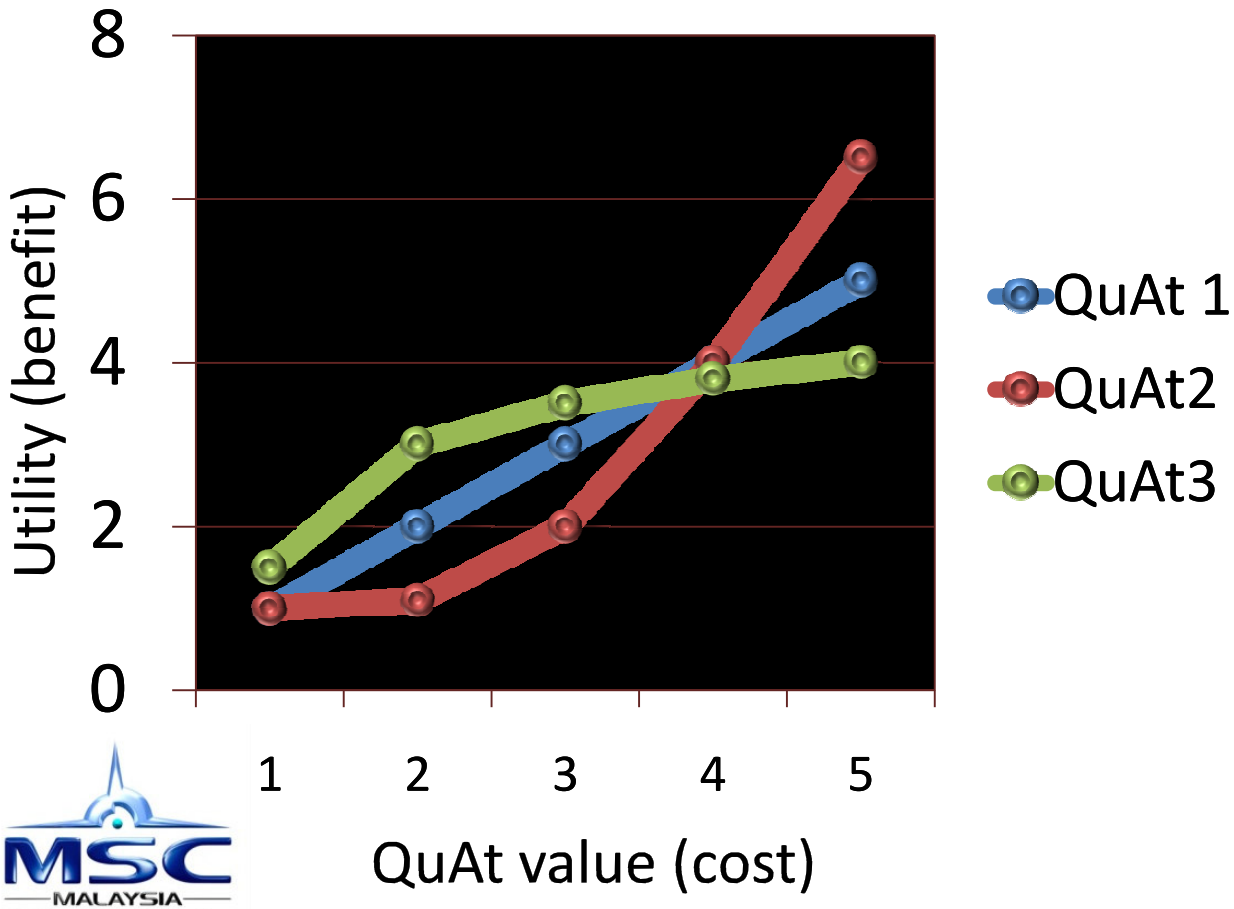
Outputs of ATAM

- A concise presentation of architecture
- Articulation of business goals
- QuAt requirements in terms of scenarios
- Mapping of arch decisions to QuAt reqs
- A set of sensitivity and trade-off points
- A set of risks and non-risks
- A set of risk themes

The use, disclosure, reproduction, modification, transfer or transmittal of this work without the written permission of IASA is strictly prohibited. @ IASA 2006

Cost Benefit Analysis Method (CBAM)

So, you identified Trade-off points:
<u>which way should you trade</u>?



Steps in CBAM:

1. Prioritize scenarios
2. Assign scenario utility profile
3. Develop arch strategies to meet QuAt response levels
4. Determine ROI of each strategy
5. Calculate the total benefit
6. Select winning strategies

MSC MALAYSIA National ICT Initiative

IASA

The use, disclosure, reproduction, modification, transfer or transmittal of this work without the written permission of IASA is strictly prohibited. @ IASA 2006

Summary

- In order to create a successful solution, you must deliver architecture that meets numerous non-functional requirements and is build for right Quality Attributes
- Make sure to have such requirements, or help to define them proactively
- Making architectural decisions at sensitivity and trade-off points is not a straight forward task. ATAM and CBAM methods can help
- Architectural decisions may be influenced by various stakeholders and can be motivated by non-technical reasons such as strategic business concerns, political considerations, or by limitations in skill set, capacity and schedules
- You can evaluate the impact of such influences and either accept them as constraints or push back explaining why such decisions may prevent achieving required levels of Quality Attributes, thus averting a disaster at a later time

MSC MALAYSIA National ICT Initiative

IASA

The use, disclosure, reproduction, modification, transfer or transmittal of this work without the written permission of IASA is strictly prohibited. @ IASA 2006

IT Architect Regional Conference

Enterprise Innovation through Architecture Resilience

16 – 17 July 2009, The Legend Hotel

Q & A

THANK YOU

MSC MALAYSIA National ICT Initiative

IASA

The use, disclosure, reproduction, modification, transfer or transmittal of this work without the written permission of IASA is strictly prohibited. @ IASA 2006