QUality Assessment of System Architectures and their Requirements (QUASAR)
Version 3.1

Donald Firesmith
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

NDIA 12th Annual Systems Engineering Conference
29 October 2009
Topics

Requirements and Architecture Challenges

Underlying Concepts

QUASAR Method

Reasons to use QUASAR
Requirements and Architecture Challenges

Requirements and Architecture are the first two Opportunities to make Major Engineering Mistakes.

Architecturally Significant Requirements are typically poorly engineered.

Architecture and associated Architecturally Significant Requirements Affect:

- Project Organization and Staffing (Conway’s Law)
- Downstream Design, Implementation, Integration, Testing, and Deployment Decisions

A common project-specific Quality Model is needed to drive the

- Quality Requirements, which drives the
- Quality of the System Architecture, which drives the
- Quality of the System
Requirements and Architecture Challenges

Architecturally-Significant, Quality-Related Requirements and their associated Architectural Decisions *Drive* the System and Component:

- Ultimate Quality
- Development Schedule
- Development Costs
- Sustainment Costs
- Maintainability and Upgradeability
- Acceptance and Usage by Stakeholders
Requirements and Architecture Challenges

It is important to identify (and thereby help Manage) Risks:

- Requirements and Architecture Risks
- System and Project Risks
- Business Risks

It is important to provide Acquirer/Management:

- Visibility into
- Oversight over
  the System and Component Requirements and Architecture

It is important to determine Compliance:

- Requirements and Architecture with Contract (Acquirer) Requirements
- Architecture with System and Component (Developer) Requirements
Topics

Requirements and Architecture Challenges

Underlying Concepts

QUASAR Method

Reasons to use QUASAR
What is Quality?

Quality

the Degree to which a Work Product (e.g., System, Subsystem, Requirements, Architecture) Exhibits a Desired or Required Amount of Useful or Needed Characteristics and Attributes

Not just lack of defects!

Question:

What Types of Characteristics and Attributes are these?

Answer:

They are the Characteristics defined by the Project Quality Model.
Quality of a Work Product is defined in terms of a **Quality Model**:

- **Quality Characteristics**  
  (a.k.a., Quality Factors, the ‘ilities’)  
  (e.g., availability, extensibility, interoperability, maintainability, performance, portability, reliability, safety, security, and usability)

- **Quality Attributes**  
  (a.k.a., Quality Subfactors)  
  (e.g., the quality attributes of performance are jitter, latency, response time, schedulability, throughput)

- **Quality Measurement Scales**  
  (e.g., milliseconds, transactions per second)
Quality Model

Architectural Components

System

Quality Model

defines the meaning of the quality of a

defines the meaning of a specific type of quality of a

Quality Characteristics

Quality Attributes

Quality Measurement Scales

Quality Measurement Method

Internal Quality Characteristics

External Quality Characteristics

are measured along

measures quality along

are measured using
Quality Model – Performance Quality Attributes

Performance Attributes:
- Jitter
- Latency
- Response Time
- Schedulability
- Throughput

Quality Characteristic:
- Mandated Threshold
  - Failure Detection
  - Failure Reaction
  - Failure Adaptation

Performance Problem Type

Performance Solution Type

Performance Attribute

Quality Attribute

is measured along a Quality Measurement Scale

Quality Model
Quality Case - Definition

Quality Case

a Cohesive Collection of *Claims, Arguments, and Evidence* that Makes the Developers’ Case that their Work Product(s) have *Sufficient Quality*

Foundational Concept underlying QUASAR

A Generalization and Specialization of Safety Cases from the Safety Community:

More) Can Address any Quality Characteristic and/or Quality Attribute

Less) May be Restricted to only Requirements or Architecture

Useful for:

• Assessing Quality

• System Certification and Accreditation (e.g., safety and security)
A Quality Case consists of the following types of Components:

1. **Claims**
   Developers’ Claims that their Work Products have *Sufficient* Quality, whereby quality is defined in terms of the qualify characteristics and quality attributes defined in the official project quality model.

2. **Arguments**
   Clear, Compelling, and Relevant Developer Arguments Justifying the Assessors’ Belief in the Developers’ Claims (e.g., decisions, inventions, trade-offs, analysis and simulation results, assumptions, and associated rationales).

3. **Evidence**
   Adequate Credible Evidence Supporting the Developers’ Arguments (e.g., official project diagrams, models, requirements specifications and architecture documents; requirements repositories; analysis and simulation reports; test results; and demonstrations witnessed by the assessors).
Quality Cases – Components

Work Product

make developers’ case for adequate quality of the

Quality Case

Claims

Arguments

Evidence

supports

is developed for

Quality Characteristic

Quality Attribute

justify belief in
Specialized QUASAR Quality Cases

QUASAR utilizes the following specialized types of Quality Cases:

1. Requirements Quality Cases
2. Architectural Quality Cases

QUASAR Version 1 only had Architectural Quality Cases.

QUASAR Versions 2 and 3 have Both Types of Quality Cases.
QUASAR Quality Case Responsibilities

Requirements Engineers and Architects’ Responsibilities:

• Prepare Quality Cases
• Provide Preparation Materials (including Presentation Materials and Quality Cases) to Assessors Prior to Assessment Meetings
• Present Quality Cases (Make their Case to the Assessors)
• Answer Assessors’ Questions

Assessor Responsibilities:

• Prepare for Assessments
• Actively Probe Quality Cases
• Develop Consensus regarding Assessment Results
• Determine and Report Assessment Results:
  — Present Outbriefs
  — Publish Reports
Quality Case Diagram Notation

Quality Factor A Supported
<<claim>>

Quality Subfactor A₁ Supported
<<claim>>
justifies belief in
Decision 1
<<argument>>
supports
Diagram 1
<<evidence>>

Quality Subfactor A₂ Supported
<<claim>>
Trade-Off 1
<<argument>>
Model 1
<<evidence>>
Diagram N
<<evidence>>

Quality Subfactor Aₙ Supported
<<claim>>
Assumption 1
<<argument>>
Document 1
<<evidence>>

…

Decision N
<<argument>>
Trade-Off N
<<argument>>
Model N
<<evidence>>
Assumption N
<<argument>>
Diagram N
<<evidence>>
Architectural Interoperability Case Diagram

Claim: Architecture Supports Interoperability Goals

Meets Requirements

Claim: Physical Interoperability
Claim: Energy Interoperability
Claim: Protocol Interoperability
Claim: Syntax Interoperability
Claim: Semantics Interoperability

Arguments (Architectural Decisions)

One-Way Connections
Layered Architecture
Open Interface Standards
Service Oriented Architecture (SOA)
Fly-By-Wire
Modular Architecture
Proxies and Wrappers

Evidence

Wiring Diagram
Context Diagram
Allocation Diagram
Layer Diagram
Interoperability Whitepaper

Supports

Hardware Schematics
Configuration Diagram
Network Diagrams
Activity or Collaboration Diagrams
Vendor-Supplied Technical Documentation

Meets Requirements

justifies belief in

Arguments (Architectural Decisions)

supports

Evidence

Hardware Schematics
Configuration Diagram
Network Diagrams
Activity or Collaboration Diagrams
Vendor-Supplied Technical Documentation
Example QUASAR Scope – Four Assessments

Tier 1
- System of Systems

Tier 2
- System 1
- System 2
- System 3
- ... System N

Tier 3
- Subsystem 1
- Subsystem 2
- Subsystem 3
- ... Subsystem N

Tier 4
- Segment 1
- Segment 2
- Segment 3
- ... Segment N

Tier 5
- Subsegment 1
- Subsegment 2
- Subsegment 3
- ... Subsegment N

Tier 6
- Assembly 1
- Assembly 2
- Assembly 3
- ... Assembly N

Tier 7
- Subassembly 1
- Subassembly 2
- Subassembly 3
- ... Subassembly N

Tier 8
- HW CI 1
- ... HW CI N
- SW CSCI 1
- ... SW CSCI N
- Data CI 1
- ... Data CI N
- Facilities
- Manual Procedures
- Roles

Tier 9
- HW C 1
- ... HW C N
- SW C 1
- ... SW C N

Tier 10
- Part 1
- ... Part N
- SW Unit 1
- ... SW Unit N
What is a System Architecture?

System Architecture

the Most Important, Pervasive, Top-Level, Strategic Decisions, Inventions, Engineering Trade-Offs, Assumptions, and associated Rationales about How a System’s Architectural Elements will collaborate to meet the System’s Derived and Allocated Requirements
What is a System Architecture?

System Architecture Includes:

- **The System’s Numerous Static and Dynamic, Logical and Physical Structures**
  (i.e., Essential Architectural Elements, their Relationships, their Associated Blackbox Characteristics and Behavior, and how they Collaborate to Support the System’s Mission and Requirements)

- **Architectural Decisions, Inventions, and Tradeoffs**
  (e.g., Styles, Patterns, and Mechanisms used to ensure that the System Achieves its Architecturally-Significant Product and Process Requirements (esp. Quality Requirements or ‘ilities’))

- **Strategic and Pervasive Design-Level Decisions**
  (e.g., using a Design Paradigm such as Object-Oriented Orientation or Mandated Widespread use of common Design Patterns)

- **Strategic and Pervasive Implementation-Level Decisions**
  (e.g., using a Safe Subset of C++)
Some Example Views of Models of Structures

Multifaceted architecture having multiple structures requiring multiple models providing multiple views

Architects must ensure view and model consistency

Software Engineering Institute

View Carnegie Mellon
Architecture vs. Design

<table>
<thead>
<tr>
<th>Architecture</th>
<th>Design</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Pervasive</strong> (Multiple Components)</td>
<td><strong>Local</strong> (Single Components)</td>
</tr>
<tr>
<td><strong>Strategic</strong> Decisions and Inventions</td>
<td><strong>Tactical</strong> Decisions and Inventions</td>
</tr>
<tr>
<td><strong>Higher-Levels</strong> of System</td>
<td><strong>Lower-Levels</strong> of System</td>
</tr>
<tr>
<td><strong>Huge Impact</strong> on Quality, Cost, &amp; Schedule</td>
<td><strong>Small Impact</strong> on Quality, Cost, &amp; Schedule</td>
</tr>
<tr>
<td><strong>Drives</strong> Design and Integration Testing</td>
<td><strong>Drives</strong> Implementation and Unit Testing</td>
</tr>
<tr>
<td><strong>Driven by</strong> Requirements and Higher-Level Architecture</td>
<td><strong>Driven by</strong> Requirements, Architecture, and Higher-Level Design</td>
</tr>
<tr>
<td><strong>Mirrors</strong> Top-Level Development Team Organization (Conway’s Law)</td>
<td><strong>No Impact</strong> on Top-Level Development Team Organization</td>
</tr>
</tbody>
</table>
Architectural Documentation Current-State

System Architecture Documents:

• Mostly natural language Text with Visio-like Diagrams (Cartoons)
• Logical (functional) and Physical Architecture

DOD Architecture Framework (DODAF):

• All-Views, Operational Views, Systems Views, and Technical Standards Views for allocating Responsibilities to Systems and Supporting System Interoperability

Models (both static and dynamic; logical and physical):

• Tailored UML becoming *de facto* Industry Standard
• SysML starting to become Popular

Visio Diagrams as Wall Posters

Whitepapers, Reports, and other Specialty-Engineering Documents:

• Performance, Fault Tolerance, Reliability, Safety, Security
Quality Requirements – Components

Quality Management includes:

- **Quality Goal**: specifies the minimum acceptable level of quality.
- **Quality Requirement**: quantifies the quality of a system.
- **Subsystem**: defines stakeholders and their minimum acceptable level of quality.
- **System**: defines the meaning of the quality of a system.
- **Condition**: is applicable during a specific period.
- **Quality Criterion**: shall exceed a specified threshold.
- **Quality Threshold**: is measured along a specific dimension.
- **Quality Metric**: is measured using a specific method.
- **Quality Attribute**: states the importance of achieving a quality goal.
- **Quality Characteristic**: determines the existence of a quality goal.
- **Quality Model**: defines the meaning of the quality of a system.

The Quality Model

- **Quality Characteristic**
- **Quality Attribute**
- **Quality Measure**
- **Quality Metric**

This diagram illustrates the components of a quality management system, showing the relationships between different quality elements and how they contribute to the overall quality of a system.
Topics

Requirements and Architecture Challenges

Underlying Concepts

QUASAR Method

Reasons to use QUASAR
### Definition

**QUality Assessment of System Architectures and their Requirements**

a Well-Documented and Proven Method based on the use of *Quality Cases* for *Independently* Assessing the *Quality* of:

- Software-intensive *System / Subsystem Architectures* and the
- *Architecturally Significant Requirements* that Drive Them
Informal *Peer Reviews* are Inadequate:

- Too Informal
- Lack of *Independent* Expert Input
- Requirements and Architecture are too Important

Quality Requirements:

- Most important Architecturally-Significant Requirements
- Largely Drive the System Architecture
- Criteria against which the System Architecture is Assessed
**QUASAR Philosophy**

**Requirements Engineers (REs) should Make Case to Assessors:**

- REs *should* know Stakeholder Needs and Goals
- REs *should* know What they Did and Why (Architecturally-Significant Requirements, Rationales, & Assumptions)
- REs *should* Know Where they Documented their Requirements Work Products

**Architects** should *Make Case* to Assessors:

- Architects *should* know Architecturally-Significant Requirements
- Architects *should* know What they Did and Why (Decisions, Inventions, Trade-Offs, Assumptions, and Rationales)
- Architects *should* know Where they Documented their Architectural Work Products
Assessors should Actively Probe Quality Cases:

- **Claims Correct and Complete?**
  Do the Claims include all relevant Quality Characteristics, Quality Attributes, Quality Goals, and Quality Requirements?

- **Arguments Correct, Complete, Clear, and Compelling?**
  Do the Arguments include all relevant Quality Characteristics, Quality Attributes, Quality Goals, Quality Requirements, Decisions, Inventions, Trade-offs, Assumptions, and Rationales?

- **Arguments Sufficient?**
  Are the Arguments Sufficient to Justify the Claims?

- **Evidence Sufficient?**
  Is the Evidence Sufficient to Support the Arguments?

- **Current Point in the Schedule?**
  Are the Claims, Arguments, and Evidence appropriate for the Current Point in the Schedule?
QUASAR Method – Three Phases

1. Quality Assessment Initiation (QAI)
2. Requirements Quality Assessment (RQA)
3. Architecture Quality Assessment (AQA)

repeat for system and each subsystem being assessed
QUASAR Phases and Tasks

Time (not to scale)

System Assessments

Phase 2) Requirements Quality Assessment (RQA)

Prep. | RQA Meeting | Follow-Through

Phase 3) Architecture Quality Assessment (AQA)

Prep. | AQA Meeting | Follow-Through

Subsystem 1 Assessments

Phase 2) Requirements Quality Assessment (RQA)

Prep. | RQA Meeting | Follow-Through

Phase 3) Architecture Quality Assessment (AQA)

Prep. | AQA Meeting | Follow-Through

Subsystem N Assessments

Phase 2) Requirements Quality Assessment (RQA)

Prep. | RQA Meeting | Follow-Through

Phase 3) Architecture Quality Assessment (AQA)

Prep. | AQA Meeting | Follow-Through
Quasar Teams and their Work Products

System Requirements Team

engineer the

leads the

Subsystem Requirements Team(s)

make their

engineer the

requirements

Quality Cases

architecturally-significant

requirements

Architecture

quality cases

Subsystem Architecture

Assessment Team(s)

assess the

requirements teams’

Subsystems

Subsystems

architecturally-significant

requirements

System-Level

Architecturally-Significant

Requirements

drive the

architecture

Subsystem

Systems

architecture

architectural

requirements

drive the

architectural

requirements

drive the

architecture

architectural

quality cases

make their

leads the

Top-Level Architecture Team

engineer the

makes its

make their

assess the

architecture teams’
Quality Assessment Initiation (QAI)

repeat for system and each subsystem being assessed

Requirements Quality Assessment → Architecture Quality Assessment
Phase 1) QAI – Objectives

Prepare Teams for Requirements and Architecture Assessments

Develop Consensus:

- Scope of Assessments
- Schedule Assessments
- Tailor the Assessment Method and associated Training Materials

Produce and Publish Meeting Outbrief and Minutes

Manage Action Items

Capture Lessons Learned

Tailor/Update QUASAR Method and Training Materials
Phase 1) QAI – Preparation Task

1. Management Team staffs Assessment Team

2. Process and Training Teams train Assessment Team

3. Assessment Team identifies:
   - System Requirements Team
   - System Architecture Team

4. Process and Training Teams train System Requirements and Architecture Teams

5. Assessment, Requirements, and Architecture Teams collaborate to Organize QAI Meeting (i.e., Attendees, Time, Location, Agenda)
Phase 1) QAI – Meeting Task

1. Assessment, System Requirements, and System Architecture
   Teams Collaborate to determine Assessment Scope:
   - Subsystems/Architectural Elements/Focus Areas to Assess (Number and Identity)
   - Quality Characteristics and Quality Attributes underlying Assessment
   - Assessment Resources (e.g., Staffing, Schedule, and Budget)

2. Teams Collaborate to develop Initial Assessment Schedule with regard to System schedule, Subsystem schedule, and associated milestones

3. Teams Collaborate to tailor QUASAR Method

4. Assessment Team captures Action Items
Phase 1) QAI – Follow-Through Task

1. Assessment Team develops and presents Meeting Outbrief
2. Assessment Team develops, reviews, and distributes Meeting Minutes
3. Assessment/Process/Training Teams tailor, internally review, and distribute:
   • QUASAR Procedure, Standards, and Templates
   • QUASAR Training Materials
4. Teams distribute Assessment Schedule
5. Teams obtain Needed Resources
6. Assessment Team Manages Action Items
7. Assessment Team captures Lessons Learned
Phase 1) QAI – Work Product Flow

1. QUASAR Training Materials
2. QUASAR Stds & Procedures
3. Questions/Answers
4. Recommendations
5. SAI Outbrief
6. SAI Minutes
7. Action Item List

System Requirements Team

System Architecture Team

QUASAR Training Materials

QUASAR Stds & Procedures

Questions/Answers

Recommendations

SAI Outbrief

SAI Minutes

Action Item List

Training Team

Process Team

System Assessment Team

Lessons Learned

**Notes:**
- * indicates a critical path
- 1, 2, 3, 4, 5, 6 indicate sequence of tasks
Phase 1) QAI – Work Products

Legend
- developer work product
- assessor work product
- influences
- aggregation

Meeting Notes
Meeting Outbrief
Meeting Minutes
Assessment Scope
Assessment Schedule
Method Tailoring
Lessons Learned
Preparatory Materials
QUASAR Training Materials
QUASAR Standards & Procedures
Phase 1) QAI – Lessons Learned

Ensure Appropriate Team Memberhips (e.g., Authority)

Ensure Adequate Resources (e.g., Staffing, Budget, and Schedule)

Obtain Consensus on:

- Assessment Objectives and Scope
- Definitions (e.g., of Quality Characteristics, Attributes, and Cases)

Provide Early Training:

- Method Training (QUASAR, Requirements Engineering, and Architecting)
- System/Subsystem Training (Requirements and Architecture)
QUASAR assessments should be Organized according to a Quality Model that defines Quality Characteristics (a.k.a., factors, “ilities”) and their Quality Attributes such as:

- Availability
- Interoperability
- Performance
  - Jitter, Response Time, Schedulability, and Throughput
- Portability
- Reliability
- Safety
- Security
- Usability
Requirements Quality Assessment (RQA)

- Quality Assessment Initiation

  repeat for system and each subsystem being assessed

  - Requirements Quality Assessment
  - Architecture Quality Assessment
Phase 2) ARA – Objectives

Use Requirements Quality Cases to:

• Independently assess Quality and Maturity of the Architecturally Significant Requirements:
  — Drive the Architecture
  — Form Foundation for Architecture Quality Assessment

• Help Requirements Engineers identify Requirements Defects and Weaknesses so that:
  — Defects and Weaknesses can be Corrected
  — The Architecture (and System) can be Improved
Phase 2) RQA – Objectives

Use Requirements Quality Cases to:

- Identify Requirements Risks so that they can be Managed
- Provide Visibility into the Status and Maturity of the Requirements
- Increase the Probability of Project Success

Ensure Architecture Team will be Prepared to Support the coming Architecture Quality Assessment.

Capture Lessons Learned.

Update QUASAR Method and associated Training Materials.
Phase 2) RQA – Preparation Task

Process/Training Team trains the Requirements and Architecture Teams significantly prior to the RQA Meeting.

Requirements and Architecture Teams provide Preparatory Materials to the Quality Assessment Team significantly prior to the RQA Meeting:

- Summary Presentation Materials
- Requirements Quality Cases (including electronic access to evidentiary materials)
- Example of Planned Architectural Quality Case

Quality Assessment Team:

- Reads Preparatory Materials
- Generates RFIs and RFAs
Phase 2) RQA – Meeting Task

1. Requirements Team presents:
   • System Overview
   • Requirements Overview
   • Requirements Quality Cases

2. Quality Assessment Team assesses Quality and Maturity of Requirements:
   • Completeness of Quality Cases
   • Quality of Quality Cases

3. Architecture Team presents Example Architectural Quality Case

4. Quality Assessment Team recommends Improvements

5. Quality Assessment Team manages Action Items
Phase 2) RQA – Follow-Through Task

Quality Assessment Team:

1. Develops Consensus Regarding Requirements Quality
2. Produces, Reviews, and Presents Meeting Outbrief
3. Produces, Reviews, and Publishes RQA Report
4. Updates and publishes the System Quality Assessment Summary Matrix
5. Captures Lessons Learned
6. Manages Action Items

Requirements Team:

Addresses Risks Raised in RQA Report

Process Team:

Updates Assessment Method (e.g., Standards and Procedures)

Training Team:

Updates Training Materials (if appropriate)
Phase 2) RQA – Work Product Workflow

1. QUASAR Training Materials
2. QUASAR Stds & Procedures
3. RQA Checklist and Report Template
4. RQA Preparatory Materials
5. RQA Presentation Materials
6. Requirements Quality Cases and Draft Representative Architectural Quality Case
7. Questions/Answers
8. Recommendations
9. RQA Outbrief
10. RQA Report
11. Action Item List

Teams:
- Requirements Team
- Architecture Team
- Assessment Team
- Training Team
- Process Team

Materials:
- QUASAR Training Materials
- QUASAR Stds & Procedures
- RQA Checklist and Report Template
- RQA Preparatory Materials
- RQA Presentation Materials
- Requirements Quality Cases and Draft Representative Architectural Quality Case
- Questions/Answers
- Recommendations
- RQA Outbrief
- RQA Report
- Action Item List

Outcomes:
- Lessons Learned
- System Quality Assessment Summary Matrix
Phase 2) RQA – Work Products

- Meeting Invitation and Agenda
- RQA Preparatory Materials
- RQA Presentation Materials
- RQA Assessment Report
- System Quality Assessment Summary Matrix
- RQA Assessment Outbrief
- Action Item List
- Assessor Notes
- Lessons Learned
- Requirements Support Matrix
- RQA Assessment Report Template
- QUASAR Standards and Procedures
- Requirements Assessment Checklist
- QUASAR Training Materials

Legend:
- requirement work product → influences
- architecture work product → aggregation
- assessor work product → specialization

Text, Diagrams, Models, Documents

System / Component Overview
- Rqmts Quality Case Diags
- Draft Architecture Quality Case

Requirements Overview
- Requirements Quality Cases

Claims
- Supporting Evidence

Justifying Arguments
- Supporting Assumptions

QUASAR Conventions

Software Engineering Institute | Carnegie Mellon

QUASAR Version 3.1 Overview
Donald Firesmith, 29 October 2009
© 2009 Carnegie Mellon University
# System Quality Assessment

## Summary Matrix

<table>
<thead>
<tr>
<th>QF 1</th>
<th>QF 2</th>
<th>QF 3</th>
<th>QF 4</th>
<th>QF 5</th>
<th>QF 6</th>
<th>QF 7</th>
<th>QF 8</th>
<th>QF n</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
</tr>
<tr>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
</tr>
<tr>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
</tr>
<tr>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
</tr>
<tr>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
</tr>
</tbody>
</table>

The matrix above represents the System Quality Assessment Summary Matrix. Each column indicates a different aspect (SYS, AC 1, AC 2, AC 3, AC 4, AC 5, AC n) and the rows correspond to different quality factors (QF 1, QF 2, QF 3, QF 4, QF 5, QF 6, QF 7, QF 8, QF n). The entries are marked with R (representing requirement) or A (representing action) depending on whether a specific aspect requirement or action is present for a given quality factor.
Architecture Quality Assessment (AQA)

Quality Assessment Initiation

repeat for system and each subsystem being assessed

Requirements Quality Assessment

Architecture Quality Assessment
Phase 3) AQA – Objectives

Use Architectural Quality Cases to:

- Independently assess Architecture Quality in terms of its Support for its Derived and Allocated Architecturally Significant Requirements
- Help Architects identify Architectural Defects and Weaknesses so that:
  - Defects and Weaknesses can be Corrected
  - The Architecture (and System) can be Improved
- Identify Architectural Risks so that they can be Managed
- Provide Visibility into the Status and Maturity of the Architecture
- Increase the Probability of Project Success
Phase 3) AQA – Preparation Task

Architecture and Quality Assessment Teams organize the AQA Assessment Meeting.

Training Team provides (at appropriate time):
- QUASAR Training (if not provided prior to RQA assessment)
- AQA Assessment Checklist and Report Template

Architecture Team makes available (min. 2 weeks before meeting):
- Any Updated Quality Requirements
- Architecture Overview
- Quality Case Diagrams
- Architecture Quality Cases (Claims, Arguments, and Evidence)

Quality Assessment Team:
- Reads Preparatory Materials
- Generates RFIs and RFAs
Phase 3) AQA – Meeting Task

Architecture Team:

1. Introduces the Architecture (e.g., Context and Major Functions)
2. Briefly reviews the Architecturally Significant Requirements
3. Briefly summarizes the Architecture (e.g., Most Important Architectural Components, Relationships, Decisions, Inventions, Trade-Offs, Assumptions, and Rationales)
4. Individually Presents Architectural Quality Cases (Quality Case Diagram, Claims, Arguments, and Evidence)

Quality Assessment Team:

1. Probes Architecture (Architectural Quality Case by Quality Case)
2. Manages Action Items
Phase 3) AQA – Follow-Through Task

Quality Assessment Team:
1. Develops Consensus regarding Architecture Quality
2. Produces, reviews, and presents Meeting Outbrief
3. Produces, reviews, and publishes AQA Report
4. Updates and republishes System Quality Assessment Summary Matrix
5. Captures Lessons Learned
6. Manages Action Items

Architecture Team:
Addresses Architectural Defects, Weaknesses, and Risks Raised in AQA Report

Process Team:
Updates Assessment Method (if appropriate)

Training Team:
Updates Training Materials (if appropriate)
Phase 3) AQA – Work Product Workflow

- **QUASAR Training Materials**
- **QUASAR Stds & Procedures**
- **AQA Checklist and Report Template**
- **AQA Preparatory Materials**
  - **Introductory Material and Architectural Quality Cases**
  - **Questions/Answers**
  - **Recommendations**
  - **AQA Outbrief**
  - **AQA Report**
  - **Action Item List**

**Architecture Team**

**Process Team**

**Training Team**

**Assessment Team**

**Lessons Learned**

**System Quality Assessment Summary Matrix**
Phase 3) AQA – Primary Work Products

Legend
- architect work product
- assessor work product
- influences
- aggregation
- specialization
Topics

Requirements and Architecture Challenges

Underlying Concepts

QUASAR Method

Reasons to use QUASAR
QUASAR Benefits

QUASAR ensures Specification of *Architecturally-Significant* Requirements.

QUASAR provides Acquirer Visibility into (and supports oversight of) the Quality of the Requirements and Architecture

QUASAR supports Certification and Accreditation

QUASAR emphasizes using a common project-specific Quality Model:

- Which drives the Quality Requirements
- Which drives the Quality of the System Architecture
- Which drives the Quality of the System
QUASAR Benefits

QUASAR Supports Process Improvement:
- Solves Major Requirements and Architecture Problems

QUASAR Provides needed Flexibility:
- Any Effective Requirements Engineering and Architecting Methods
- Uses Existing Requirements and Architecture Work Products (i.e., almost no new work products required)
- Any Subsystems based in Need and Risk (i.e., fits any system size, budget, schedule, and tier)
- Any Quality Characteristics and Quality Attributes

QUASAR Helps:
- Requirements Engineers Succeed
- Architects Succeed
- Program Succeed
How the SEI Can Help You

QUASAR is Ready for Use Now.

QUASAR Handbook and Training Materials can be downloaded from SEI Website.

The SEI Acquisition Support Program (ASP) offers QUASAR as a Service:

- Consulting and Training
- Facilitation of QUASAR Assessments
- Recommended RFP and Contract Language
Questions?

For more information:

Donald Firesmith
Acquisition Support Program
Software Engineering Institute
dgf@sei.cmu.edu


Quasar Tutorial (1 day) :