*QU*ality *A*ssessment of *S*ystem *A*rchitectures and their *R*equirements (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$_1$

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$_2$

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$_3$

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:

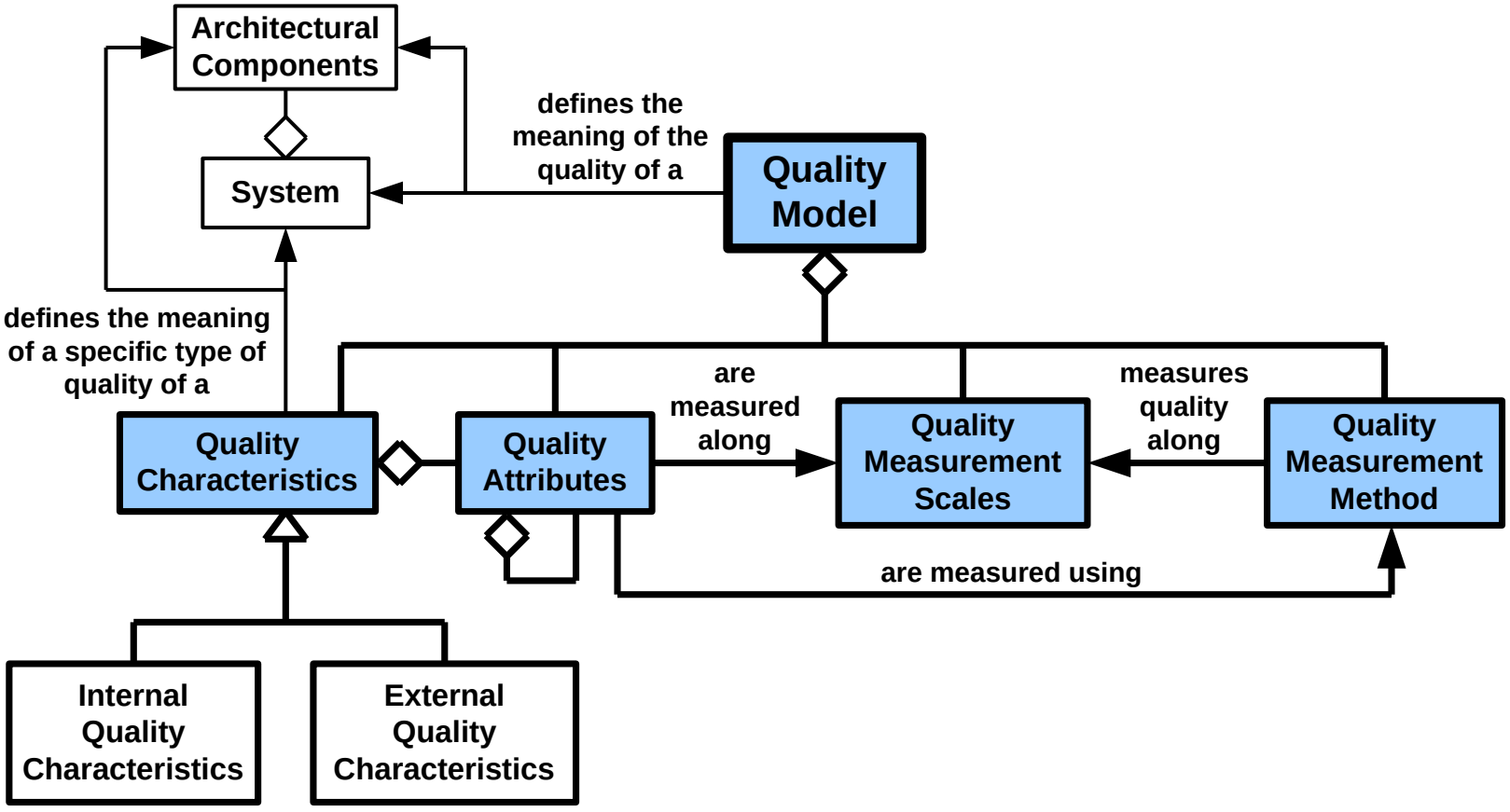
> They are the Characteristics defined by the Project Quality Model.

Quality Model$_1$

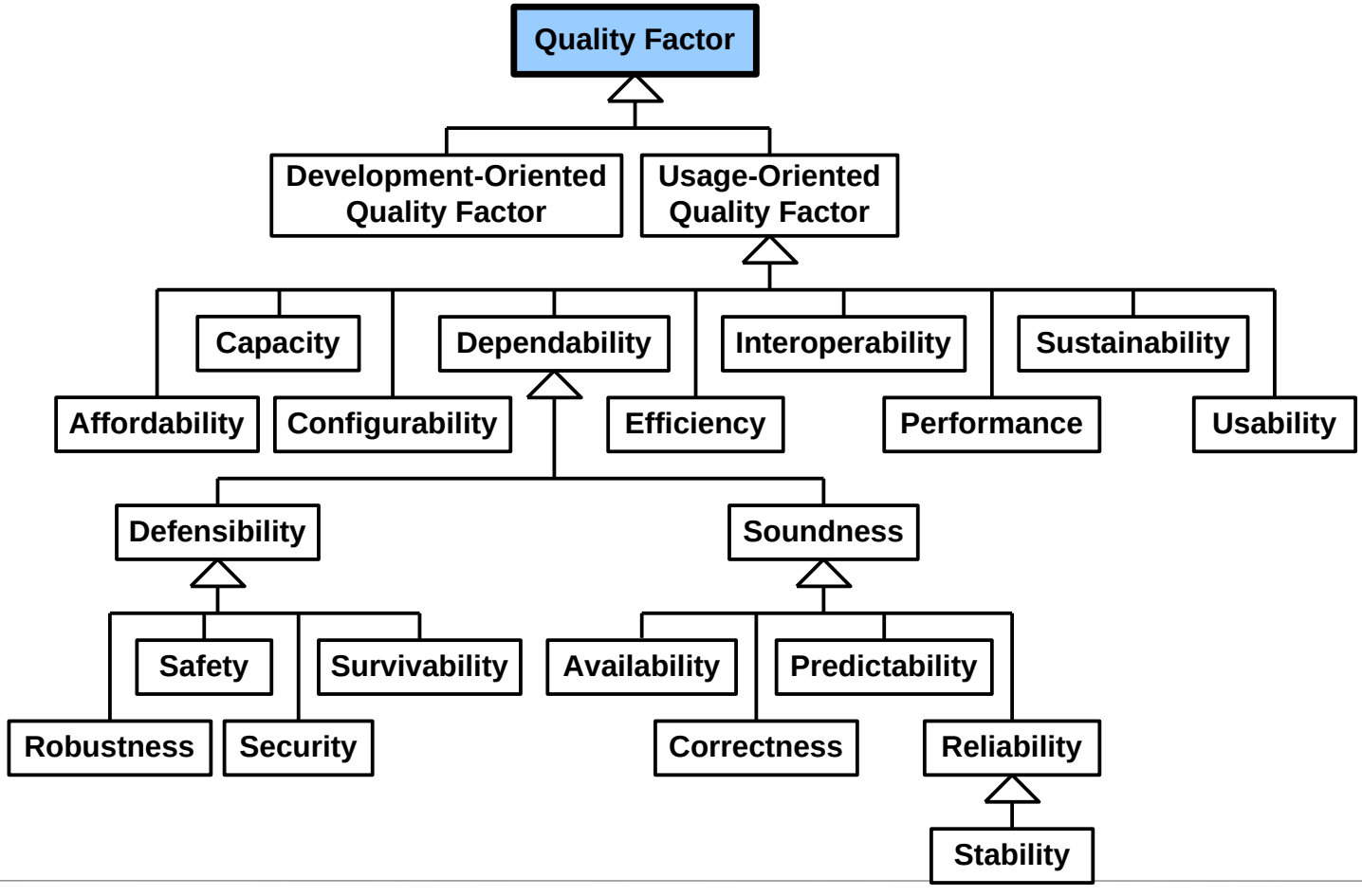
Quality of a Work Product is defined in terms of a **Quality Model**:

- **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)

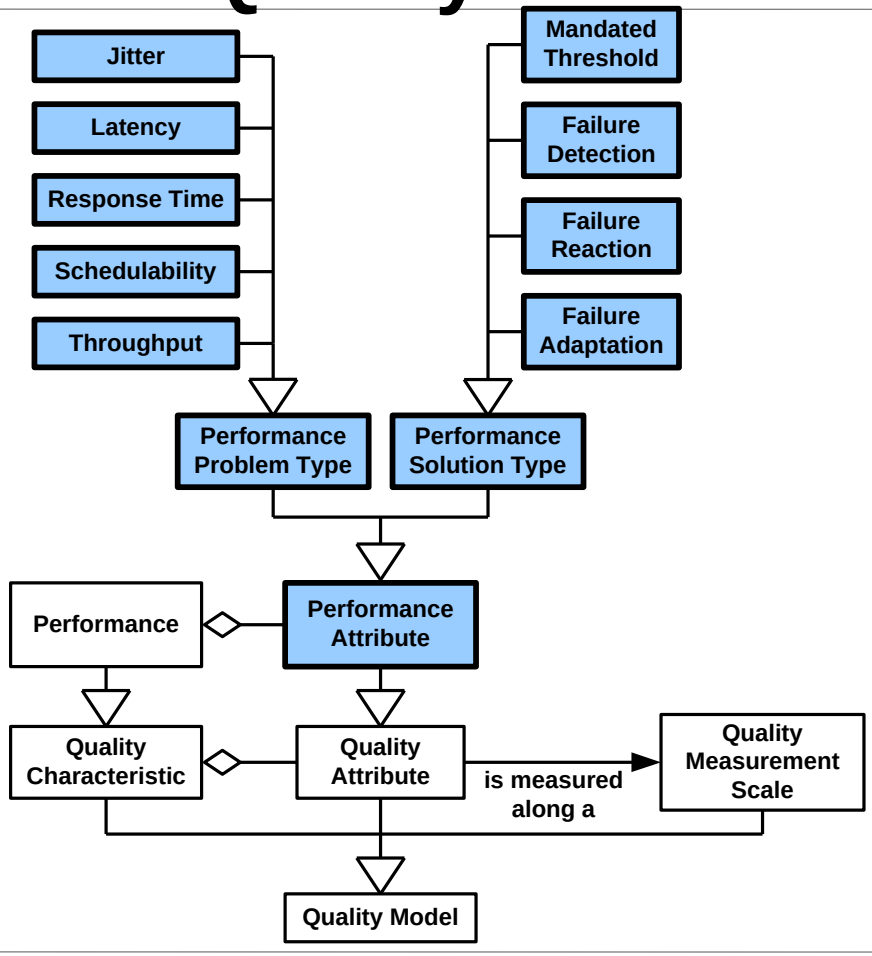
Quality Model$_2$



Quality Model – Quality Characteristics



Quality Model – Performance Quality Attributes



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:

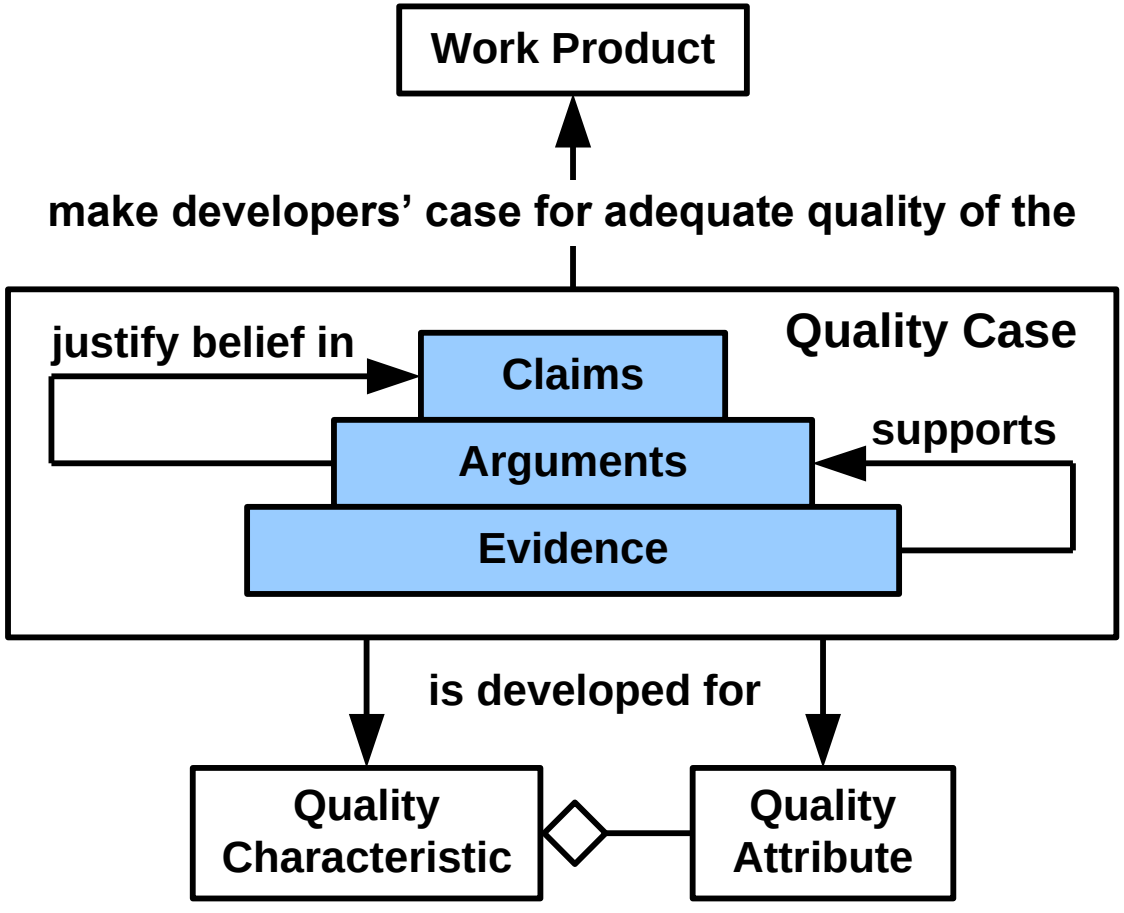
- Assessing Quality
- System Certification and Accreditation (e.g., safety and security)

Quality Cases – Components$_1$

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$_2$



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

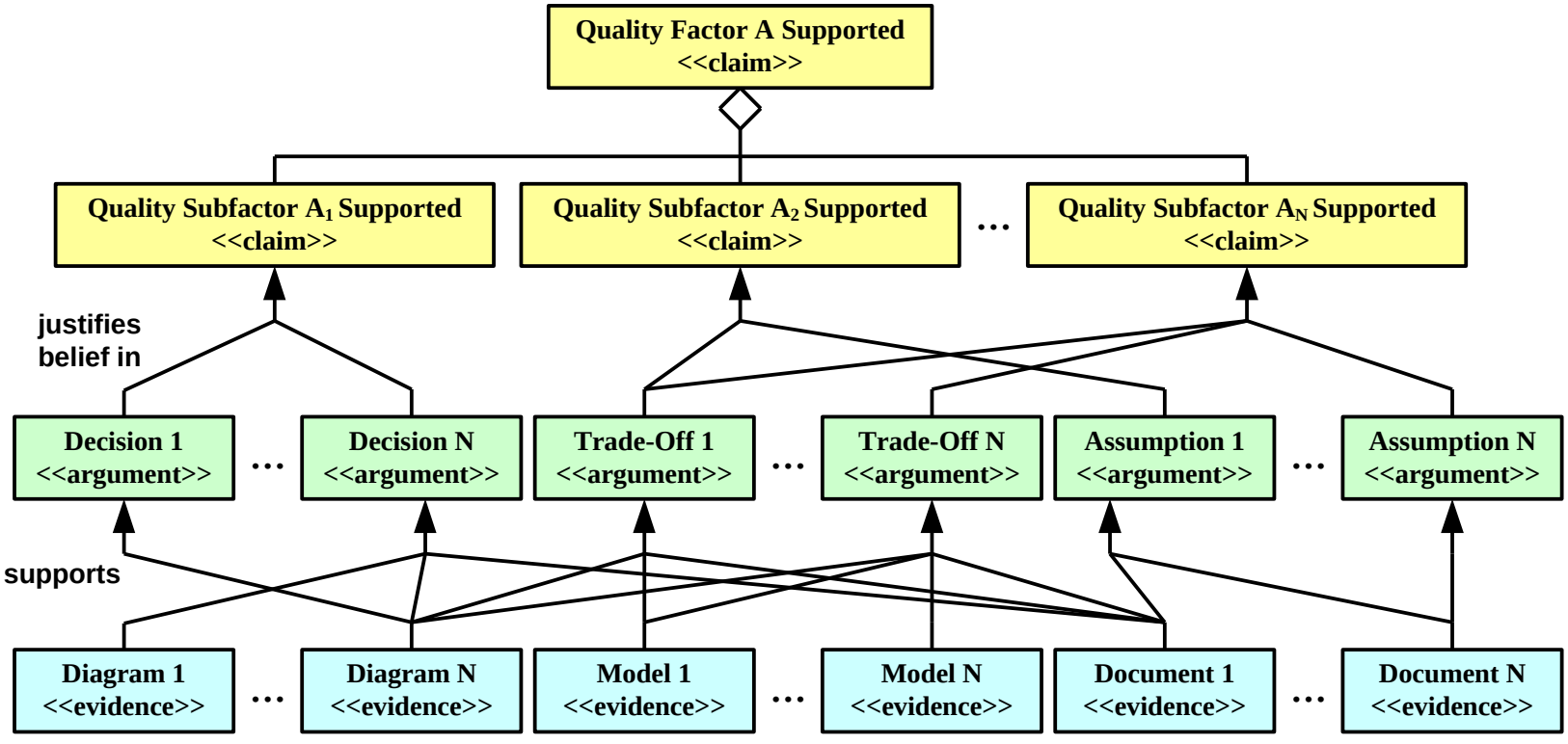
Requirements Engineers and Architects' Responsibilities:

- 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

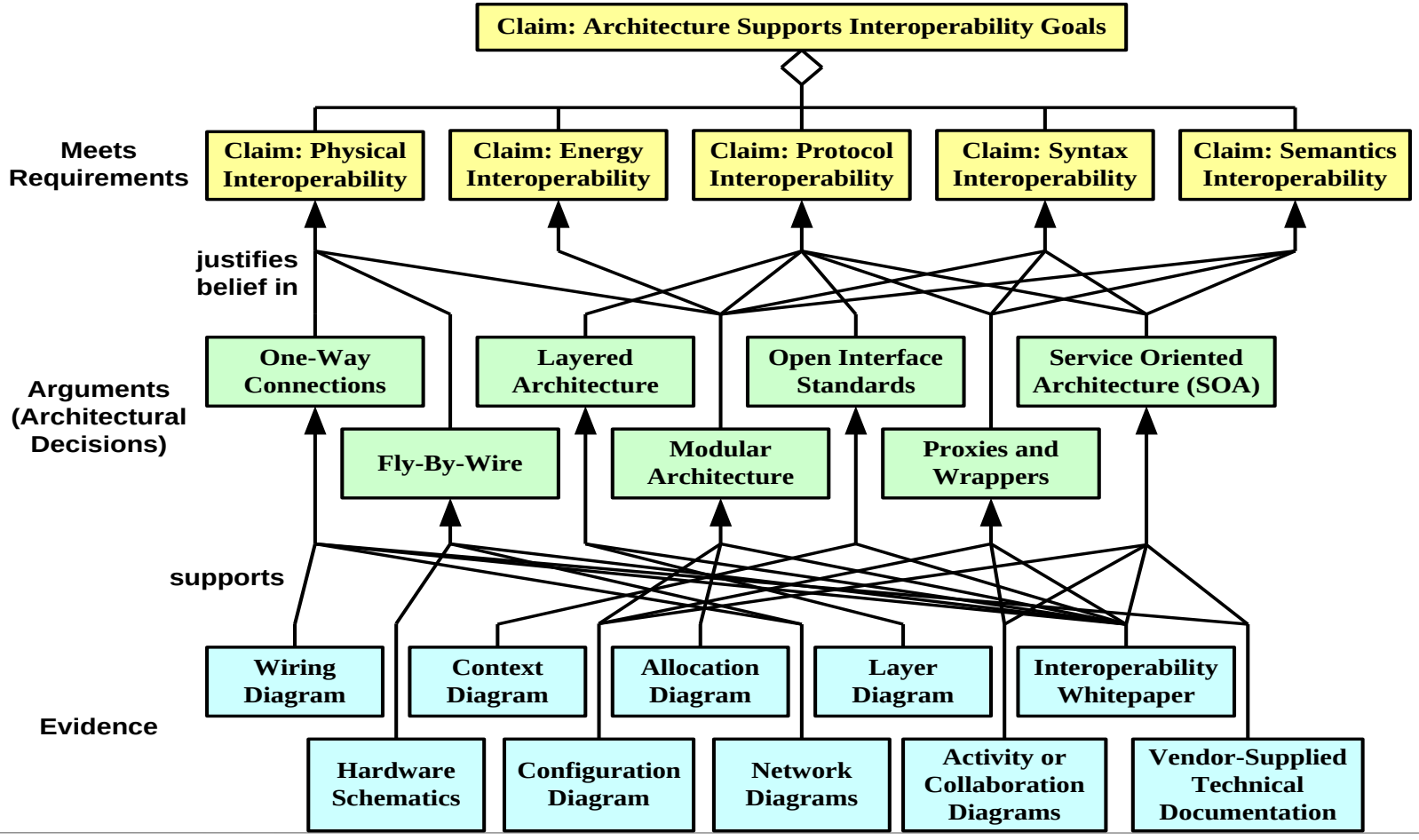
Assessor Responsibilities:

- Prepare for Assessments
- *Actively* Probe Quality Cases
- Develop Consensus regarding Assessment Results
- Determine and Report Assessment Results:
  - Present Outbriefs
  - Publish Reports

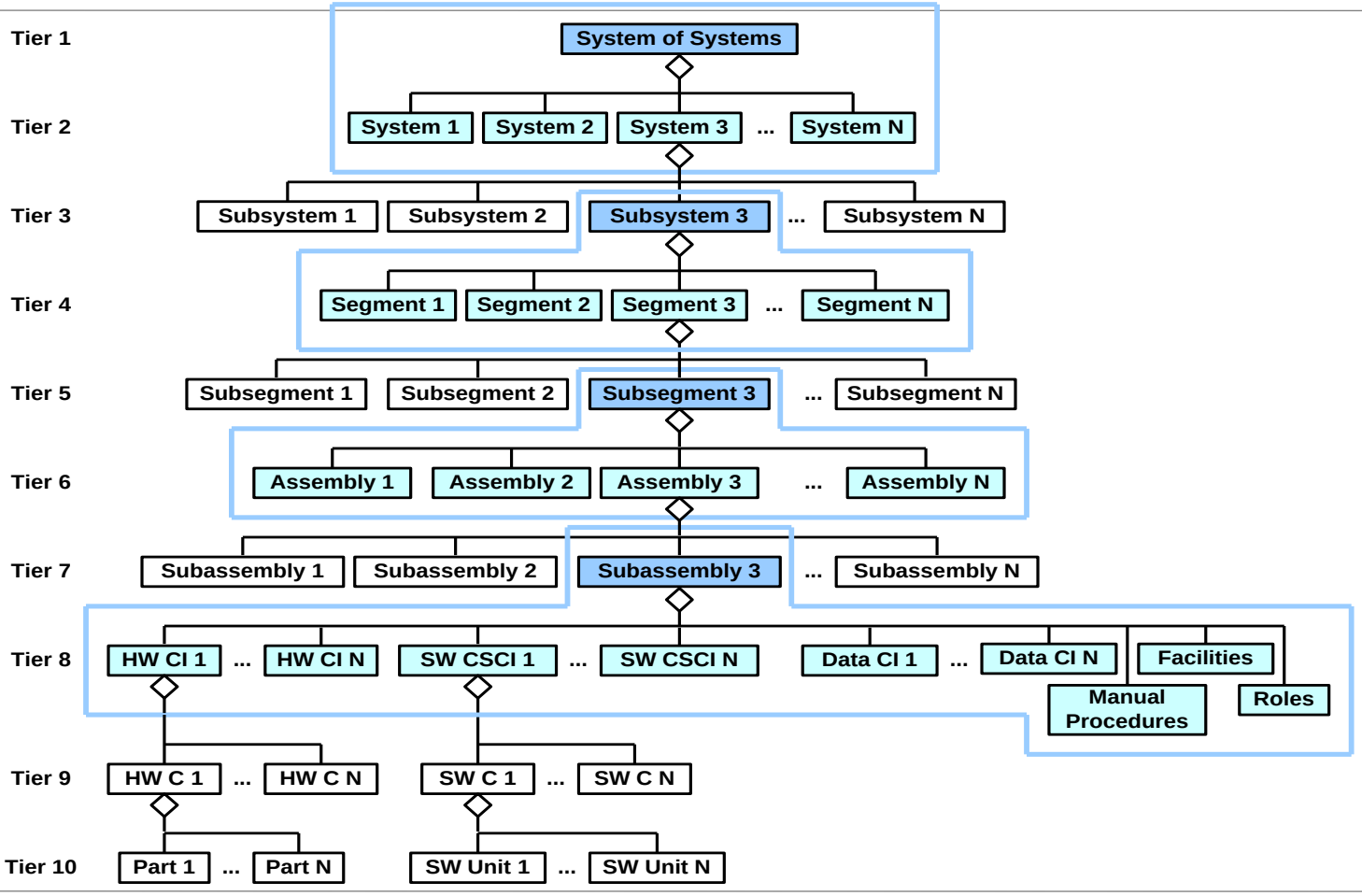
Quality Case Diagram Notation



Architectural Interoperability Case Diagram



Example QUASAR Scope – Four Assessments



What is a System Architecture?[1]

System Architecture

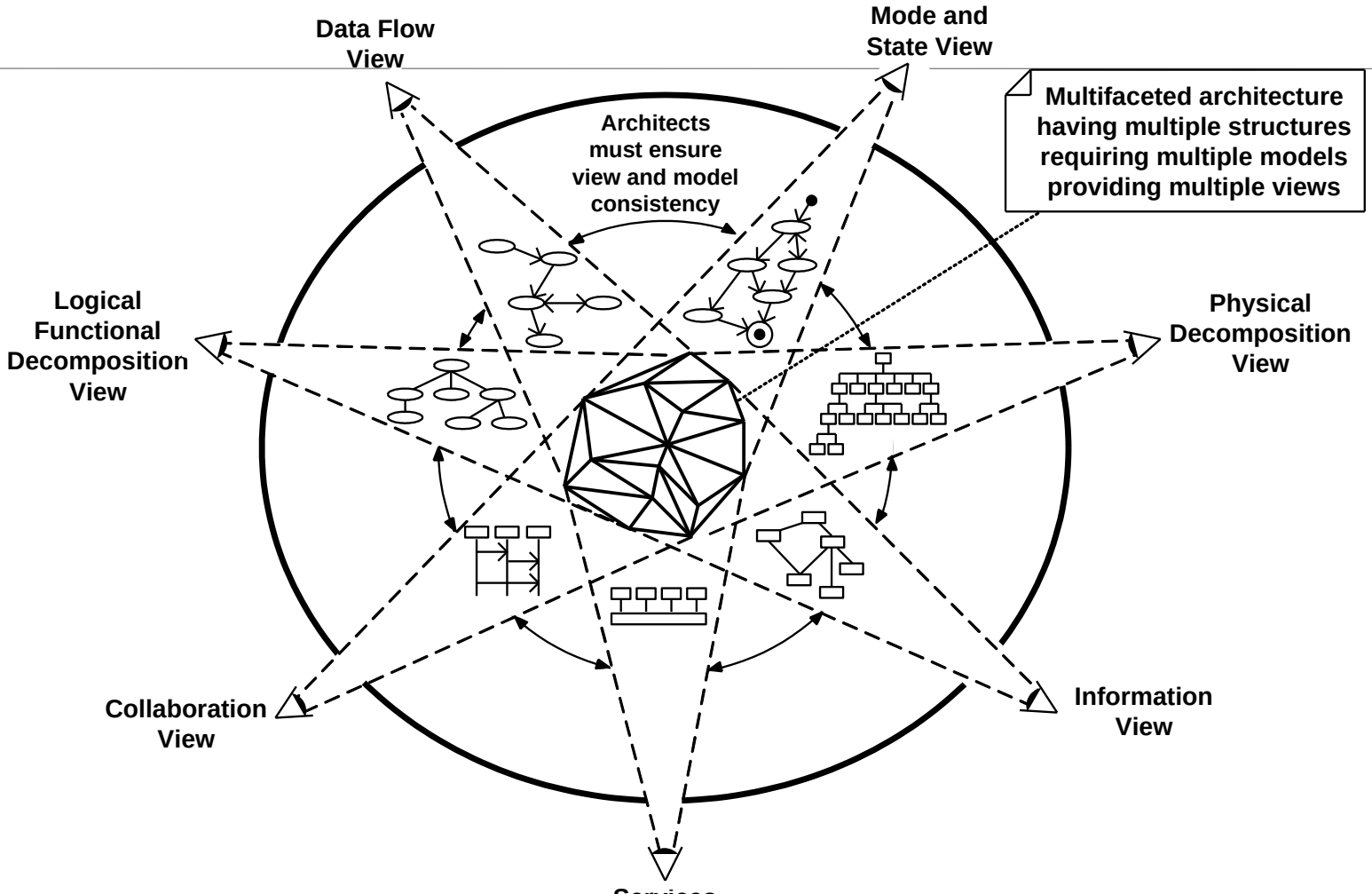
> the *Most Important, Pervasive, Top-Level, Strategic Decisions, Inventions, Engineering Trade-Offs, Assumptions,* and associated *Rationales* about *How* a System's Architectural Elements will collaborate to meet the System's Derived and Allocated Requirements

What is a System Architecture?$_2$

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-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



Architecture vs. Design

Architecture	Design
Pervasive **(Multiple Components)**	***Local*** **(Single Components)**
Strategic **Decisions and Inventions**	***Tactical*** **Decisions and Inventions**
Higher-Levels **of System**	***Lower-Levels*** **of System**
Huge Impact **on Quality, Cost, & Schedule**	***Small Impact*** **on Quality, Cost, & Schedule**
Drives **Design and Integration Testing**	***Drives*** **Implementation and Unit Testing**
Driven by **Requirements and Higher-Level Architecture**	***Driven by*** **Requirements, Architecture, and Higher-Level Design**
Mirrors **Top-Level Development Team Organization (Conway's Law)**	***No*** **Impact on Top-Level Development Team Organization**

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):

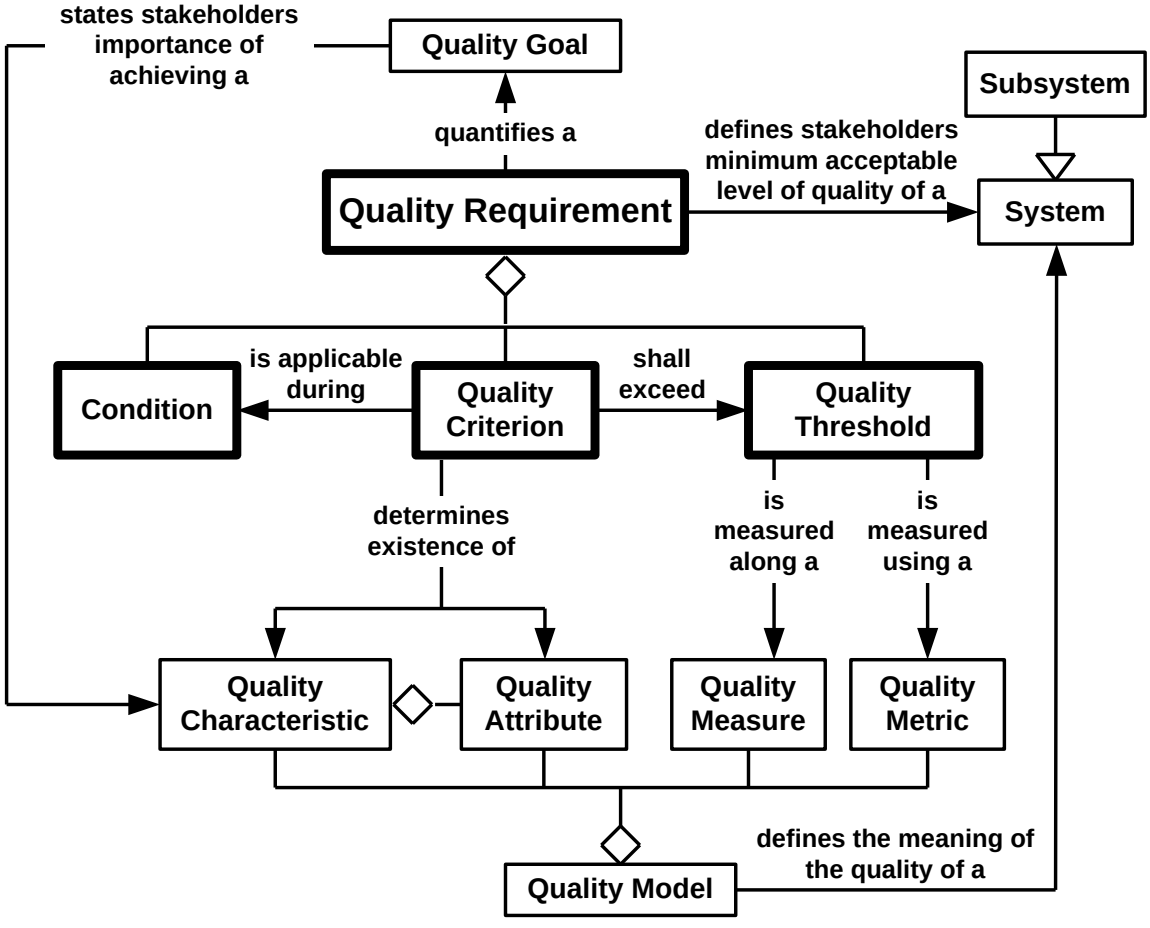
- Tailored UML becoming *de facto* Industry Standard
- SysML starting to become Popular

Visio Diagrams as Wall Posters

Whitepapers, Reports, and other Specialty-Engineering Documents:

- Performance, Fault Tolerance, Reliability, Safety, Security

Quality Requirements – Components



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

QUASAR Philosophy$_1$

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$_2$

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

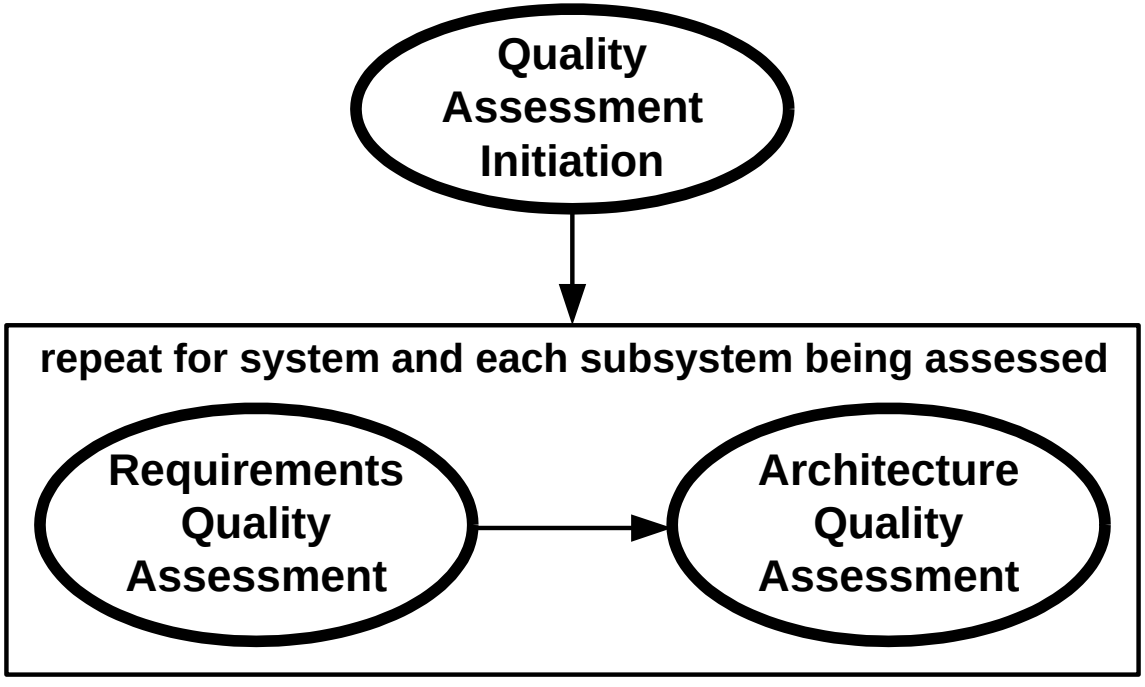
QUASAR Philosophy$_3$

Assessors should *Actively* Probe Quality Cases:

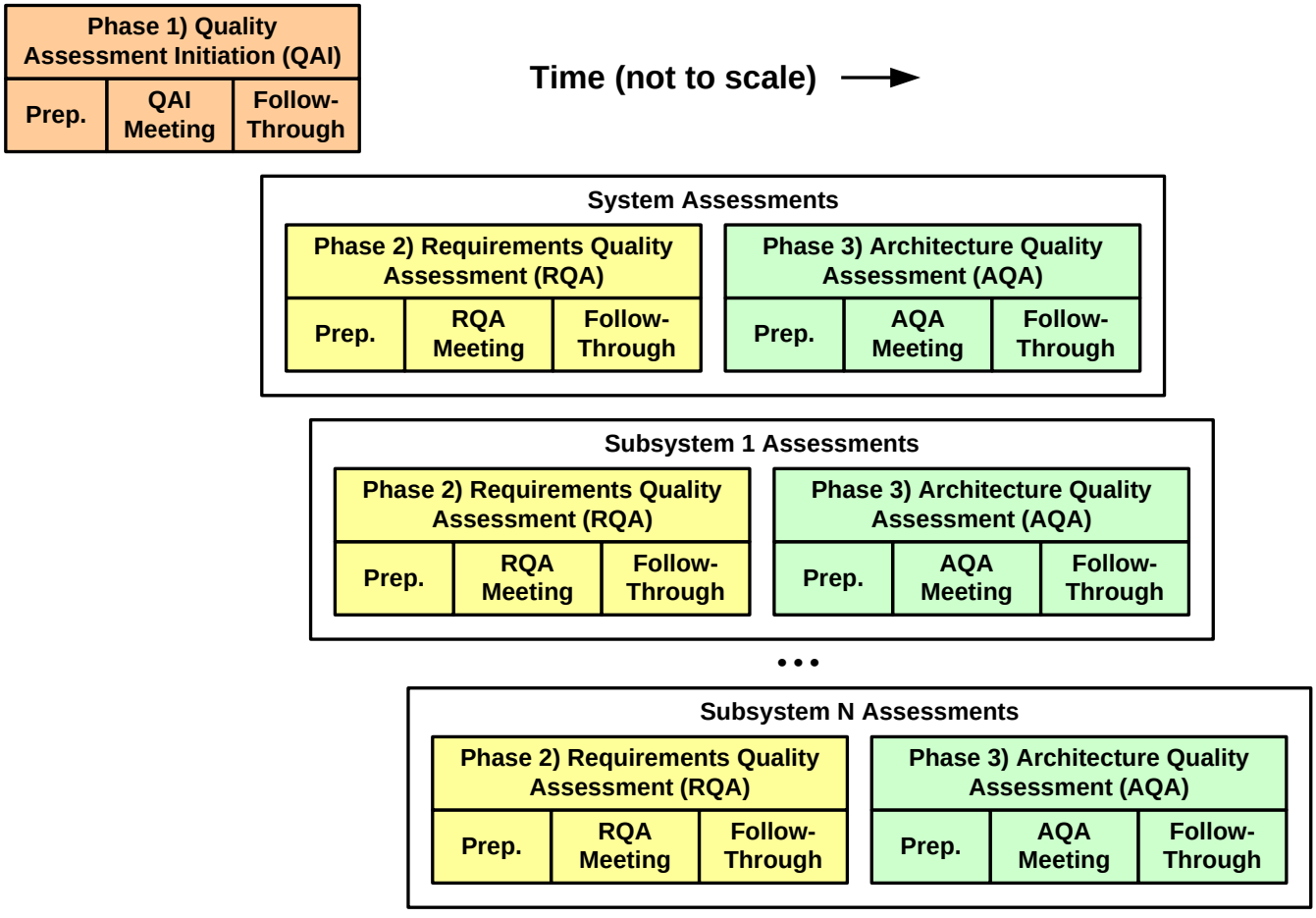
- **Claims Correct and Complete?**
  Do the Claims include *all* relevant Quality Characteristics, Quality Attributes, Quality Goals, and Quality Requirements?
- **Arguments Correct, Complete, Clear, and Compelling?**
  Do the Arguments include *all* relevant Quality Characteristics, Quality Attributes,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

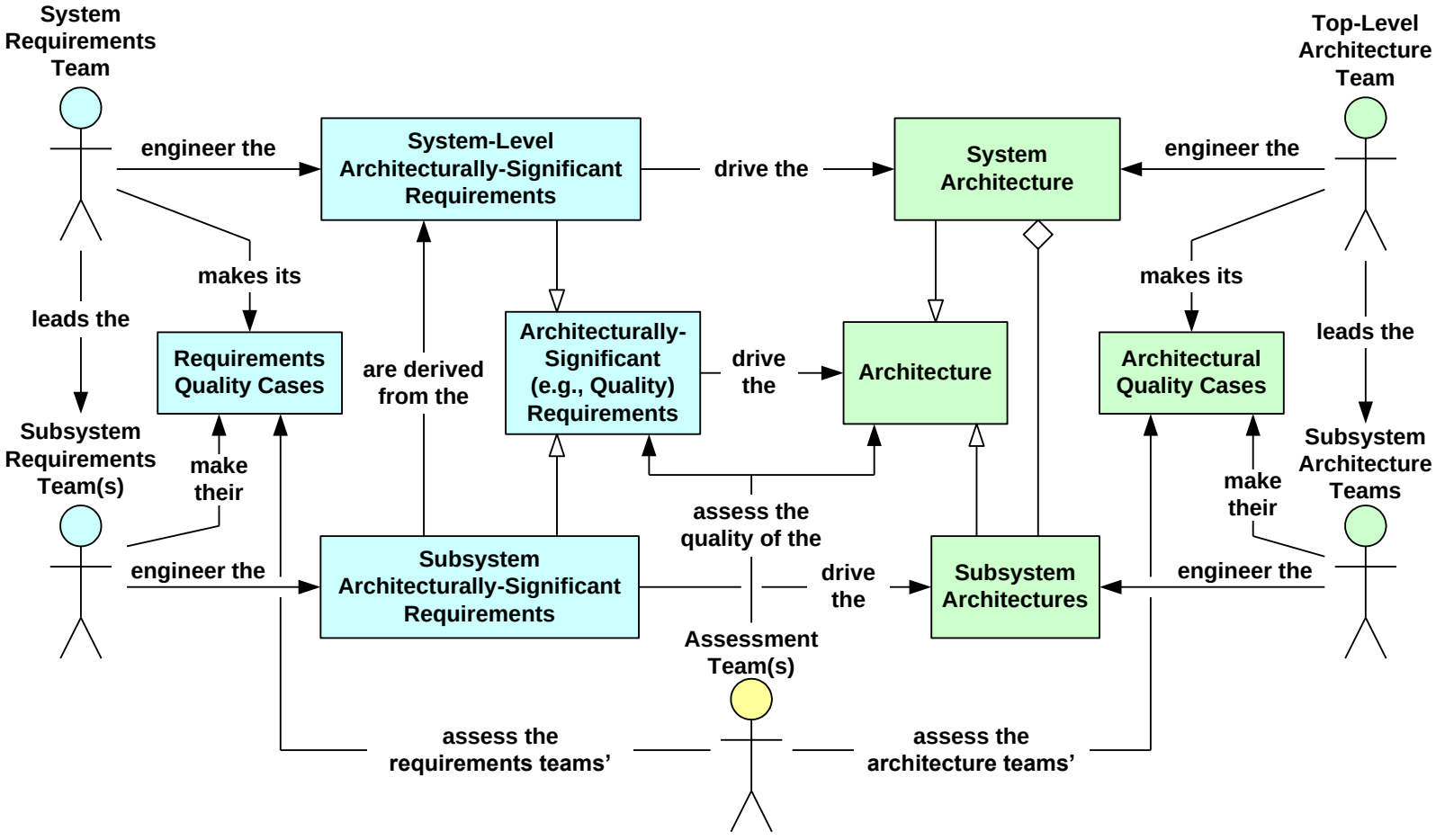
1. Quality Assessment Initiation (QAI)
2. Requirements Quality Assessment (RQA)
3. Architecture Quality Assessment (AQA)



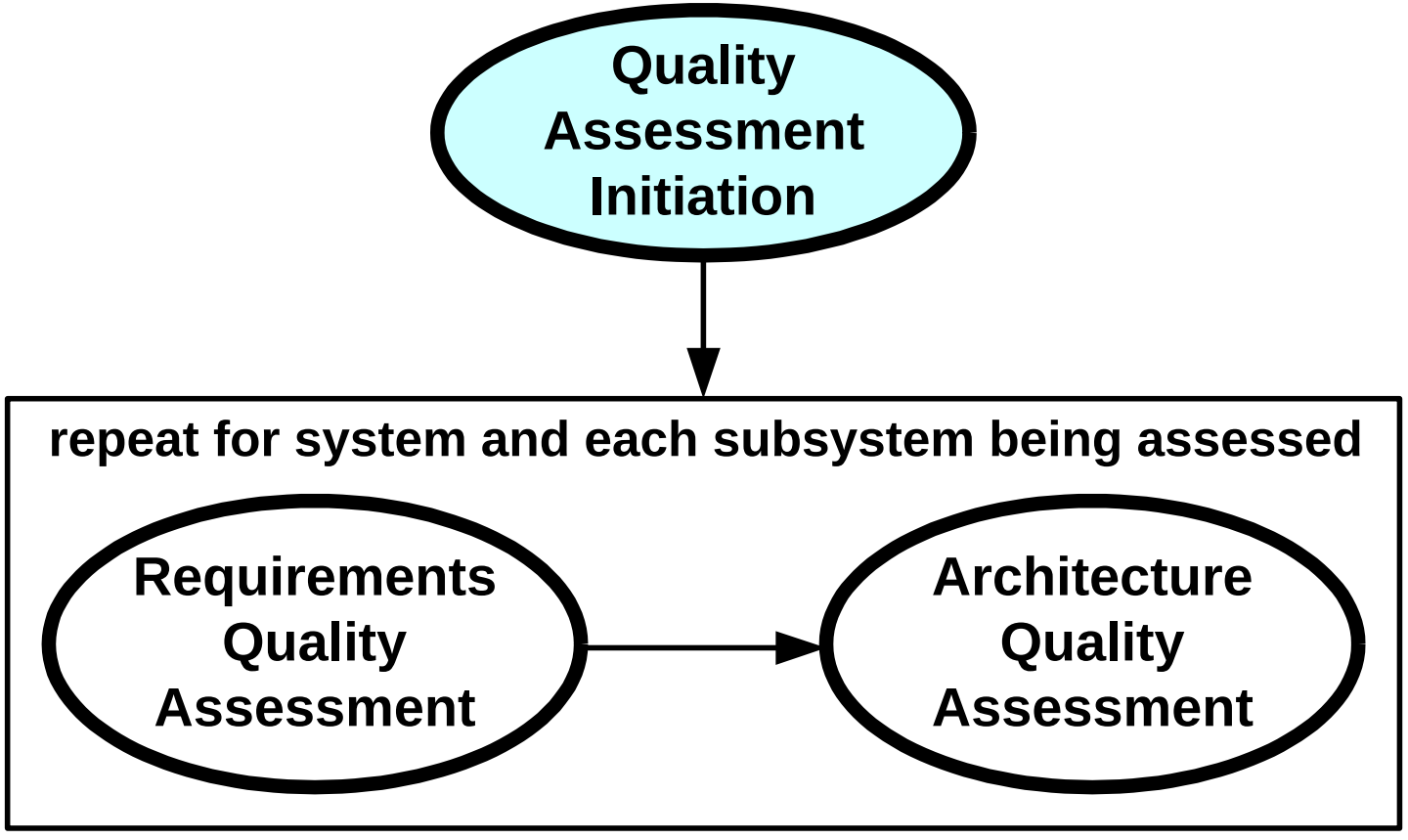
QUASAR Phases and Tasks



Quasar Teams and their Work Products



Quality Assessment Initiation (QAI)



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)

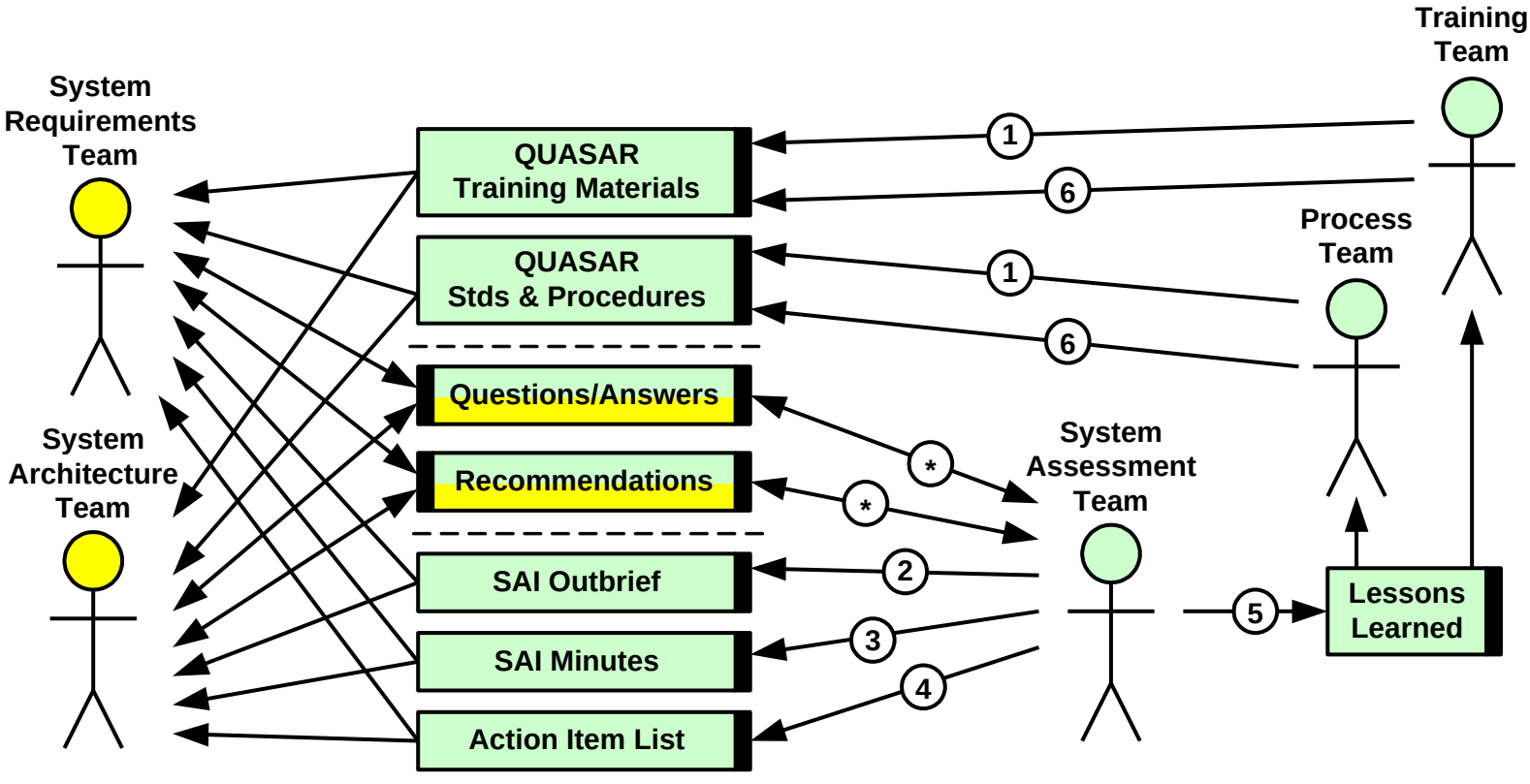
Phase 1) QAI – Meeting Task

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

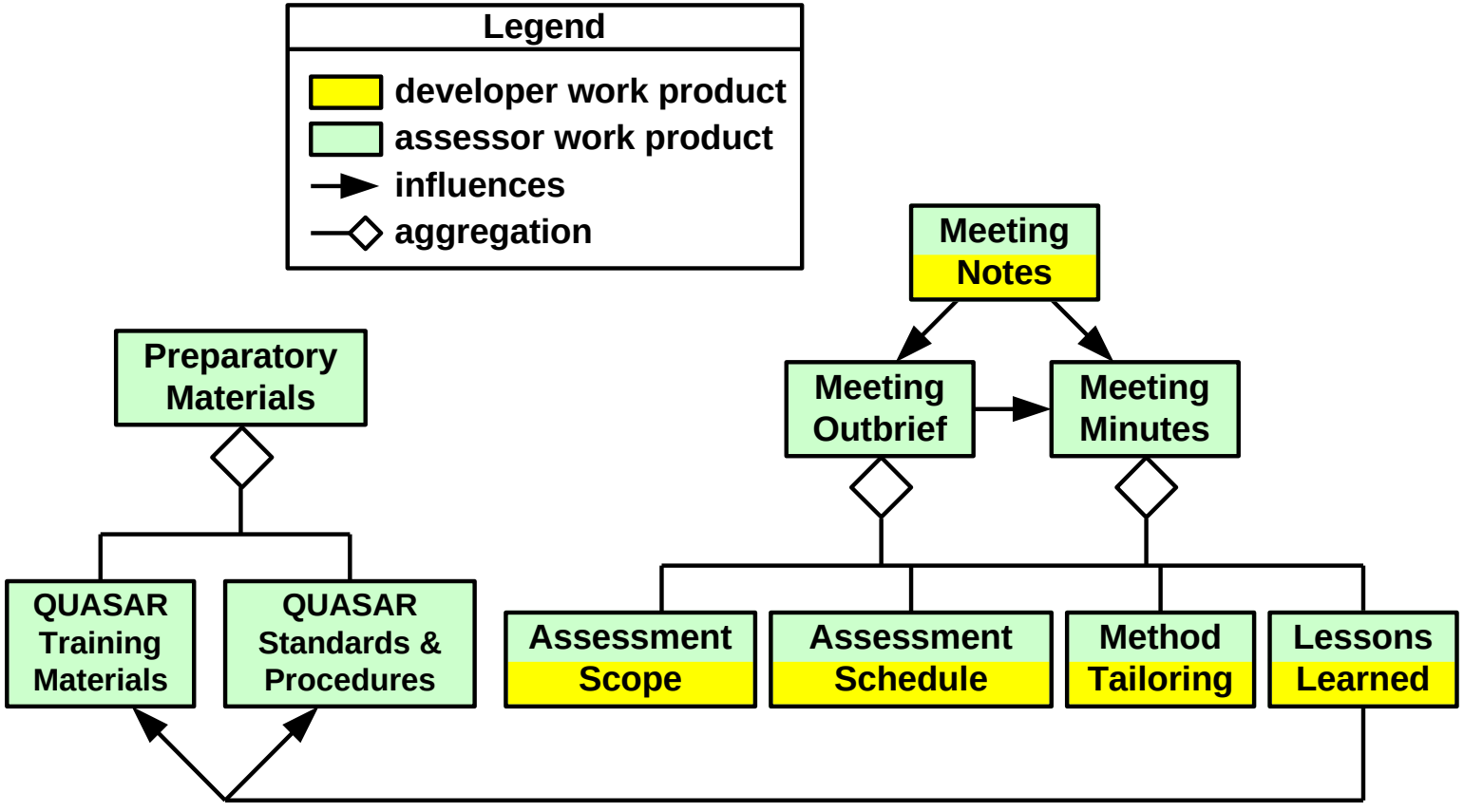
Phase 1) QAI – Follow-Through Task

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



Phase 1) QAI – Work Products



Phase 1) QAI – Lessons Learned$_1$

Ensure Appropriate Team Memberships (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:

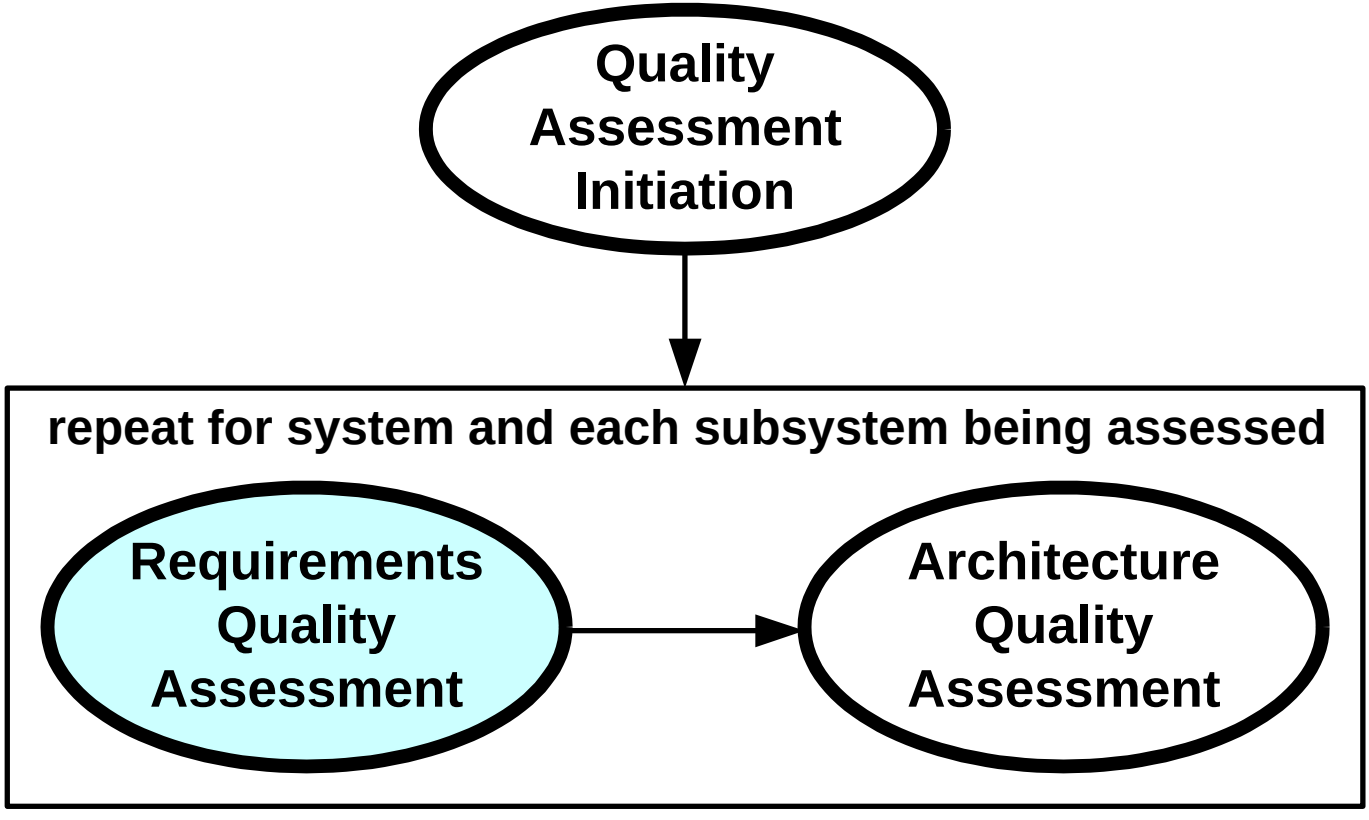
- Method Training
  (QUASAR, Requirements Engineering, and Architecting)
- System/Subsystem Training
  (Requirements and Architecture)

Phase 1) QAI – Lessons Learned$_2$

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)



Phase 2) ARA – Objectives$_1$

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$_2$

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

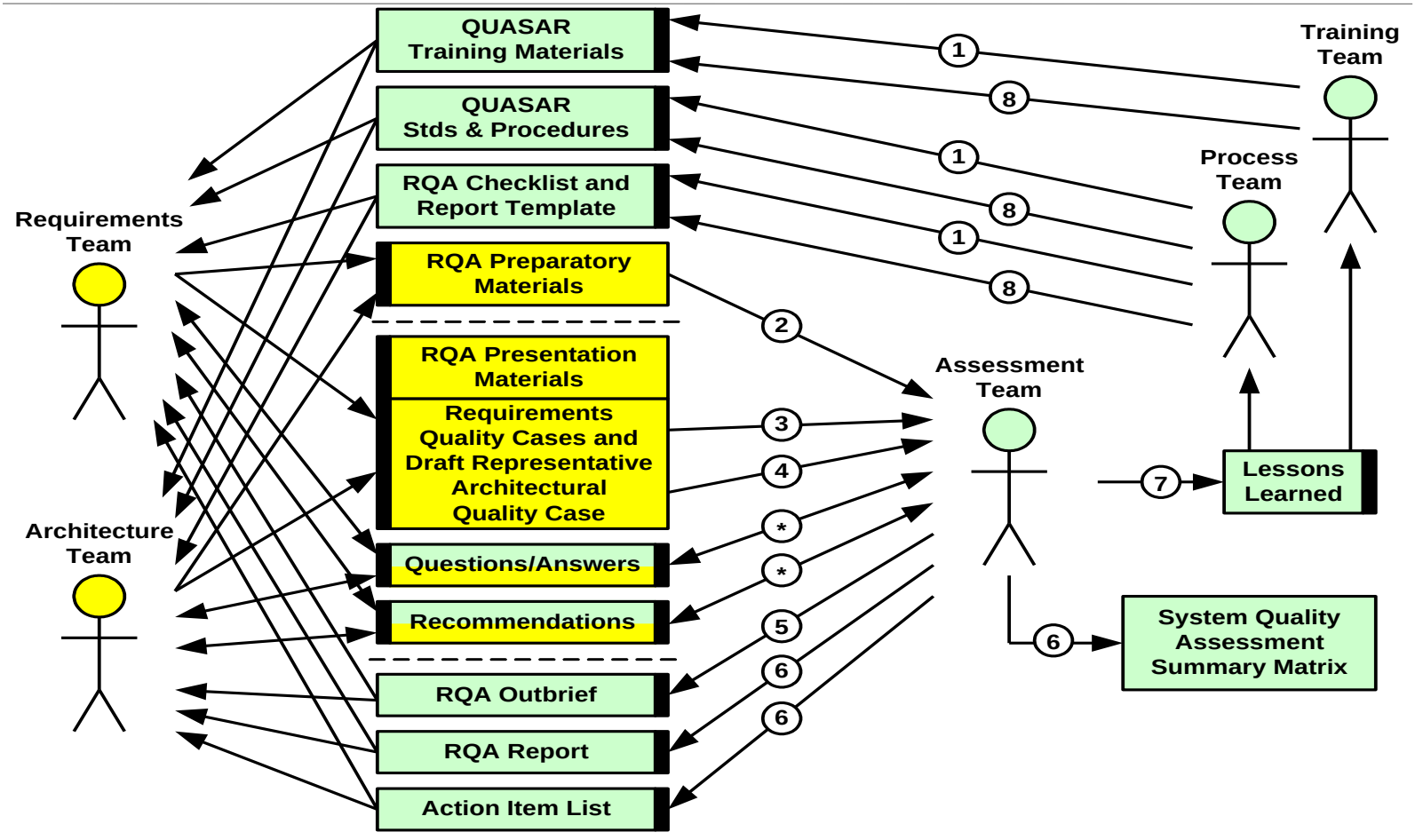
Process Team:

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

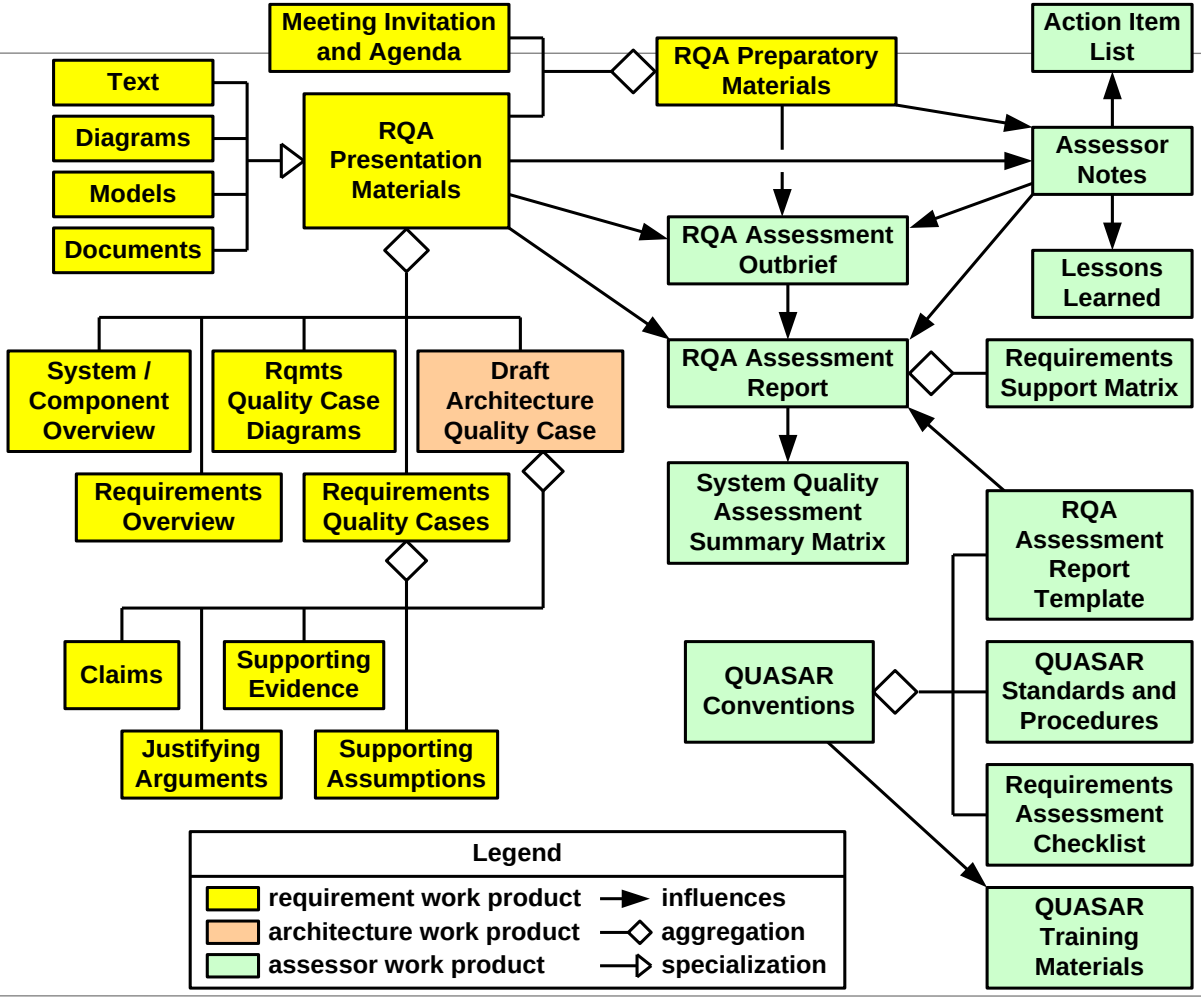
Training Team:

Updates Training Materials (if appropriate)

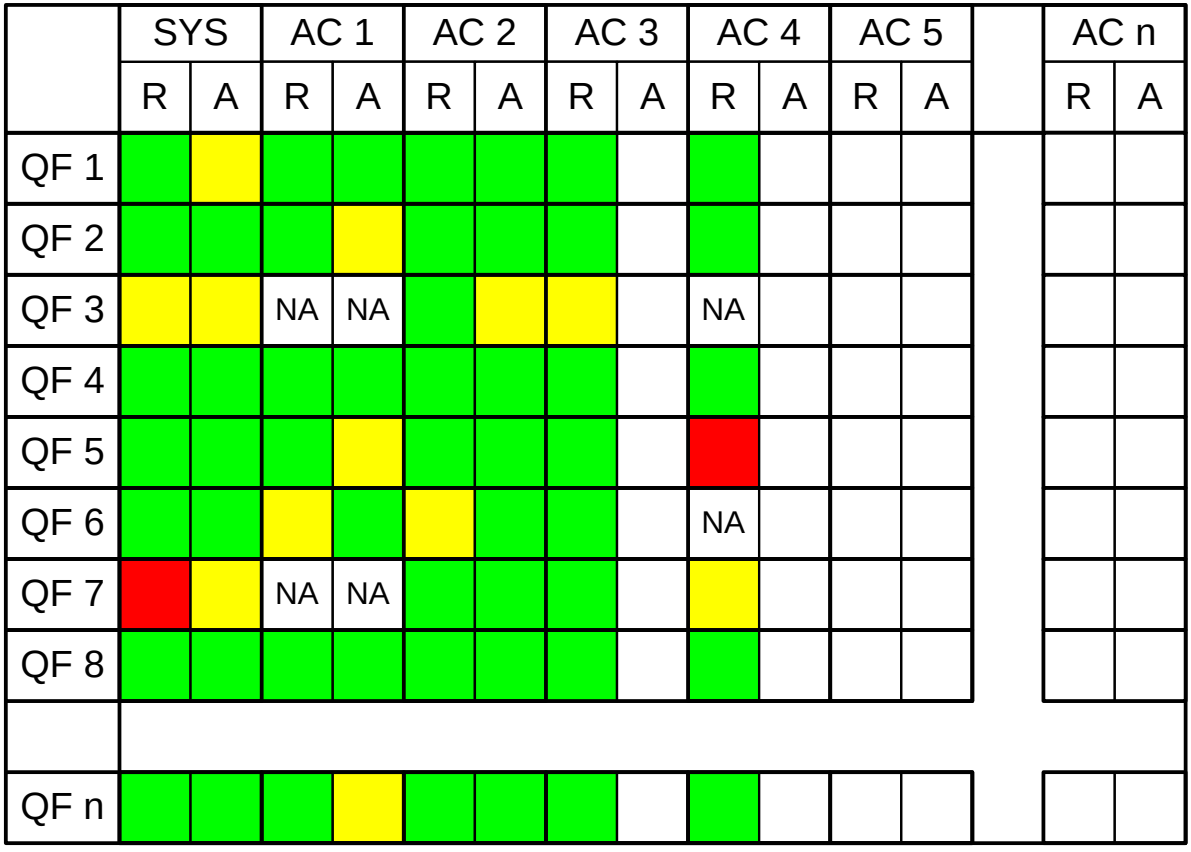
Phase 2) RQA – Work Product Workflow



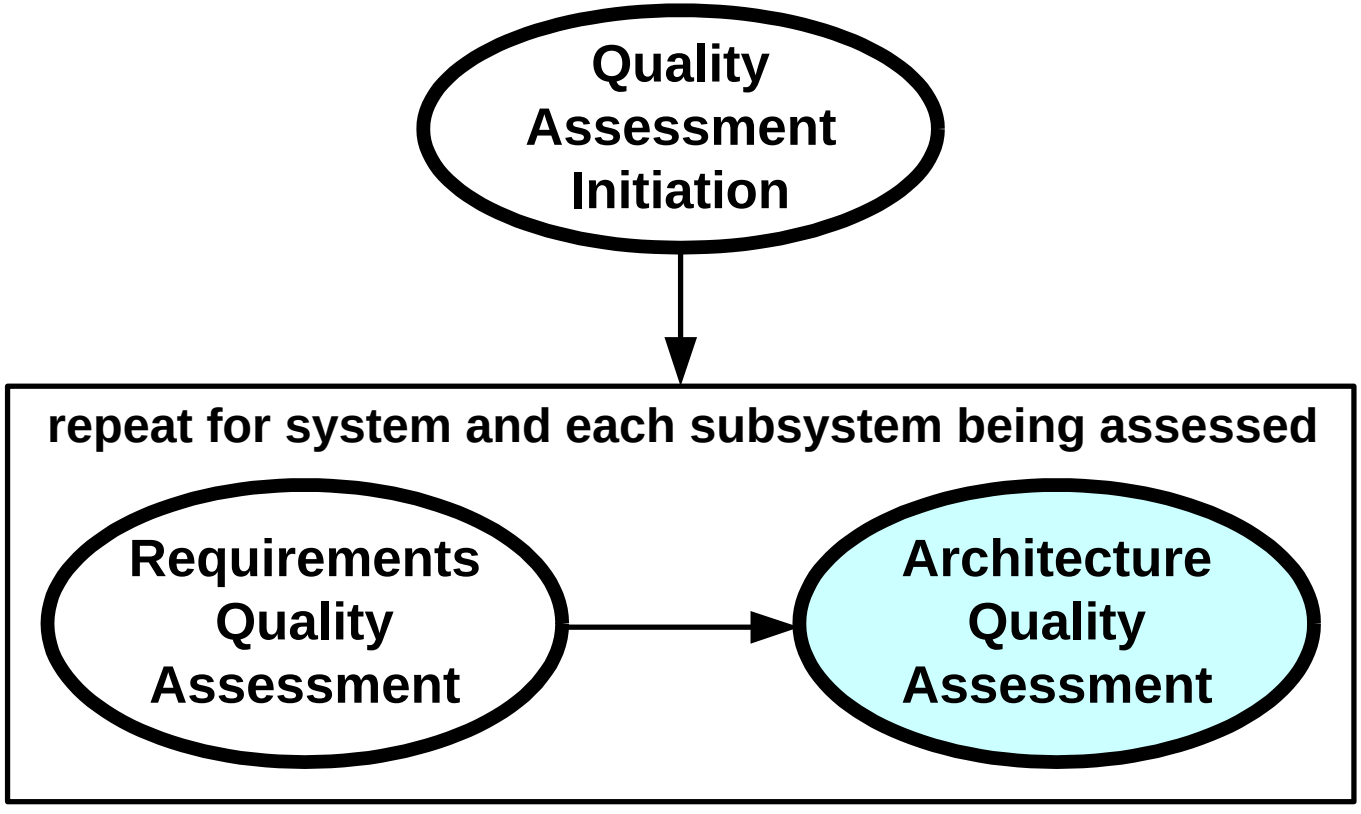
Phase 2) RQA – Work Products



System Quality Assessment Summary Matrix



Architecture Quality Assessment (AQA)



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

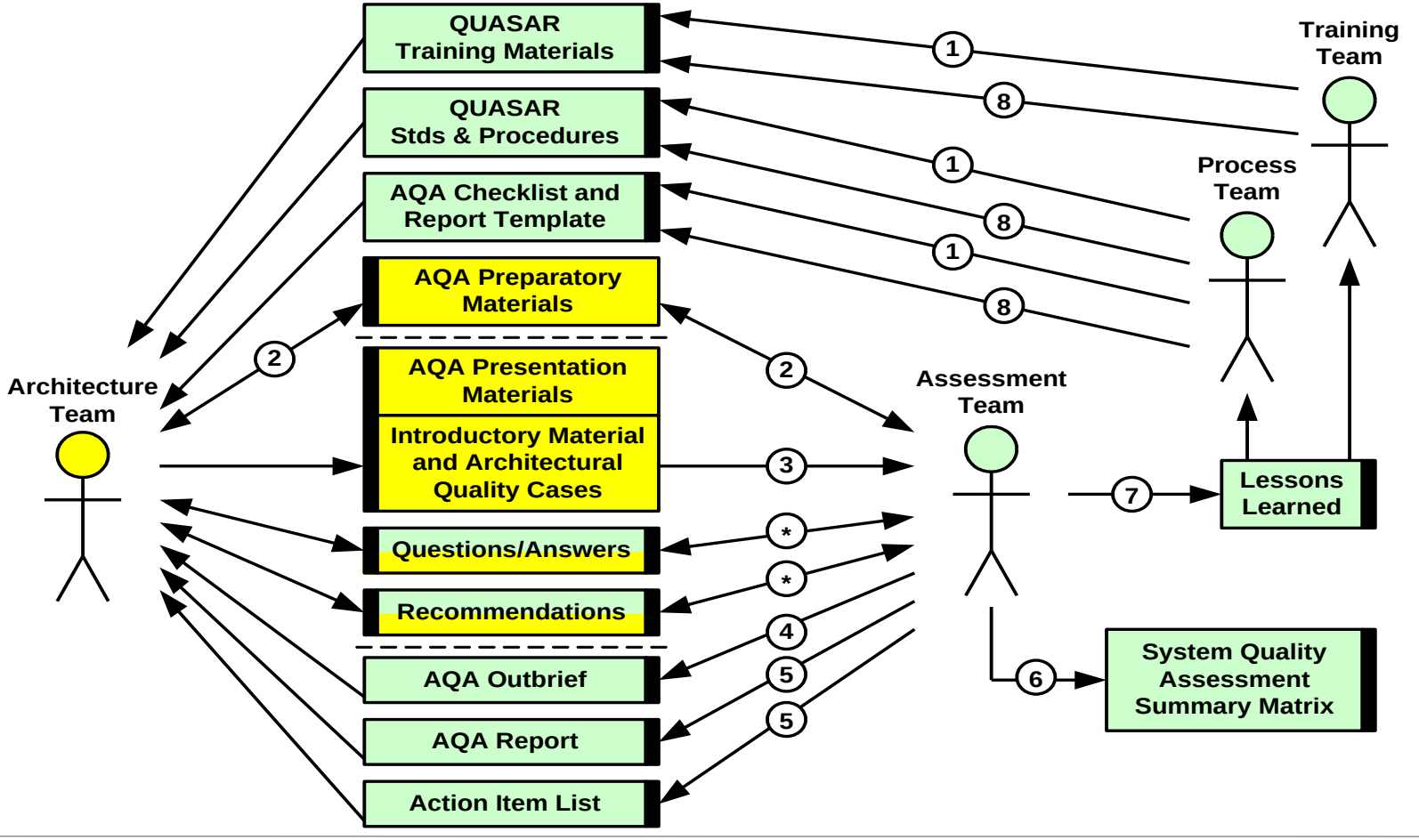
Process Team:

- Updates Assessment Method (if appropriate)

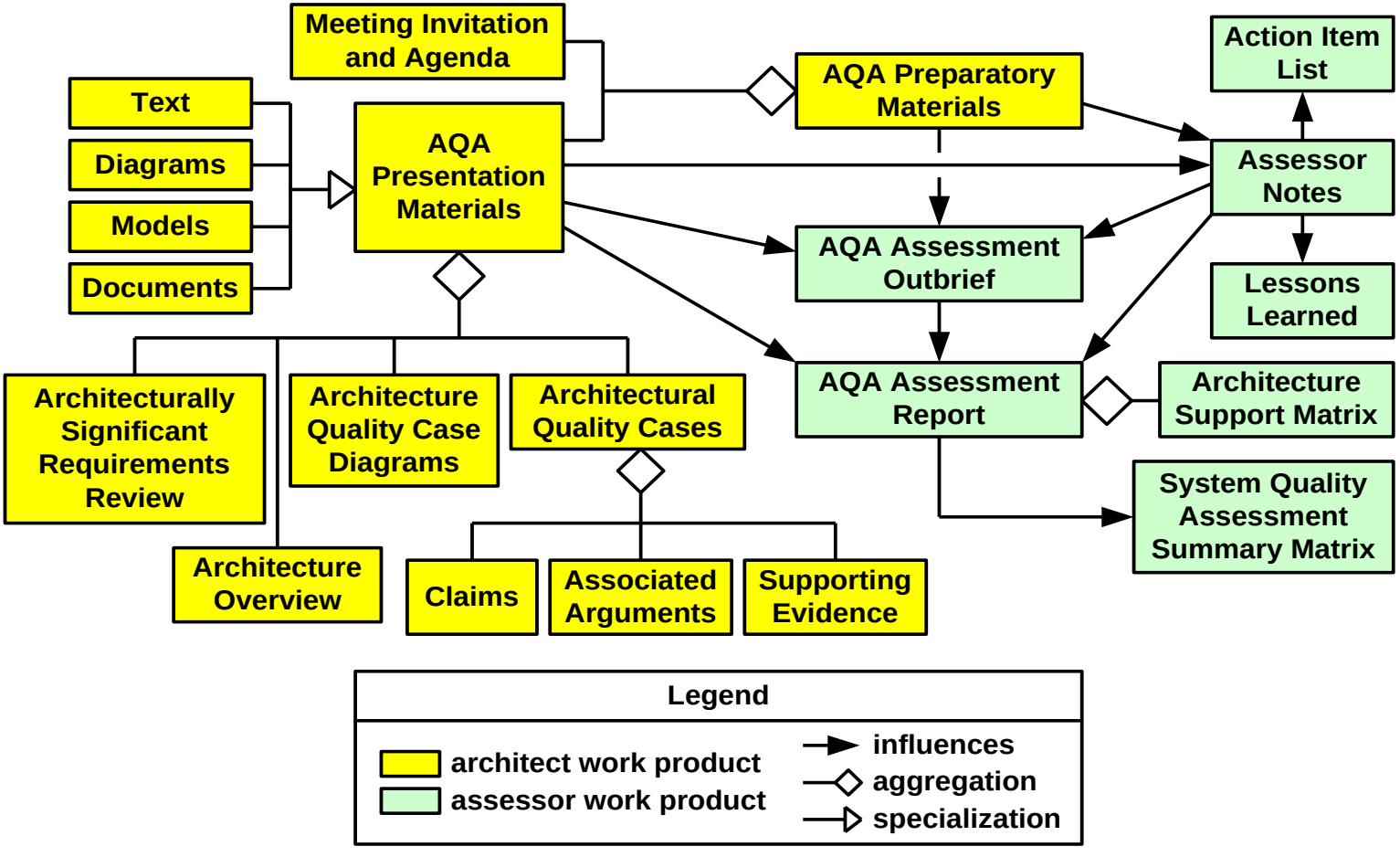
Training Team:

- Updates Training Materials (if appropriate)

Phase 3) AQA – Work Product Workflow



Phase 3) AQA – Primary Work Products



Topics

Requirements and Architecture Challenges

Underlying Concepts

QUASAR Method

Reasons to use QUASAR

QUASAR Benefits$_1$

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$_2$

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

The Method Framework for Engineering System Architectures (MFESA), Donald Firesmith et al., Auerbach Publishing, November 2008

Quasar Tutorial (1 day) : http://www.sei.cmu.edu/library/abstracts/presentations/quasartutorial2008.cfm

Software Engineering Institute | Carnegie Mellon