



**Carnegie Mellon
Software Engineering Institute**

Pittsburgh, PA 15213-3890

Architecture Business Cycle Ensuring Product Qualities

**Linda M. Northrop
Director, Product Line Systems Program
Software Engineering Institute**

**Sponsored by the U.S. Department of Defense
© 2004 by Carnegie Mellon University**



Comments on Software Architecture

Software architecture involves more than just technical requirements for a system. It also involves non-technical factors, such as

- the architect's background
- the development environment
- the business goals of the sponsoring organization

Software architecture influences the factors that affect it.

- Architects learn from experience.
- The development environment is expanded and altered.
- Businesses gain new marketing possibilities.



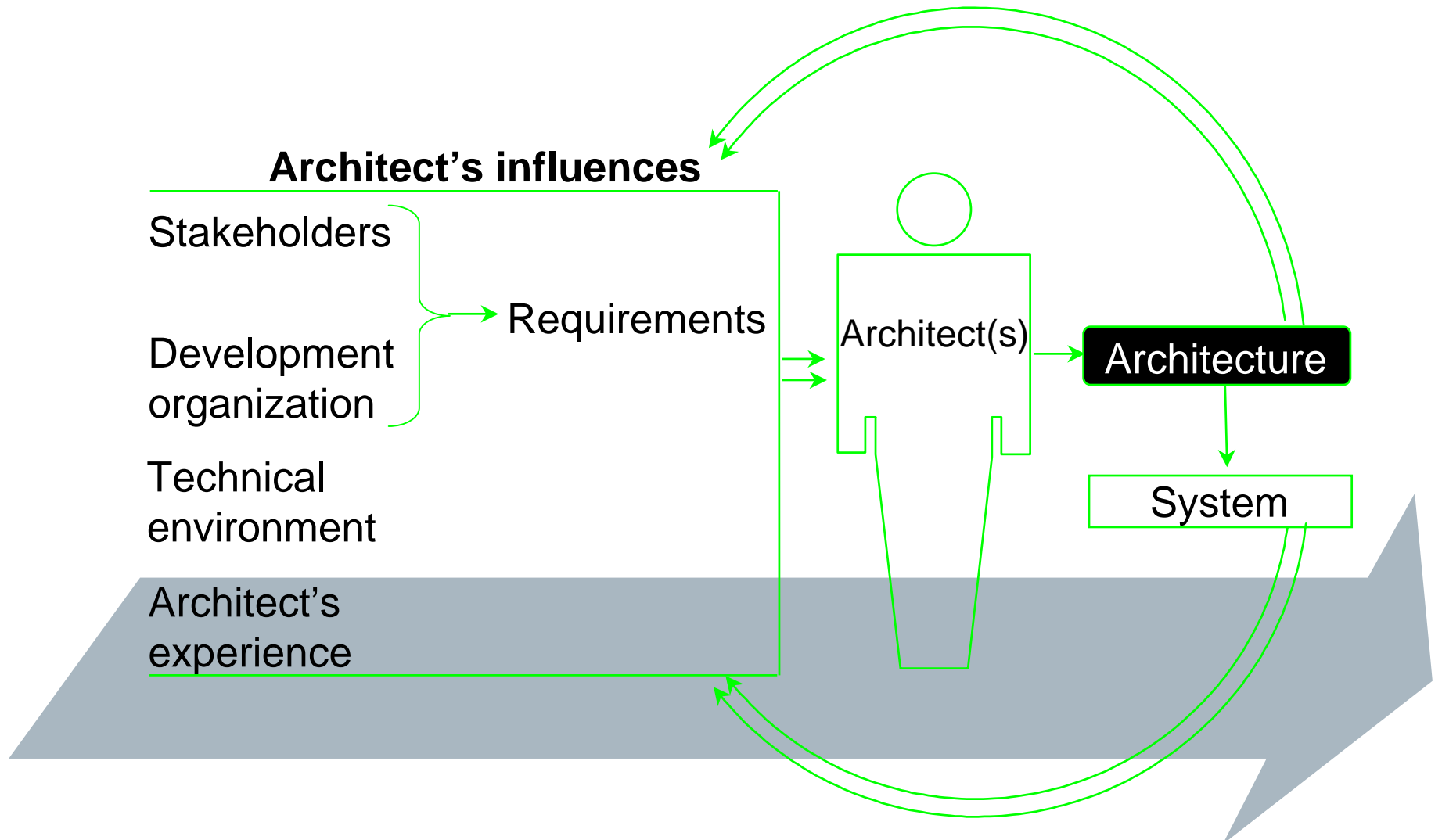
A Cycle of Influences

The relationships among business goals, product requirements, architects' experience, architectures and fielded systems form a cycle with feedback loops that an organization can manage to

- handle growth
- expand its enterprise area
- take advantage of previous investments in architectures and system building



Architecture Business Cycle (ABC)



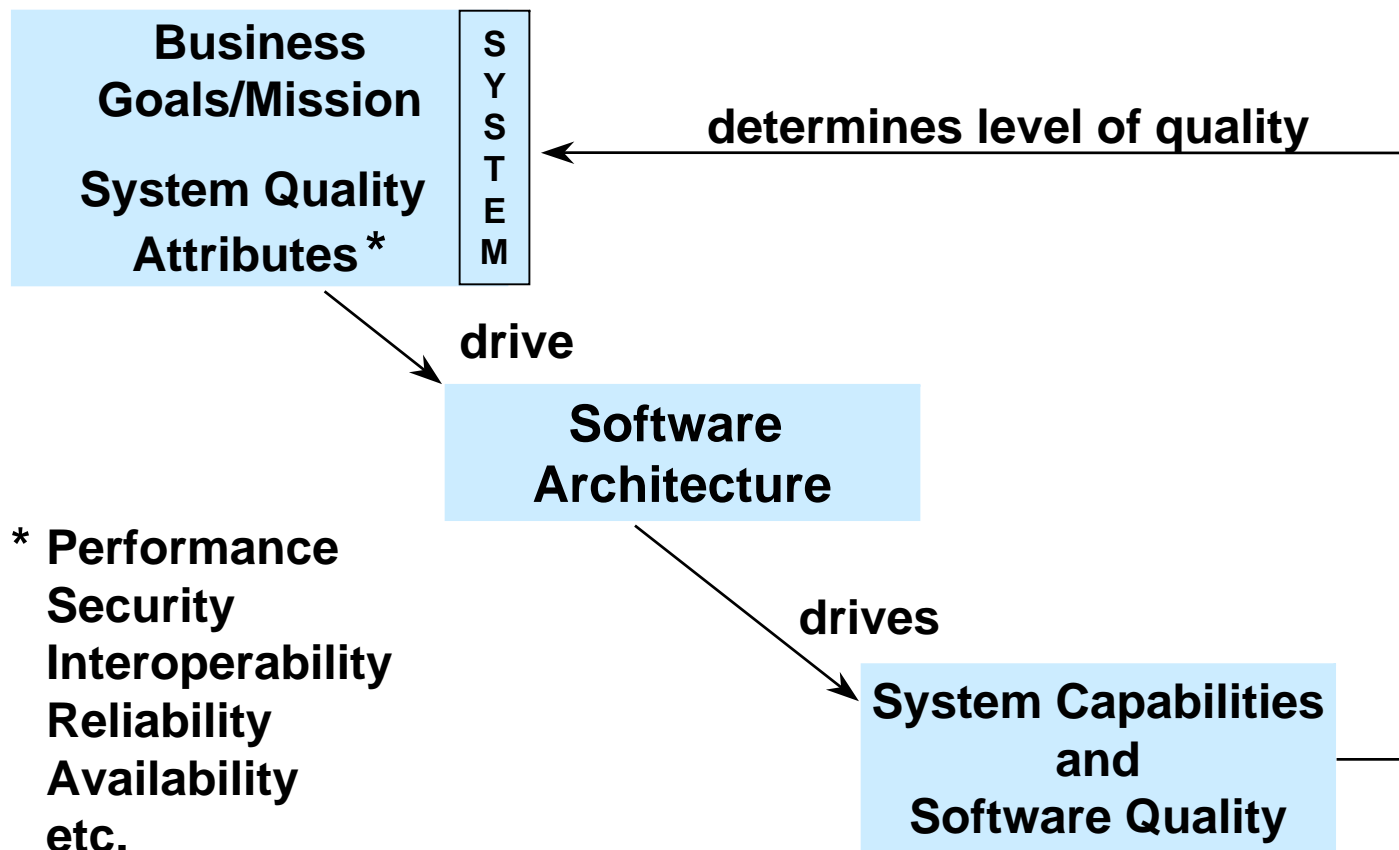


Software Architecture Axioms

1. Quality attribute requirements drive the software architecture.
2. Architecture-centric activities drive the software system life cycle.



System Qualities and Software Architecture





Describing Quality Attributes

Quality attribute names by themselves are ***not*** enough.

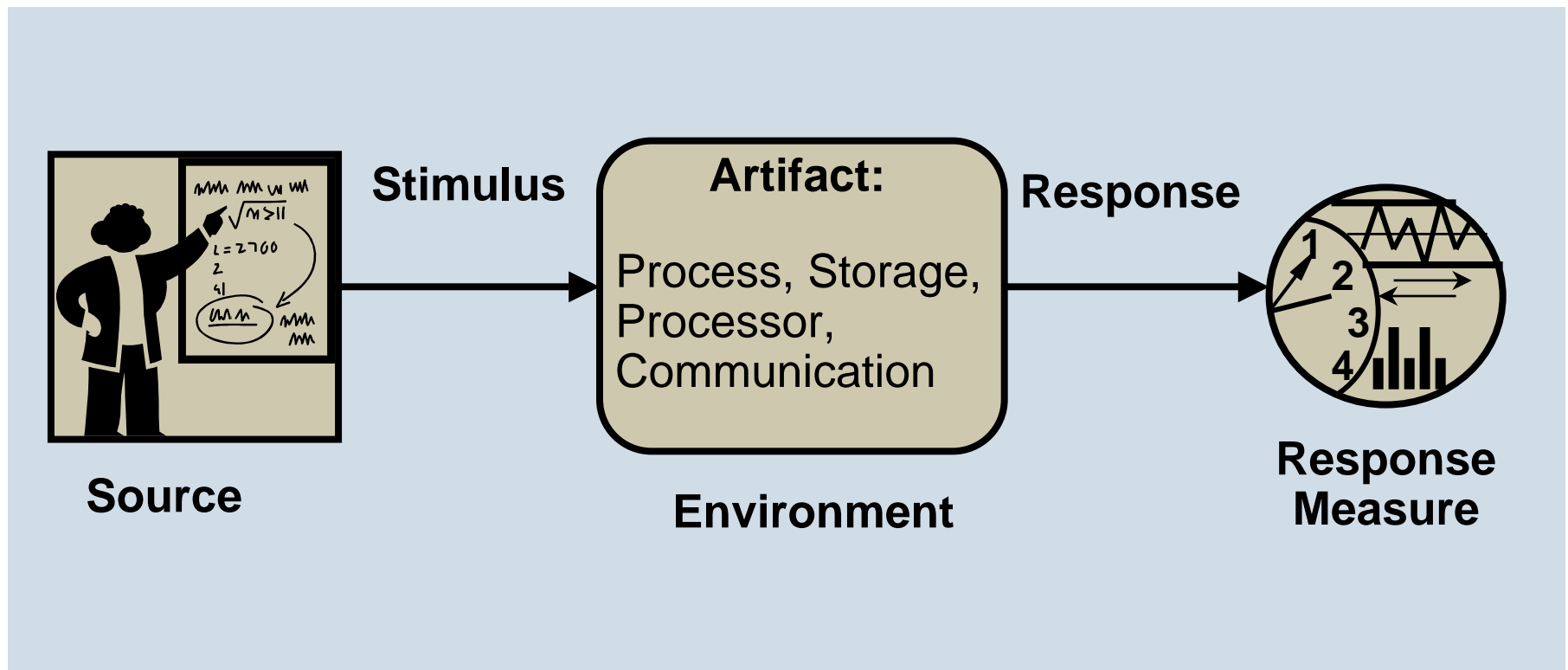
- It is meaningless to say that the system shall be "modifiable."
 - Every system is modifiable with respect to some set of changes and not with respect to some other set of changes.

A solution to the problem of describing quality attributes is to use **quality attribute scenarios** as a means to better characterize quality attributes.

Scenarios also are a powerful way to represent stakeholder views and their underlying quality attribute concerns.



Parts of a Quality Attribute Scenario





Effects of Architectural Decisions on Quality Attributes

The degree to which a system meets its quality attribute requirements is dependent on architectural decisions.

- A change in structure improving one quality often affects the other qualities.
- Architecture is critical to the realization of quality attributes.
- These product qualities should be designed into the architecture.
- Architecture can only permit, not guarantee, any quality attribute.



Software Architecture Corollaries

1. Quality attribute requirements drive the software architecture.
 - Quality attribute requirements stem from business/mission goals.
 - Scenarios are a powerful way to characterize quality attributes and represent stakeholder views.
2. Architecture-centric activities drive the software system life cycle.



Architecture-centric Development Activities

Within the Architecture Business Cycle, architecture-centric activities include the following:

- creating the **business case** for the system
- understanding the **requirements**
- **creating and/or selecting** the architecture
- **documenting and communicating** the architecture
- **analyzing or evaluating** the architecture
- **implementing** the system based on the architecture
- ensuring that the implementation **conforms** to the architecture



Quality Attributes and Software Architecture Practices

The architecture-centric activities must

- directly link to business and mission goals
- explicitly focus on quality attributes
- explicitly involve system stakeholders
- be grounded in state-of-the-art quality attribute models and reasoning frameworks



Software Architecture Corollaries

1. Quality attribute requirements drive the software architecture.
 - Quality attribute requirements stem from business/mission goals.
 - Scenarios are a powerful way to characterize quality attributes and represent stakeholder views.
2. Architecture-centric activities drive the software system life cycle.
 - These activities must have an explicit focus on quality attributes.
 - These activities must directly involve stakeholders.
 - The architecture must be descriptive and prescriptive.



Some SEI Techniques and Methods

- creating the **business case** for the system
- understanding the **requirements**
 - *Quality Attribute Workshop (QAW)*
- **creating and/or selecting** the architecture
 - *Attribute-Driven Design (ADD)*
- **documenting and communicating** the architecture
 - *Views and Beyond Approach*
- **analyzing or evaluating** the architecture
 - *Architecture Tradeoff Analysis Method (ATAM)*
 - *Cost Benefit Analysis Method (CBAM)*
- **implementing** the system based on the architecture
- ensuring that the implementation **conforms** to the architecture
 - *ARMIN (and DiscoTect)*



SEI Methods and Quality Attributes

QAW
ADD
Views and Beyond
ATAM
CBAM
ARMIN
(Discotech)



- are explicitly focused on quality attributes
- directly link to business and mission goals
- explicitly involve system stakeholders
- are grounded in state-of-the-art quality attribute models and reasoning frameworks
- are documented for practitioner consumption
- are applicable to DoD challenges and DoD systems



Relationship to ACE2 Objectives

Understandability
Maintainability
Extensibility
Executability



are all quality
attributes.

Any set of software architecture practices that explicitly focus on quality attributes can focus on these four and can be used to support all of the ACE2 stated objectives for architectures.