Achieving Critical System Qualities Through Software Architecture

Dr. Stuart Faulk
Computer and Information Science
University of Oregon

Overview

- What is “software architecture” (and what isn’t)?
- The Role of Architecture
  - System and organizational properties are influenced by architectural choices
  - Sources of architectural requirements
- The Role of Software Engineering in Disciplined Development
  - A disciplined process
  - Using architectural structures to achieve design goals

Working Definition

"The software architecture of a program or computing system is the structure or structures of the system, which comprise software components, the externally visible properties of those components, and the relationships among them."

From *Software Architecture in Practice,* Bass, Clements, Kazman

Examples

- **An architecture comprises a set of**
  - **Software components**
  - **Component interfaces**
  - **Relationships among them**
- **Examples**

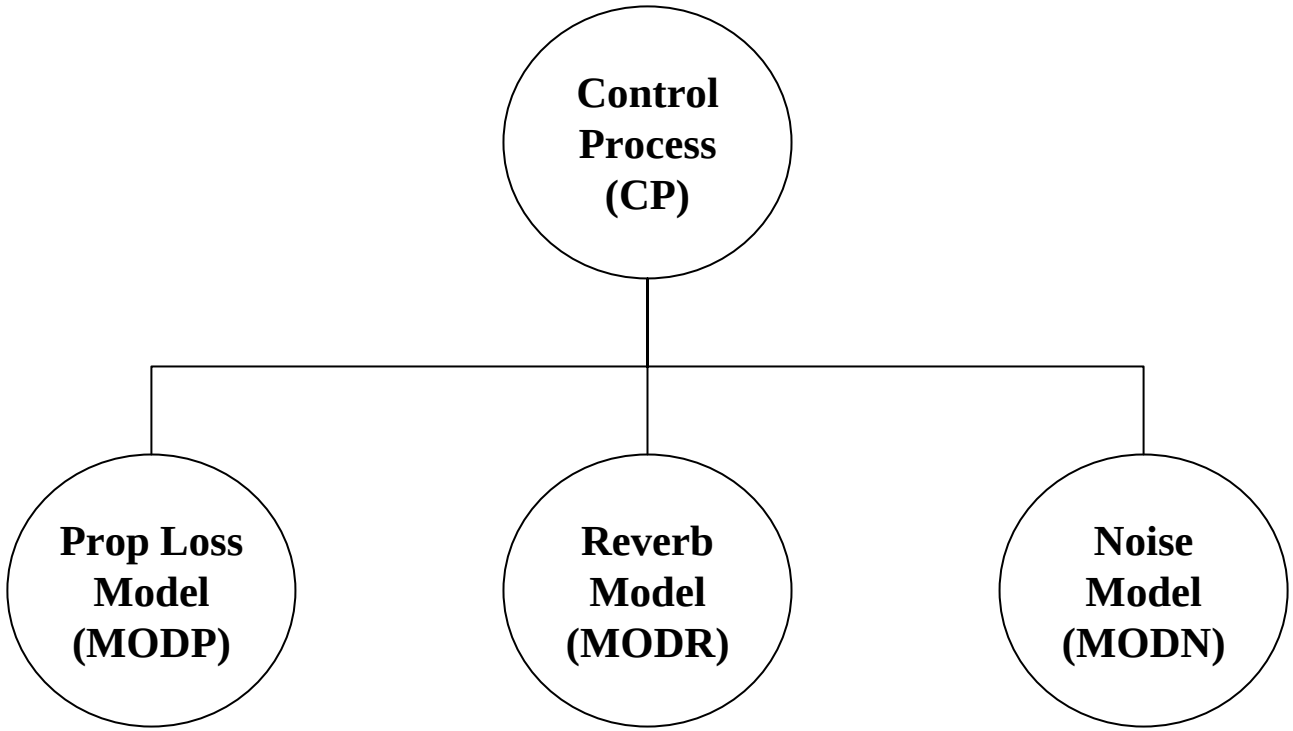
Structure	Components	Interfaces	Relationships
Calls Structure	Programs	Program interface and parameter declarations.	Invokes with parameters (A calls B)
Data Flow	Functional tasks	Data types or structures	Sends-data-to
Process	Sequential program (process, thread, task)	Scheduling and synchronization constraints	Runs-concurrently-with, excludes, precedes

Implications of the Definition

> "The software architecture of a program or computing system is the structure or structures of the system, which comprise software components, the externally visible properties of those components, and the relationships among them." - Bass, Clements, Kazman

- Systems typically comprise more than one architecture
  - There is more than one useful decomposition into components and relationships
  - Each addresses different system properties or design goals
- It exists whether any thought goes into it or not!
  - Decisions are necessarily made if only implicitly
  - Issue is who makes them and when
- Many "architectural specifications" aren't

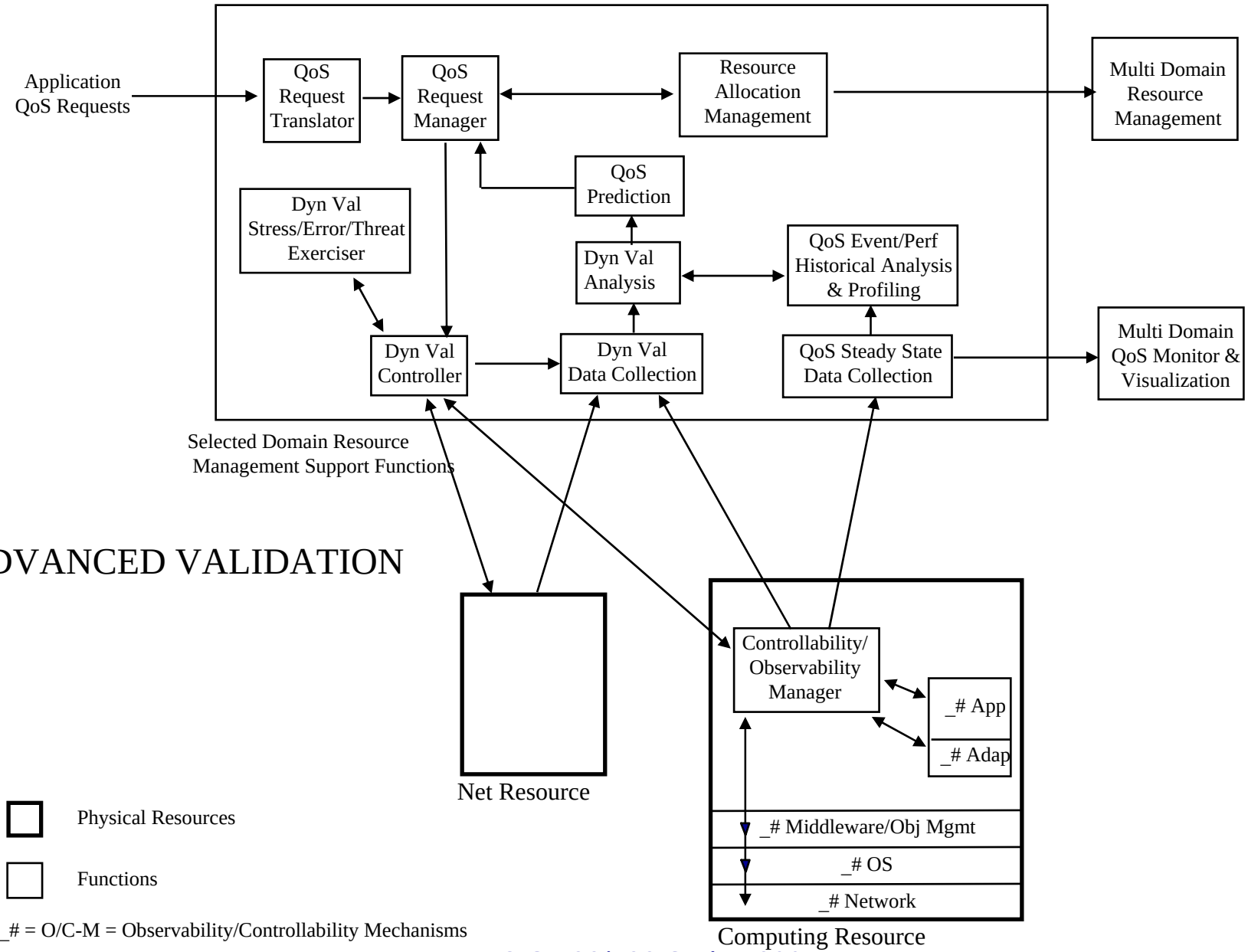
Is it Architecture?



Typical (but uninformative) architectural diagram

- What is the nature of the components?
- What is the significance of the link?
- What is the significance of the layout?

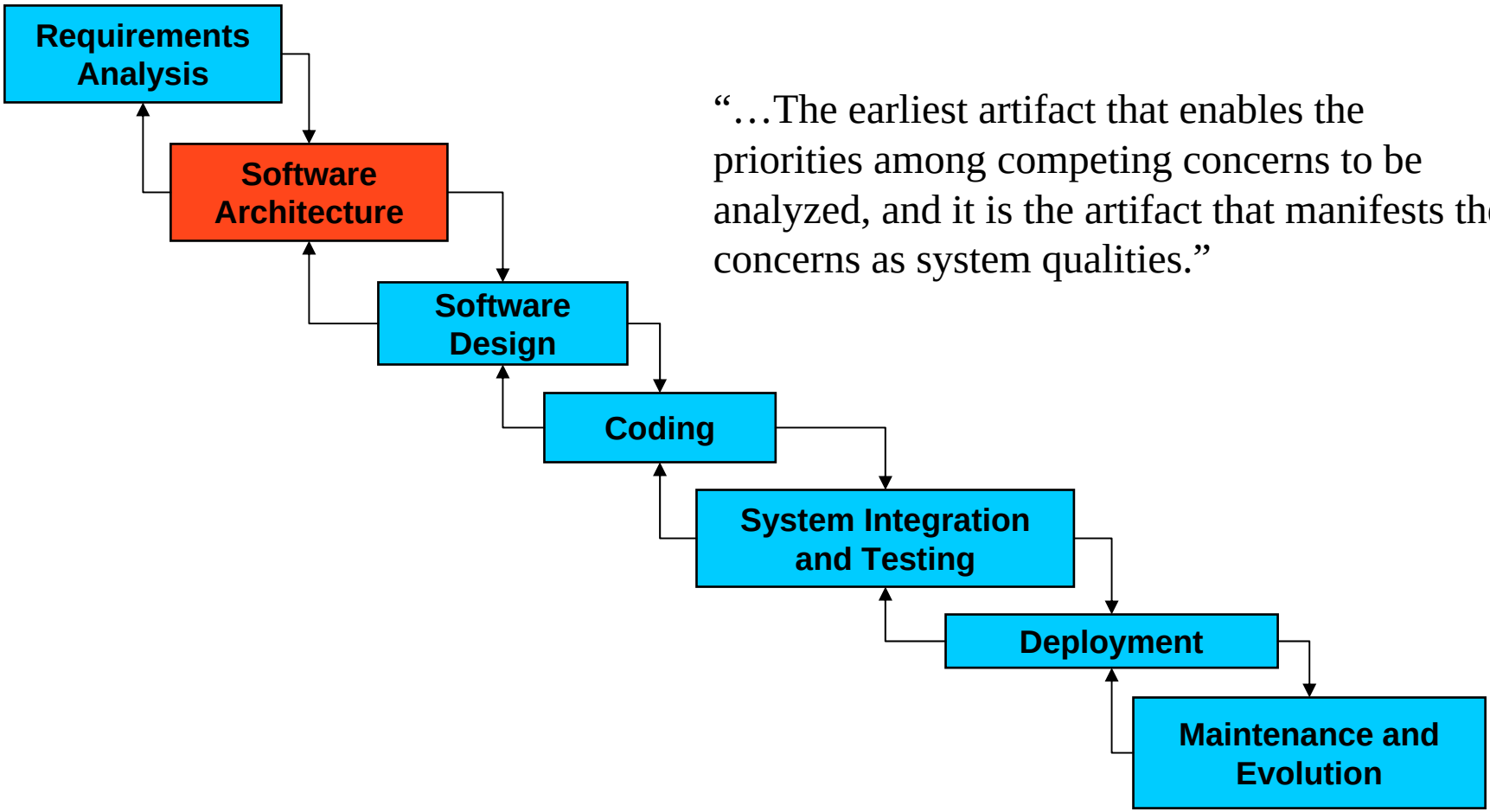
Is it architecture?



The Role of Architecture

- Which system or development characteristics are determined by the architecture?
- Where do architectural requirements originate?

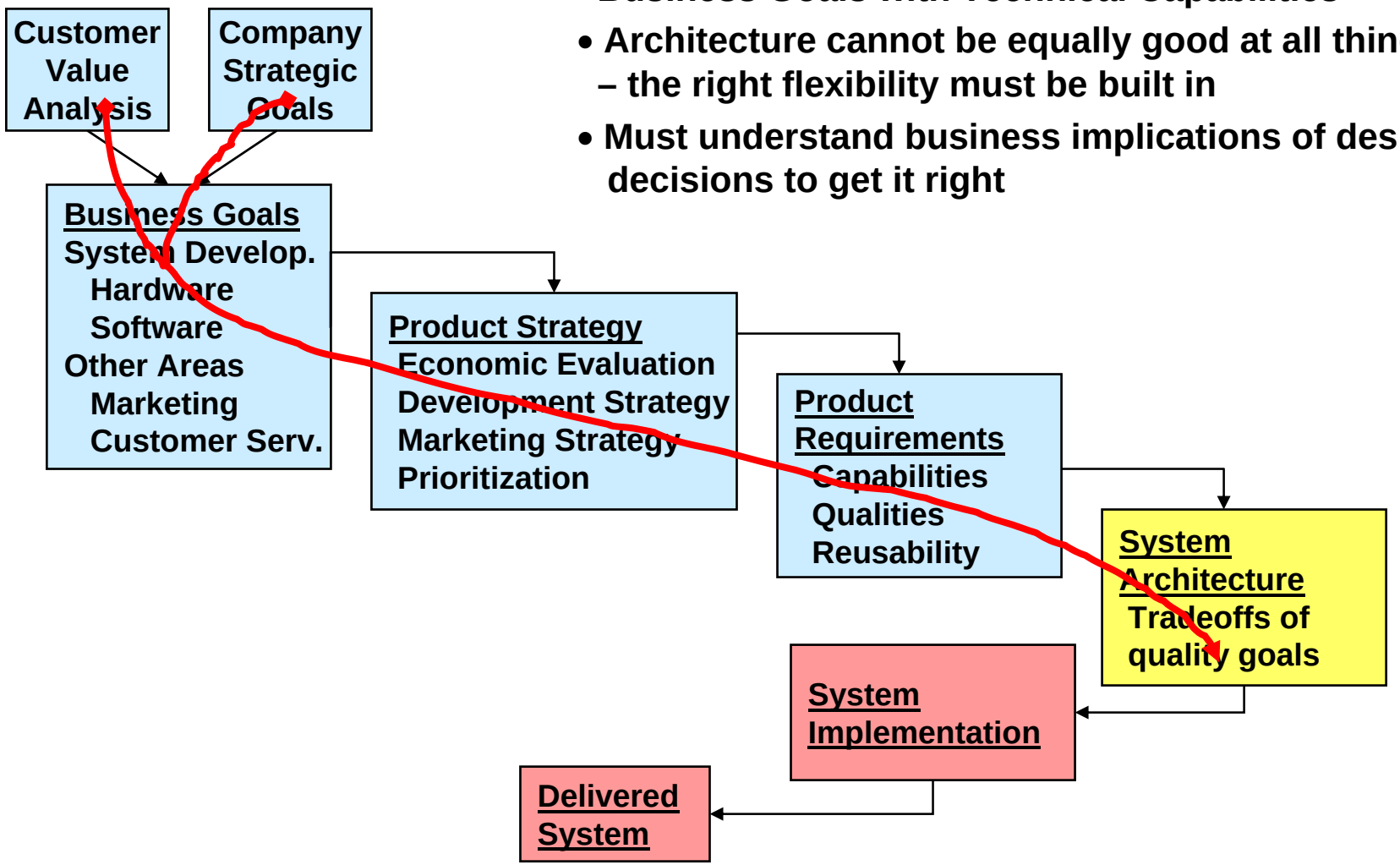
Fit in the Development Cycle



"…The earliest artifact that enables the priorities among competing concerns to be analyzed, and it is the artifact that manifests the concerns as system qualities."

Product Development Cycle

- **Goal: Development process must synchronize Business Goals with Technical Capabilities**
- **Architecture cannot be equally good at all things – the right flexibility must be built in**
- **Must understand business implications of design decisions to get it right**



Effects of Architectural Decisions (What?)

- What kinds of system and development properties are affected by the system structure(s)?
- System run-time properties
- System static properties
- Production properties? (effects on project)
- Business/Organizational properties?

Affects of Architectural Decisions (What?)

- What kinds of system and development properties are affected by the system structure(s)?
- System run-time properties
  - Performance, Security, Availability, Usability
- System static properties
  - Modifiability, Portability, Reusability, Testability
- Production properties? (effects on project)
  - Project cost, time to market
- Business/Organizational properties?
  - Lifespan, Versioning, Interoperability, Target market

Effects of Architectural Decisions (Who?)

- Who has a stake in what choices are made between architectural properties?

Affects of Architectural Decisions (Who?)

- Organizations that have a stake in what choices are made between those properties
  - Developing organization's management, marketing, end users
  - Maintenance organization, IV&V, Customers
  - Regulatory agencies (e.g., FAA)
- There are many interested parties (*stakeholders*) with many diverse and often conflicting interests
- **Important because their interests defy mutual satisfaction**
  - **There are inherently tradeoffs in most architectural choices**
  - **E.g. Performance vs. security, initial cost vs. maintainability**
- Making successful tradeoffs requires understanding the nature, source and priority of these constraints.

Importance of Stakeholders

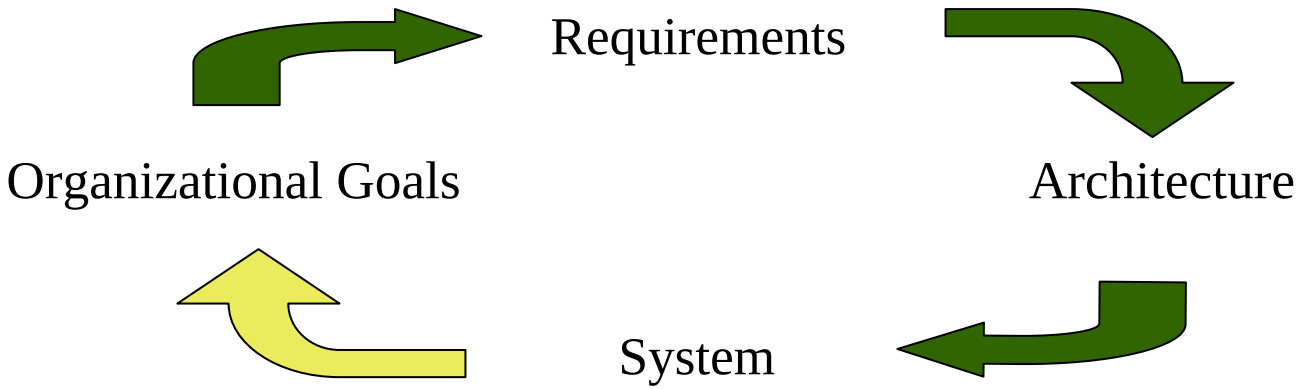
- There are many interested parties (stakeholders) with many diverse and often conflicting interests
- **Important because their interests defy mutual satisfaction**
  - There are inherently tradeoffs in most architectural choices
  - E.g. Performance vs. security, initial cost vs. maintainability
- Making successful tradeoffs requires understanding the nature, source and priority of these constraints.

Architectural Feedback

Architecture influences the things that influence it

- Architecture influences organization
  - Influences organizational structure by work breakdown
  - Changing architectures requires reorganization
- Architecture influences future architectures
  - People understand what they have experience with
  - Organizational inertia
- Architecture influences business decisions
  - Facilitates some changes, discourages others
  - May influence customer requirements since it affects the cost and speed of subsequent development

Architecture Business Cycle



- Viewed in the business context, cycle, not waterfall
- Implies need to think about long term implications of architectural design decisions

Summary

- Earliest set of design decisions – hence, most influential and hardest to change
- Determines a wide range of critical system, production, and business properties
- A product of tradeoffs between conflicting demands by different stakeholders
- Requirements come from product/business goals and subsequently affect them
- Implication: good design is a balance of technical, business and social influences
  - Must understand the context
  - Must communicate effectively
  - Must negotiate the requirements
  - Must think strategically about the effects of decisions

Architectural Qualities

Terminology

- Avoid “functional” and non-functional" classification
- Behavioral Requirements - Any and all information necessary to determine if the run-time behavior of a given implementation constitutes an acceptable system.
  - All quantitative constraints on the system's run-time behavior
  - Other objective measures (safety, performance, fault-tolerance)
  - In theory all can be validated by observing the running system and measuring the results
- Developmental Quality Attributes - Any constraints on the system's static construction
  - Testing ease (testability), ease of change (mutability), maintainability, reusability
  - Measures of these qualities are necessarily relativistic (I.e., in comparison to something else

Behavioral vs. Developmental

Behavioral (observable)

- Performance
- Security
- Availability (fault-tolerance)
- Security
- Usability (responsiveness?)

Properties resulting from the properties of components, connectors and interfaces that exist at run time.

Developmental Qualities

- Modifiability
- Portability
- Reusability
- Integrability
- Testability
- Understandability*

Properties resulting from the properties components, connectors and interfaces that exist at design time *whether or not they have any distinct run-time manifestation*.

Behavioral vs. Developmental (2)

Behavioral (observable)

- Performance
- Security
- Availability (fault-tolerance)
- Usability (responsiveness?)

Developmental Qualities

- Modifiability
- Portability
- Reusability
- Integrability
- Testability

- **Usefully viewed as distinct concerns**
  - Visible at different times
  - Can focus on one at a time
- **Often not easy to separate in practice**
  - Design time mechanisms often carried into run-time structures
  - Real separation requires careful engineering
  - Many mechanisms bind more than one attribute at a time or abstract from what we want to control. Examples?
- **The "art" of design includes finding structures that:**
  - Address multiple concerns concurrently and/or
  - Cleanly separate design from run-time constraints

End