QA Plan

- Create a plan for your project
- For non-code artifacts
  - Identify artifacts to be reviewed, by whom, and when
  - Define the goal of each review
  - Specify the review method: try *active review*
  - Record the results
- For code artifacts
  - Goals of testing
  - Testing strategies deployed
  - Test cases and results
- Use Cyclone plan as model (adapt to project!)
- Review in instructor meeting

Achieving System Qualities Through Software Architecture

What is “software architecture?”

Role in determining system qualities

Architectural views

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

Remember as: ***Components, Interfaces,*** *and* ***Relations***

Examples

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

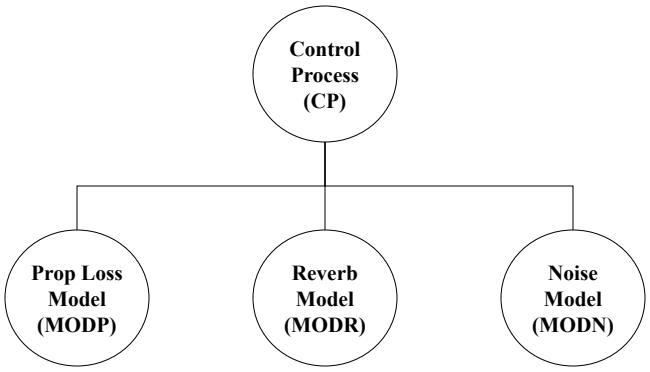
Structure	Components	Interfaces	Relationships
Calls Structure	Programs	Program interface and parameter declarations.	Invokes with parameters (A calls B)
Data Flow	Functional tasks	Data types or structures	Sends-data-to
Process	Sequential program (process, thread, task)	Scheduling and synchronization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

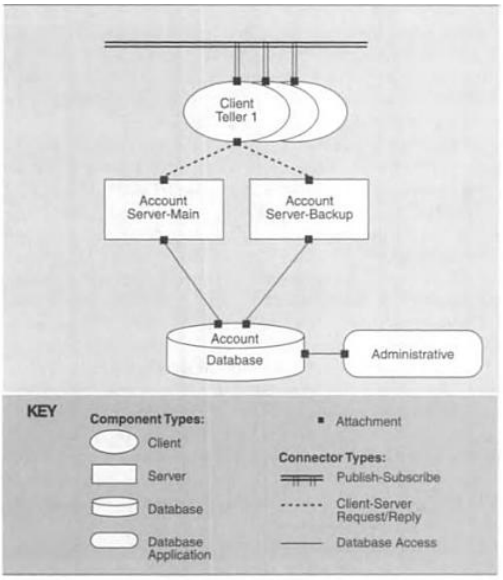


Typical (but uninformative) architectural diagram

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

Better

A bird's-eye-view of a system as it might appear during runtime. This system contains a shared repository that is accessed by servers and an administrative component. A set of client tellers can interact with the account repository servers and communicate among themselves through a publish-subscribe connector.

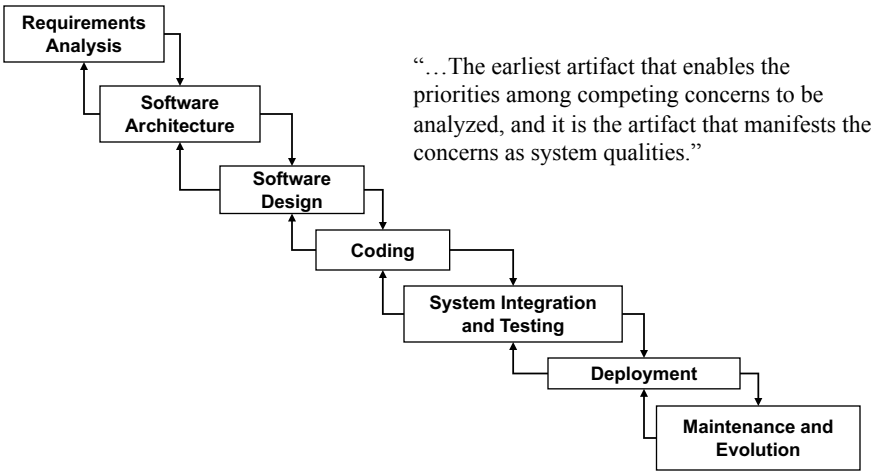


The Role of Architecture

Which system or development characteristics are determined by architecture?

What is the source of requirements?

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."

Effects of Architectural Decisions

- What kinds of system and development properties are and are not affected by architecture?
- System run-time properties
  - –
- System static properties
  - –
- Production properties? (effects on project)
  - –
- Business/Organizational properties?
- What is not affected?

Effects of Architectural Decisions

- What kinds of system and development properties are and are not affected by architecture?
- System run-time properties
  - Performance, Security, Availability
- System static properties
  - Modifiability, Portability, Reusability, Testability
- Production properties? (effects on project)
  - Concurrent development, Scheduling, Time-to-market
- Business/Organizational properties?
  - Lifespan, Versioning, Interoperability

Quality Requirements

Behavioral (observable)

- Performance
- Security
- Availability
- Reliability
- Fault-tolerance

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

Developmental Qualities

- Modifiability(ease of change)
- Portability
- Reusability
- Ease of integration
- Understandability
- Provide independent work assignments

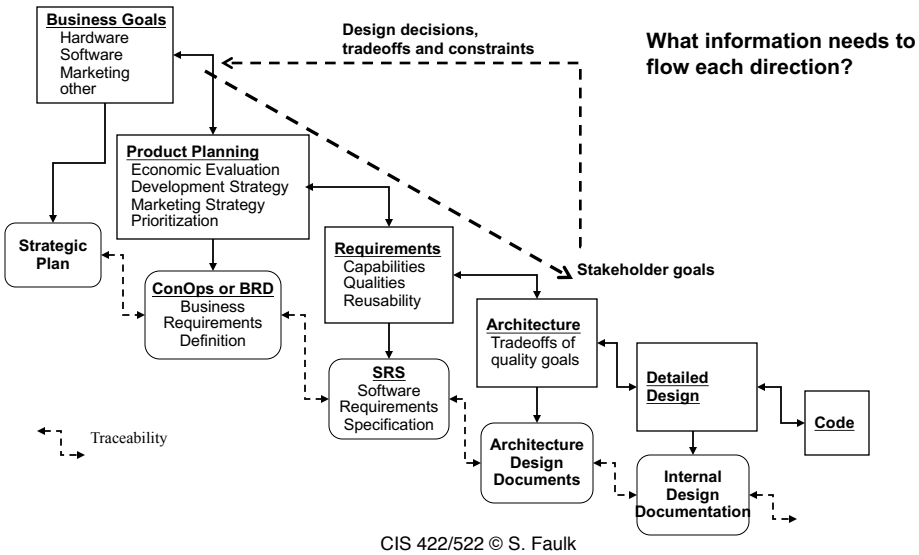
Properties resulting from the properties components, connectors and interfaces that exist at design time *whether or not they have any distinct run-time manifestation*.

Importance to Stakeholders

- Which stakeholders have a vested interest in the architectural design?
  - 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 may 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

Role of Architecture in Controlled Development

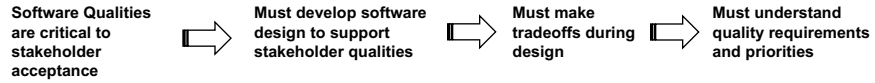
Product Development Cycle and Architecture



Engineering Software Architecture

- What are we trying to gain/maintain control of in the Architectural Design phase?
  - Profoundly effect system and business qualities
  - Requires making tradeoffs
- Control implies *achieving system qualities by choice not chance*
  - Understanding what the tradeoffs are
  - Understanding the consequences of each choice
  - Making appropriate choices at appropriate times
  - Choices made by people with appropriate skills and authority

Conceptual Flow



Intellectual Control Thread

Implications for the Development Process

Implies need to address architectural concerns throughout the development process:

- Understanding the “business case” for the system
- Understanding the quality requirements
- Designing the architecture to meet quality goals
- Representing and communicating the architecture
- Analyzing or evaluating the architecture
- Implementing the system based on the architecture
- Ensuring the implementation conforms to the architecture

Related Design Questions

- Create business case for the system
  - What is the “business” rationale or goal?
- Understanding the quality requirements
  - What are the design goals?
- Creating or selecting the architecture
  - What are appropriate components and relations?
  - What are the decomposition principles?
- Representing and communicating the architecture
  - How are the components and relations represented?
- Analyzing or evaluating the architecture
  - How do we decide if the architecture is any good?

Architectural Views

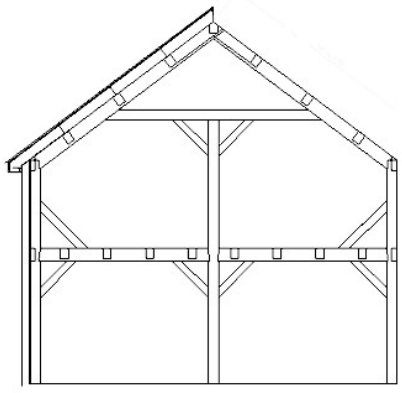
Which structures should we use?

Structure	Components	Interfaces	Relationships
Calls Structure	Programs (methods, services)	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

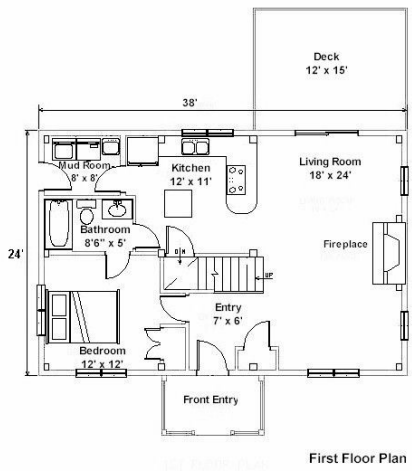
- Choice of structure depends the *specific design goals*
- Compare to architectural blueprints
  - Different blueprint for load-bearing structures, electrical, mechanical, plumbing

Elevation/Structural

East View

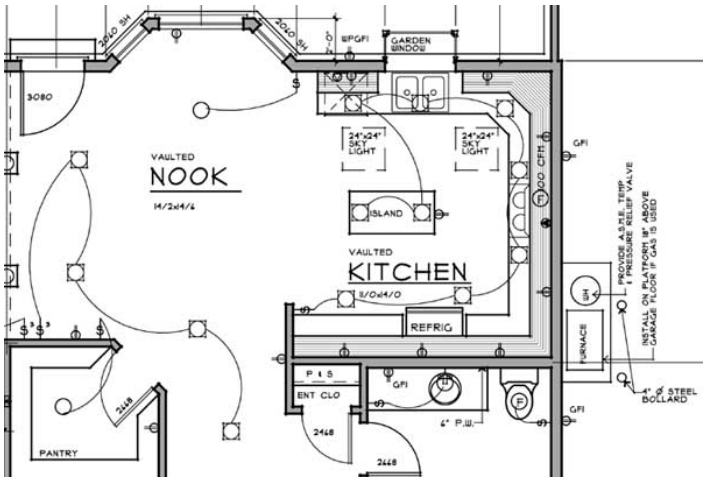


Floor Plan



First Floor Plan

Electrical Plan



Models/Views

- Each is a view of the same house
- Different views answer different kinds of questions
  - How many electrical outlets are available in the kitchen?
  - What happens if we put a window here?
- Designing for particular software qualities also requires the right architectural model or “view”
  - Any model can present only a subset of system structures and properties
  - Different models allows us to answer different kinds of questions about system properties
  - Need a model that makes the properties of interest and the consequences of design choices visible to the designer, e.g.
    - Process structure for run-time property like performance
    - Module structure for development property like maintainability

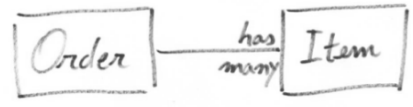


Figure 2: Conceptual Data Model – First Draft

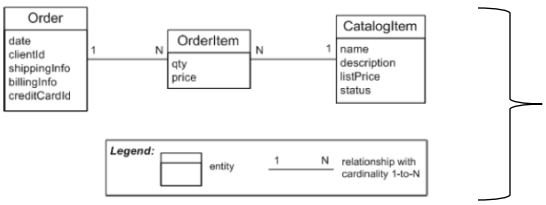


Figure 3: Logical Data Model

Example: Data Model View

- Data Model Architecture
  - Entities: data structures
  - Relations: cardinality, aggregation, generalization/specialization
  - Interface: attributes
- Model/communicate structure of complex data
  - What data is kept?
  - How is it related?
  - How is it structured and accessed in the system?

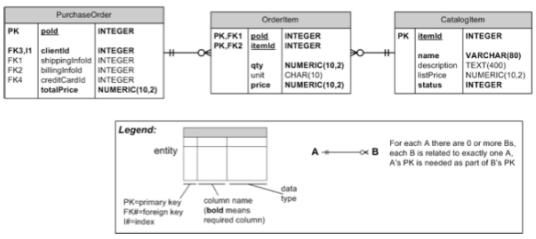


Figure 4: Physical Data Model

For Your Projects

Behavioral Qualities	Developmental Qualities
• Performance • Security • Availability • Reliability • Fault-tolerance	• Modifiability(ease of change) • Portability • Reusability • Ease of integration • Understandability • Independent work assignments • Subsetability (extend/contract)

Which qualities are of interest for your projects?
Which structures should you use?

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
- Realized at design time in different views

Questions?

Examples of Key Architectural Structures

- Module Structure
  - Decomposition of the system into work assignments or information hiding modules
  - Most influential design time structure
    - Modifiability, work assignments, maintainability, reusability, understandability, etc.
- Uses Structure
  - Determine which modules may use one another's services
  - Determines subsetability, ease of integration (e.g. for increments)
- Process Structure
  - Decomposition of the runtime code into threads of control
  - Determines potential concurrency, real-time behavior