Enterprise Architecture Principles

Business processes should drive the technical architecture

- Define dependencies and relationships among users and suborganizations of an organization
- Other stakeholders

- Message-driven approaches are desirable because they decouple system components
- Event-driven approaches are desirable because they help make a system responsive to events that are potentially visible and significant to users
Architecture Modules: Applications

Often most visible to users

- Application deployment
- Data modeling and integrity
- Business intelligence: decision support and analytics
  - Interoperation and cooperation
    - **Ontologies**: representations of domain knowledge
- Component and model repositories
- Business process management

© Munindar P. Singh, CSC 513. Spring 2010
Architecture Modules: Infrastructure

- Connectivity
- Platform: hardware and operating systems
- Storage
- System management
Architecture Modules: Systems

Functionality used by multiple applications

- Middleware: enabling interoperation, e.g., via messaging
- Identity management
- Security and audit
- Accessibility
- Policy repositories and engines
GOVERNANCE: RELATES ORG OF USER AND OTHER STAKEHOLDERS TO SYSTEM ARCHITECTURE
3/22/10

**STAKEHOLDERS IN A TYPICAL ENTERPRISE SETTING**

Specifically, NCSU's IT System

- UNIV ADMINISTRATOR
- FINANCE
- STUDENTS
- INVESTORS (?)
- RESEARCH SPONSORS
- IT SUPPORT
- SECURITY

PUBLIC SPONSORS

IT

- USERS
  - DEV
  - ADMIN

PUBLIC SYSTEM

- OPERATING
  - DAT
  - FIN
  - ST FCE
  - FACILITIES
  - SCHOOLS

STAKEHOLDERS OF A SYSTEM

- ALONG WITH AN INTEREST (STAKE)
- IN THE SYSTEM BEING CONSIDERED
- THE SYSTEM FOR WHICH THE SYSTEM IS CONSTRUCTED
- BUT IF IT IS BUILT TO SUPPORT BUSINESS PROCESS
Enterprise Functionalities: 1

It helps to separate the key classes of functionality in a working software system

- **Presentation**: user interaction
  - A large variety of concerns about device constraints and usage scenarios

- **Business logic**
  - Application logic
  - General rules
Enterprise Functionalities: 2

- Data management
  - Ensuring integrity, e.g., entity and referential integrity (richer than storage-level integrity)
  - Enabling access under various kinds of problems, e.g., network partitions
  - Supporting recovery, e.g., application, operating system, or hardware failures
Enterprise Functionalities: 3

Bases for choosing the above three-way partitioning as opposed to some other

- Size of implementations
- Organizational structure: who owns what and who needs what
- Staff skill sets
  - User Interface: usability and design
  - Programming
  - Database
  - Policy tools
- Products available in the marketplace
One-Tier and Two-Tier Architectures

- One tier: monolithic systems; intertwined in the code base
  - Historically the first
  - Common in legacy systems
  - Difficult to maintain and scale up
- Two-tier: separate data from presentation and business logic
  - Classical client-server (or fat client) approaches
  - Mix presentation with business rules
  - Change management

Munindar P. Singh, CSC 513, Spring 2010
Three-Tier Architecture: 1

- Presentation tier or frontend
  - Provides a view to user and takes inputs
  - Invokes the same business logic regardless of interface modalities: voice, Web, small screen, ...

- Business logic tier or middle tier
  - Specifies application logic
  - Specifies business rules
    - Application-level policies
    - Inspectable
    - Modifiable
Three-Tier Architecture: 2

- **Data** tier or backend
  - Stores and provides access to data
  - Protects integrity of data via concurrency control and recovery
Multitier Architecture

Also known as n-tier (sometimes treated synonymously with three-tier)

- Best understood as a componentized version of three-tier architecture where
  - Functionality is assembled from parts, which may themselves be assembled
  - Supports greater reuse and enables greater dynamism
  - But only if the semantics is characterized properly

- Famous subclass: service-oriented architecture