Tiers

One
Two
Three (multi)

Pros & Cons of Each

Performance
- Scalability
- Availability
- Cost

Maintenance
- Overhead
- Due to replication
- Supports

Well-defined functionality for each tier
- Why is it an advantage?
- Easier management?
- Maintainability

Latency increases for more tiers?
- Throughput
- Depends on an implementation arch (app server, DBMS, replication)
- Enabled by these tiers, not required by those tiers arch

© North Carolina State University. All Rights Reserved
IT MANAGEMENT ASPECTS

- HARDWARE
  - INSTALLATION
  - MAINTENANCE
  - UPDATE

SOFTWARE

- FEWER SOFTWARE VERSIONS

TWO: INCOMPATIBILITIES

THREE:
**Development Case (Design)**

- Maintainability
- Reuse

**Performance**

- Latency
- Throughput
- Scalability
- Availability

**Administration**

- Installation
- Maintenance
- Upgrades

**IT Staff (Operations)**

- IT Staff
- IT Staff
- Users

**IT Operations Staff**

- Users

**User Advocate**

- Finance (CFO)
- Negotiation
- Marketing
- IT Staff
Functional versus Nonfunctional (Concerns or Requirements)

Deeply domain-dependent e.g., airbag
  stock trading

Data type

Queue
  [CREATE]
  [ENQUE]
  [DEQUE]
  Failure to do these => NOT enqueue
  Functional

REST are non-functional

Software arch (in particular)
  - All of them must support func. requirement
  - Distinctions among them are OOP-based only on nonfunc. reqs