# Software Architecture

# Quality Attributes

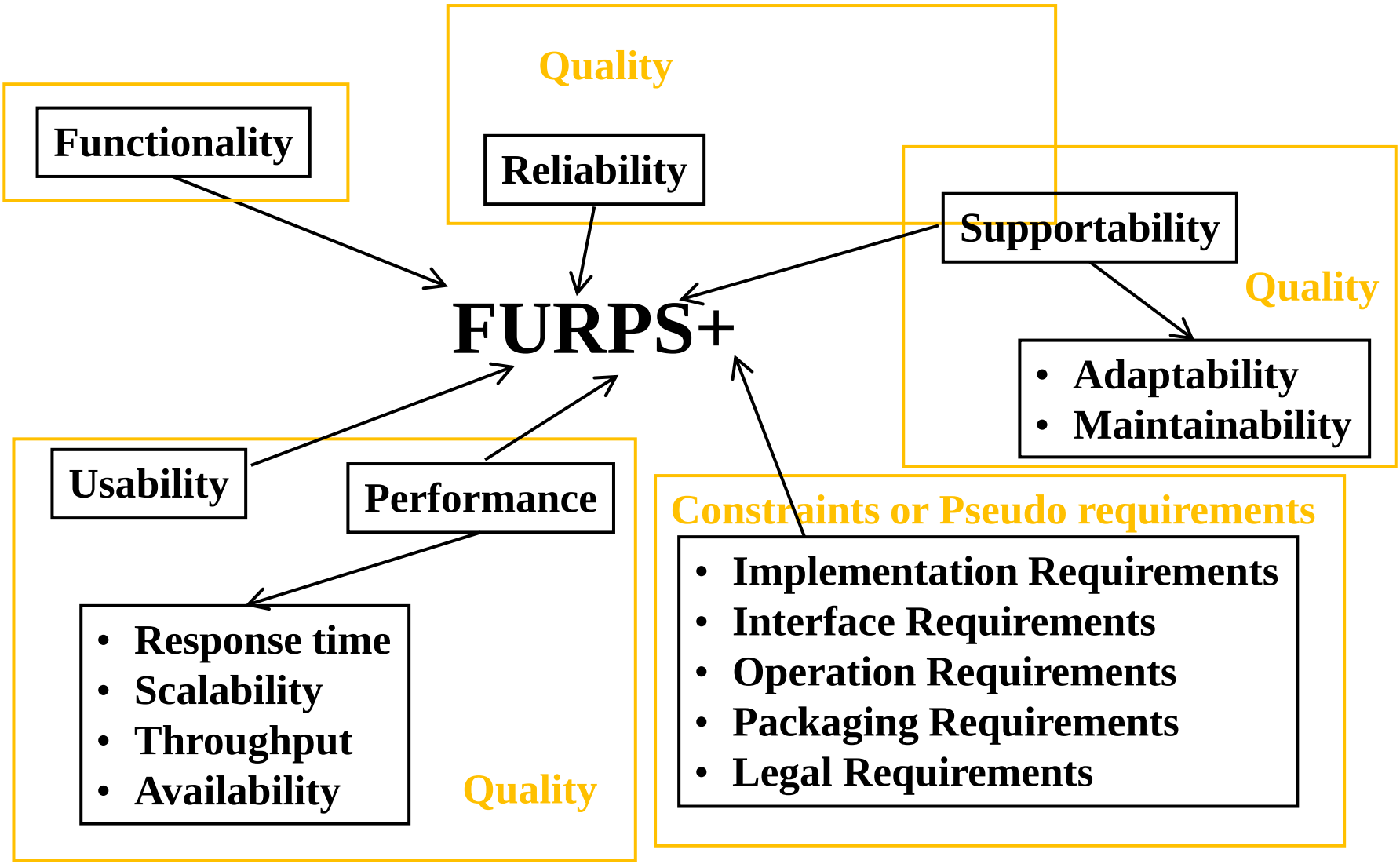
Soha Makady

s.makady@fci-cu.edu.eg

# Types of Requirements

- Functional requirements
  - Describe the interactions between the system and its environment independent from the implementation
    - "An operator must be able to define a new game. "
- Nonfunctional requirements
  - Aspects not directly related to functional behavior.
  - Capture many facets of how the functional requirements are achieved.
    - "The response time must be less than 1 second"
- Constraints
  - Imposed by the client or the environment
    - "The implementation language must be Java "
  - Called "Pseudo requirements" in the text book.

# The FURPS+ model



# Functional vs. Nonfunctional Requirements

## Functional Requirements

- Describe user tasks that the system needs to support
- Phrased as actions
  - "Advertise a new league"
  - "Schedule tournament"
  - "Notify an interest group"

## Nonfunctional Requirements

- Describe properties of the system or the domain
- Phrased as constraints or negative assertions
  - "All user inputs should be acknowledged within 1 second"
  - "A system crash should not result in data loss".

# Types of Nonfunctional Requirements

- Performance: A performance quality attribute defines a metric that states the amount of work an application must perform in a given time, and/or a deadline that must be met.
- Performance requirements are particularly fatal in real-time applications (like what?)
- Performance includes:
  - Response time
  - Scalability
  - Throughput
  - Availability

# Types of Nonfunctional Requirements

- Performance includes:
  - Throughput:
    - A measure of the amount of work the system must perform in unit time.
    - Work could be measured in transactions per second (tps), or messages processed per second (mps)
    - "An online banking application needs to guarantee that it can execute 1000 tps from Internet banking customers"
  - Average throughput requirement vs. peak throughput requirement? (Horse racing example)

# Types of Nonfunctional Requirements

- Performance includes:
  - Availability: the degree to which a system or component is operational and accessible when required for use

# Types of Nonfunctional Requirements

- Performance includes:
  - Response time: a measure of the latency an application exhibits in processing a business transaction.
    - Response time is most often (but not exclusively) associated with the time an application takes to respond to some input.
    - Guaranteed versus average response time?
    - “95% of all requests must be processed in less than 4 s, and no requests must take more than 15s.”

# Types of Nonfunctional Requirements

- Performance includes
  - Scalability: Scalability is a non-functional property of a system that describes the ability to appropriately handle increasing (and decreasing) workloads.
    - "How well a solution to some problem will work when the size of the problem increases"

# Types of Nonfunctional Requirements

- Performance includes Scalability:
- What is expected to get bigger?
  - Request load: Based on some defined mix of requests on a given hardware platform, an architecture for a server application may be designed to support 100 tps at peak load, with an average 1 s response time.
  - If this request load were to grow by ten times, can the architecture support this increased load?
  -

# Types of Nonfunctional Requirements

- Performance includes Scalability:
- What is expected to get bigger?
  - Request load: Without any additional hardware capacity:
    - Application throughput should remain constant (i.e., 100 tps)
    - Response time per request should increase only linearly (i.e., 10 s).
    - What could be a scalable solution?
      - Scale up versus scale out?

# Request load – Scale up vs.

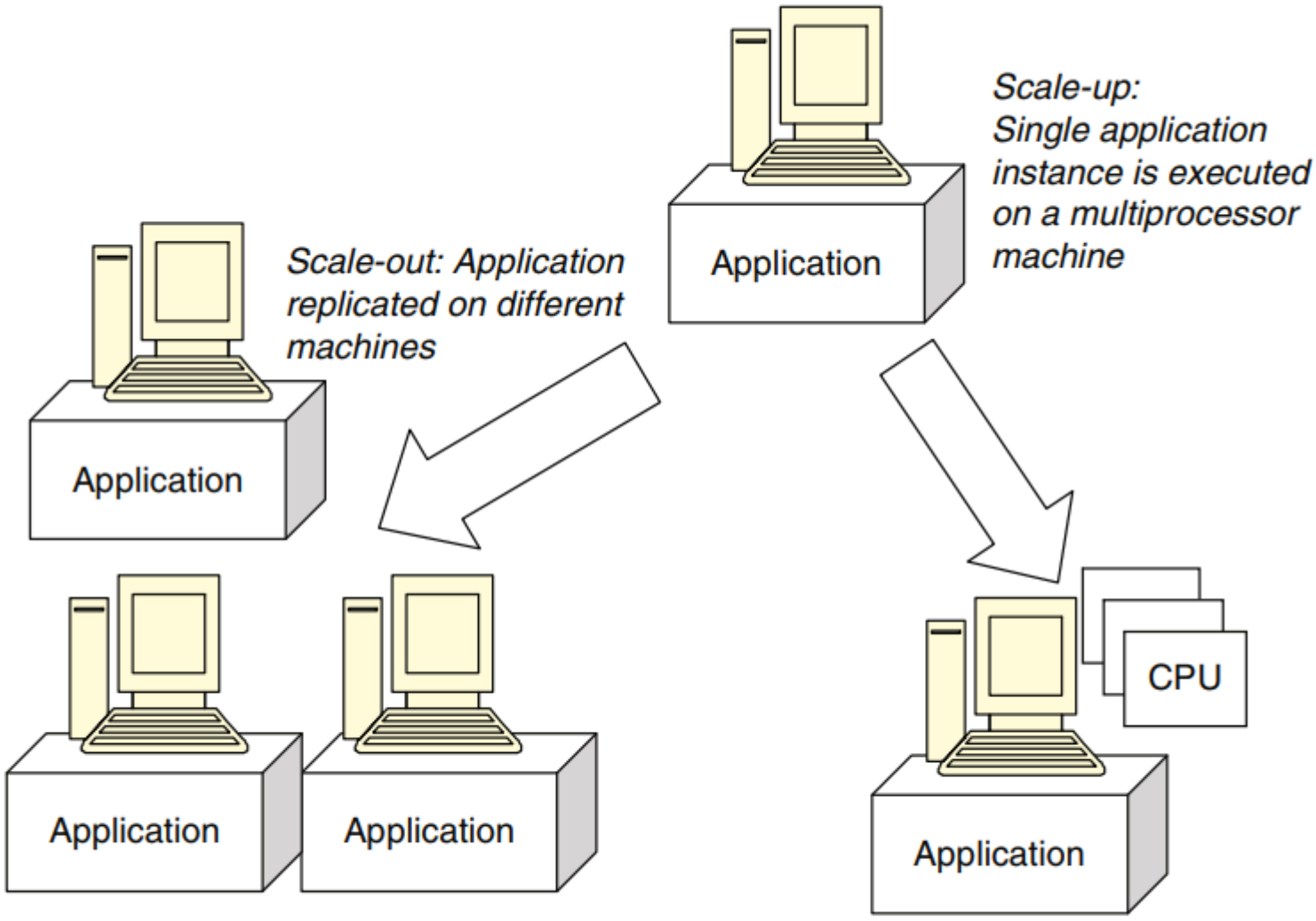



**Fig. 3.1** Scale out versus scale up

# Request load – Scale up vs. Scale out

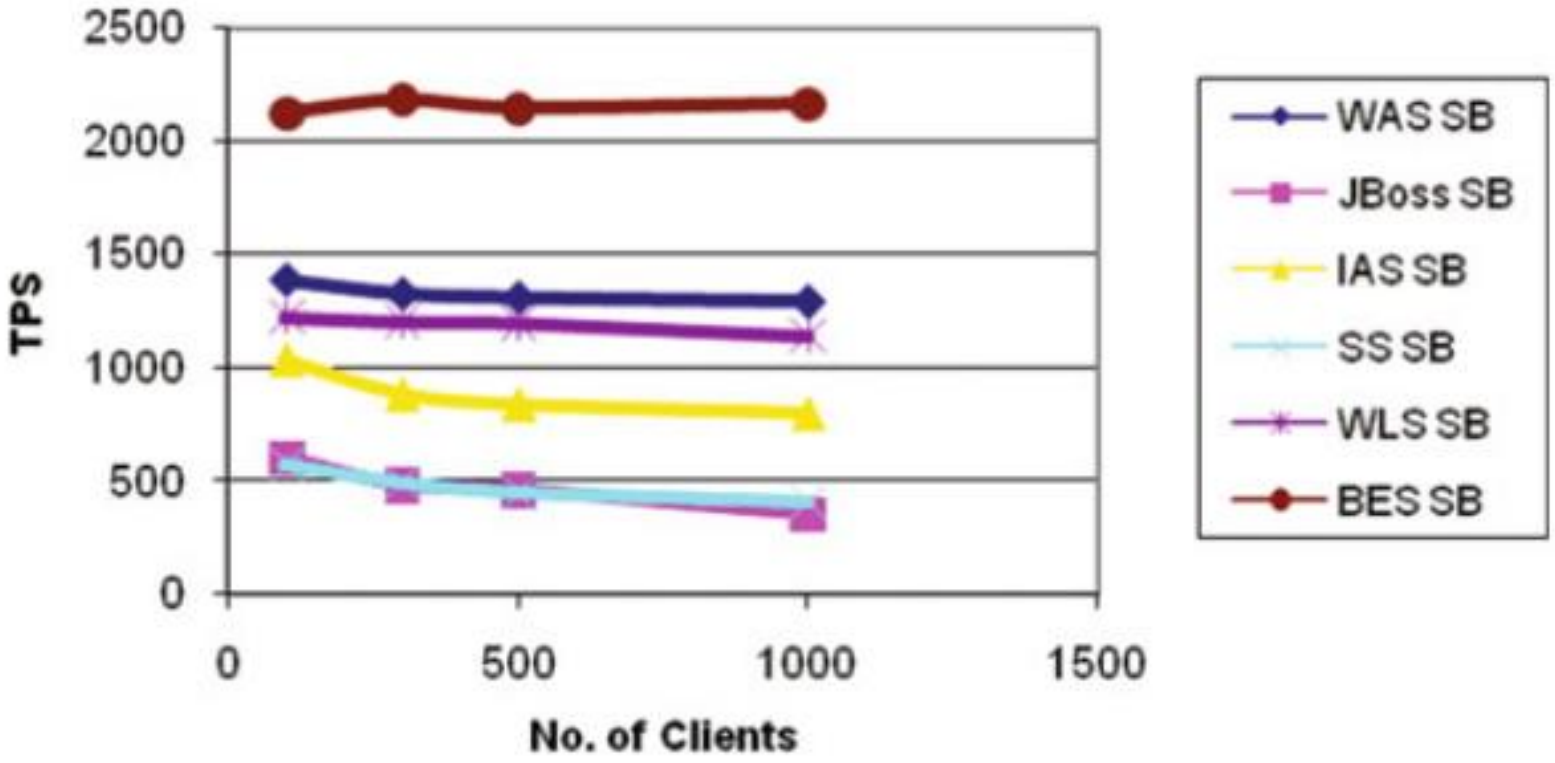



**Fig. 3.2** Effects of increasing client request load on JEE platforms

# Types of Nonfunctional Requirements

- Performance includes Scalability:
- What is expected to get bigger?
  - Simultaneous connection: An architecture may be designed to support 1000 concurrent users. How does the architecture respond if the number of users grows a lot?

# Types of Nonfunctional Requirements

- Performance includes Scalability:
- What is expected to get bigger?
  - Data size: An application processes messages of an average 56K size. How well will the architecture scale if the message size became 10 MB
  - Deployment: How does the effort involved in deploying or modifying an application to an increasing user base grow? (CD versus online installation? Which one scales better?)

# Types of Nonfunctional Requirements

- Usability: is the ease with which a user can learn to operate, prepare inputs for, and interpret outputs of a system or component.
- Reliability
  - Failures in applications cause them to be unavailable.
  - Failures impact on an application's reliability
  - Reliability vs. Availability?
  - **Percentage of availability = (total elapsed time – sum of downtime)/total elapsed time**
  - **MTBF = (total elapsed time – sum of downtime)/number of failures**
-

# Other Non-functional Requirements

- Security: At the architectural level, security boils down to understanding the precise security requirements for an application, and devising mechanisms to support them. These include:
  - Authentication
  - Authorization
  - Encryption
  - Integrity
  - Non-repudiation

# Other Non-functional Requirements

- Authentication: Applications can verify the identity of their users and other applications with which they communicate
- Authorization: Authenticated users and applications have defined access rights to the resources of the system.
- Encryption: The messages sent to/from the application are encrypted.
- Integrity: This ensures the contents of a message are not altered in transit.
- Nonrepudiation: The sender of a message has proof of delivery and the receiver is assured of the sender's identity.

# Required Readings

- Chapter 3 from “Essential Software Architecture” textbook of Ian Gorton, 2011 Edition.