Software Architecture Design Structure Quality Attributes

WEEK 1-1

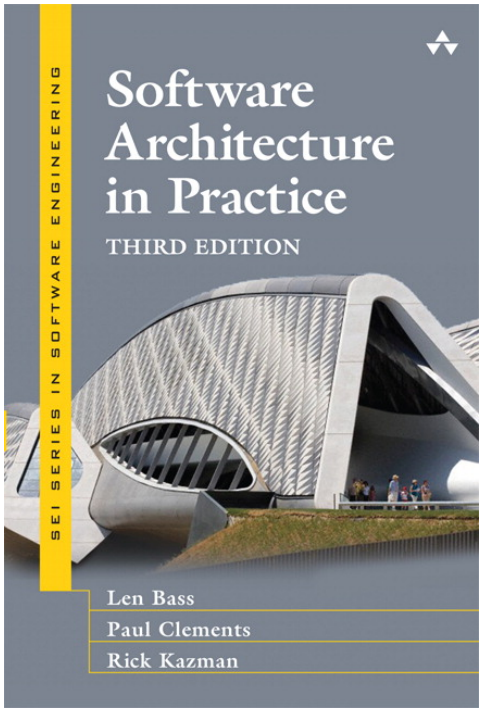
About the Course

Course management using Moodle

https://moodle.rose-hulman.edu/course/view.php?id=29772

Textbook

Len Bass, Paul Clements, Rick Kazman, Software Architecture in Practice, 3rd Edition, Addison-Wesley, ISBN 978-0321815736.



Please read the Syllabus, if you haven't already done so.

Today

Software Architecture

Design structures

Quality Attributes

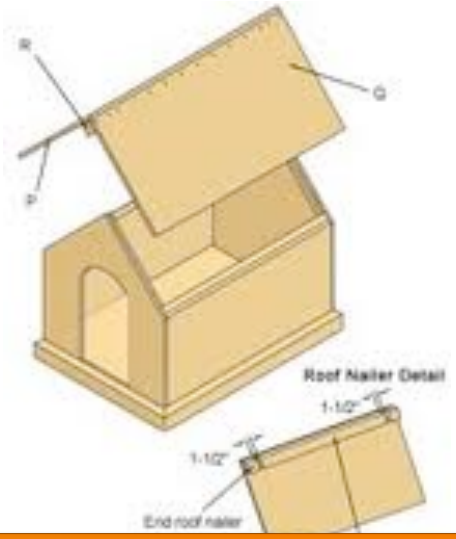
An Example of Architecture

The Louvre pyramid, built in 1989, designed by architect I.M. Pei.
http://en.wikipedia.org/wiki/Louvre_Pyramid

Why is Architecture Important?

Architects are Important ☺

Building a Dog House



Can be built by one person and requires:

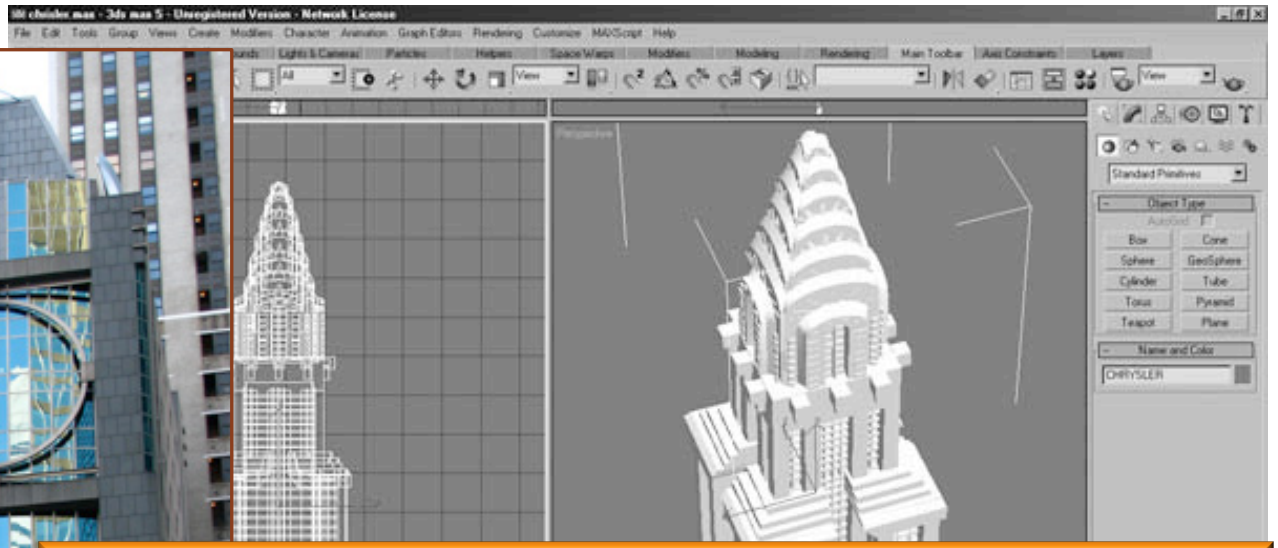
- Minimal modeling
- Simple plan
- Simple tools
- Risk/Constraints (Low)

Building a Family House

Built efficiently by a team and requires:

- Design/Modeling
- Organized process/plan
- Power tools
- Risks/Constraints (med.)

How About a Skyscraper?



Architectural feats built by teams of teams and requires:

- Systems of Systems Design/Modeling
- Well-defined, mature process/plan
- Advanced design & construction tools
- Risks/Constraints (Enormous!)

So, Who is Software Architect?

One who

provides technical leadership

makes high-level design choices

establishes coding standards, tools, and platforms

Software Architecture

What is Software Architecture?

- Set of structures needed to reason about the system
- Is made up of three entities
  - Software elements
  - Relation among them
  - Properties of both elements and relations

Types of Structure

Modules

Component and Connector

Allocation Structure

Modules

Modules are static structure such as: implementation units for project teams, business logic, and user interfaces

Types of Module Structure

Decomposition structure : is a sub-module of relation

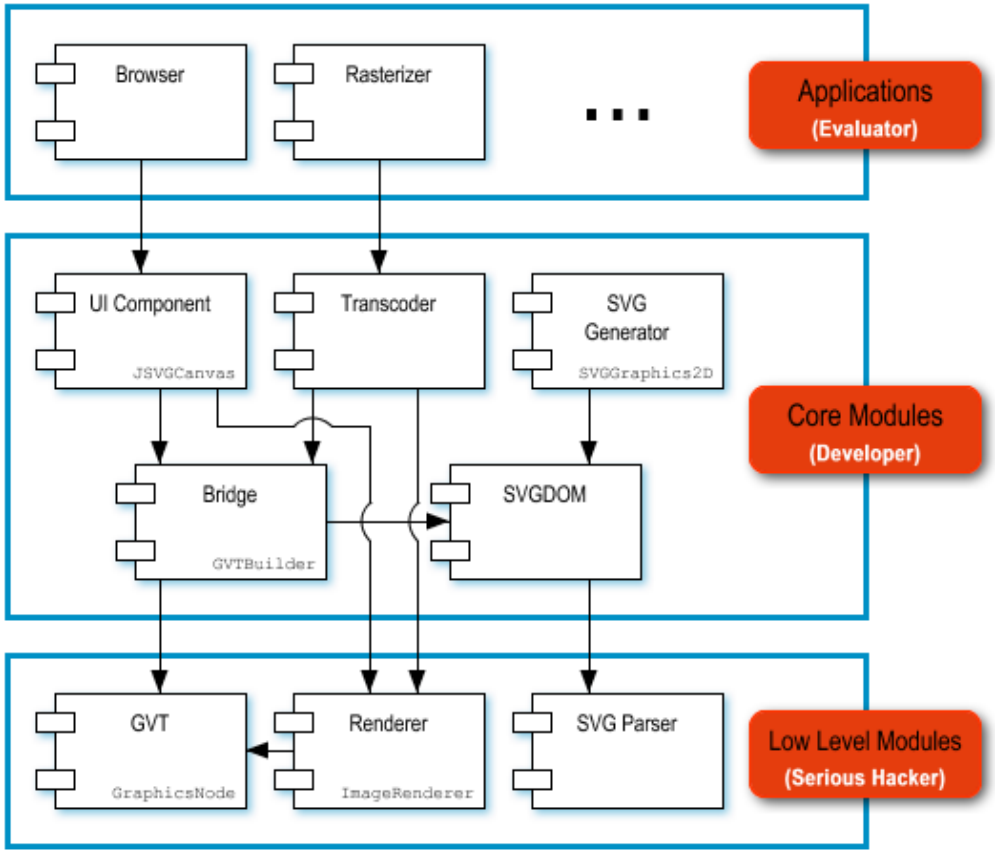
Uses structure: dependency relation

Layer structure: abstraction for portability

Class structure: is an instance of relation

Data model: Domain model, entity relation

An Example …



The Apache Batik Project

http://xmlgraphics.apache.org/batik/using/architecture.html

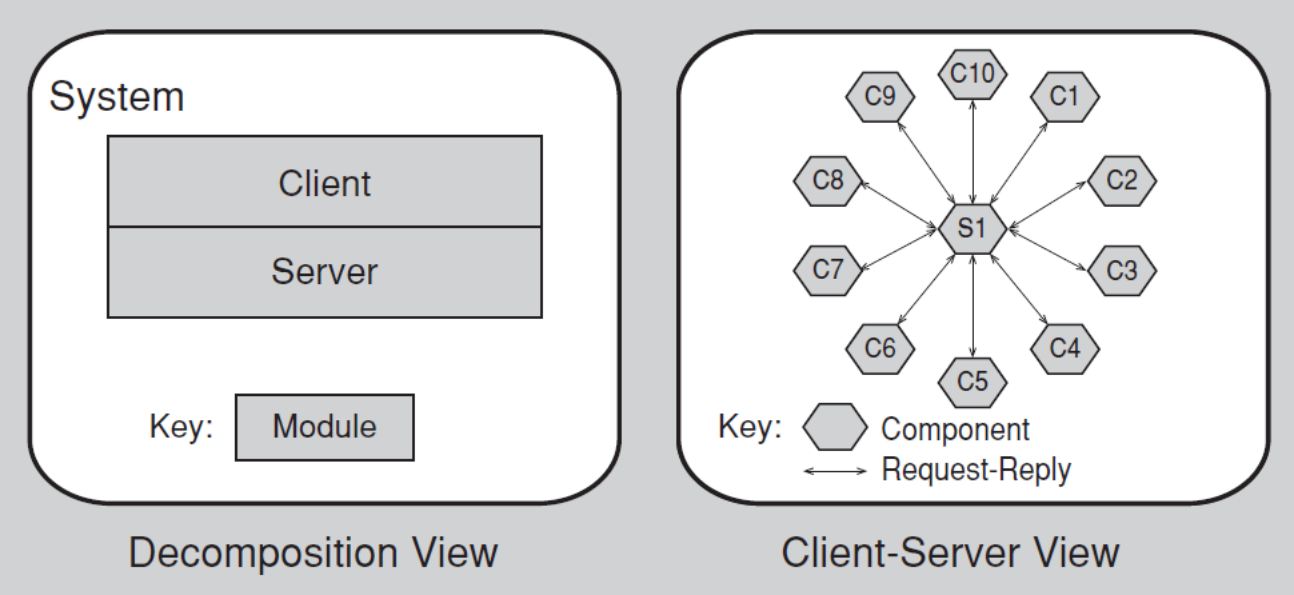
Component and Connector

Interaction of elements at runtime

Structures are the same as module structure but they deal with dynamic aspects of the system

E.g. Client-Server, Peer-to-Peer, and Pipe and Filters

Relating One Structure to Another



Allocation Structure

Mapping of Module or C&C structures into entities that are not software

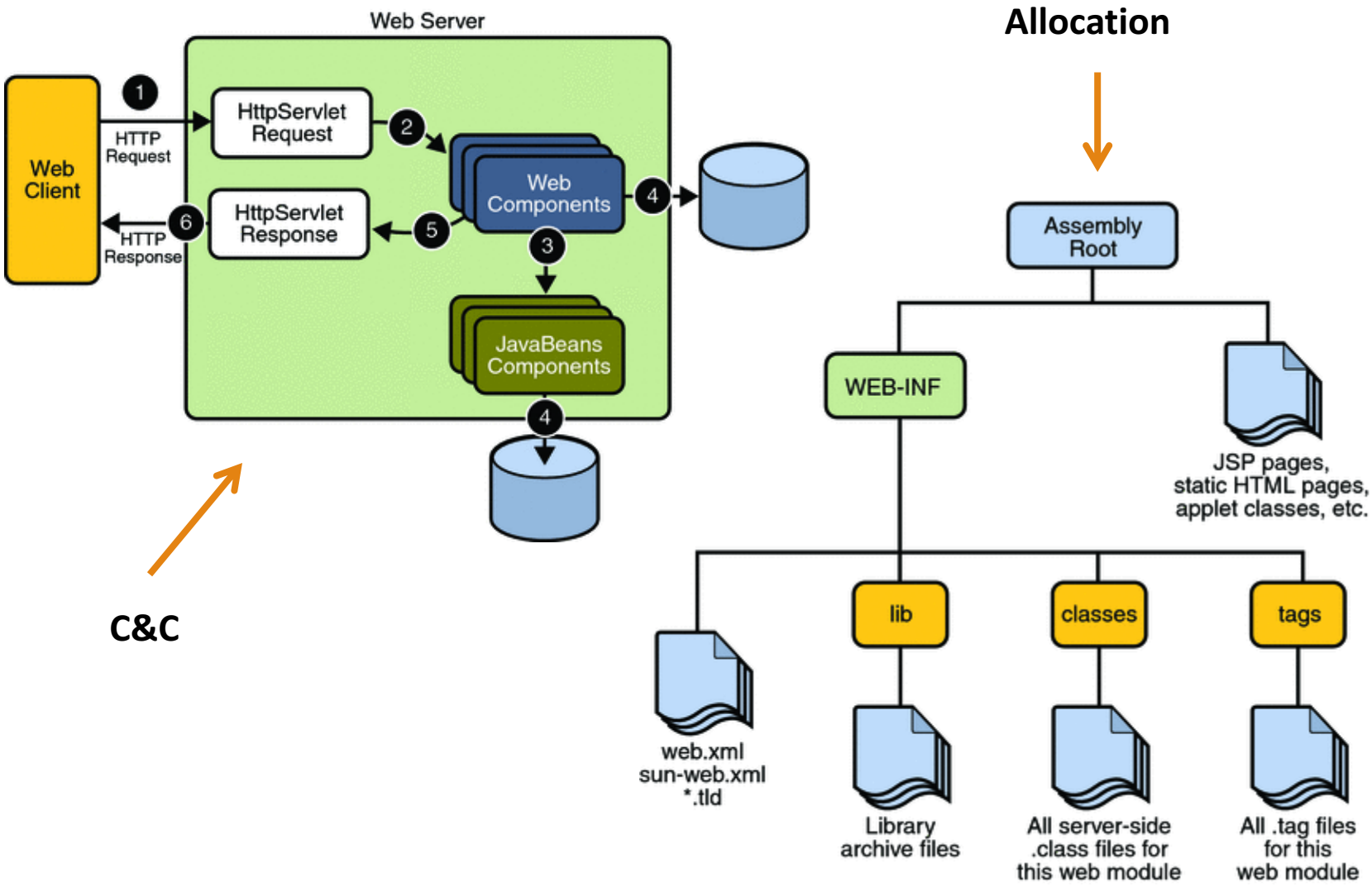
Types of Allocation Structure

Deployment structure: allocated-to and migrates-to relations

Implementation structure: software element to file structures

Work assignment structure: Responsibility for implementation

Example …


Web Server
Web Client
1
HTTP Request
HttpServlet Request
2
Web Components
4
3
JavaBeans Components
4
6
HTTP Response
HttpServlet Response
5
C&C
Allocation
Assembly Root
WEB-INF
JSP pages, static HTML pages, applet classes, etc.
lib
classes
tags
web.xml
sun-web.xml
*.tld
Library archive files
All server-side .class files for this web module
All .tag files for this web module

Are All Structures Necessary?

Fewer is better!

Choose something that would

- Help communicate core ideas of the system
- Help reason about important quality attributes of the system

Quality Attribute (QA)

"QA is a measurable or **testable property** of a system that is used to indicate how well the system satisfies the **needs of a product** along some dimension of **interest to a stake holder.**"

-Bass et al.

Specifying Quality Attributes

A quality attribute requirement should be **testable**

QA can be expressed using scenarios

Scenarios have Six Parts

Source of stimulus

Stimulus

Environment

Artifact

Response

Response Measure

Parts of Scenarios 1/3

Source of stimulus

- Entity that generates stimulus
- e.g. a human, a computer system, an actuator, etc.

Stimulus

- A condition that requires a response
- e.g. user operation, an attack, a file transfer complete event, etc.

Parts of Scenarios 2/3

Environment

- Specifies the mode in which the system is executing
- e.g. normal operation, overload condition, system freeze, etc.

Artifact

- The portion of the system to which the requirement applies
- e.g. Data store, UI, middleware, etc.

Parts of Scenarios 3/3

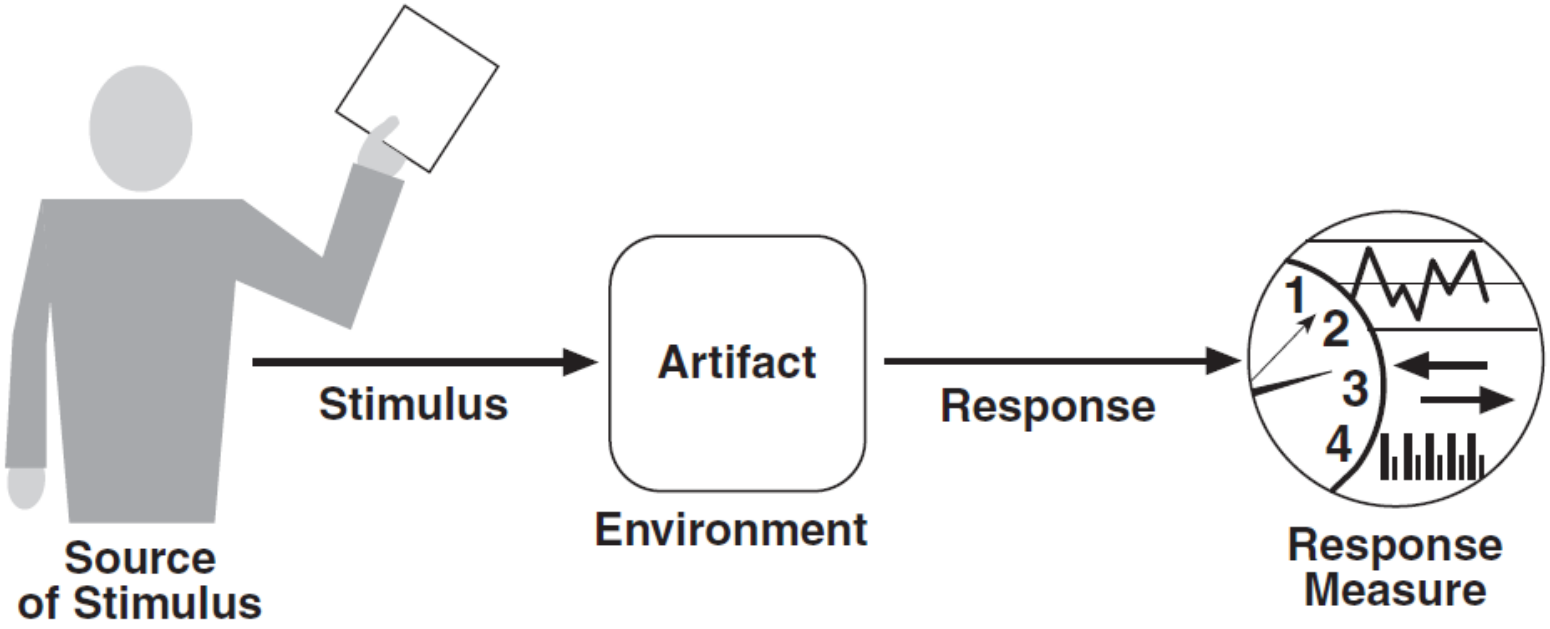
Response

- Responsibilities that the system (for runtime qualities) or the developers (for development-time qualities) should perform in response to the stimulus
- e.g. event handling, feature implementation, etc.

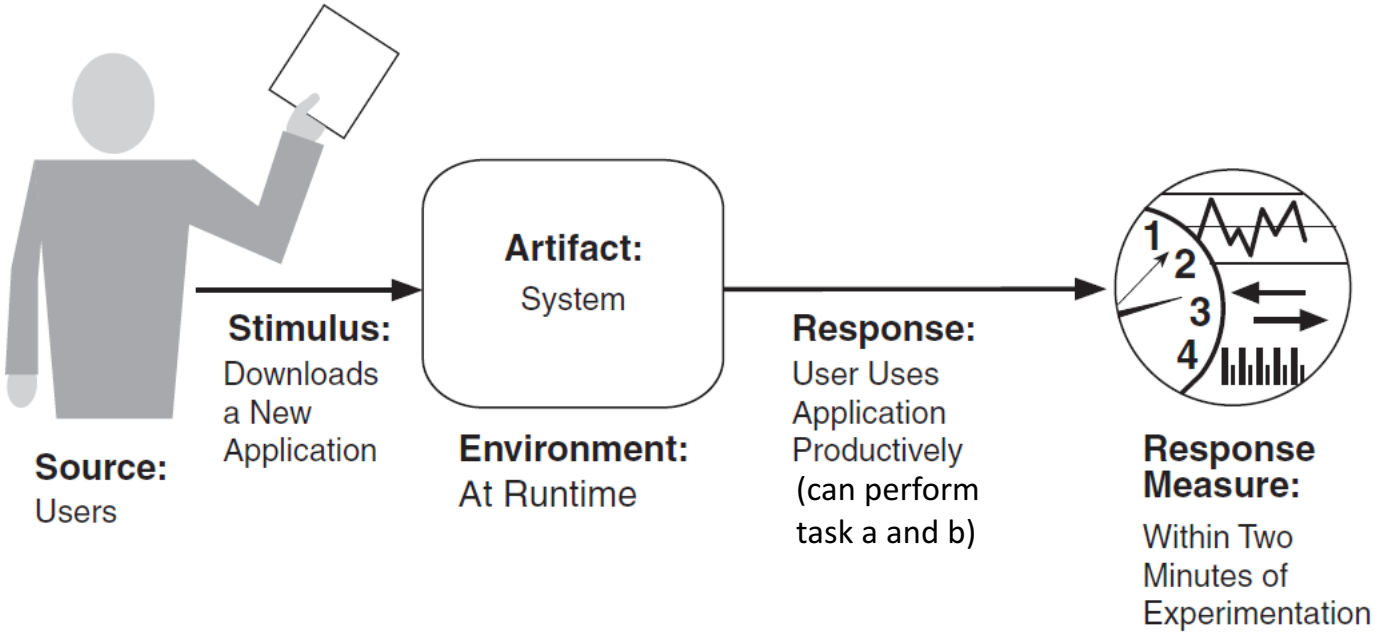
Response measure

- Determines whether a response is satisfactory or whether a requirement is satisfied
- e.g. latency, jitter, throughput, labor or clock time to make change, time to test, time to deploy modification

Summing Up Scenario



A Concrete Usability Scenario



"User after downloading a new application in their system should be able to use the application productively within two minutes of usage."

Next …

Things Due

- Paper Review 1 – Due 8:00 am, Thursday

Concepts

- Modifiability (Read Ch 7)
- The Layered Pattern (Ch 13 pp. 205-210)