# SWE 530:
# Software Design Process
## Software Design Process

Dr. H. Birkan YILMAZ
Department of Computer Engineering
Boğaziçi University

(birkan.yilmaz@bogazici.edu.tr )

Adapted from slides of Dr. Albert Ali Salah

# Recap

- The design process is concerned with describing ***how*** a requirement is to be met by the design product;
- Design ***representation*** forms provide means of modelling ideas about a design, and also of presenting the design plans to the programmer;
- ***Abstraction*** is used in problem-solving, and is used to help separate the *logical* and *physical* aspects of the design process;

# Overview

- What is software?
- Building models
- Transferring design knowledge
- Constraints upon the design process and product
- Recording design decisions
- Designing with others

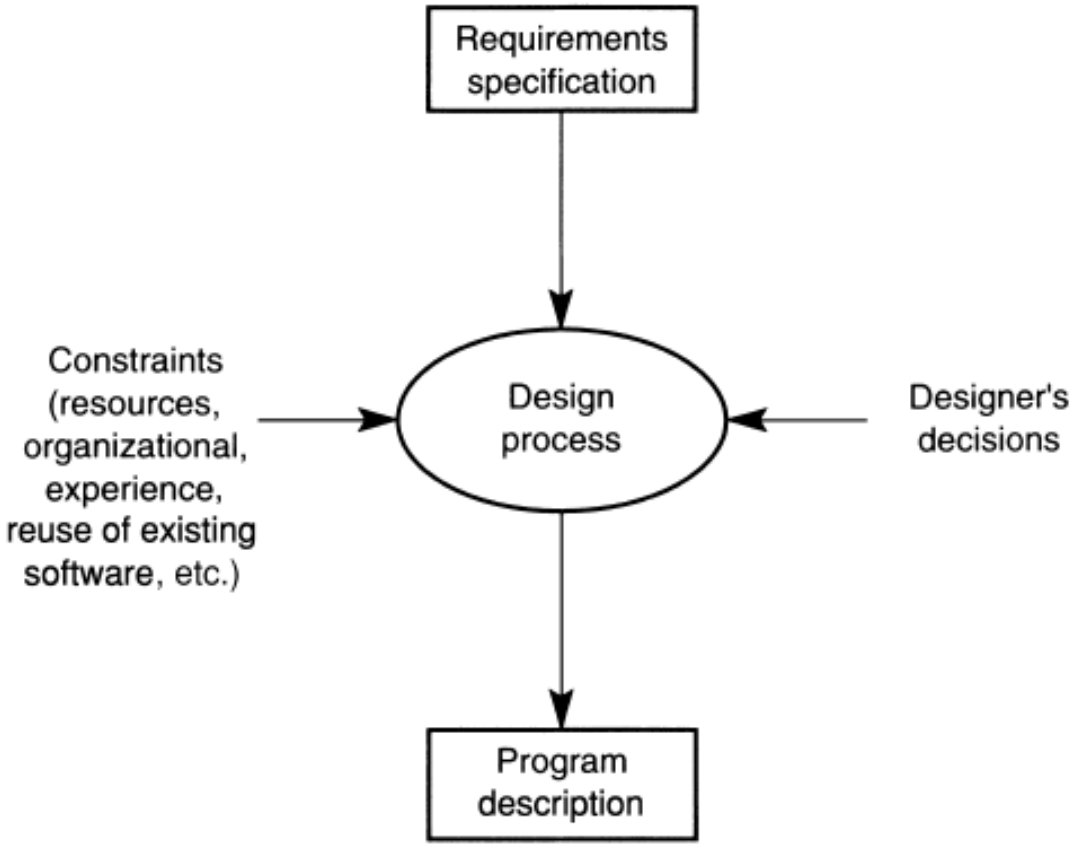
# What is software?

- Not just some binary code!
- Includes distribution of code, e.g. Client-server applications
- Includes scripting forms (HTML, XML)
- Different architectures (e.g. mobile platforms, Web, stand-alone app.)
- The move to higher-level languages

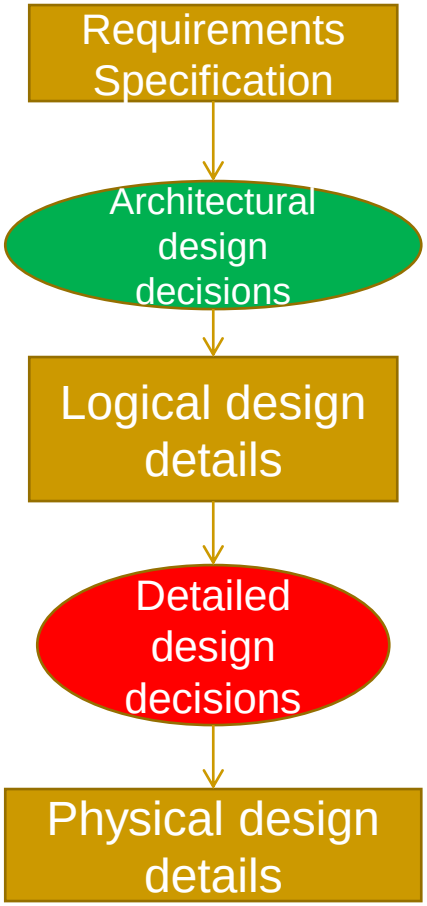
# The software design process

- Designer formulates and develops an **abstract design model representative** of the **solution**
- Why is this process not understood as well as other forms of design?
  - The complexity of software
  - The problem of conformity
  - The (apparent) ease of changeability
  - The invisibility of software

# General model of software design process



# Phases of software design process



# Practice of design

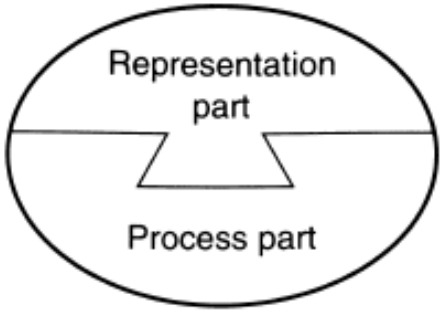
- The **use of abstract 'mental models'** by the designer to simulate the dynamic behaviour.
- **Expanding** the **detail of a model in a systematic manner** by keeping all elements of the design at the same level of detail.
- The need to **make** any **constraints** affecting the design as **explicit** as possible when handling an unfamiliar problem.
- **Reuse** of previous design plans.
- Making notes about **future** (detailed) intentions.

# Transferring design knowledge

- Codifying and exchanging experiences about the processes involved in design and resulting design features that have proved effectively gaining design knowledge.
- The characteristics of an **exceptional designer**:
  - Familiarity with the application domain
  - Skill in communicating technical vision to other project members.
  - Identification with project performance

# Design methods

- Design methods (70s)
  - The process part
  - The representation part
  - Heuristics
- Design patterns (90s)
- Design methods help in:
  - Identification of design actions
  - Use of representation forms
  - Procedures for making transformations between representations
  - Verification and validation operations
  - Quality measures
  - Identification of certain constraints



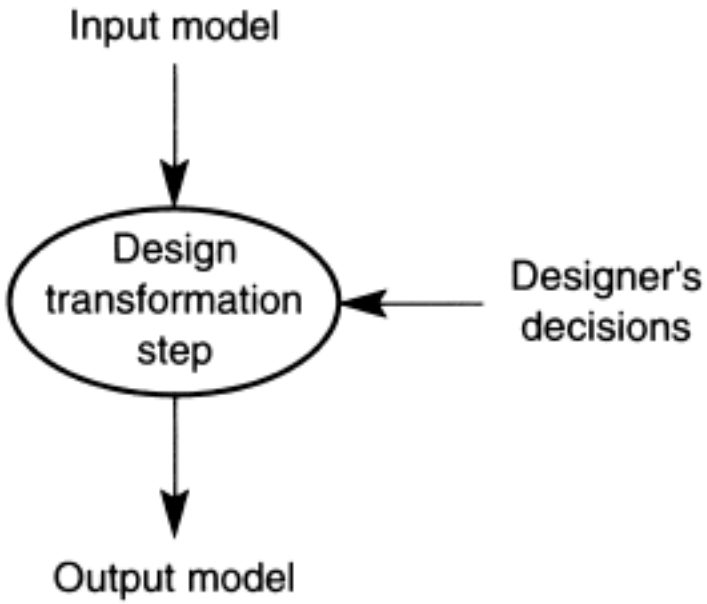

'How to describe a designer's ideas'

'What to do in order to produce a design'

# Classic design conditions

- According to Akin (1990), there are three classic conditions observed in creative acts of design:
  - The **recognition step**: Aha! moment, recognition of **a solution**.
  - The problem **restructuring step**: change of viewpoint allows major breakthrough
  - The **development of procedural knowledge**: generalization to similar problems, gaining of expertise

# Transformation model of design activity




- two principal forms of design transformation:
  - the ***refinement*** (or *elaboration*) of structures, in which the input and output forms of the model are the same, but extra detail is added;
  - the ***transformation of viewpoint***, which may involve a change of representation form or introduce a different interpretation of the representation form.

# Design constraints

- Designing software is rarely an unconstrained process
- Examples of constraints
  - Programming language to be used (!)
  - Execution environment or Operating System
  - Performance expectations
  - User interface needs

# Recording design decisions

- The need to record the design decisions from the viewpoint of **design** and **maintenance** tasks
- The design and maintenance can be extended and modified by making use of design decisions
- **Quality control** is the main intension to record the design decisions
- Benefits to the **new members** of design and maintenance teams
- Generally design decisions are represented through the **diagrams** or **other notation**

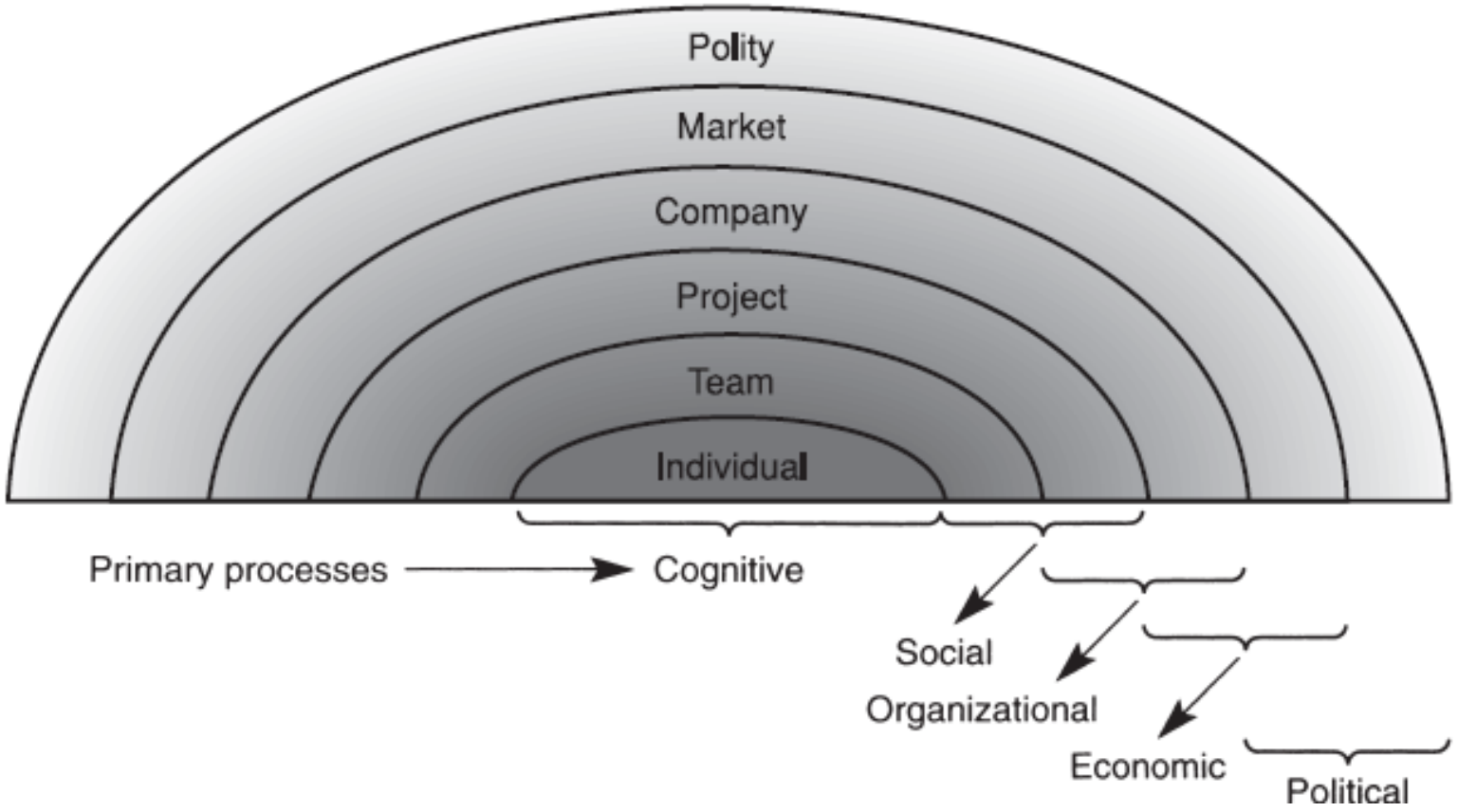
# Designing with others

- Major issues:
  - **how to split** the design task among the team, and to determine the **interfaces** between the parts;
  - **how to integrate** the individual contributions to the design, which may well involve a process of negotiation between the members of the team.

# Designing with others

- Factors:
  - the **size** of a team (there seem to be pointers that a size of 10–12 members is probably an upper limit for productive working);
  - the large impact that may be exerted by a small subset of the **members** of the team who possess superior **application domain knowledge**;
  - the influence of organizational issues within a company (and particularly the need to maintain a bridge between the developers and the customer).

# Factors influencing software design process



# Summary

- The complexity of the model-building processes for software systems, with their need to consider static forms as well as the dynamic behaviour of the eventual system;
- The influence of the invisible nature of software upon any attempts to describe it;
- The need for domain knowledge on the part of the designer;
- How the observed practices of software designers relate to the model of the general design process and the use of opportunistic design practices by designers;

# Summary - continued

- The general form of a design method, and its three major components: the representation part, the process part, and the heuristics;
- How to go about recording the results of the design process, presenting an ideal view of design development by ‘faking’ an ideal development process;
- Some of the factors that affect the operation of design teams, and how this differs from individual design practices.

# Questions?