NPTEL Course on

Human Computer Interaction
- An Introduction

Dr. Pradeep Yammiyavar
Professor,
Dept. of Design,
IIT Guwahati,
Assam, India

Dr. Samit Bhattacharya
Assistant Professor,
Dept. of Computer Science
and Engineering,
IIT Guwahati, Assam, India

Indian Institute of Technology Guwahati

Module 2:
Interactive System Design

Lecture 2:
HCI and Software Engineering

Dr. Samit Bhattacharya

Objective

- In the previous lecture, we have learned about the idea of “usability”
  - While designing an interactive system, what we should do to take care of usability?
- In this lecture, we shall learn about the answer to the above question

Objective

- In particular, we shall learn about the following
  - The difference between a software design and an interactive system design
  - User-centered and participatory design
  - The interactive system design life cycle

The Central Idea

- Suppose you are designing a database management system (DBMS): what are your design objectives
  - Efficient storage of large databases (storage)
  - Efficient way to retrieve results of a query from database (retrieval)
  - Allowing the user to access the database (interaction)

The Central Idea

- Note that this is a scenario where the user interacts with the system (database)

- However, the user is a “computer expert”, who has “technical knowledge” about the system
  - Through some query language, the user can access, manipulate and update the database

The Central Idea

- Now consider a tourist information system
- In the background, it is nothing but a database of various tourist-related information
- However, its users may or may not be “computer experts”
  - They do not care about what goes on inside
  - They just want to “get” the information “easily”

The Central Idea

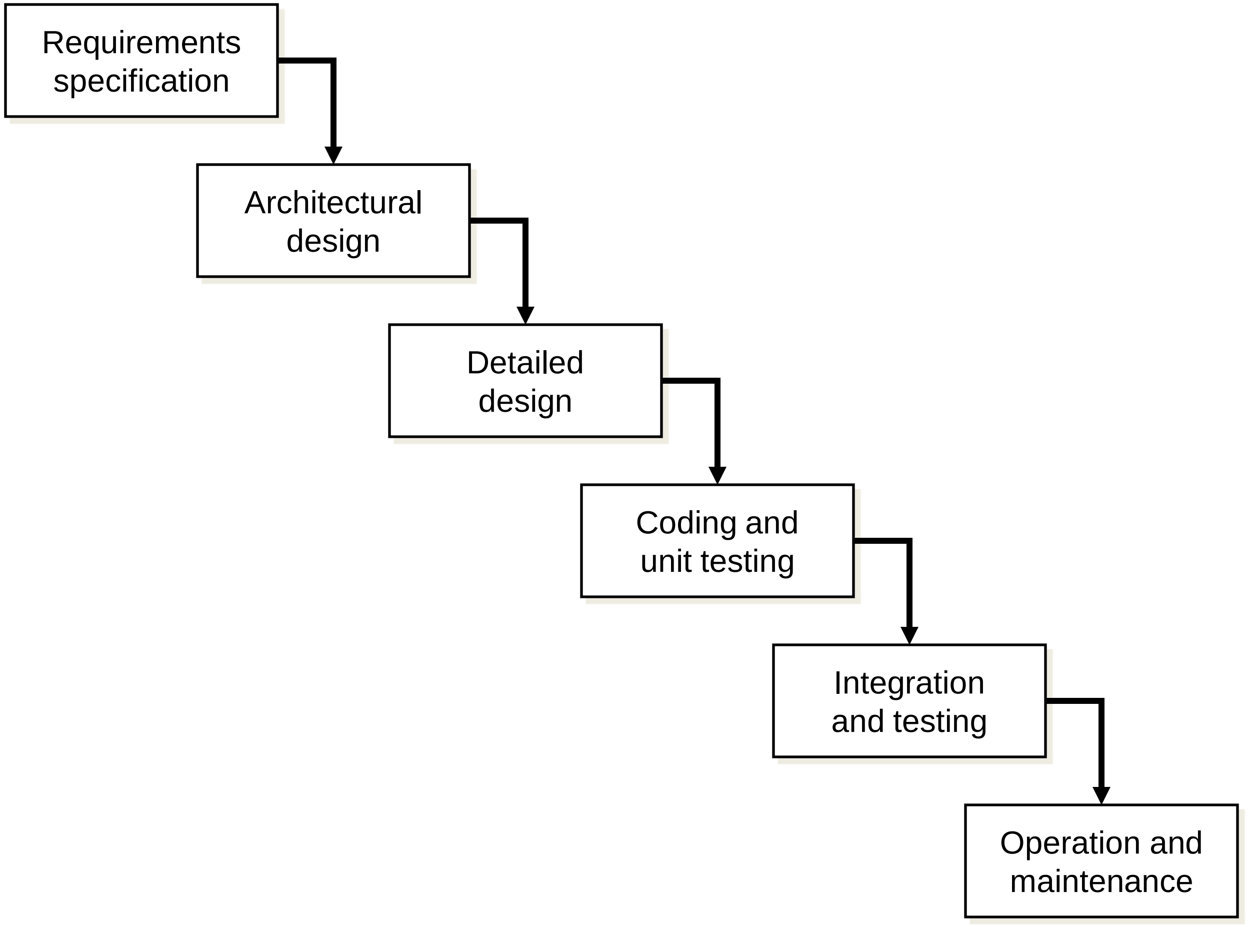
- The term “easily” is very significant
  - It means, we need to have an interface and interaction mechanism that do not require any specialized knowledge
- That is, we need a “usable” system
- **Design goal of an interactive system: increase usability**

What Happens in Software Engineering

- The waterfall model: the simplest and typical way to visualize software design

- Design process composed of a series of sub-stages
  - Each sub-stage follows the previous stage and precedes the next stage (looks like a waterfall)

The Waterfall Model



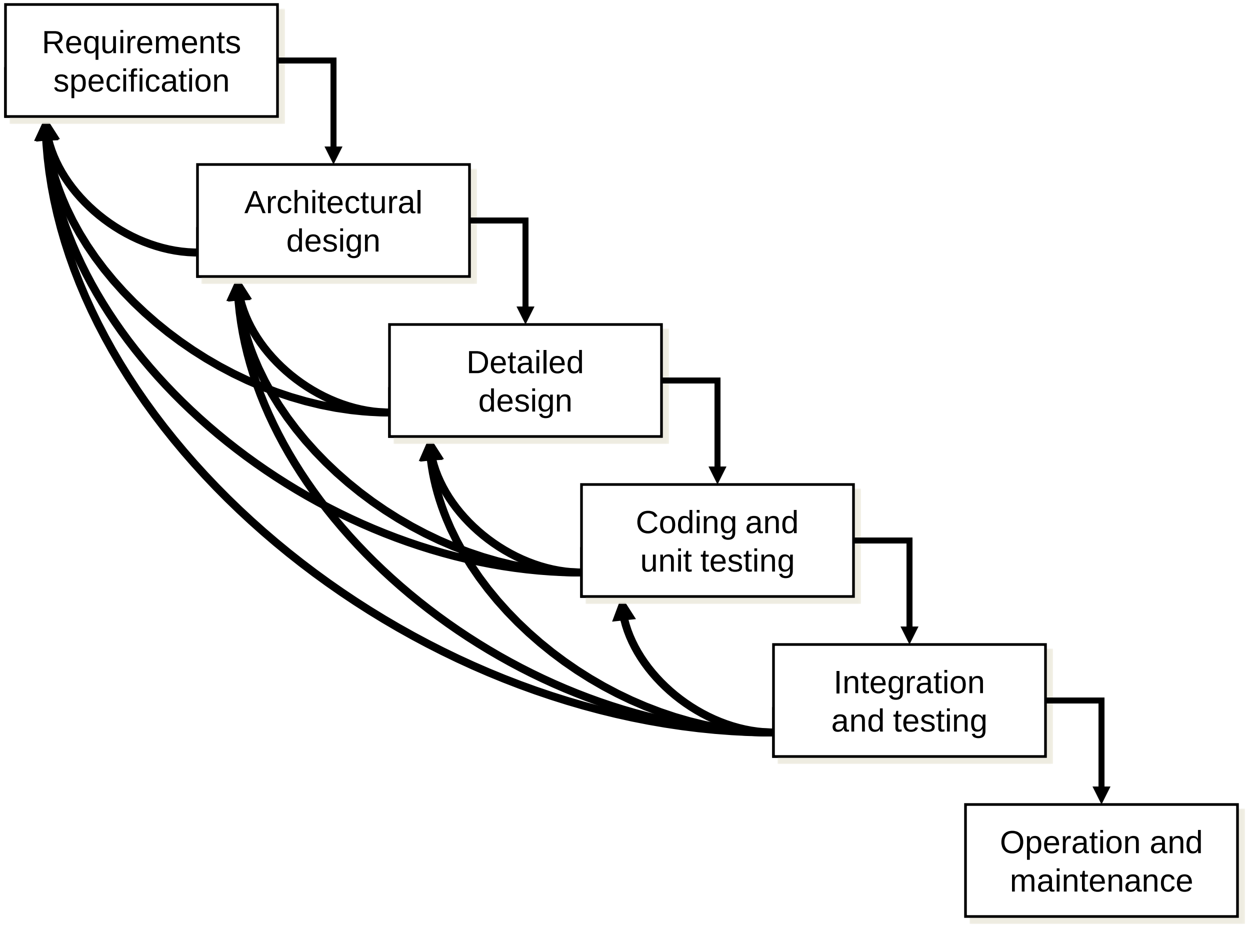
What Happens in Software Engineering

- Note the uni-directional flow (that’s how real waterfalls work anyway!!)
- In other words,
  - Each stage depends on the previous stages but not vice-versa

Interactive System Design

- The uni-directional flow is not appropriate for interactive system design
- In other words,
  - Each stage depends on the previous stages. It may also depend on the next stages (feedback).
- It is no longer the (linear) waterfall model

Interactive System Design



Why This Difference

- We are trying to design “for” the user
  - Not for a programmer’s convenience or expert’s use
- What should we consider
  - Abilities and needs of the users
  - Their usage context
  - Their work setting
- In other words, we have to “know the user”

Need: Know The User

- A never ending process because there is so much to know and because the users keep changing
- An interactive system designer should consider the human factors that characterize users

The Human Factors

- Perception: our ability to perceive our surroundings
  - Can be visual, auditory or haptic (touch)
- Cognition: the way we process the perceived information in our “mind” and take decisions
- Motor action: this is the mechanism through which we interact with the surrounding
  - Example: hand movement, eyeball movement, speech

Need: Know The User

- These factors (user characteristics) vary with
  - Age, gender, physical and cognitive abilities, personality
  - Education, cultural or ethnic background
  - Training, motivation, goals
- An interactive system designer should recognize this diversity

Need: Recognize Diversity

- Systems used by several communities of users
  - No single design can satisfy all users and situations
- Designer faces real challenge to cater to the need of each community
  - Designers must characterize users and situations as precisely and completely as possible

A Generic User Characterization

- Novice or first time users
  - Know nothing about the task or interface concepts
  - Often anxious about the computer and its functionality
- Knowledgeable or intermediate users
  - Have stable task concepts and broad interface concepts

A Generic User Characterization

- Expert users
  - Thoroughly familiar with the task and interface concepts
  - Want to get their job done quickly

So, Why The Difference?

- Designer must know the user
  - This knowledge can not be captured at once
- Design involves acquiring new knowledge and using it to refine design in continuous cycle (till some “acceptable” design is found)
  - The reason for so many “feedbacks” in the waterfall model

User Centered Design (UCD)

- The design process, where designer collects feedback about the design from users and use this to refine design, is known as “user centered design” or UCD

- UCD is based on understanding the domain of work or play in which people are engaged and in which they interact with computers

User Centered Design (UCD)

- Assumptions
  - Result of a good design is a *satisfied user*
  - Process of design is a *collaboration between designers and user*.
  - Design *evolves and adapts* to users' changing concerns, and the process produces a specification as an important byproduct
  - The user and designer are in *constant communication* during the entire process

UCD Drawbacks

- In UCD, user involvement is “passive”
  - The designer elicits feedback from user (through interviews, informal discussions etc)
  - Prepares specification on the basis of user response
  - Take feedback on the design and makes refinements

UCD Drawbacks

- Problems with “passive” involvement of user
  - User intuition about a new interface may not be correct (feedback not reliable)
  - The interview process itself may not be formulated properly (designer asks wrong questions)
  - It is not possible for the designer to identify all possible issues to take feedback from users, as the designer’s knowledge about the user may not be complete

Participatory Design

- Solution: make (representative) users a part of the design team

- Such a design process, where end users are part of the design team, is known as “participatory design”

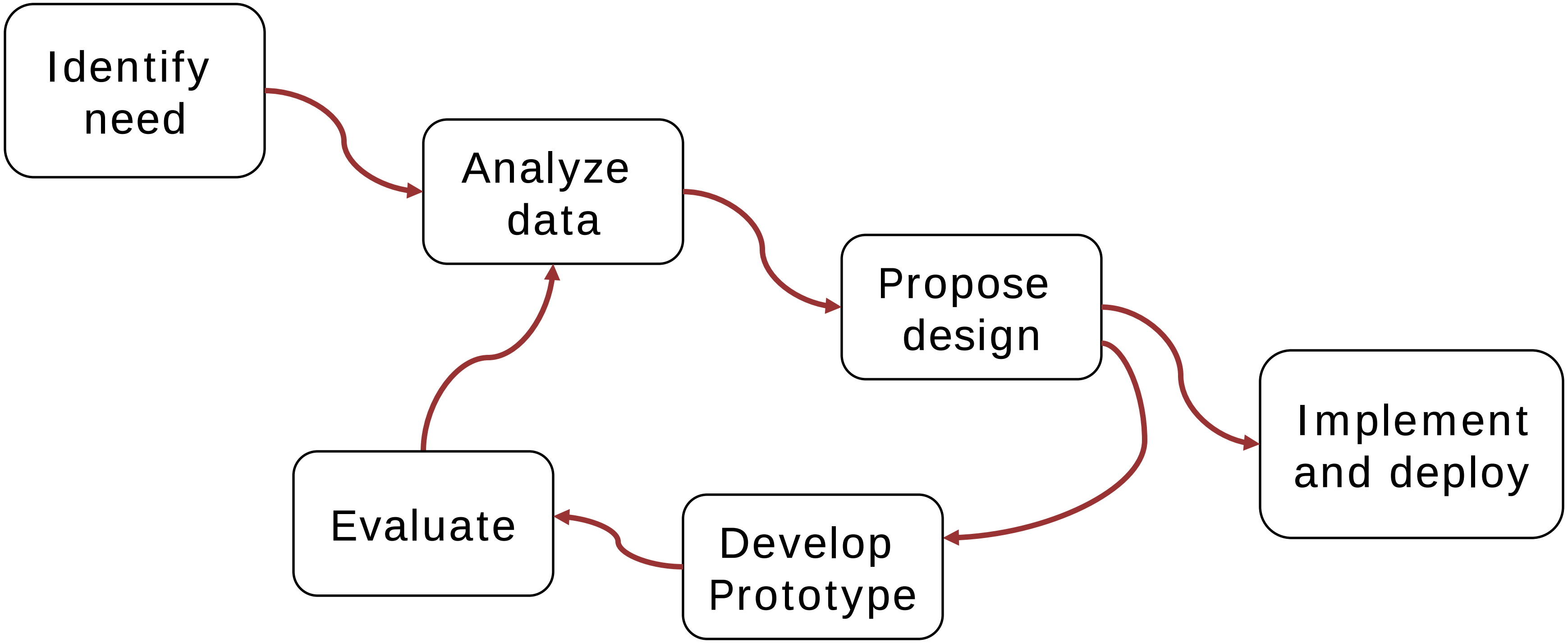
Participatory Design: Key Points

- Users are first-class members of the design team
  - As opposed to their passive involvement in UCD
- Users are considered subject experts
  - Know all about their work context
- Iterative design process
  - All design stages are subject to revision

Interactive System Design Life Cycle (ISLC)

- Key stages
  - Know the user, propose design, evaluate design by users, refine design
- Iterative design process
  - The above stages are iterated till an acceptable (determined from user feedback) solution is obtained

ISLC:
A Consolidated Diagram



Identify need

Analyze data

Propose design

Implement and deploy

Develop Prototype

Evaluate

Life Cycle Stage: Identify Need

- What is wanted – identify users and their need
- Designers make use of one or more methods to identify the requirements

Life Cycle Stage: Identify Need

- Such methods include
  - Interview (structured, semi-structured, unstructured)
  - Contextual inquiry
  - Cultural probes
  - Ethnography
  - User models

Life Cycle Stage: Analyze Data

- Analysis of the data collected
- Two types of analysis are performed
  - Scenario analysis: analyze data collected from the user on one or more usage scenario of the system

Life Cycle Stage: Analyze Data

- Analysis of the data collected
- Two types of analysis are performed
  - Task analysis: analyze tasks required to be carried out by the user to operate the system
    - System level task analysis: analysis of external tasks required to operate the system

Life Cycle Stage: Analyze Data

- Analysis of the data collected
- Two types of analysis are performed
  - Task analysis: analyze tasks required to be carried out by the user to operate the system
    - Cognitive task analysis: analysis of tasks performed in the mind of the user

Life Cycle Stage: Propose Design

- Design proposal arrived at from the analysis of collected data
  - Guidelines and principles help in the development of initial design
  - Several sets of guidelines (both general and specific) are there to cater to specific interface design context

Life Cycle Stage: Develop Prototype

- Implement a prototype of the design for collecting user feedback
- A spectrum of techniques is used in developing prototypes
  - Paper prototype (one extreme)
  - Complete software (other extreme)
  - Lots in between …

Life Cycle Stage: Evaluate Design

- Evaluation of the design by users
- In the initial design phase, evaluation is done on prototypes
  - Cost effective and easier to perform
  - Suitable for iterative design process where the evaluation is performed many times

Life Cycle Stage: Evaluate Design

- Evaluation of the design by users
- The full system is typically evaluated at the end
  - Full system evaluation is costly in terms of money, manpower, time and effort
  - Hence, typically done once or a limited number of times

Life Cycle Stage: Evaluate Design

- Several evaluation methods are available
  - Checklist/guideline based evaluation
    - Heuristic evaluation, cognitive walkthrough
  - Model-based evaluation: employs models (of the system or user or hybrid) for evaluation
    - Hybrid models are essentially models that combines the features of both the system and the user

Life Cycle Stage: Evaluate Design

- Several evaluation methods are available
  - Empirical evaluation – evaluate with real users
    - Involve implementation of the system with full functionalities

Note

- In subsequent lectures, these stages and methods of the ISLC will be discussed in details