

Lecture 2

HCI in the software process

Chapter 6

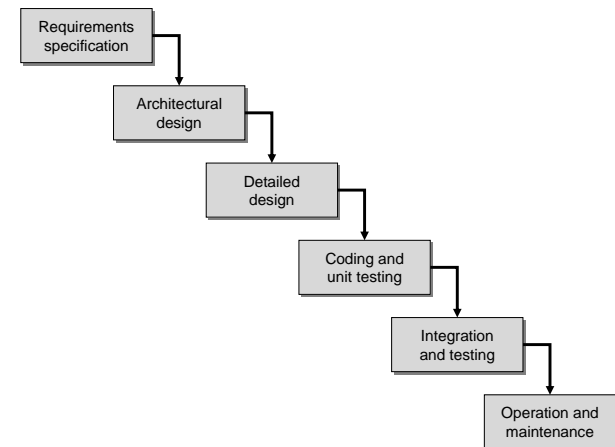
Agenda

- Software engineering and the design process for interactive systems
- Iterative design and prototyping
- When is each best?
- Self study – read and take your own notes
 - Usability engineering
 - Design rationale

The software lifecycle

- Software engineering is the discipline for understanding the software design process, or life cycle
- Designing for usability occurs at all stages of the life cycle, not as a single isolated activity
- There are many models of the software life cycle we will look at the 2 main ones
 - Waterfall
 - Prototyping

The waterfall model



Activities in the life cycle

- Requirements specification
 - designer and customer try capture what the system is expected to provide can be expressed in natural language or more precise languages, such as a task analysis would provide
 - Informal design and scenario based design will result in better requirements analysis



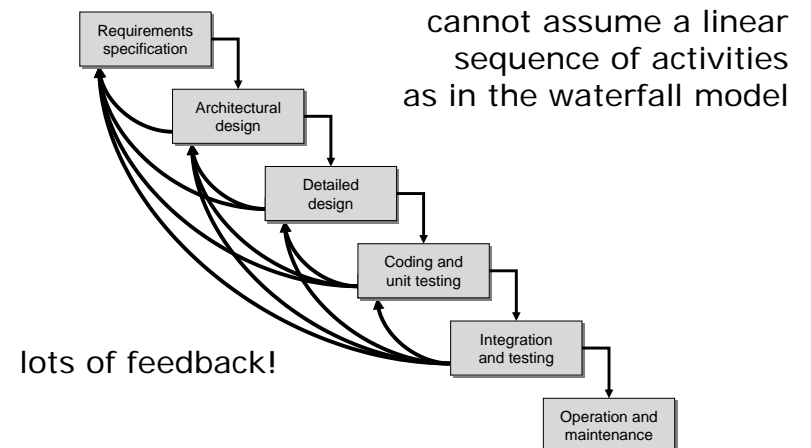
Detailed design

- Detailed design of the interface
- Move from informal to formal specification
- Separation of layers
 - A layered approach to software development will provide for more flexibility
 - Data
 - Logic
 - Interface

Testing

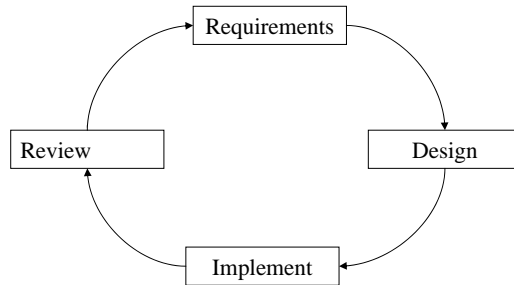
- Testing is not only about functionality of code
- Usability testing
 - There are some basics that are nearly always important
 - Layout
 - Language
 - Number of click/steps to perform task
 - Choose rather than remember
 - More detail in weeks 7 & 8

The life cycle for interactive systems



Iterative design and prototyping

- Iterative design overcomes inherent problems of incomplete requirements



- Prototypes
 - simulate or animate some features of intended system
 - different types of prototypes
 - throw-away
 - incremental
 - evolutionary
- Management issues
 - time
 - planning
 - non-functional features
 - contracts

Techniques for prototyping

Storyboards

- need not be computer-based
- can be animated

Limited functionality simulations

- some part of system functionality provided by designers
- tools like HyperCard are common for these
- Wizard of Oz technique

Warning about iterative design

- design inertia – early bad decisions stay bad
- diagnosing real usability problems in prototypes....
 - and not just the symptoms

Research example

- We wanted to build a paperless assignment grading product with pen annotation of assignments
 - New paradigm
 - Few studies
 - Technical challenges
- Build a prototype



Waterfall or Prototype

- Waterfall
 - Interaction paradigm 'standard' and well understood?
 - The problem is well understood?
 - Data centric systems
 - Information systems
 - Data warehouse
- Prototype
 - The interaction paradigm new or poorly understood?
 - The problem definition is incomplete or poorly defined?
 - Interface centric systems
 - games
 - Modelling
 - Design tools

Waterfall or prototype

- It doesn't have to be a one or the other decision
- Many systems are a blend
 - With some parts are prototyped to elicit requirements
- There isn't one 'best way'
- Nor is there a 'silver bullet'

Summary

The software engineering life cycle

- distinct activities and the consequences for interactive system design

Design rationale

- recording design knowledge
- process vs. structure