

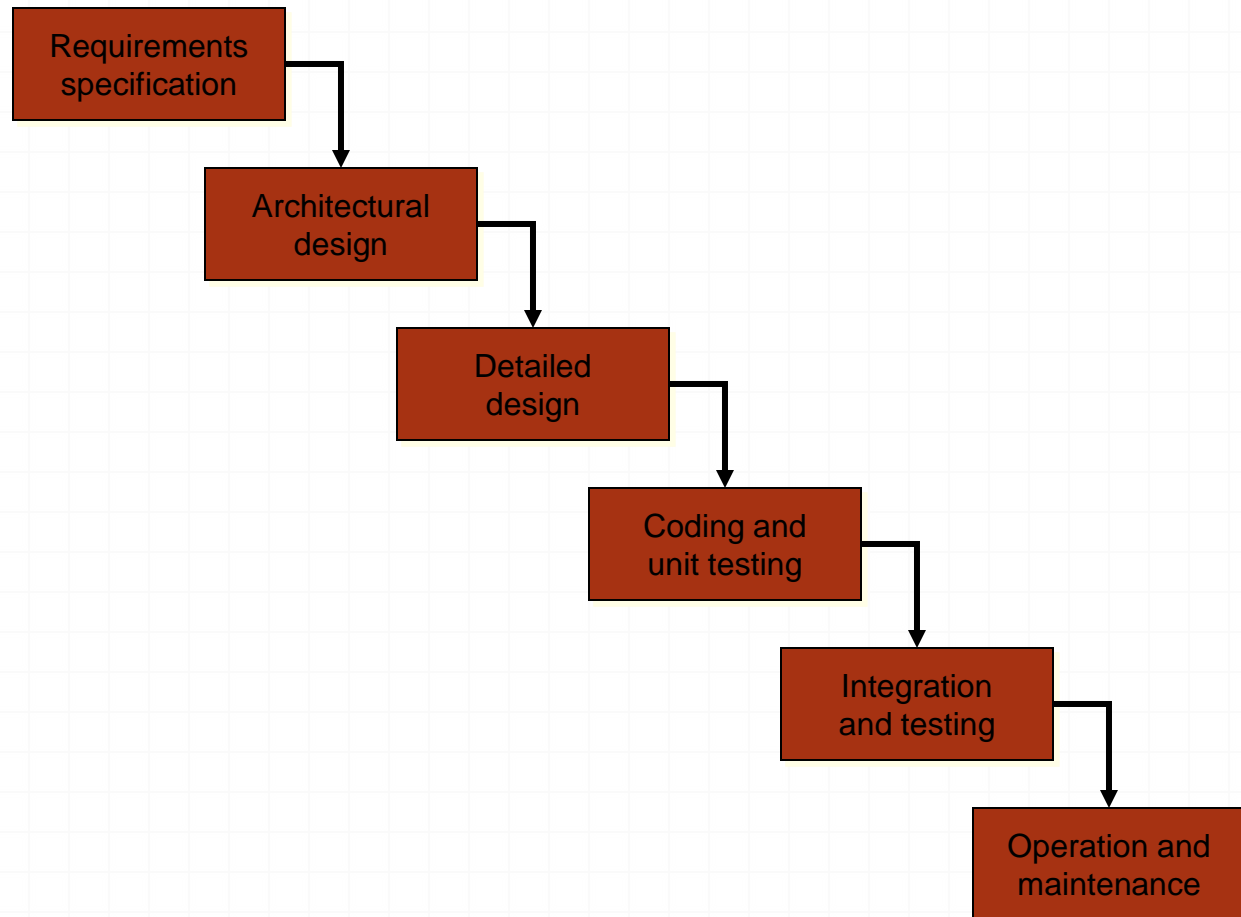
Chapter 6

HCI in the software
process

Software Engineering

- Software design process or life cycle called Software Engineering
 - that addresses the management and technical issues of the development of software systems.
 - The software life cycle is an attempt to identify every activity that occur in software development from the initial concept formation for a software system up until its eventual phasing out and replacement.
 - These activities must then be ordered in time in any development Project

The waterfall model



Activities in the life cycle

o Requirements specification

- o Designers try to capture what functionality is expected from the system
- o “HOW” is not concern of this activity.
- o Can be expressed in natural language or more precise languages like Unified Modeling Language.
- o begins at the start of product development and the requirements are from the customer’s perspective
- o It involves eliciting information from the customer about the work environment, or domain, in which the final product will function.

Activities in the life cycle

◦ Architectural Design:

- a decomposition of the system that allows independent development of separate components
- These components will later be integrated
- satisfying both functional and non-functional requirements after integration

◦ Detailed Design:

- Designer is responsible to ensure complete functionality of the component(s) provided by the architectural design phase.

Activities in the life cycle

o **Coding & Testing:**

- o Transform detailed design in a programming language.
- o Component testing

o **Integration & Testing:**

- o Completed components should be tested individually
- o they must be integrated as described in the architectural design.
- o Integration testing is performed to ensure correct behavior and acceptable use of any shared resources e.g. memory

Activities in the life cycle

o **Maintenance:**

- o involves the correction of errors in the system which are discovered after release
- o System can be revised to satisfy requirements that were not realized during previous development.
- o maintenance provides feedback to all of the other activities in the life cycle

Verification and validation

o Verification

- o Are you building the product right?
- o Software must conform to its specification

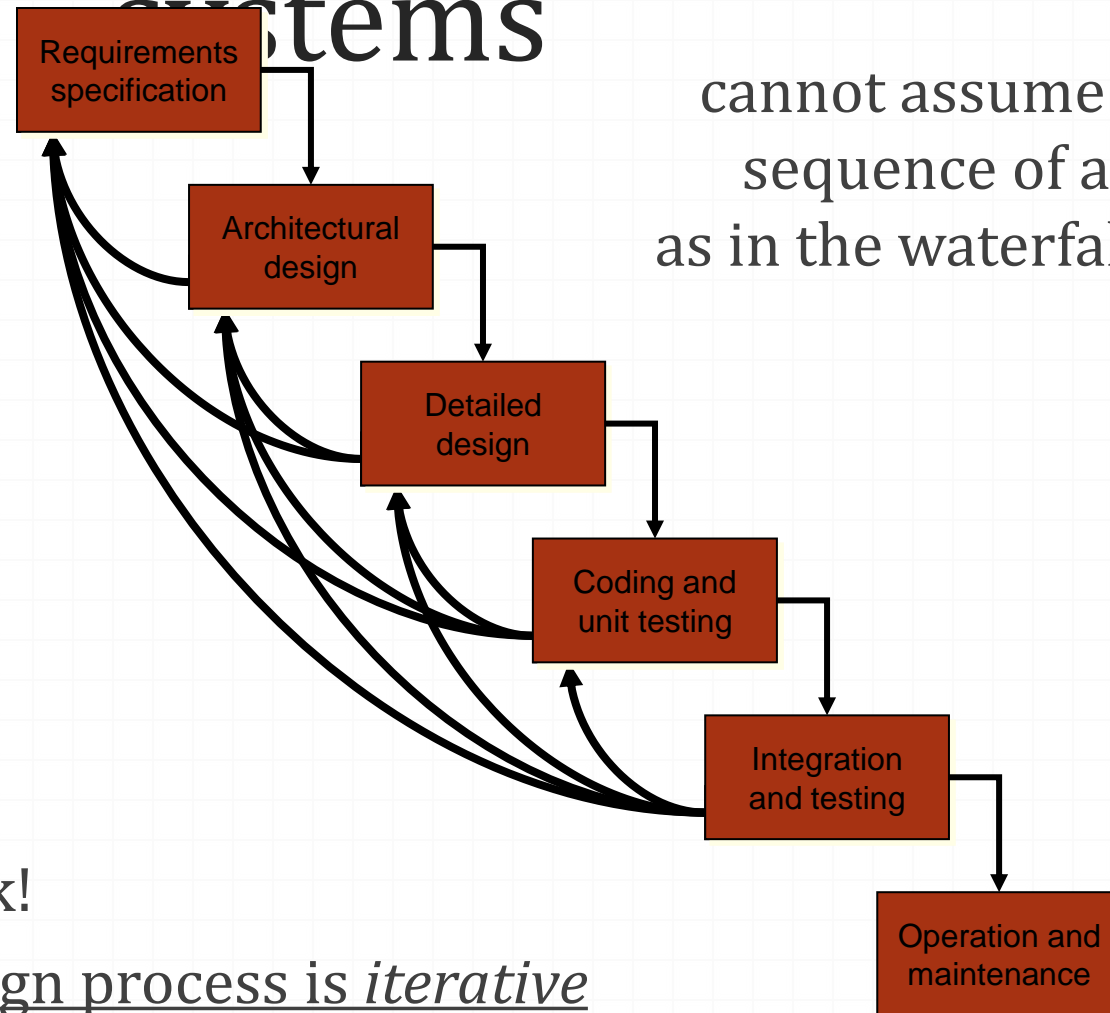
o Validation

- o Are you building the right product?
- o Software should do what the user really requires

o Validation & Verification relies on some proof

- o Subjective means of proof i.e User signed Requirements/system design
- o Management contracts
 - o Time
 - o Economical issues
 - o Recourses e.g. Manpower, hardware etc

The life cycle of interactive systems



cannot assume a linear sequence of activities as in the waterfall model

o lots of feedback!

o The actual design process is *iterative*

ISO usability standard 9241

Usability has following categories:

- effectiveness

 - can you achieve what you want to?

- efficiency

 - can you do it without wasting effort?

- satisfaction

 - do you enjoy the process?

ISO usability standard 9241

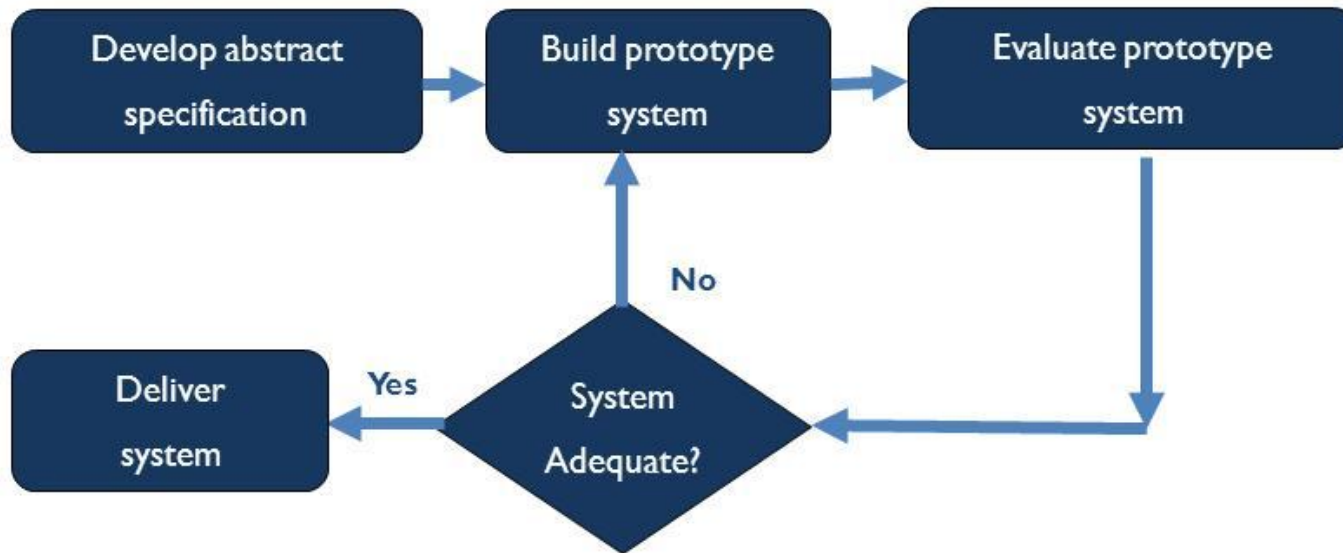
Usability objective	Effectiveness measures	Efficiency measures	Satisfaction measures
Suitability for the task	Percentage of goals achieved	Time to complete a task	Rating scale for satisfaction
Appropriate for trained users	Number of power features used	Relative efficiency compared with an expert user	Rating scale for satisfaction with power features
Learnability	Percentage of functions learned	Time to learn criterion	Rating scale for ease of learning
Error tolerance	Percentage of errors corrected successfully	Time spent on correcting errors	Rating scale for error handling

Iterative design and prototyping

- Iterative design overcomes inherent problems of incomplete requirements
- **Prototypes**
 - Simulate or animate some but not all features of intended system
- Types of prototypes:
 - throw-away/Rapid Prototyping
 - Incremental Prototyping
 - Evolutionary Prototyping

Throw Away/Rapid Prototyping

- o The prototype is built and tested.
- o It is likely to *inform* the final solution, but the prototype itself will not become part of the final solution.



Techniques of Prototyping

o **Storyboards:**

- o is a graphical depiction of the interface without accompanying system functionality.
- o can be animated, If not animated, storyboards usually include annotations and scripts indicating how the interaction will occur.

o **Limited functionality simulation:**

- o some part of system functionality provided by designers
- o tools like HyperCard & Wizard of Oz technique