Overview

- Software engineering and the design process for interactive systems
- Standards and guidelines as design rules
- Usability engineering
- Iterative design and prototyping
- Design rationale

Introduction

Paradigms and principles concentrated on examining the product of interactive system design.

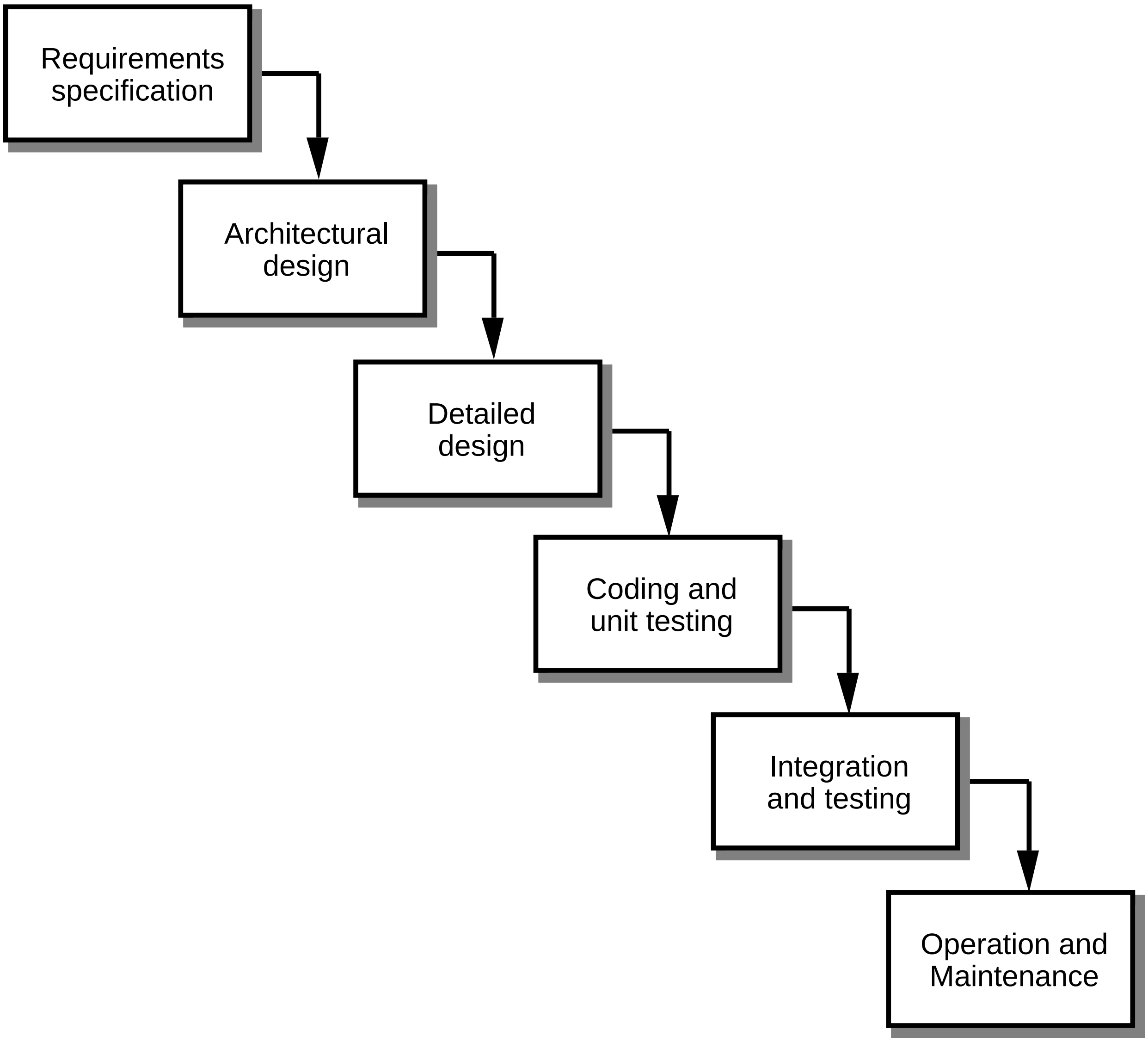
Now we focus on the process of design.

Software engineering is the emerging discipline for understanding the design process, or life cycle.

Designing for usability occurs at all stages of the life cycle, not as a single isolated activity

The software life cycle

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

Architectural design

high-level description of *how* the system will provide the services required

factor system into major components of the system and how they are interrelated

needs to satisfy both functional and nonfunctional requirements

Detailed design

refinement of architectural components and interrelations to identify modules to be implemented separately

the refinement is governed by the nonfunctional requirements

Activities in the life cycle (cont'd)

Coding and unit testing

implementing and testing the individual modules in some executable programming language

Integration and testing

combining modules to produce components from the architectural description

Operation and maintenance

product is delivered to customer and any problems/enhancements are provided by designer while product is still live

the largest share of the life cycle

Verification and validation

Verification

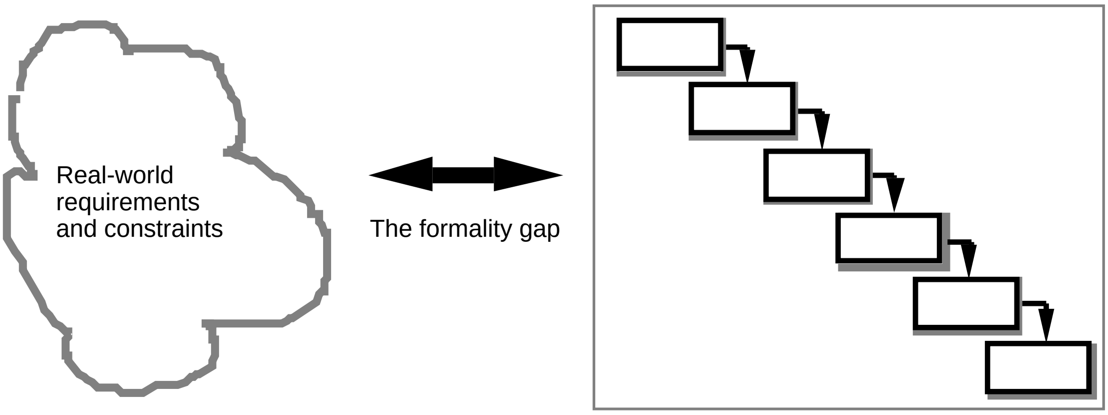
designing the product right

Validation

designing the right product

The formality gap

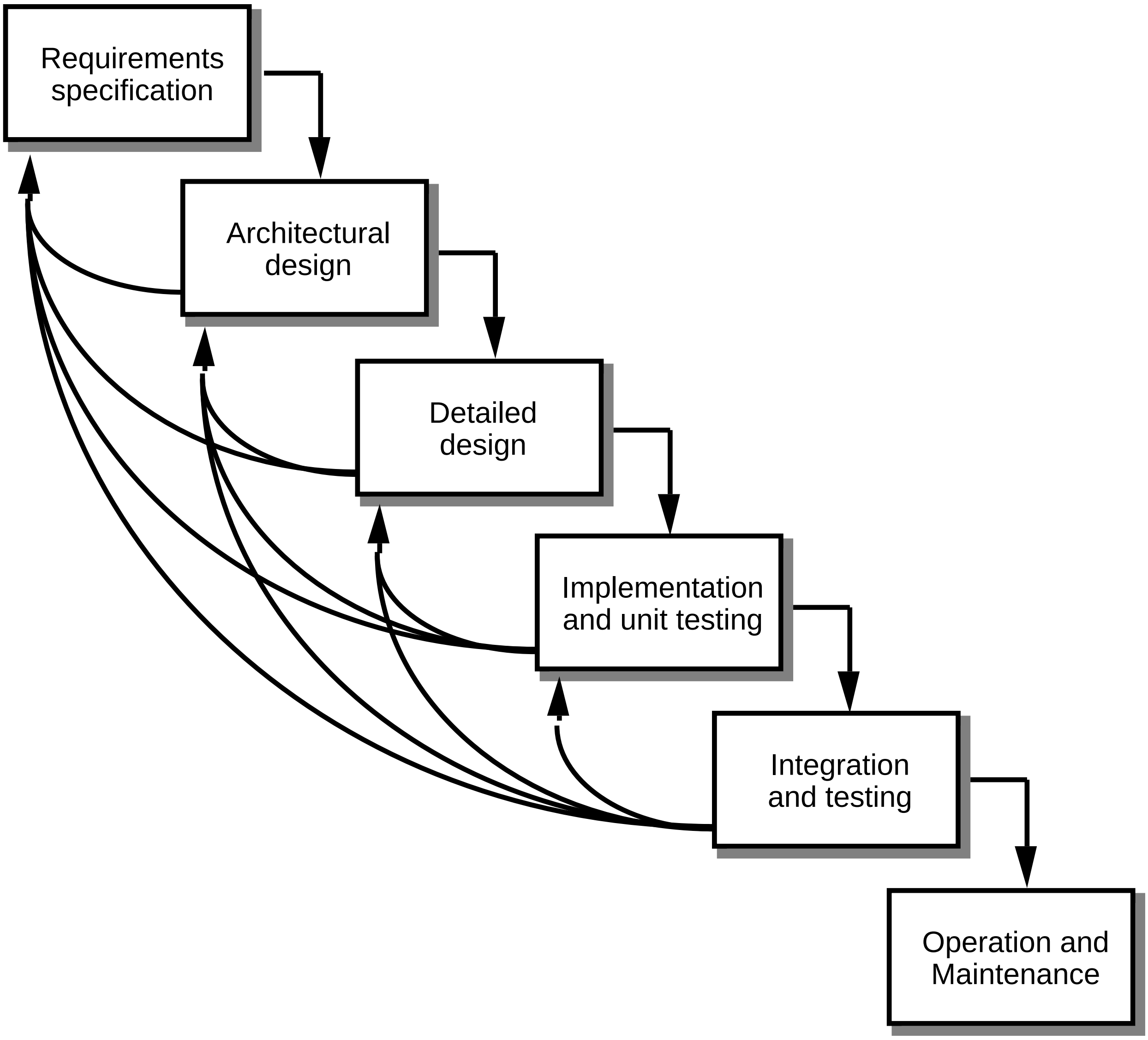
validation will always rely to some extent on subjective means of proof



Management and contractual issues

design in commercial and legal contexts

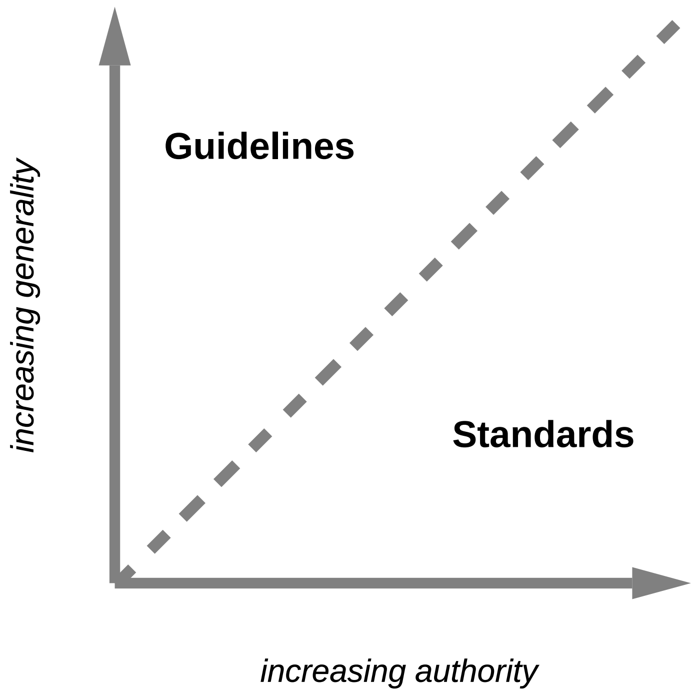
The life cycle for interactive systems


Requirements specification
Architectural design
Detailed design
Implementation and unit testing
Integration and testing
Operation and Maintenance

Cannot assume a simple linear sequence of activities as assumed by the waterfall model

Using design rules

Design rules suggest how to increase usability



Standards

set by national or international bodies to ensure compliance by a large community of designers

standards require sound underlying theory and slowly changing technology

hardware standards more common than software

high authority and low level of detail

ISO 9241 defines usability as effectiveness, efficiency and satisfaction with which users accomplish tasks

Using design rules (cont'd)

Guidelines

more suggestive and general

many textbooks and reports full of guidelines

abstract guidelines (principles) applicable during early life cycle activities

detailed guidelines (style guides) applicable during later life cycle activities

understanding justification for guidelines aids in resolving conflicts

Usability engineering

The ultimate test of usability based on measurement of user experience

Usability engineering demands that specific usability measures be made explicit as requirements

Usability specification

usability attribute/principle

measuring concept

measuring method

now level/ worst case/ planned level/ best case

Problems

usability specification requires level of detail that may not be possible early in design

satisfying a usability specification does not necessarily satisfy usability

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

Design rationale

Design rationale is information that explains why a computer system is the way it is.

Benefits of design rationale

- communication throughout life cycle
- reuse of design knowledge across products
- enforces design discipline
- presents arguments for design trade-offs
- organizes potentially large design space
- capturing contextual information

Process-oriented

preserves order of deliberation and decision-making

Structure-oriented

emphasizes *post hoc* structuring of considered design alternatives

Design rationale techniques

Issue-based information system (IBIS)

- basis for much of design rationale research
- process-oriented
- hierarchical structure of issues, with one root issue
- positions are potential resolutions of an issue
- arguments modify the relationship between positions and issues
- gIBIS is a graphical version

Design space analysis

- structure-oriented
- QOC – hierarchical structure
  - questions (and sub-questions) represent major issues of a design
  - options provide alternative solutions to the question
  - criteria are the means of assessing the various options in order to make a choice
- DRL – similar to QOC with a larger language and more formal semantics

Design rationale techniques (cont'd)

Psychological design rationale

to support task-artefact cycle in which user tasks are affected by the systems they use

aims to make explicit consequences of design for users

designers identify tasks system will support

scenarios are suggested to test task

users are observed on system

psychological claims of system made explicit

negative aspects of design can be used to improve next iteration of design

Summary

The software engineering life cycle

distinct activities and the consequences for interactive system design

Using design rules

standards and guidelines to direct design activity

Usability engineering

making usability measurements explicit as requirements

Iterative design and prototyping

limited functionality simulations and animations

Design rationale

recording design knowledge

process vs. structure