

Software Design and Architecture

1

Introduction

2

Objective

- To be introduced to principles of good design, and techniques for the evaluation of software design quality.
- To cover the principal architectural issues associated with the design and construction of software systems including architectural design and documentation, component models and technologies, and frameworks.
- To introduce the students to a number of design patterns and their applications.

3

Motivation

4

- A staircase that leads right into a wall!!!



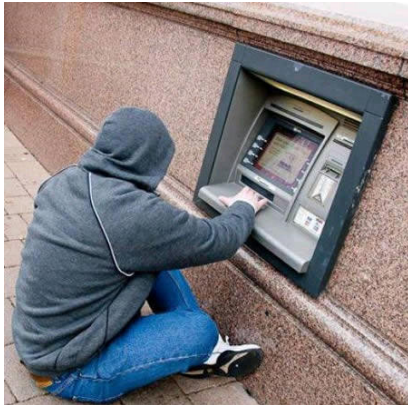
5

- A door that would drop you 10 feet down!



6

- An impossible to use ATM machine



7



8

- What do you think is wrong in these real life scenarios?
- The requirements are correct!
 - A staircase next to the outer wall
 - A door on the first floor
 - An ATM outside the bank branch
 - The bridge
- The design is flawed!
 - The execution based on the design results in disaster.

9

- Software are no different.
- For a useful software, it has to be 'engineered'; which involves giving specific attention to every phase of software development.

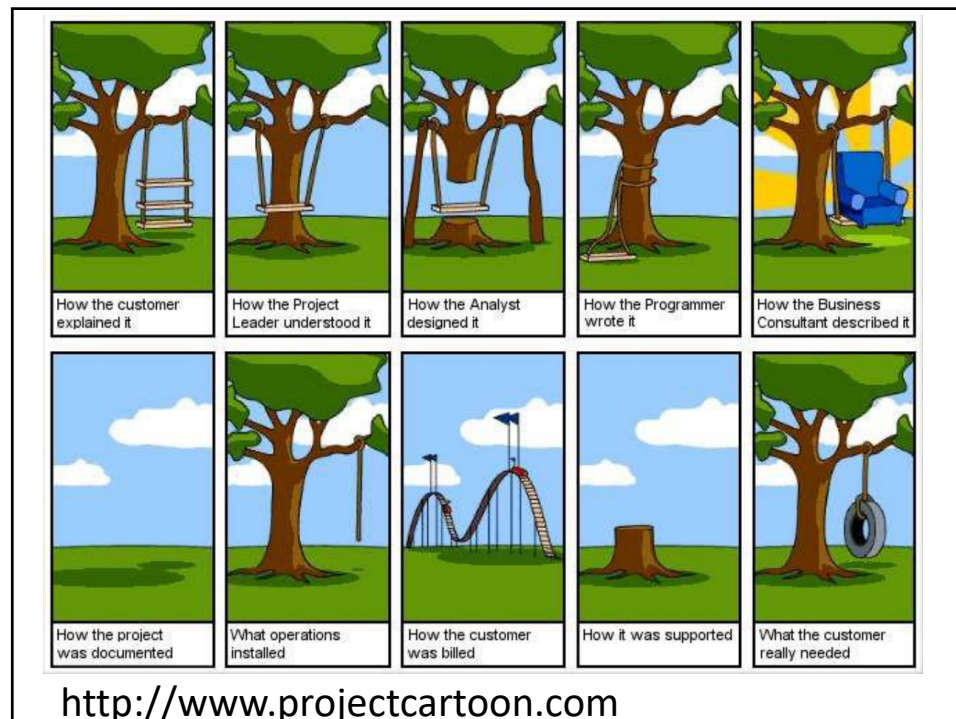
10

Problems in software development

• Common issues

- The final software does not fulfill the needs of the customer
- Hard to extend and improve: if you want to add a functionality later its mission impossible
- Bad documentation
- Bad quality: frequent errors, hard to use, ...
- More time and costs than expected

11



12

Ariane 5 Flight 501 Failure

Ariane flight V88 was the failed maiden flight of the Arianespace Ariane 5 rocket, vehicle no. 501, on 4 June 1996. It carried the Cluster spacecraft, a constellation of four European Space Agency research satellites. The launch ended in failure due to multiple errors in the software design: dead code, intended only for Ariane 4, with inadequate protection against integer overflow led to an exception handled inappropriately—halting the whole inertial navigation system that otherwise would have been unaffected. This resulted in the rocket veering off its flight path 37 seconds after launch, beginning to disintegrate under high aerodynamic forces, and finally self-destructing via its automated flight termination system. **The failure has become known as one of the most infamous and expensive software bugs in history.** The failure resulted in a loss of more than US\$370 million.

13

Software Design

14

What is Design?

- Design is the first step in the development phase for any engineered product or system.
- Design is about HOW the system will perform its functions.

15

Software Design

- A software design is a meaningful engineering representation of some software product that is to be built.

“The process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient detail to permit its physical realization” [TAY59]

16

Software Design

- The literature on design methods began to appear in the 1950s and 60s.
- Since then, design methodology has become an independent discipline of scientific study.
 - What are the essential characteristics of design?
 - What processes are used by designers?
 - Is one process better than another, constituting 'right' and 'wrong' ways to design?
 - Why are some processes favorable over others?
 - Do different processes lead to different qualities of results?

17

17

Software Design

- As one of the most complex man-made artifacts, computer software is very difficult to design.
- There are many factors that affect designs and many stakeholders,
 - i.e. people who participate in the design process, play various different roles in the design processes and influence the design of software.

18

18

Software Design - Simplified

Requirements specification was about the WHAT the system will do

Design is about HOW the system will perform its functions

- provides the overall decomposition of the system
- allows to split the work among a team of developers
- also lays down the groundwork for achieving non-functional requirements (performance, maintainability, reusability, etc.)
- takes target technology into account (e.g., kind of middleware, database design, etc.)

19

Why Design is important?

- Without a proper design, we risk building an unstable system
 - one that will fail when small changes are made
 - one that may be difficult to test
 - one whose quality cannot be assessed until late in the software process, perhaps when critical deadlines are approaching, and much capital has already been invested into the product.

20

Major Design Challenges

- Design team should not do too much
 - Detailed design should not become code
- Design team should not do too little
 - It is essential for the design team to produce a complete detailed design

21

Software Development Activities

- **Requirements Elicitation**
- **Requirements Analysis** (e.g., Structured Analysis, OO Analysis)
 - analyzing requirements and working towards a conceptual model without taking the target implementation technology into account
 - useful if the conceptual gap between requirements and implementation is large
 - part of requirements engineering (but may produce more than what is going to be part of the requirement spec)
- **Design**
 - coming up with solution models taking the target implementation technology into account
- **Implementation**
- **Testing**

22

Software Design in SDLC

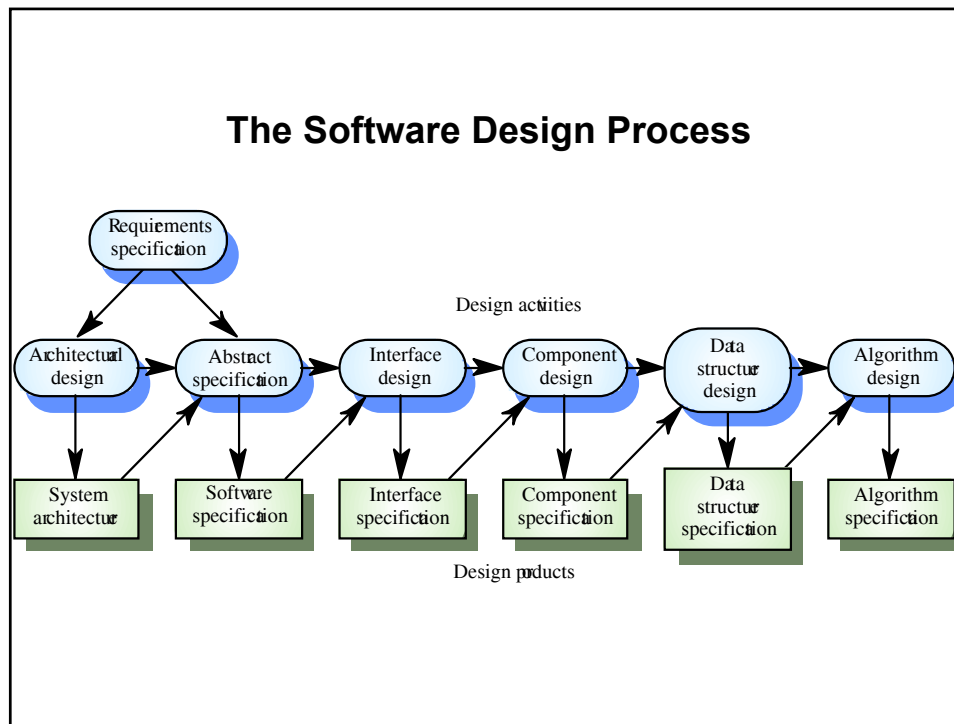
- In SDLC (Software Development Life Cycle), Design phase is one of the most important phases.
- In the software engineering context, design focuses on four major areas of concern: data, architecture, interfaces and components.

23

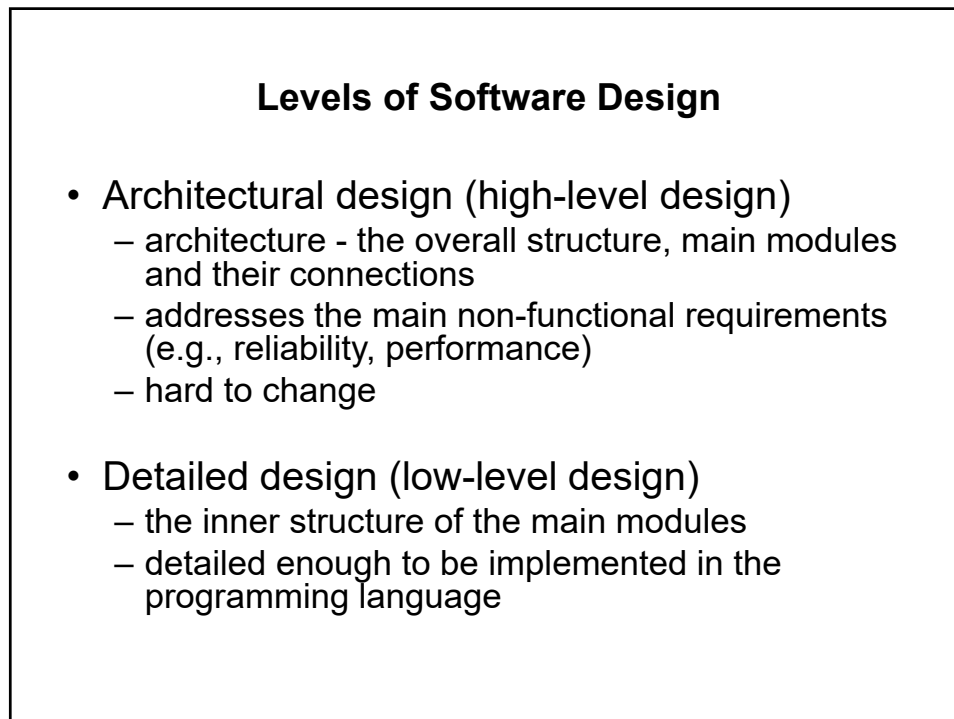
Design Process Activities

- Architectural design
 - Modules, inter-relationships etc
- Abstract specification
 - Services of each sub-system, constraints etc
- Interface design
 - Interface to other sub-system or outside environment
- Component design
 - Services allocated to components and their interfaces designed
- Data structure design
- Algorithm design

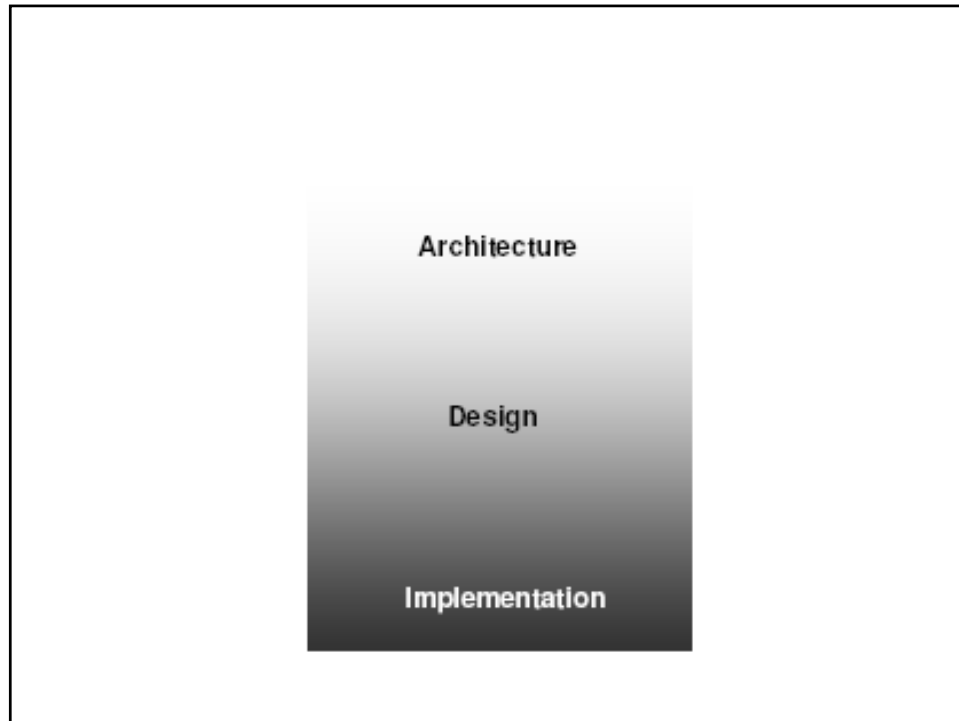
24



25



26



27

Design vs. Architecture

- Architecture is concerned with the selection of architectural elements, their interaction, and the constraints on those elements and their interactions
- Design is concerned with the modularization and detailed interfaces of the design elements, their algorithms and procedures, and the data types needed to support the architecture and to satisfy the requirements.
- Architecture...is specifically not about...details of implementations (e.g., algorithms and data structures.)

28

Design vs. Architecture (cont.)

S.No.	SOFTWARE DESIGN	SOFTWARE ARCHITECTURE
01.	Software design is about designing individual modules/components.	Software architecture is about the complete architecture of the overall system.
02.	Software design defines the detailed properties.	Software architecture defines the fundamental properties.
03.	In general, it refers to the process of creating a specification of software artifact which will help to developers to implement the software.	In general, it refers to the process of creating high level structure of a software system.

29

Design vs. Architecture (cont.)

S.No.	SOFTWARE DESIGN	SOFTWARE ARCHITECTURE
05.	Software design avoids uncertainty.	Software architecture manages uncertainty.
06.	Software design is more about on individual module/component.	Software architecture is more about the design of entire system.
07.	It is considered as one initial phase of Software Development Cycle (SSDLC) and it gives detailed idea to developers to implement consistent software.	It is a plan which constrains software design to avoid known mistakes and it achieves one organizations business and technology strategy.

30

Design vs. Architecture (cont.)

S.No.	SOFTWARE DESIGN	SOFTWARE ARCHITECTURE
08.	Some of software design patterns are creational, structural and behavioral.	Some of software architecture patterns are microservice, server less and event driven.
09.	In one word the level of software design is implementation.	In one word the level of software architecture is structure.
10.	How we are building is software design.	What we are building is software architecture.