

# SOFTWARE DESIGN & ANALYSIS (Week-1)

USAMA MUSHARAF

MS-CS (Software Engineering)

*LECTURER (Department of Computer Science)*

*FAST-NUCES PESHAWAR*

# COURSE CONTENT

- Principles of Object Technology. OOP Review. Principles of Modeling.
- OOA&D Overview. OO Development Process.
- Requirements Engineering, Analysis, and Specification: Requirements Engineering, Use Cases, Prototyping,
- Class Models. Interaction Diagrams. Verification and Validation.
- Architectural and Detailed Design.
- Class Diagrams. Interaction Diagrams. State Machines and Diagrams. Implementation, Package Diagrams. Activity Diagrams.
- OO Patterns,
- Verification and Validation.
- Note: Students may also be introduced to Object Diagram, Component Diagram, Package Diagram, Deployment Diagram, Network Diagram.

# RECOMMENDED BOOKS

## Recommended Books

- 1. Applying UML and patterns: An introduction to Object-Oriented Analysis and Design and Iterative Development by Craig Larman, Prentice Hall; 3rd Edition (October 30, 2004). ISBN-10: 0131489062
- 2. Using UML: Software Engineering with Objects and Components by Perdita Stevens, Addison-Wesley; 2nd Edition (February 13, 2006). ISBN-10: 0321269675
- 3. Fundamental of Object-Oriented Design in UML by Meiler Page-Jones, Addison Wesley, 2000. ISBN: 020169946X.
- 4. The Unified Modeling Language User Guide by G. Booch, J. Rumbaugh and I. Jakobson, Addison-Wesley Professional; 2nd Edition (2005). ISBN- 10: 0321267974.

# OBJECTIVE OF THIS COURSE

- The course will start with requirements gathering, & end with implementation in object oriented programming language.(JAVA)

# EVALUATION CRITERIA

Assessment	Weightage
Assignment	10%
Quiz	10%
Project	10%
Sessional 1	15%
Sessional 2	15%
Final Exam	40%

# WEEKLY PLAN

Week	Topics
Week 1 - 5	<ol style="list-style-type: none"><li>1. Intro to Software Design and Analysis, Brief overview of Software Engineering Phases, Problems in Software Development, Software Design Process, Difference between Design and Architecture,</li><li>2. Software Process Models. Software Requirement Engineering, Structure Analysis</li><li>3. Requirement Engineering (Cont...), Analysis Modeling, Object Oriented Analysis, Modeling with UML (Static and Dynamic Models), Design Principles and Concepts</li><li>4. Design Principles and Concepts (Cont..), Object Oriented Design Concepts, Object Oriented Design with a Case Study,</li><li>5. Case Study with in depth Requirement Analysis &amp; Design</li></ol>

# WEEKLY PLAN

Week	Topics
Week- 6 - 9	<b>Software Development</b> Intro to Java Programming, Environment Setup, Basic Syntax and Programming language constructs. Control Structures, Arrays, Multi-Dimensional Arrays, Practice with some small case studies. <b>Software Development (Design Mapping to Code)</b> Object-Orientation, Classes , Objects. Constructors. Class Association (Inheritance), Polymorphism, Object Association (Composition, Aggregation), Abstract Classes, Interfaces, SOLID Principles-Hands-on. Exception Handling,
Week 10 - 12	<ul style="list-style-type: none"><li>▪ GUI, Graphical User Interface Design in Java,</li><li>▪ Event Driven Programming in Java,</li><li>▪ Database Connectivity</li></ul>
Week 13 - 15	<b>Problem Solving Using Design Patterns,</b> Software Design Patterns, Design Patterns Implementation in Java.

# AGENDA OF WEEK # I

1. Intro to Software Design and Analysis,
2. Brief overview of Software Engineering Phases,
3. Software Design Process,
4. Difference between Design and Architecture,
5. Problems in Software Development





Software can have huge impact in  
any aspect of our society

# WHERE CAN WE FIND SOFTWARE?



# SOME POPULAR ONES...

**facebook**

☐ Remember Me

[Forgot your password?](#)

Email

Password

Login

**Facebook helps you connect and share with the people in your life.**



### Sign Up

It's free and anyone can join

Full Name:

Your Email:

New Password:

I am: 

Select Sex:

Birthday: 

Month:

Day:

Year:

Why do I need to provide this?

[To create a page for a celebrity, band or business, click here.](#)

# SOME POPULAR ONES...



Google Search

I'm Feeling Lucky

Google.com.pk offered in: اردو

# AND EVEN IN...



# CONCLUSION

Software is almost everywhere!!!

# SOFTWARE APPLICATIONS

- ✓ Personal Computer Software
- ✓ Business Software
- ✓ System Software
- ✓ Real Time Software
- ✓ Engineering & Scientific Software
- ✓ Embedded Software
- ✓ Web Based Software
- ✓ Artificial Intelligence Software

# 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





*A clever person solves a  
problem.*

*A wise person avoids it.*

- Albert Einstein



**SOLUTION**

**SOFTWARE ENGINEERING**

# SE HISTORY

- ⊙ SE introduced first in 1968 – conference about “software crisis” when the introduction of third generation computer hardware led more complex software systems than before.
- ⊙ Early approaches based on informal methodologies leading to
  - 
  - 
  -
- ⊙ Need for new methods and techniques to manage the production of complex software.

# WHAT IS SOFTWARE ENGINEERING?

- Systematic approach for developing software
- Methods and techniques to develop and maintain quality software to solve problems.
- Study of the principles and methodologies for developing and maintaining software systems.

# WHAT IS SOFTWARE?

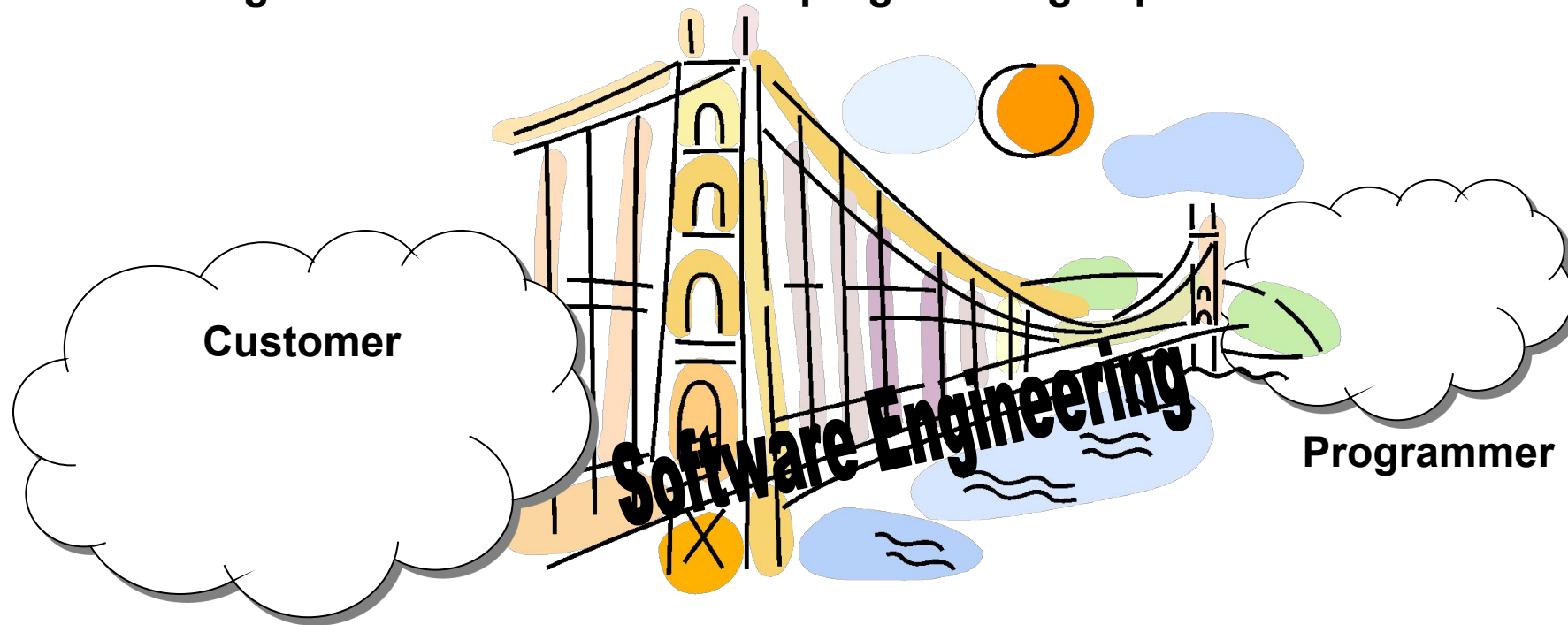
According to the IEEE

Software is:

*“Computer programs, procedures, and possibly associated documentation and data pertaining to the operation of a computer system”.*

# THE ROLE OF SOFTWARE ENGINEERING

A bridge from customer needs to programming implementation

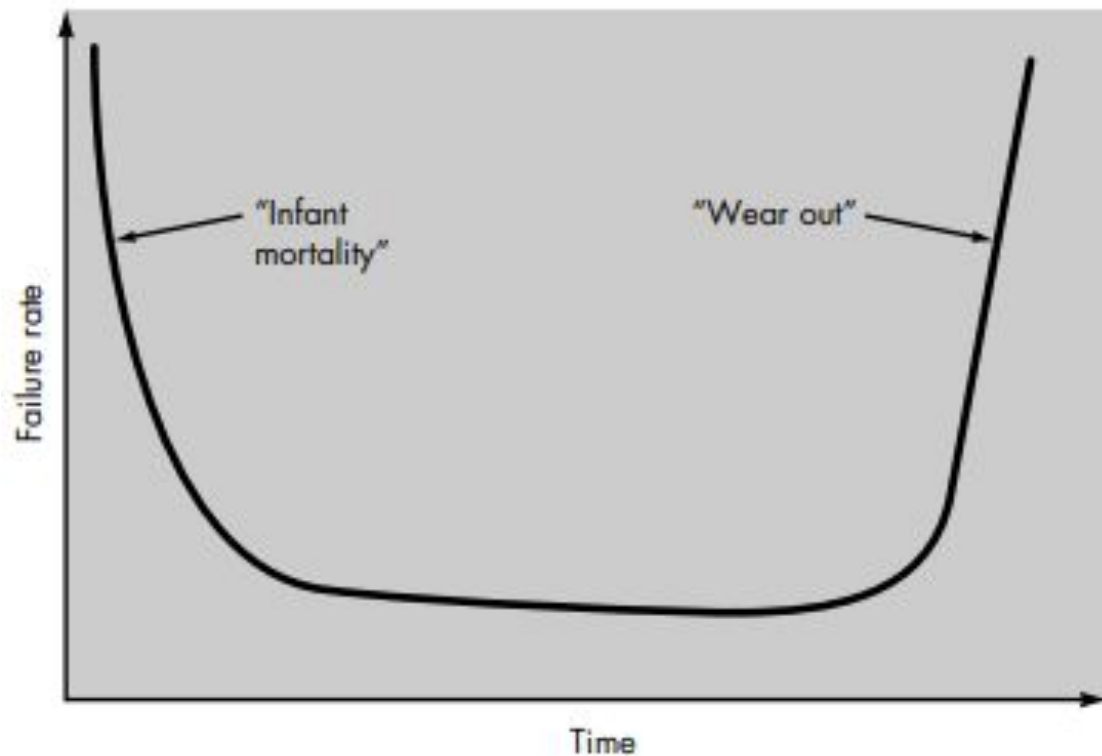


## First law of software engineering

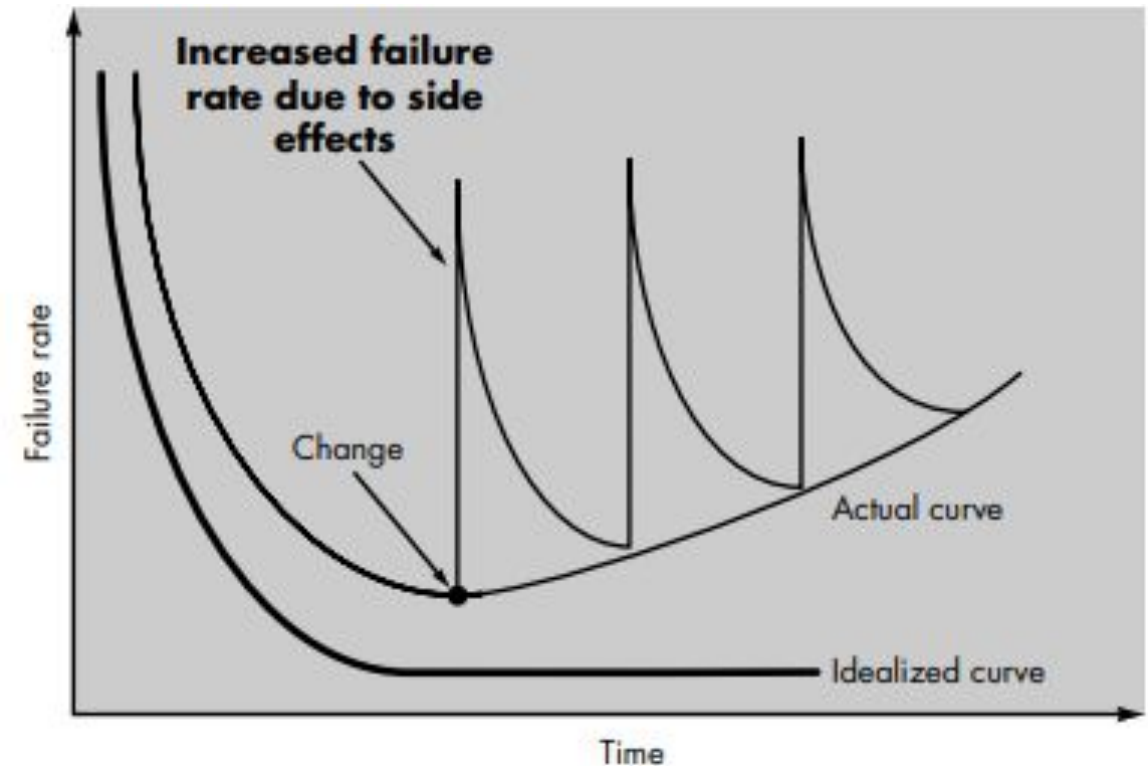
Software engineer is willing to learn the problem domain  
(problem cannot be solved without understanding it first)

# HARDWARE VS SOFTWARE

Failure curve for hardware



Idealized and actual failure curves for software

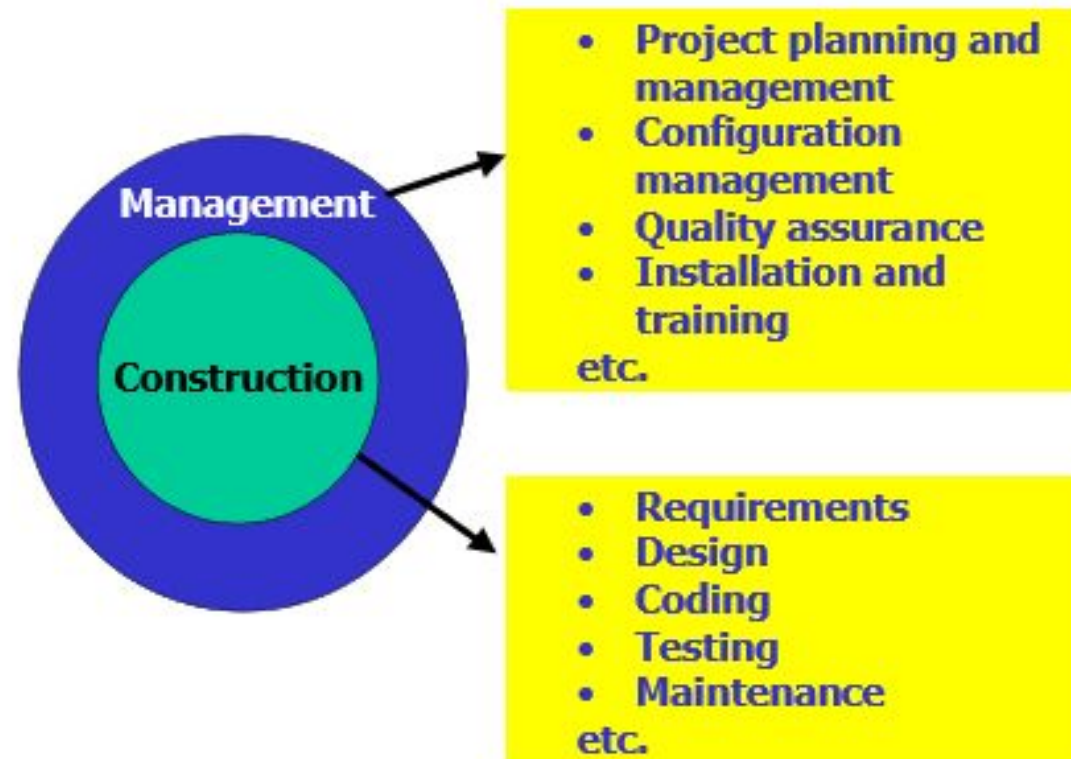




WHAT ARE THE ATTRIBUTES OF GOOD SOFTWARE?



# SOFTWARE DEVELOPMENT



# SOFTWARE ENGINEERING PHASES

- Definition: What?
- Development: How?
- Maintenance: Managing change
- Umbrella Activities: Throughout lifecycle

# DEFINITION

## REQUIREMENTS DEFINITION AND ANALYSIS

Developer must understand

- Application domain
- Required functionality
- Required performance
- User interface

## DEFINITION (CONT.)

- Project planning
  - Allocate resources
  - Estimate costs
  - Define work tasks
  - Define schedule
- System analysis
  - Allocate system resources to
    - Hardware
    - Software
    - Users

# DEVELOPMENT

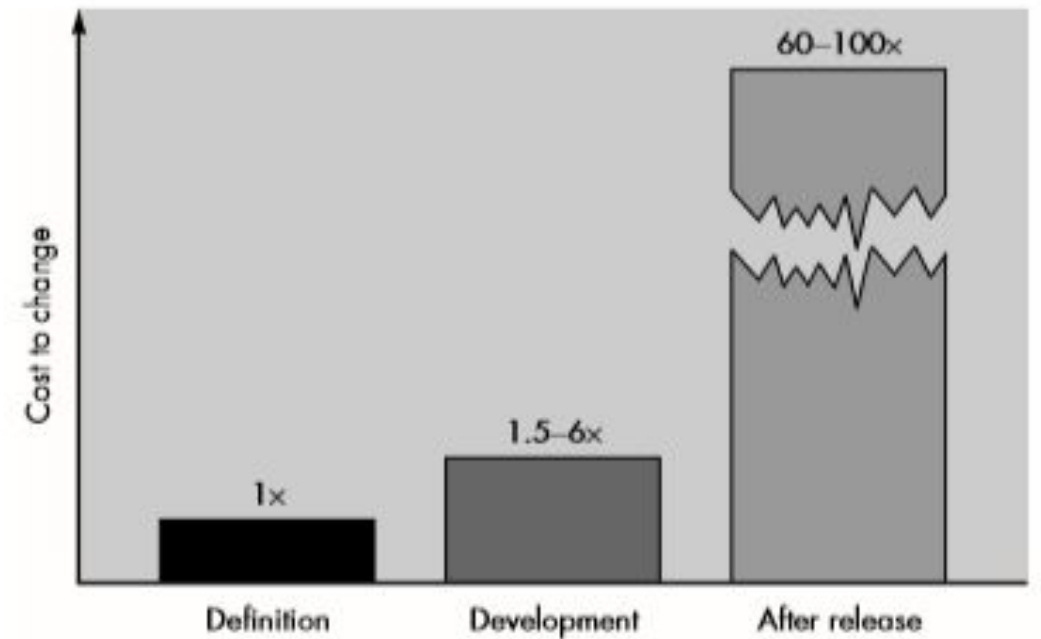
## SOFTWARE DESIGN

- User interface design
- High-level design
  - Define modular components
  - Define major data structures
- Detailed design/Low level Design
  - Define algorithms and procedural detail

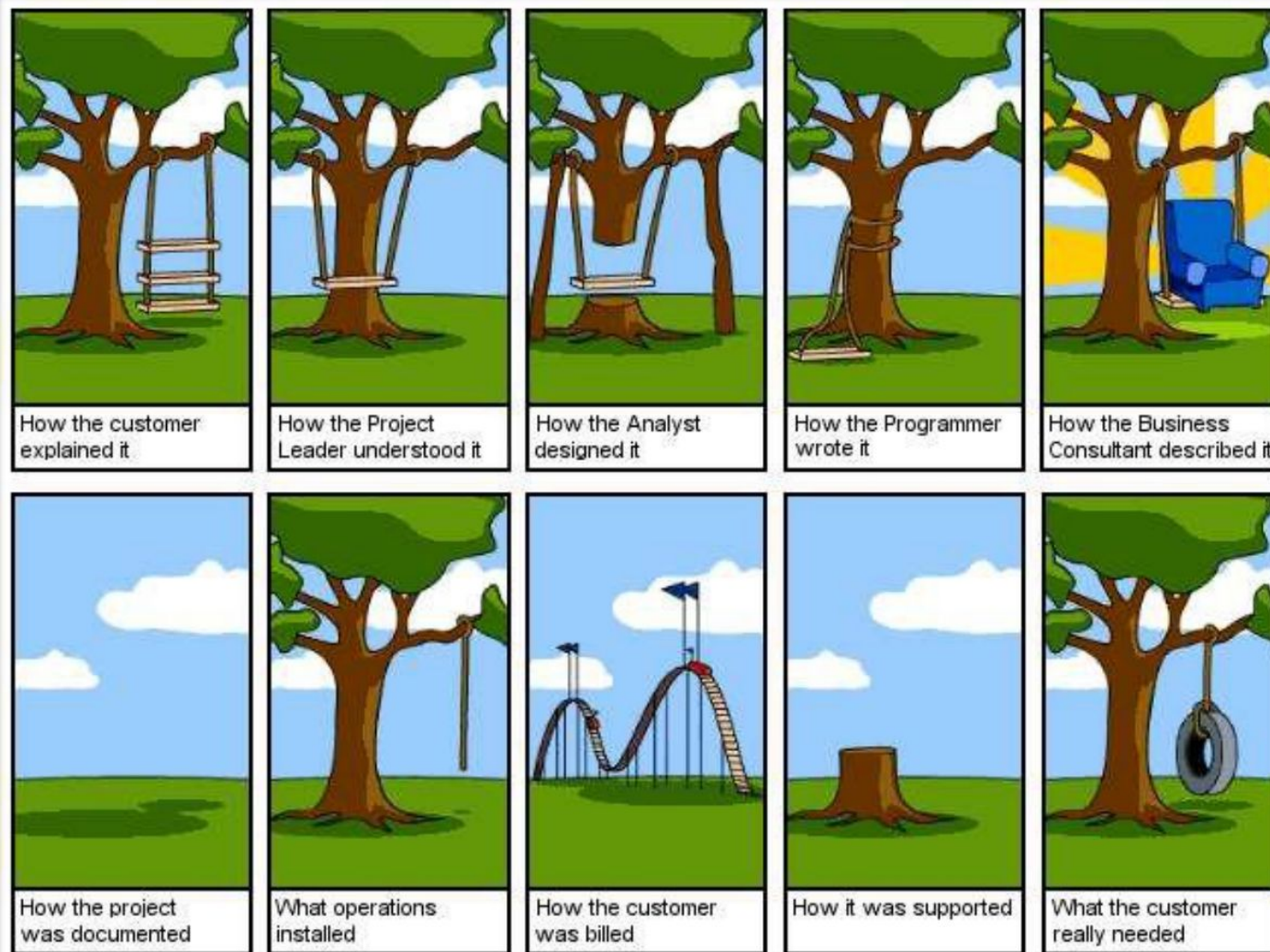
# WHY IS SOFTWARE DEVELOPMENT SO DIFFICULT?

## Changing requirements

- 5 x cost during development
- up to 100 x cost during maintenance



# MAJOR PROBLEMS IN SOFTWARE DEVELOPMENTS





# WHAT IS DESIGN?





# 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.

# 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”.

# SOFTWARE DESIGN - SIMPLIFIED

Requirements specification was about the WHAT the system will do

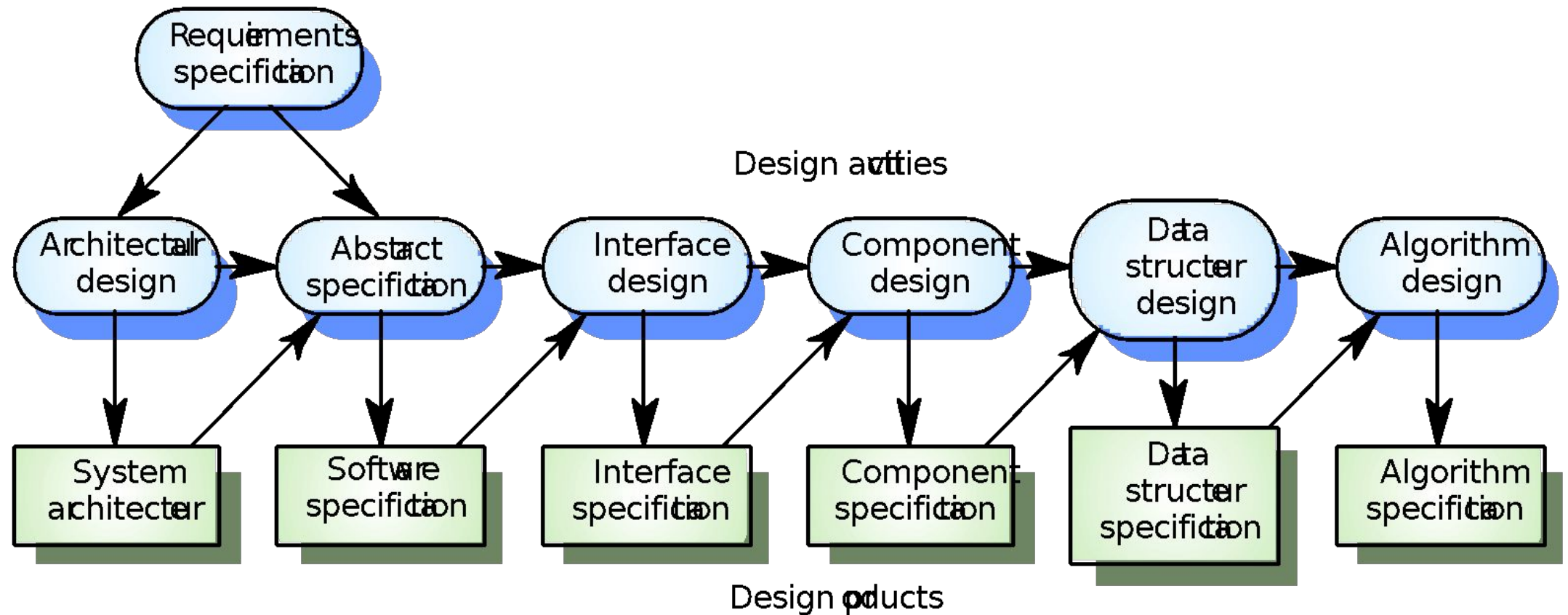
Design is about the HOW the system will perform its functions

- 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.)

# 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

# THE SOFTWARE DESIGN PROCESS



# LEVELS OF SOFTWARE DESIGN

## Architectural design (high-level design)

- architecture - the overall structure, main modules and their connections
- addresses the main non-functional requirements (e.g., reliability, performance)
- hard to change

## Detailed design (low-level design)

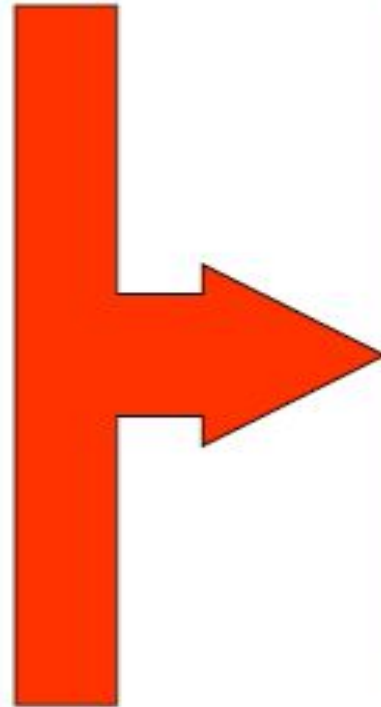
- the inner structure of the main modules
- detailed enough to be implemented in the programming language

# 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.)

# Software Development

- Lists
- Arrays
- Class
- Object
- Procedures
- Functions
- Algorithms
- Etc.



```
//*****
import InventoryItem;
import java.util.StringTokenizer;
import java.io.*;

public class Inventory
{
    // _____
    // Reads data about a store inventory from an input file,
    // creating an array of InventoryItem objects, then prints them.
    // _____
    public static void main (String[] args)
    {
        final int MAX = 100;
        InventoryItem[] items = new InventoryItem[MAX];
        StringTokenizer tokenizer;
        String line, name, file="Inventory.dat";
        int units, count = 0;
        float price;

        try
        for (int scan = 0; scan < count; scan++)
            System.out.println (items[scan]);
        }
        catch (FileNotFoundException exception)
        {
            System.out.println ("The file " + file + " was not found.");
        }
        catch (IOException exception)
        {
            System.out.println (exception);
        }
    }
}
```





## Large-scale, complex software systems...

- Large (Distributed) System
- Many people working on the same problem
- Overly Complex
- Millions of Code...
- Should be delivered on time and within budget!



# Coding only will not do...

- Lists
- Arrays
- Class
- Object
- Procedures
- Functions
- Algorithms
- Etc.



# More programmers...?



---

# Software Architecture

The software architecture of a program or computing system is the structure or structures of the system, which comprise software **components**, the externally visible **properties** of those components, and the **relationships** between them.



**HAVE A GOOD DAY!**