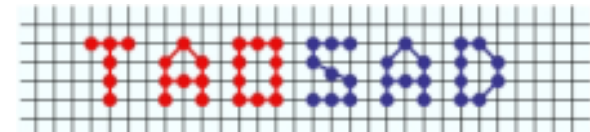


Rationale for Software Architecture



Bedir Tekinerdoğan

University of Twente
Department of Computer Science
Enschede, The Netherlands
e:mail – bedir@cs.utwente.nl
<http://www.cs.utwente.nl/~bedir/>



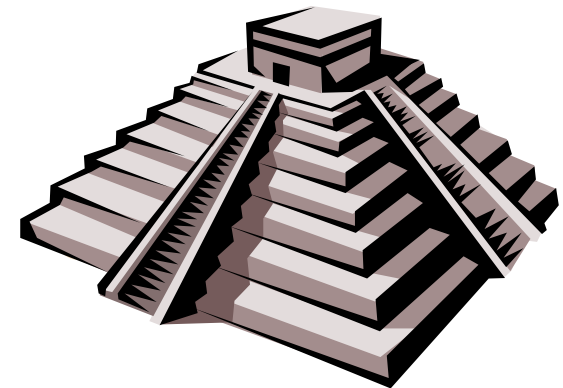
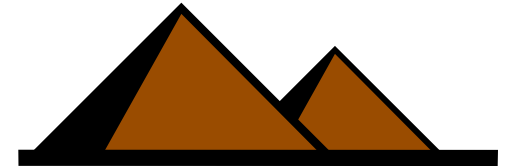
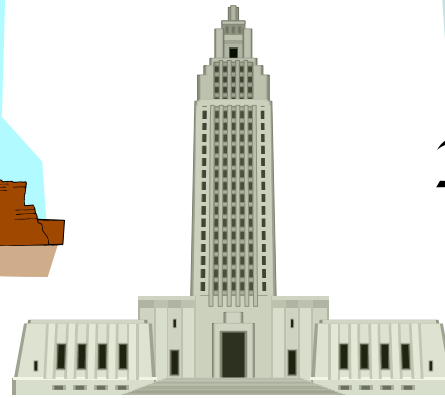
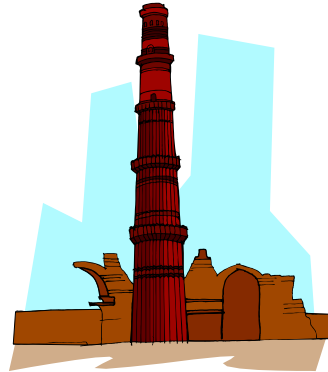
<http://trese.cs.utwente.nl/taosad/>

What is Architecture?

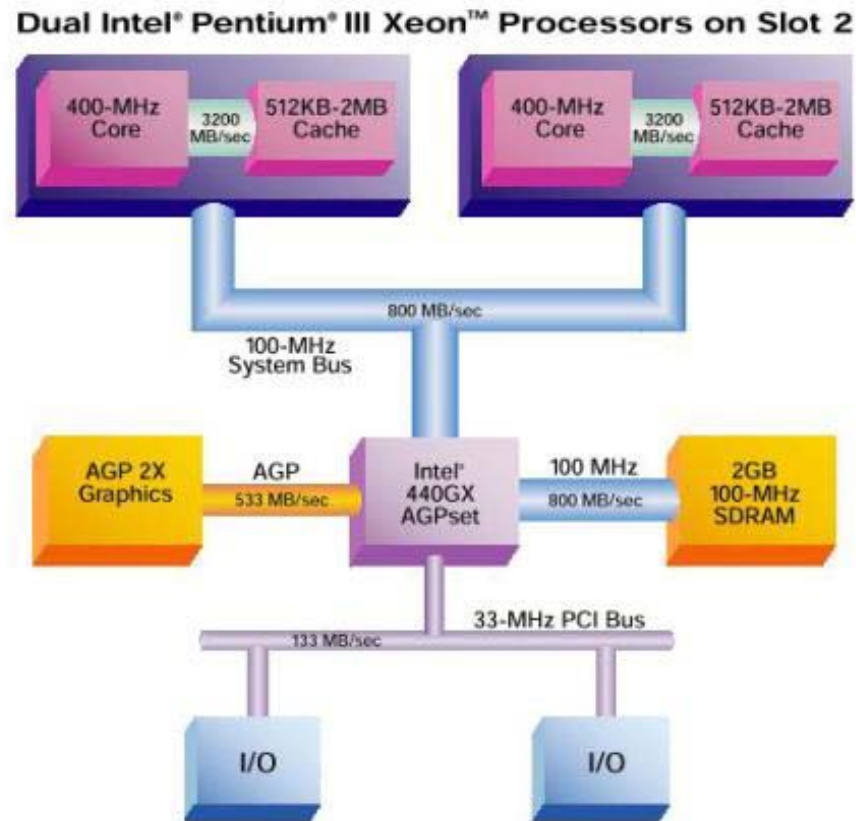
- **Architecture (Webster's):**
 - 1: the art or science of building; specifically: the art or practice of **designing and building structures** and especially habitable ones
 - 2 a>:formation or **construction** as or as if as the result of conscious act <the architecture of the garden> b: a unifying or **coherent form or structure** <the novel lacks architecture>
 - 3: architectural **product** or work
 - 4: a **method** or style of building
 - 5: the **manner** in which the components of a computer or computer system are organized and integrated



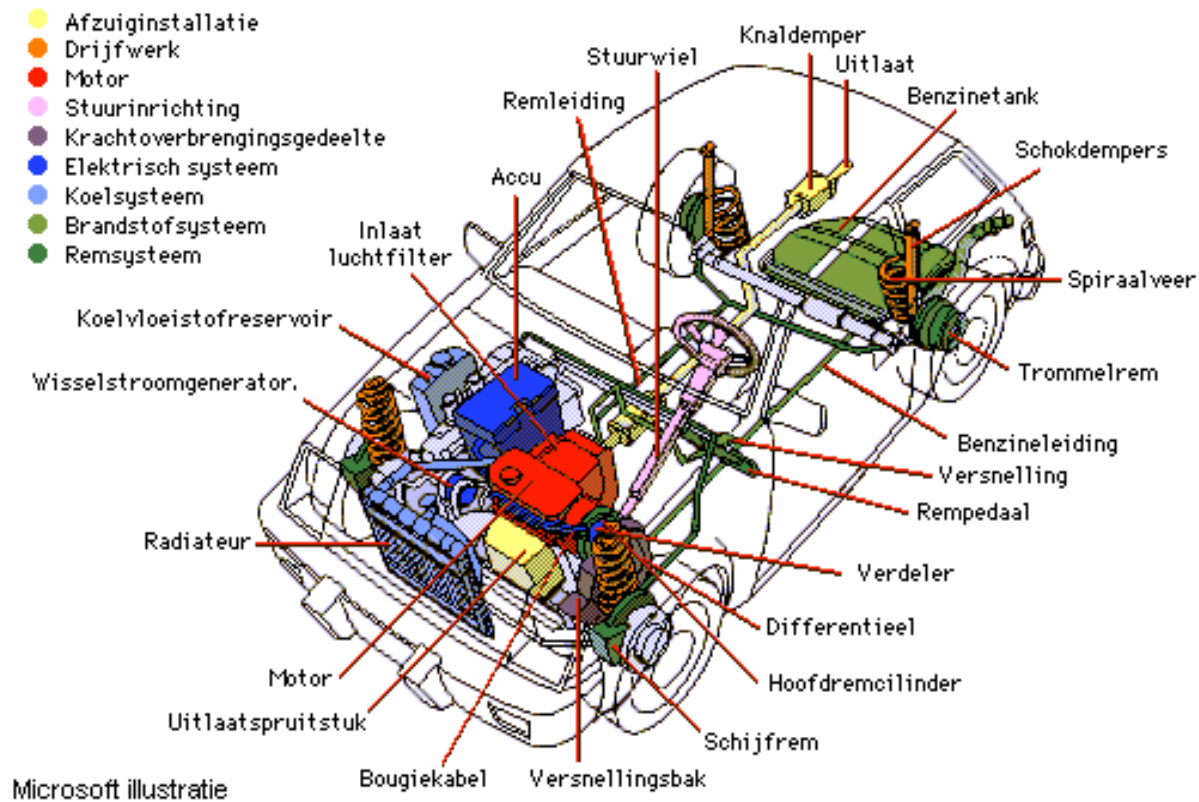
Building Architectures



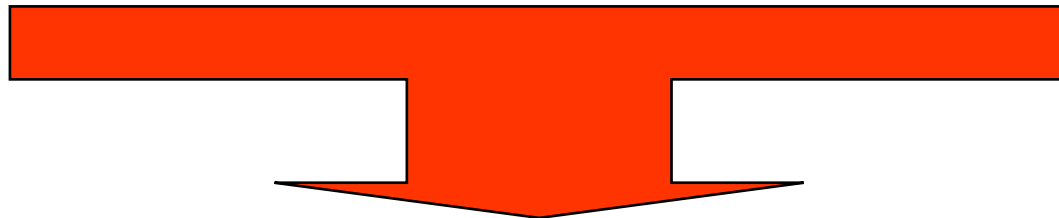
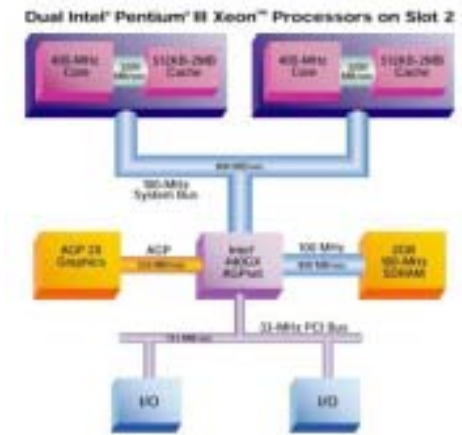
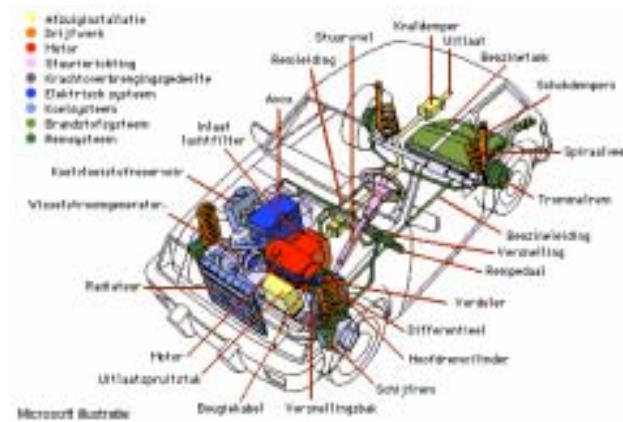
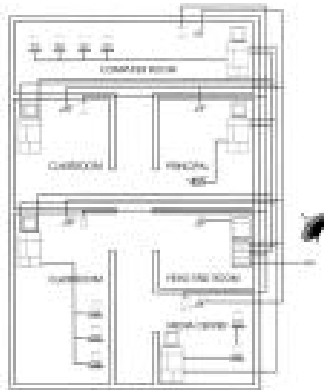
Computer Architecture



Car Architecture



Architecture



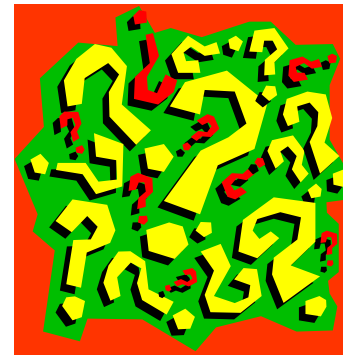
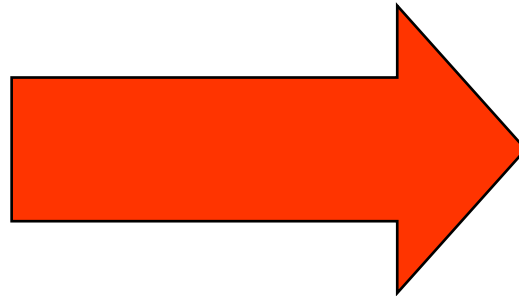
Describes (is) Structure
Consisting of components
which forms the foundation of the whole....

Taj Mahal



Taj Mahal : Built by Mogol Shah Cihan in memory of his beloved wife Mümtaz Mahal 1648

Bricks are not enough...

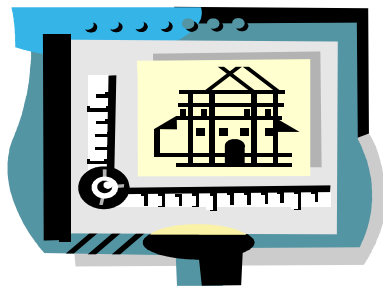
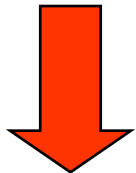


Build me a
palace!

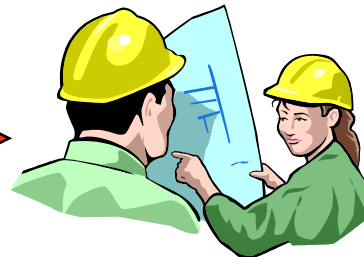
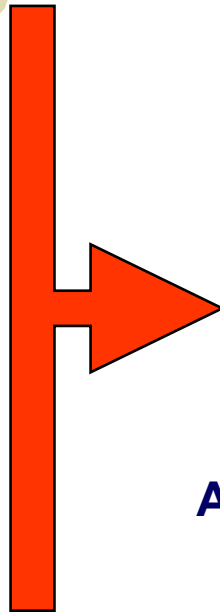
You need an architect first..



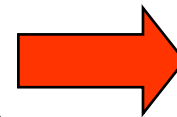
Architecture Design



Architecture



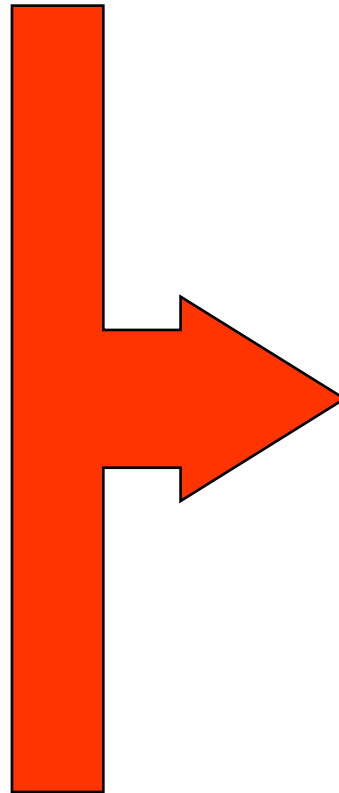
Architecture Realization



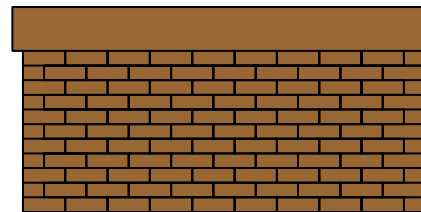
Building

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, complex software systems...

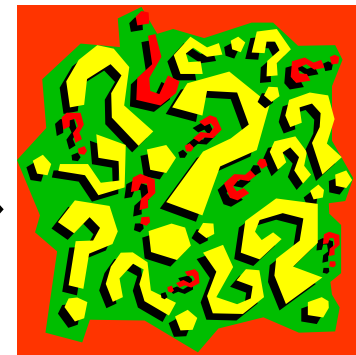
- Large (Distributed) System
- Many people working on the same problem
- Overly Complex
- Millions of Code...
- Should be delivered on time and within budget!



Programming
in the Large

Coding is not enough...

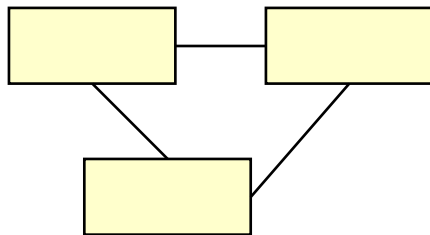
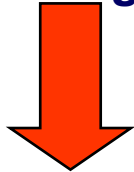
- Lists
- Arrays
- Class
- Object
- Procedures
- Functions
- Algorithms
- Etc.



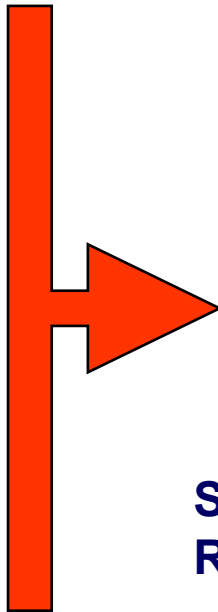
A Software Architect?



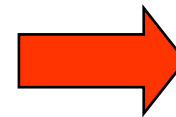
Software Architecture Design



Software Architecture



Software Architecture Realization

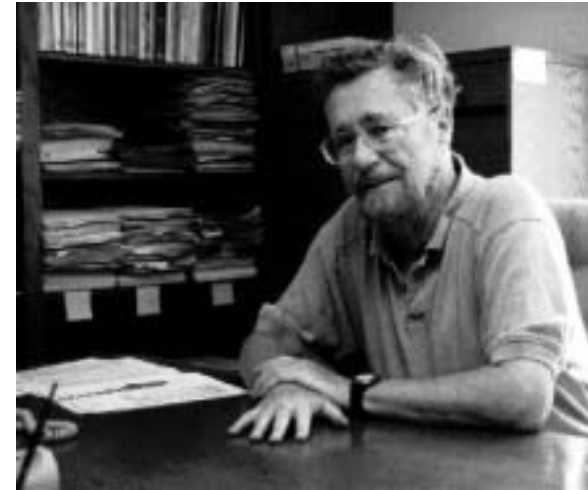


```
.....  
import InventoryItem;  
import java.util.StringTokenizer;  
import java.io.*;  
  
public class Inventory  
{  
    //-----  
    // Reads data about a store inventory from an  
    input file,  
    // creating an array of InventoryItem objects, then  
    prints them.  
    //-----  
    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);  
        }  
    }  
}
```

Application

Does Software have Structure?!

- **Yes!!**
- **Dijkstra, 1968:**
 - “...Correct arrangement of the **structure** of software systems before simple programming...”
- **Parnas, 1972:**
 - “...selected criteria for the decomposition of the system impact the structure of the programs and several **design principles** must be followed to provide a **good structure**...”



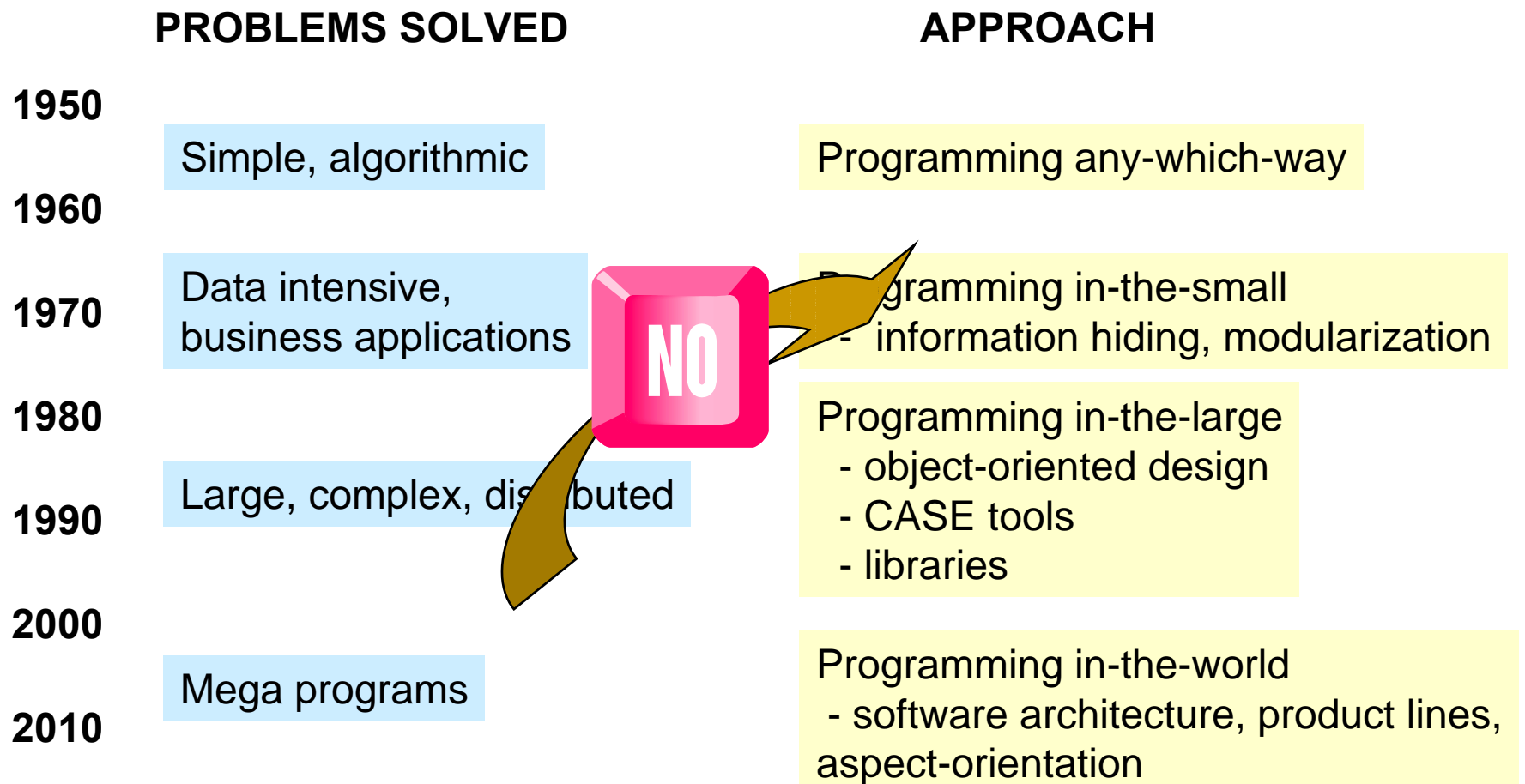
Edsger Dijkstra 1930-2002

Structure in Software

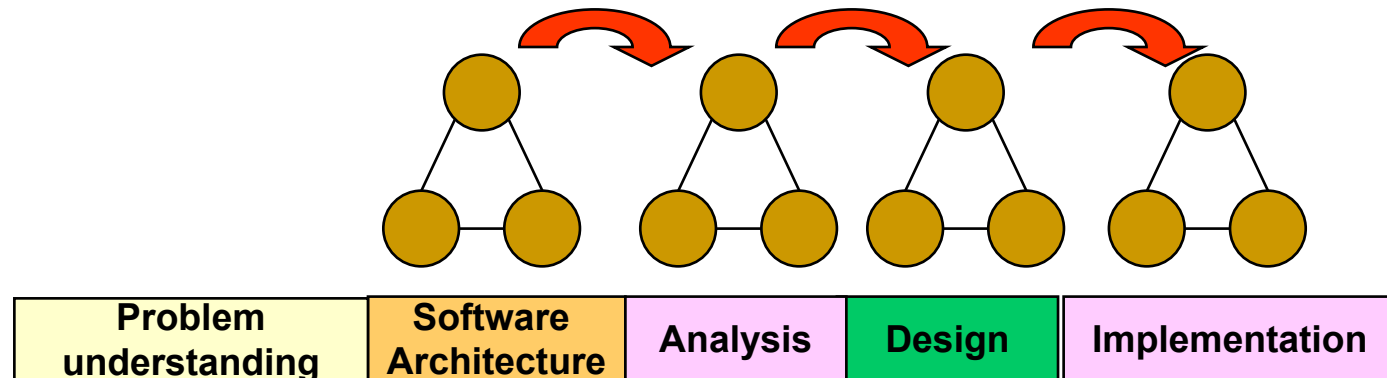
- 1960s - Structured Programming
 - Functional/algorithmic decomposition
- 1970s - Structured Design
 - Methodology/guidelines for dividing programs into subroutines.
- 1980s – Modular programming languages
 - Modular (object-based) programming
 - Grouping of sub-routines into modules with data.
 - Object decomposition
- 1990s – Towards Software Architectures
 - Object-Oriented Analysis/Design/Programming started being commonly used
 - Software Architecture Design
- 2000s- Software Product Line Architectures



Structure in Software



Rationale for Software Architecture



- Architecture is **explicit representation** of global structure of system
- Improves communication among different stakeholders
- Guides analysis and design phases
- Can predict global performance
- Gross level reuse



Summary on Structure in History

- The more complex the problems the more the need to provide an explicit structure.
- Increased consciousness on *structure* of software
- Structure idea did not stop at programming level but moved up to design methods.
- This has logically culminated in *Software Architecture*
- Successfully applied in industry which have specific *architecture design teams/divisions*.
- Active Software Architecture Design community with its own conferences, workshops, publications...
- Is one of the most fundamental concepts in software ***engineering***.