



# Software Design

---

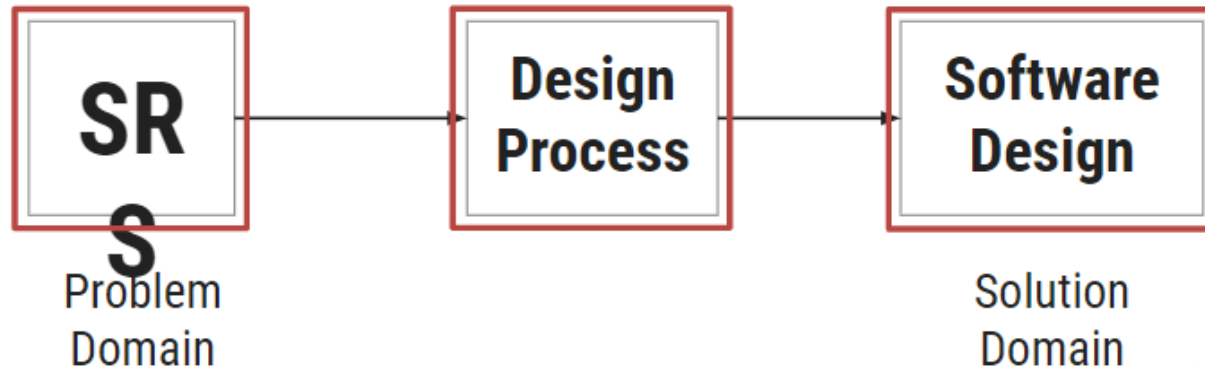
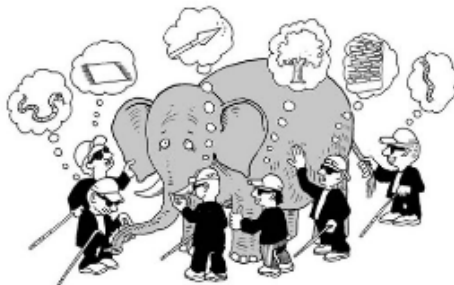
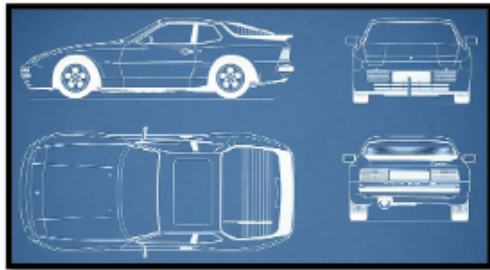
# What is Design?

---

A **meaningful representation** of something to be built

It's a **process** by which **requirements** are **translated** into **blueprint** for constructing a software

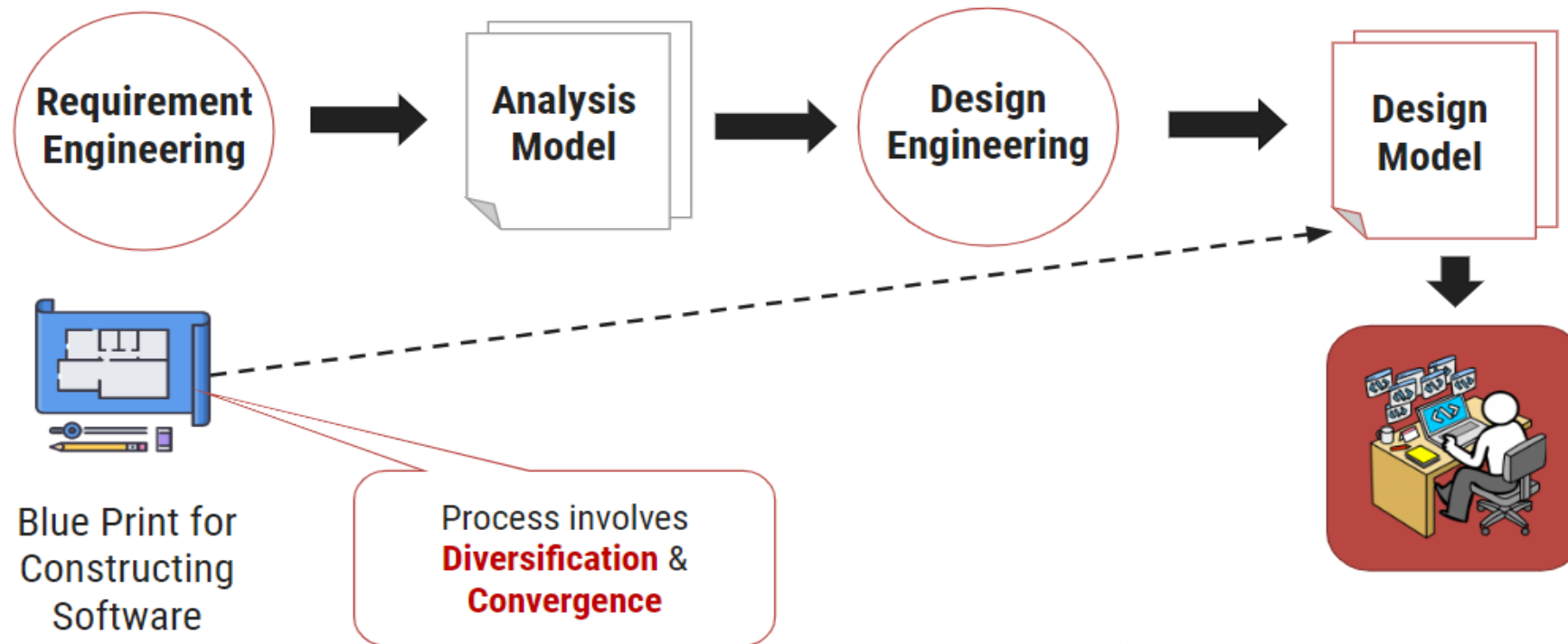
**Blueprint** gives us the **holistic view** (entire view) of a **software**



# Software Design Process

---

It is the **most Creative** part of the development **process**.



# Software design work products

For a design to be easily implemented in a conventional programming language, the following items must be designed during the design phase.

**Different modules** required to implement the design solution.

**Control relationship among** the identified **modules**. The relationship is also known as the call relationship or invocation relationship among modules.

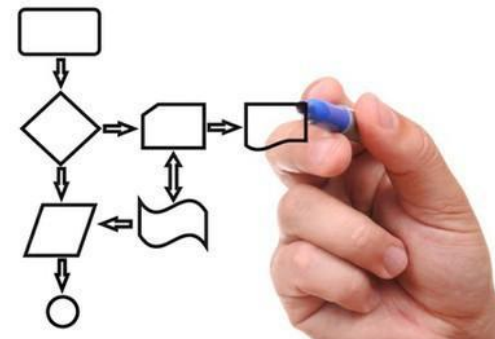
**Interface among** different **modules**. The interface among different modules identifies the exact data items exchanged among the modules.

**Algorithms** required **to implement** each individual **module**.

**Data structures of** the individual **modules**.

# Characteristics of good Design

- ▶ The design must **implement all explicit requirements** available in requirement model
- ▶ The design must **accommodate all implicit requirements** given by stakeholders
- ▶ The design must be **readable & understandable**
- ▶ The good design should **provide complete picture of the software**, addressing the **data, functional** and **behavioral** domains.

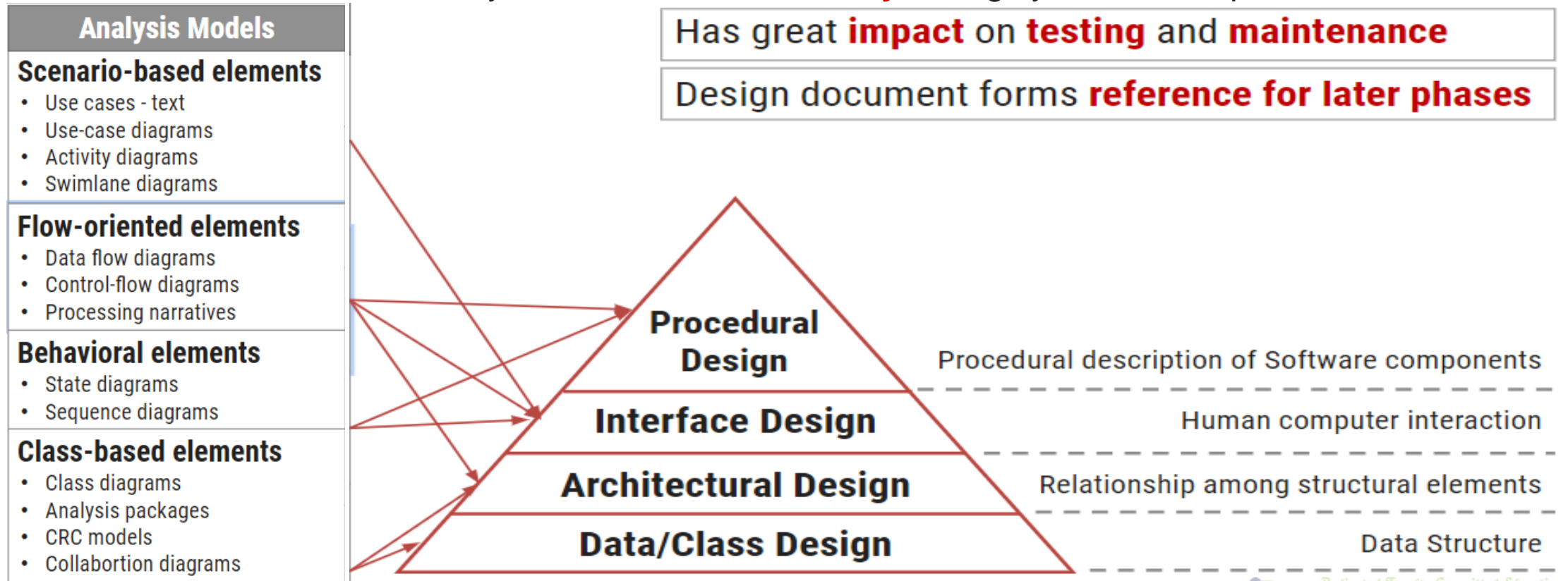


# Design Models

It is **creative** activity. It is most **critical activity** during system development

Has great **impact** on **testing** and **maintenance**

Design document forms **reference for later phases**





# Design Models

## Data Design



It **transforms class models** into **design class** realization and **prepares data structure (data design)** required to implement the software.

## Architectural Design



It **defines** the **relationship between** major **structural elements** of the software

## Interface Design



It defines **how software communicates** with **systems** & with **humans**. An interface implies flow of information & behavior.

## Procedural Design



It **transforms structural elements** of software into **procedural description** of software components

# Quality Attributes of software Design

---

## **F** Functionality

assessed by **feature set** and **capabilities of the program, generality of the functions & security of overall system**

## **R** Reliability

assessed by measuring **frequency & severity of failures, accuracy of outputs, mean-time-of-failure (MTTF), ability to recover from errors**

## **S** Supportability

**Ability to extend program**, adaptability, serviceability, testability, compatibility

## **U** Usability

assessed by considering **human factors, overall aesthetics, consistency & documentations**

## **P** Performance

measured by **processing speed, response time, resource consumption, throughput and efficiency**

# Design Concepts

---

The beginning of **wisdom** for a **software engineer** is to **recognize** the **difference** between **getting program to work** and **getting it right**.

**Fundamental  
software design  
concepts provide  
the necessary  
framework  
for  
“getting it right.”**

Each design concept helps to answer the following questions

1. What criteria can be used to **partition** software into individual **components**?
2. How is **function** or **data structure** detail **separated** from a **conceptual representation** of the software?
3. What uniform criteria **define the technical quality** of a software design?



# Architectural Design

---

## Software Architecture & Design

- ▶ **Large systems** are **decomposed** into **subsystems**
- ▶ **Sub-systems** provide **related services**
- ▶ Initial design process includes
  - Identifying sub-systems
  - Establishing a framework for sub-system control and communication

## Why to document the Architecture?

- ▶ **Stakeholder Communication:** High-level presentation of system
- ▶ **System Analysis:** Big effect on performance, reliability, maintainability and other -ilities (**Usability, Maintainability, Scalability, Reliability, Extensibility, Security, Portability**)
- ▶ **Large-scale Reuse:** Similar requirements similar architecture



# Software Architecture & Design

---

Architectural design represents the **structure of data and program components**

## It considers

**Architectural style** that the system will take

**Structure** and **properties** of the **components** that constitute the system

**Interrelationships** that occur among all architectural components of a system

- ▶ Representations of software architecture are an **enabler for communication between all parties** (stakeholders).
- ▶ Architecture “constitutes a relatively small, intellectually graspable model of how the system is structured and how its components work together”

# Architectural Style

---

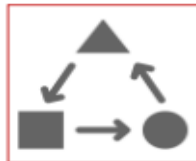
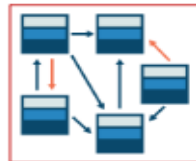
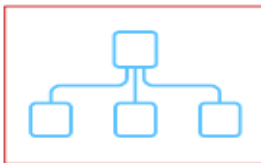
Data-centered architecture style

Data-flow architectures

Call and return architecture

Object-oriented architecture

Layered architecture



Each style describes a system category that encompasses

A **set of components** (Ex., a database, computational modules) that **perform a function** required by a system.

A set of **connectors** that **enable “communication, coordination and cooperation”** among components.

**Constraints** that define **how components can be integrated** to form the system.

**Semantic models** that enable a designer **to understand the overall properties** of a system.