- Most of the Systems in the world are inherently complex in nature
- A personal computer is a device of moderate complexity
- It is composed of a CPU, memory, a monitor, a keyboard, and some sort of secondary storage device
- CPU : an ALU, registers and a bus to which peripheral devices are attached.
- Each of these parts may in turn be further decomposed into elements, such as NAND gates, flip-flops, and so on.
- Here we see **the hierarchic nature** of a complex system.

- A personal **computer functions** properly only because of

  ->the collaborative activity of each of its major parts.

- **We can design** a computer that works only because we can decompose it into parts that we can study separately.
- We may study the operation of different parts (monitor , keyboard, ALU etc.) independently
- **Not only are complex systems hierarchic, but the levels of this hierarchy represent different levels of abstraction**
- At each level of abstraction, we find a collection of devices that collaborate to provide services to higher layers.

The structure of social institutions

- Groups of people join together to accomplish tasks that cannot be done by individuals.
- As organizations grow larger, we see a distinct hierarchy emerge.
- Multinational corporations contain companies, which in turn are made up of divisions, which in turn contain branches, which in turn encompass local offices, and so on
- The boundaries among these parts may change, and over time, a new, more stable hierarchy may emerge.

- The relationships among the various parts of a large organization are just like relationships between the components of a computer
- The degree of interaction among employees within an individual office is greater than that between employees of different offices
- But different levels are **unified by common mechanisms.**
- A mail clerk and the chief executive officer of the company

Why Software Is Inherently Complex [1]

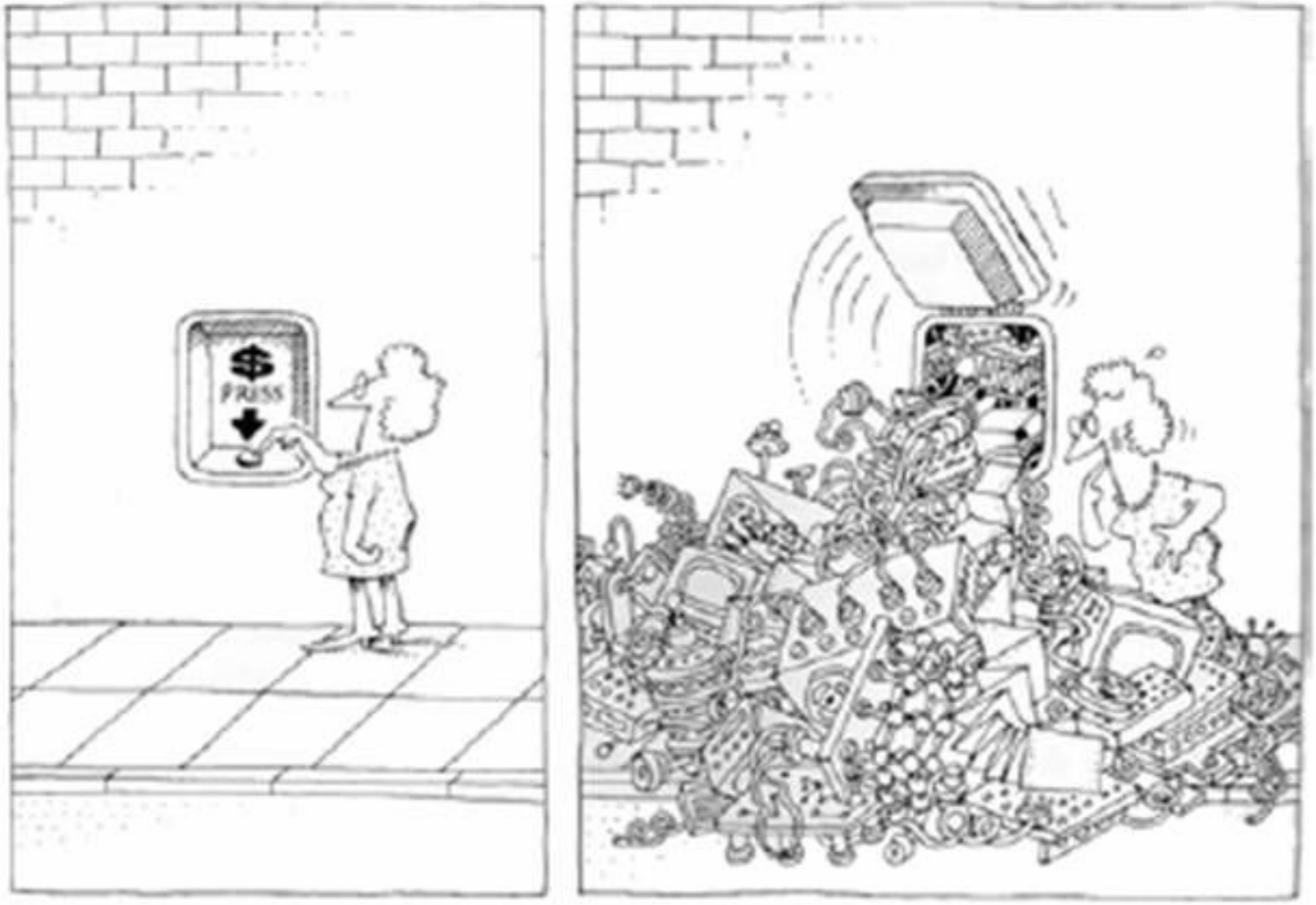
- ***The Complexity of the Problem Domain***
  - ✓ Communication gap" between the users of a system and its developers
  - ✓ Lack of instruments for precisely capturing user requirements.

What is the IEEE recommended practice for Software Requirement Specifications (SRS)?

- ✓ The common way to express requirements is with large volumes of text, occasionally accompanied by a few drawings.
- ✓ Such documents are difficult to comprehend
- ✓ Are open to varying interpretations, and
- ✓ Often contain elements that are designs rather than essential requirements.
- ✓ **Requirements of a software system often change during its development**

The Difficulty of Managing the Development Process

- ✓ The fundamental task of the software development team is to engineer the illusion of simplicity



The Difficulty of Managing the Development Process

- ✓ Most delivered systems contain thousands or even millions of lines of code
- ✓ No one person can ever understand such a system completely.
- ✓ Use a team of developers
- ✓ Often team is geographically dispersed
- ✓ Communication and coordination among team members and between teams
- ✓ With a team of developers, the key management challenge is always to maintain a unity and integrity of design.

- ***The Flexibility Possible through Software***
  - ✓ Craft all the primitive building blocks
  - ✓ Very few standards exist in the software industry …

- ***The Problems of Characterizing the Behavior of Discrete Systems***
  - ✓ Predicting the path of a ball tossed into the air
  - ✓ under normal conditions (certain laws of physics apply)
    - ✓ It will never happen that just because we threw the ball a little harder, halfway through its flight it suddenly stopped and shot straight up into the air
  - ✓ In a not-quite-debugged software simulation of this ball's motion, this kind of behavior can easily occur

- ***The Complexity of the Problem Domain***
- ***The Difficulty of Managing the Development Process***
- ***The Flexibility Possible through Software***
- ***The Problems of Characterizing the Behavior of Discrete Systems***

The Five Attributes of a Complex System

- ✓ **Hierarchic Nature**

  All systems have subsystems and all systems are parts of larger systems. . . . The value added by a system must come from the relationships between the parts, not from the parts per se

- ✓ **Relative primitives**

  The choice of what components in a system are primitive is relatively arbitrary and is largely up to the discretion of the observer of the system

The Five Attributes of a Complex System

- ✓ **Separation of concerns**
  - ✓ Intra-component linkages are generally stronger than inter-component linkages
  - ✓ This difference between intra- and inter-component interactions *makes it possible to* study each part in relative isolation
- ✓ **Common Patterns**
  - ✓ Hierarchic systems are usually composed of only a few different kinds of subsystems in various combinations and arrangements

The Five Attributes of a Complex System

- ✓ **Stable Intermediate forms**
  - ✓ complex systems will evolve from simple systems much more rapidly if there are stable intermediate forms than if there are not