**SOFTWARE ENGINEERING APPROACH**

# SOFTWARE ENGINEERING APPROACH

The model used for the development of this software is waterfall model or software life cycle. The principal stages of the model map onto fundamental development activities is shown in the figure 3.1 as,

Requirements definition

System and software design

Implementation and unit testing

Integration and system testing

Operation and maintenance

Figure 3.1 Software life cycle

## Requirement Analysis and definition

The system’s services, constraints and goals are established by consultation with system users. They are then defined in detail and serve as a system specification.

### Software requirement specification

The requirement for a system is the descriptions of the services provided by the system and its operational constraints. The process of finding out, analyzing, documenting and checking these services and constraints is called requirement engineering. The requirement engineering process can be separated as user requirements to means high level abstract requirements and system requirements to means the detailed description of what the system should do.

### User Requirements definition

The security problem is considered here as the problem of keeping communication over the network private. In other words, a secure network allows only the intended recipient to intercept and read a message addressed to her/him. Thus, protection of information is required against possible violations that can compromise its secrecy (or confidentiality). Secrecy is compromised if information is disclosed to users not authorized to access it.

### System Requirement Specification

The system’s services, constraints and goals are established by consultation with system users. They are then defined in detail and serve as a system specification.

## System and Software design

The systems design process partitions requirements to either hardware or software systems. It establishes overall system architecture. Software design involves identifying and describing the fundamental software system abstraction and their relationships.