Requirements

Starting a software development project involves determining what the customer requires in terms of software functionality. The hardest single part of building a software system is deciding what to build.

Requirements are the basis of all software applications as they focus on the user tasks that the software must support. Requirements are gathered/collected/elicited from the customer to outline what he or she wishes the software to do, and then the requirements are recorded in a way that the customer understands.

- The **functionality** of the proposed software is expressed as **requirements**. How do developers discover the requirements that the software must support? – They engage in an activity known as *Requirements Gathering.*
- Requirements gathering is challenging – it requires collaboration of groups of people with different backgrounds.
- The client and users are experts in their application domain and have a general idea of what the software should do, but they often have no experience in building software applications.
- Need to use developers with good inter-personal, and psychology skills for requirements gathering.

Requirements Gathering

Requirements Gathering often begins with the developers receiving a problem statement for the customer/client.

Example – Problem Statement

We need a bank application that allow bank customers to open a bank account; deposit money into, and withdraw money from, the bank account; and various other things.

Loose, informal, unstructured language only gives the “big picture”. Missing details, and probably some extra, unstated requirements.

Requirements Gathering – The problem statement is usually not good enough to do any serious development work. -> Developers must determine more precisely/accurately the requirements that the proposed software needs to deliver.

Examples of requirements which are not verifiable:

1) The software will have a good user interface. (Good is not defined)
2) The software will be error free. (The software will be error free)
3) The software will respond to the user within 1 second for most cases. (“Most cases” is not defined)

Do not have any requirements like this!

Categories of Requirements

1) Functional Requirements (FR), describe the specific functions or tasks that the software should perform. They describe the intended usage of the software – the services, reactions, and the behavior of the software to user input.

2) Non-functional requirements (NFR) specify conditions that the developed software needs to conform to. (security, privacy, usability, reliability, availability and performance issues). Etc.

3) Constraints are specific NFRs that restrict the development of the system.