Introduction

Lecture 01

What to do today

Question of the day

Can you walk on the water?

Think and discuss at the end of the lecture

This week

- **Today**
- Aims & Intro to course
- Syllabus
- Reading
- What are requirements?

This week

- **Next Class**
- Discussion
- Assignment due

Introduction

- What should we learn in this course about RE
  - The role of requirements in software development
  - The system development view, the user view, and their intersection
  - How to collect report and manage requirements
  - What is the state of the art of RE research
  - User’s and developer’s point of view

Organization

- The course contains
  - Working in group of 2-4 Students
  - Reviewing the work – of others and giving feedback
  - Surveying literature
  - Presenting in seminars
  - Understanding that there is no single right way

Communication

- **Communication**
  - The best place to discuss the course is during lecture times.
  - The main communication tool for the class will be gmail group.
  - Students will be expected to read new postings on a daily basis and post questions, answers, and comments.
  - All assignments will be placed on the course folder and collectively handover to instructor through course senior web site and/or announced in lecture.
  - Most class materials are available on the course folder; be sure to check regularly.
  - Email will only be used for special circumstances, such as communicating time sensitive information.

Workshop

- Collective effort
  - Need to organize a one day workshop during 8th week of the semester
  - Organize as:
    - Invitation Team
    - Organizing Team
    - Publicity Team
    - Research Team
    - Administrative & Fund generation Team
  - Names must reach to instructor by 2 Jul 1700 hrs

Quizzes

- Unannounced
  - Element of surprise
  - Four - Five quizzes of 10-15 min duration
  - Close book, notes, discussion etc
  - No makeup quiz, no best of, class average etc
  - No fixed frequency
- Total 10 grade points

Late Submission Policy

- **Late Work Policies**
- A software engineer has a responsibility to manage time effectively and turn in work on time
  - If you are having a problem, discuss it as early as possible before the due date.
  - As a rule of thumb
    - Homework assignments are due at 1700 hrs on the date specified.
    - Late submission will be given a maximum of half credit and accepted only until the next class period following their due date.

Tentative Schedule

- Course outline
  - Generally followed subject to some minor changes
  - link

Grading Key

- 2 projects 10%
  - A - Literature survey on a given RE topic
  - B - Requirements specification for a given system
    - Presentation of A, compulsory, 5%
    - Presentation of B, compulsory, 5%
      - Incl feedback on B of some other group, 5%
- Home Assignments weekly X tasks, 10%
- 3 -4 Surprise quizzes 10%
- Mid semester Exam hard 30%
- Final exam – very hard, 40%

Background

- Development of failed software since 60s:
  - Systems being delivered late over budget
  - They don't really do what user wanted to
  - Never been used to their full effectiveness by people
- Reason ?
  - No single reason / single solution to the problem
  - Major contributory factor – " Difficulties with system requirement"
- System requirement
  - What the system is required to do and the circumstances under which it is required to operate
  - A requirement is a necessary attribute in a system

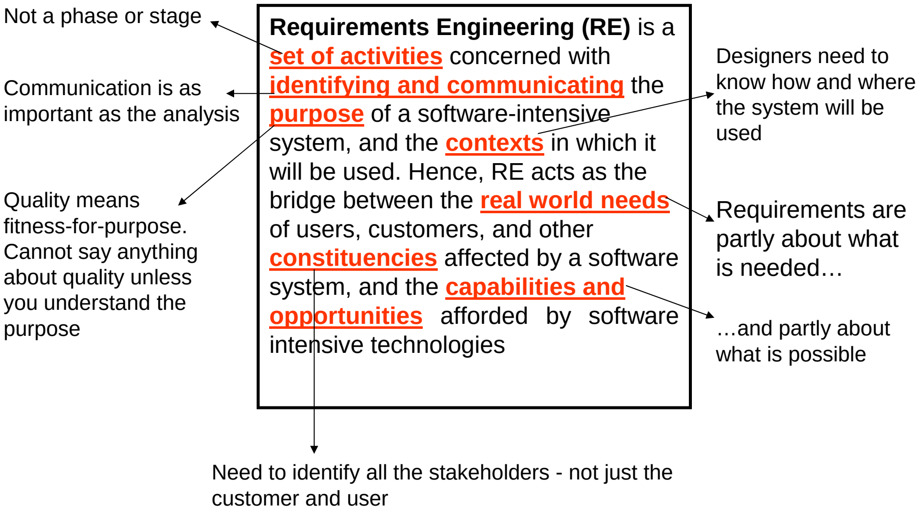
Requirements

- What are requirements?
  - A statement of a system service or constraint
  - Defined early as specification of
    - what should be implemented
    - Description of how the system should behave, domain information, constraints on system operation
  - So the requirement might describe
    - A user level facility - spell checker and correction
    - A very general system property - personal information disclosure
    - A specific constraint on system - sensor must be polled 10 times a second
    - How to carry out some computation - some formula

What Vs How

- Some suggests - requirement should always be statement of “WHAT” a system should do and not “HOW” it should do.
  - – What does a system do? (requirements)
  - – What is its structure? (architecture)
  - – How does a system behave? (requirements)
  - – How is it structured? (architecture)

RE Definition



Requirements Engineering (RE) is a **set of activities** concerned with **identifying and communicating** the **purpose** of a software-intensive system, and the **contexts** in which it will be used. Hence, RE acts as the bridge between the **real world needs** of users, customers, and other **constituencies** affected by a software system, and the **capabilities and opportunities** afforded by software intensive technologies

Not a phase or stage

Communication is as important as the analysis

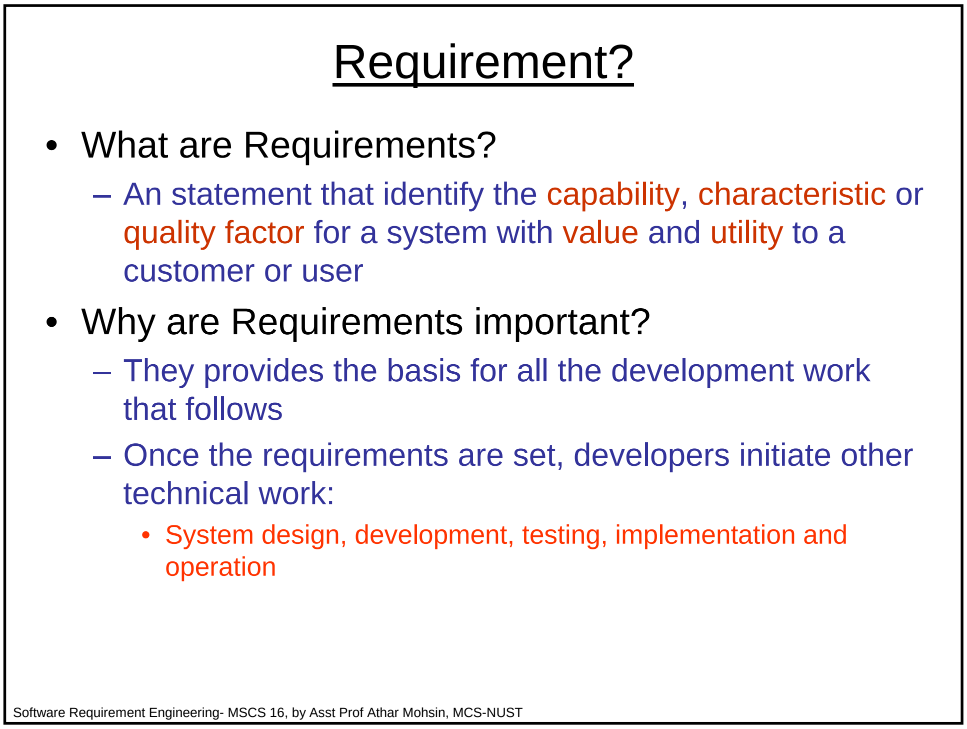
Quality means fitness-for-purpose. Cannot say anything about quality unless you understand the purpose

Designers need to know how and where the system will be used

Requirements are partly about what is needed…

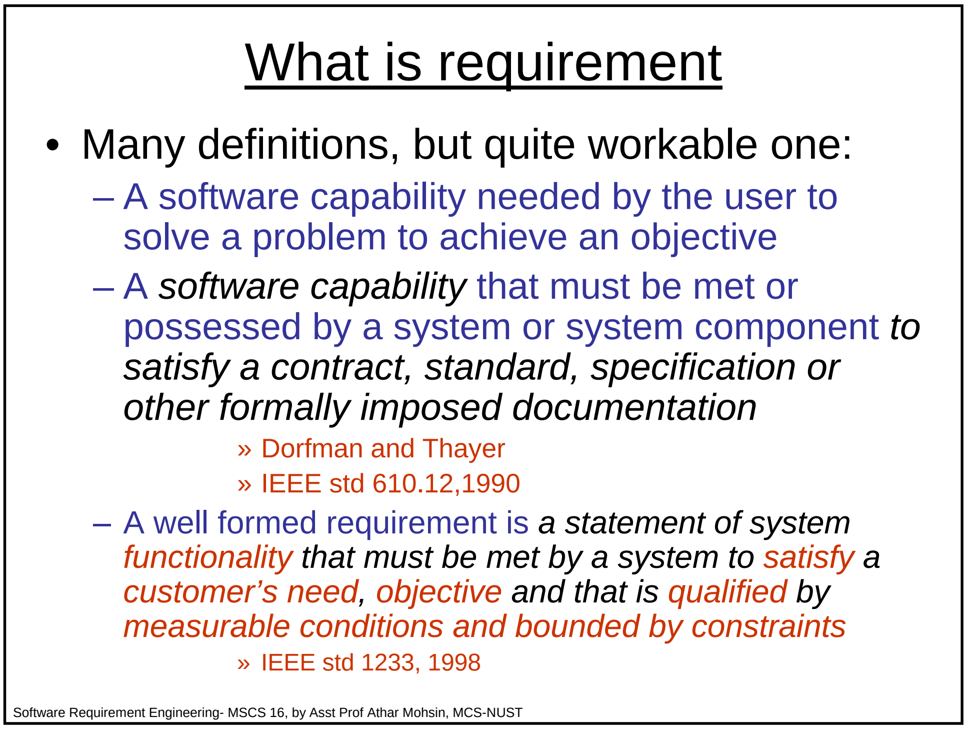
…and partly about what is possible

Need to identify all the stakeholders - not just the customer and user



Requirement?

- What are Requirements?
  - An statement that identify the capability, characteristic or quality factor for a system with value and utility to a customer or user
- Why are Requirements important?
  - They provides the basis for all the development work that follows
  - Once the requirements are set, developers initiate other technical work:
    - System design, development, testing, implementation and operation



What is requirement

- Many definitions, but quite workable one:
  - A software capability needed by the user to solve a problem to achieve an objective
  - A *software capability* that must be met or possessed by a system or system component *to satisfy a contract, standard, specification or other formally imposed documentation*
    - » Dorfman and Thayer
    - » IEEE std 610.12,1990
  - A well formed requirement is *a statement of system functionality that must be met by a system to satisfy a customer's need, objective and that is qualified by measurable conditions and bounded by constraints*
    - » IEEE std 1233, 1998

Requirement, a definition

- Definitions vary, but generally along these lines:

  A capability that the system must deliver or a condition that it must satisfy in order to address a need of a stakeholder.

  (Adopted from [Larman, 2002])

> • **Requirements** are what is wanted
> • **Engineering** is calculated manipulation (*Webster's*).
>
> So RE is calculated manipulation of what is wanted?

Identifying Problem

Boilerplate

Boundaries • **Which** problem needs to be solved?

• **Where** is the problem? Problem Domain

Stakeholder • **Whose** problem is it?

• **Why** does it need solving? Stakeholders Goal

Scenarios • **How** might a software system help?

• **When** does it need solving? Development Constraints

Feasibility • **What** might prevent us from solving it?

Requirement Classification

- Requirements are commonly classified as (IEEE std 830, 1998): (Need to be discussed in next class)
  - Functional:
    - A requirement that specifies an action that a system MUST be able to perform, without considering physical constraints
      - A requirement that specifies input/output behavior of the system
  - Non-Functional:
    - A requirement that specifies system PROPERTIES, such as environmental and implementation constraints, performance, dependencies, maintainability, extensibility and reliability.
    - Often classified as:
      - Performance Requirements
      - External interface requirements
      - Design constraints
      - Quality attributes

Requirements problems

- Many system engineering problem stem from problems with the system requirement
- Common problems are:
  - The requirements don’t reflect the real needs of the customer for the system.
  - Requirements are inconsistent and/or incomplete.
    - The system shall allow users to search for an item by title, author, or by ISBN? – is the requirement incomplete
  - It is expensive to make changes to requirements after they have been agreed.
  - There are misunderstandings between customers, those developing the system requirements and software engineers developing or maintaining the system.

Requirement Dilemma

- Significant Difference – Asking & Accepting
- Often huge difference – stated & real requirement
  - Stated Requirement
    - Those provided by a customer at the beginning of software development effort
      - Example:
        - in a request for information, proposal or quote
        - In statement of work - SOW
  - Real Requirement
    - Those that reflect the verified needs of user for a particular system
      - Identification is interactive and Iterative requirement process
- Analysis of stated requirement helps to determine
  - The refine real needs and expectations of the user from the delivered system

Requirement management

- Requirement define
  - capabilities that the system must deliver
  - Conformance or lack of conformance to a set of requirements to determine the success of failure of the project
- To find what requirements are:
  - Write them down
  - Organize them
  - Track them if these change

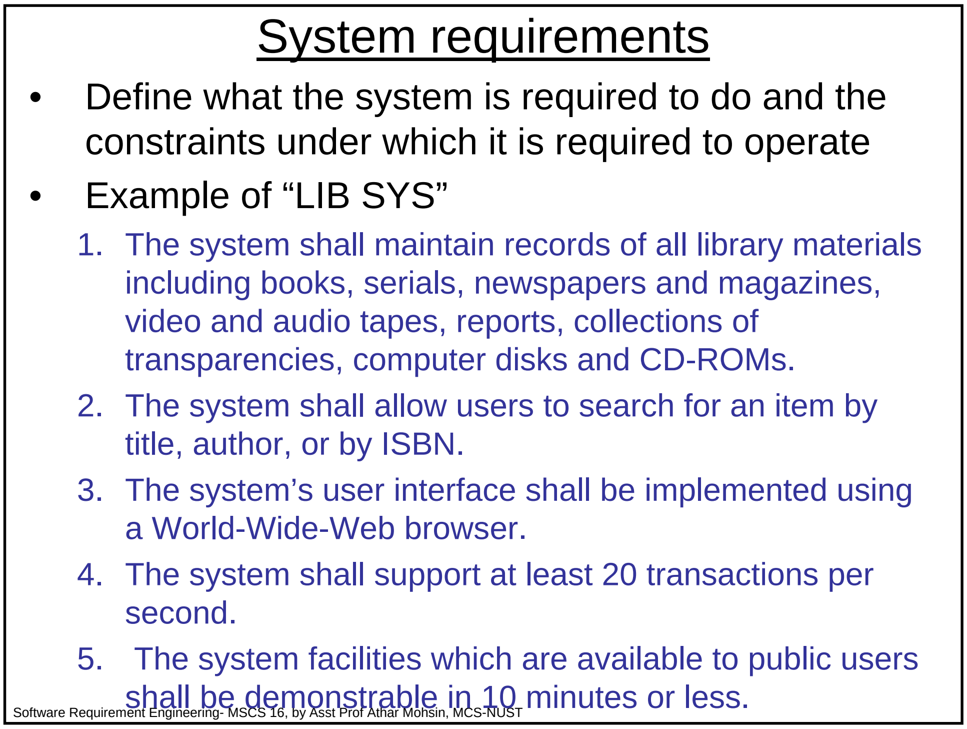
Requirements Management Definitions

- Requirements Management
  - The purpose is to:
    - establish a common understanding between the customer and the customer’s requirements that will be addressed through the software project.

- Requirement Management
  - “A systematic approach to:
    - **eliciting**, **organizing**, and **documenting** the requirements of the system, as well as a **process** that establishes and **maintains agreement** between the customer and the project team on the changing requirements of the system.”
      - » [Leffingwell and Widrig, 2003]

System Requirement

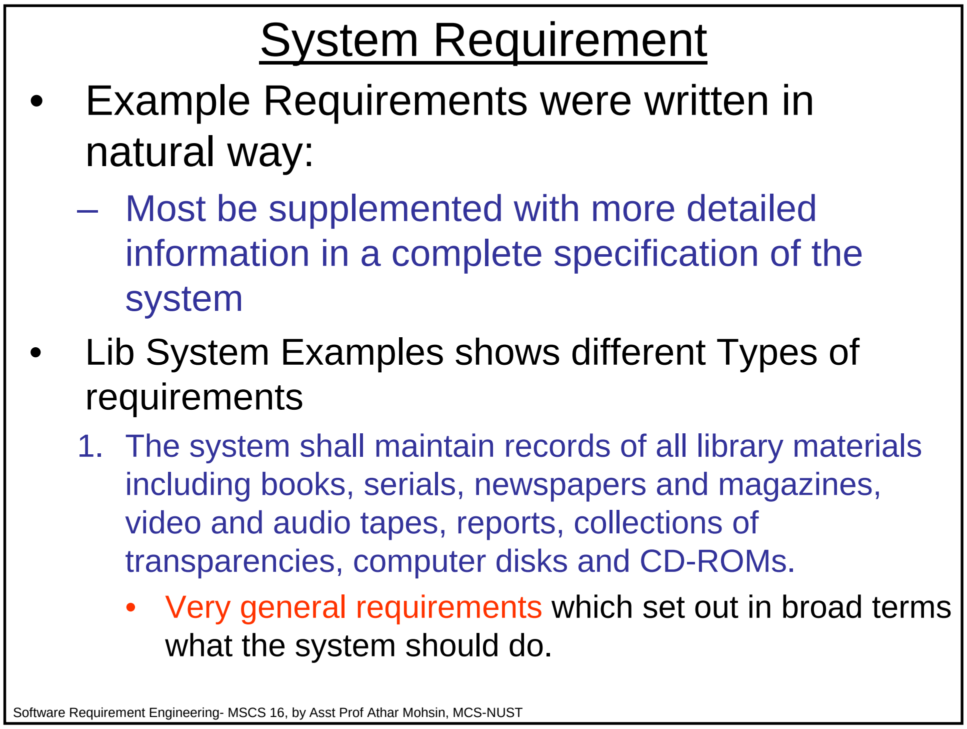
- Software Development faces “PROBLEMS”
  - System Requirement defines:
    - What the system is required to do and the circumstances under which it required to operate
  - Requirement Defines:
    - The services that the system should provide and they set out constraints on the system’s operation
- Tendency to want to start real work – Programming
  - Quick start indicates that the progress is being made
    - Insufficient time and effort spent on requirement related activity
- Better approach
  - Invest more time in requirement gathering, analysis and management activities
    - Additional time is needed to identify real requirement and plan requirement related activities



System requirements

- Define what the system is required to do and the constraints under which it is required to operate
- Example of “LIB SYS”

1. The system shall maintain records of all library materials including books, serials, newspapers and magazines, video and audio tapes, reports, collections of transparencies, computer disks and CD-ROMs.
2. The system shall allow users to search for an item by title, author, or by ISBN.
3. The system’s user interface shall be implemented using a World-Wide-Web browser.
4. The system shall support at least 20 transactions per second.
5. The system facilities which are available to public users shall be demonstrable in 10 minutes or less.



System Requirement

- Example Requirements were written in natural way:
  - Most be supplemented with more detailed information in a complete specification of the system
- Lib System Examples shows different Types of requirements

1. The system shall maintain records of all library materials including books, serials, newspapers and magazines, video and audio tapes, reports, collections of transparencies, computer disks and CD-ROMs.
   - Very general requirements which set out in broad terms what the system should do.

System Requirement

2. The system shall allow users to search for an item by title, author, or by ISBN.
   - Functional requirements which define part of the system's functionality.
3. The system's user interface shall be implemented using a World-Wide-Web browser.
   - Implementation requirements which state how the system must be implemented.
4. The system shall support at least 20 transactions per second.
   - Performance requirements which specify a minimum acceptable performance for the system.
5. The system facilities which are available to public users shall be demonstrable in 10 minutes or less.
   - Usability requirements which specify the maximum acceptable time to demonstrate the use of the system.

Types of Requirements

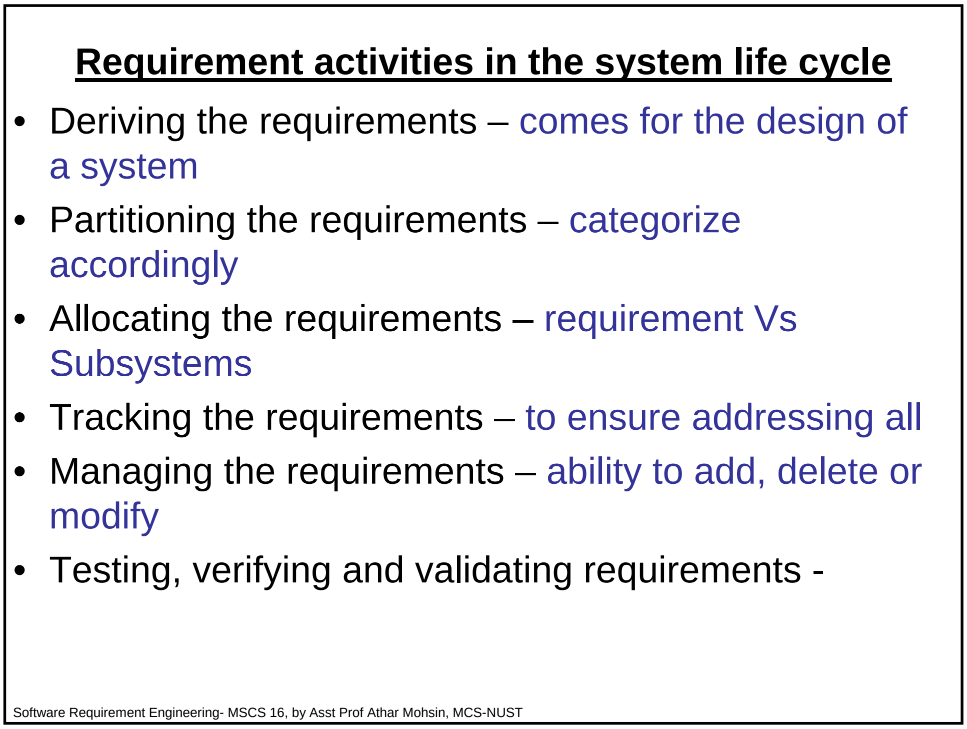
- The system shall maintain records of all library materials including books, serials, newspapers and magazines, video and audio tapes, reports, collections of transparencies, computer disks and CD-ROMs.
  - Very general requirements
- The system shall allow users to search for an item by title, author, or by ISBN.
  - Functional requirements
- The system's user interface shall be implemented using a World-Wide-Web browser.
  - Implementation requirements
- The system shall support at least 20 transactions per second.
  - Performance requirements
- The system facilities which are available to public users shall be demonstrable in 10 minutes or less.
  - Usability requirements

Requirement Problem

- So many different types of requirement
  - – Not possible to
    - Describe a standard way of writing requirement
    - Define best way to specify requirement
  - – It depends on:
    - Who is writing the requirement
    - Who is likely to read the requirement
    - General practices of the developing organization
    - The application domain of the system

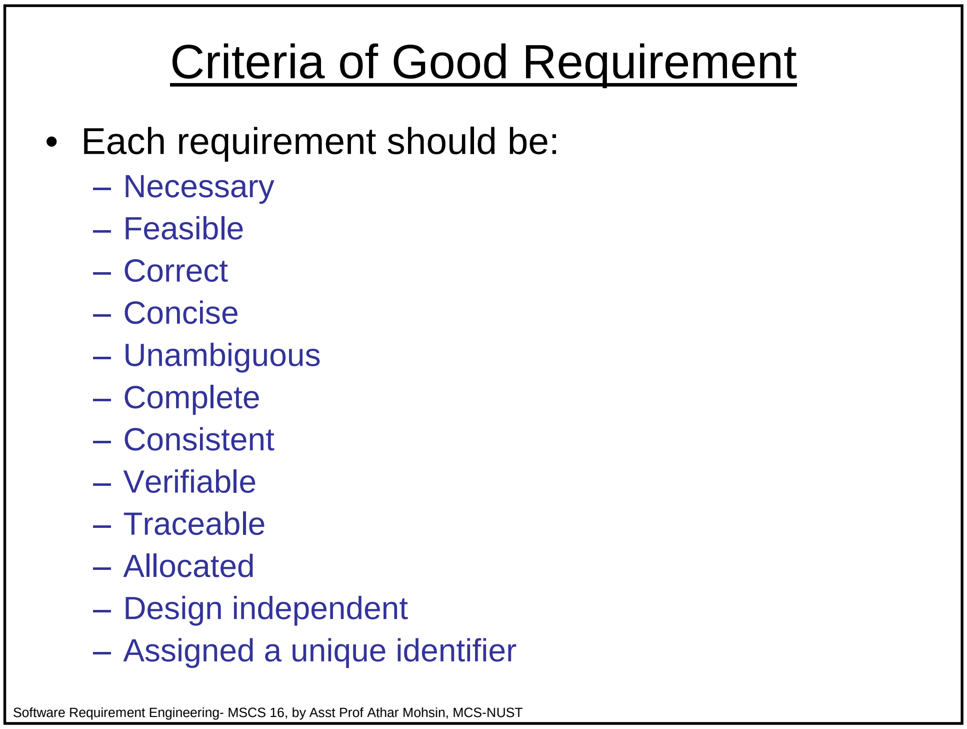
Requirement activities in the system life cycle

- Identify the stakeholder – anyone has an interest
- Identify requirements- stated or simple sentences
- Clarifying & restating requirements – real needs
- Analyzing the requirements – well defined
- Defining the requirement in a way that it means the same thing to all of the stakeholders – work for common understanding
- Specifying the requirements – precise details included in SRS
- Prioritizing the requirements – opportunity to address highest priority first



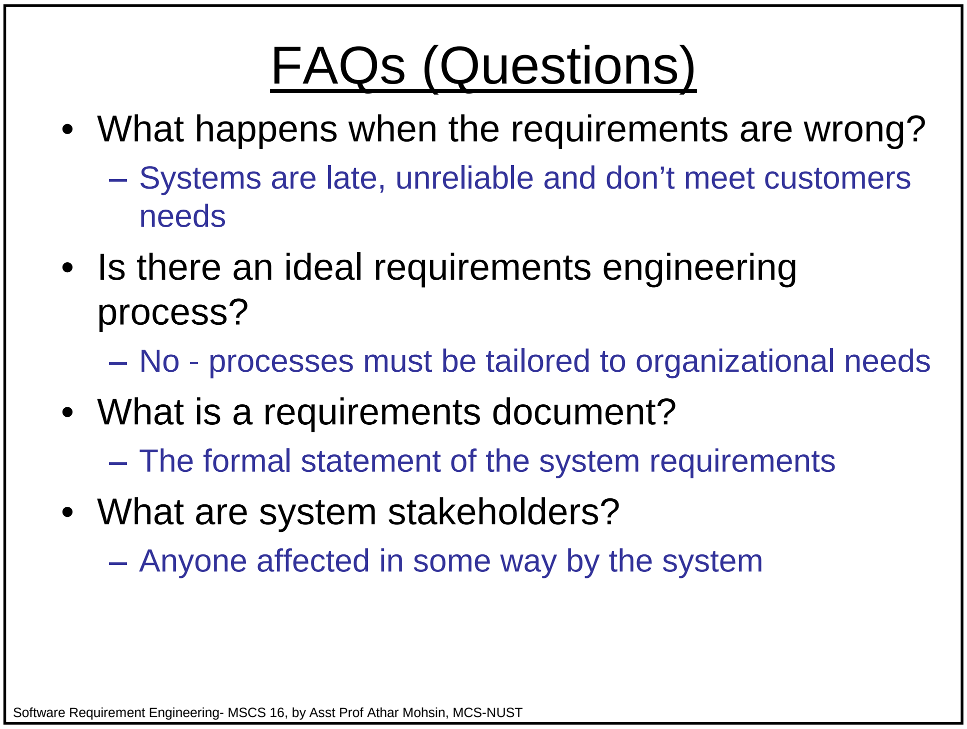
Requirement activities in the system life cycle

- Deriving the requirements – comes for the design of a system
- Partitioning the requirements – categorize accordingly
- Allocating the requirements – requirement Vs Subsystems
- Tracking the requirements – to ensure addressing all
- Managing the requirements – ability to add, delete or modify
- Testing, verifying and validating requirements -



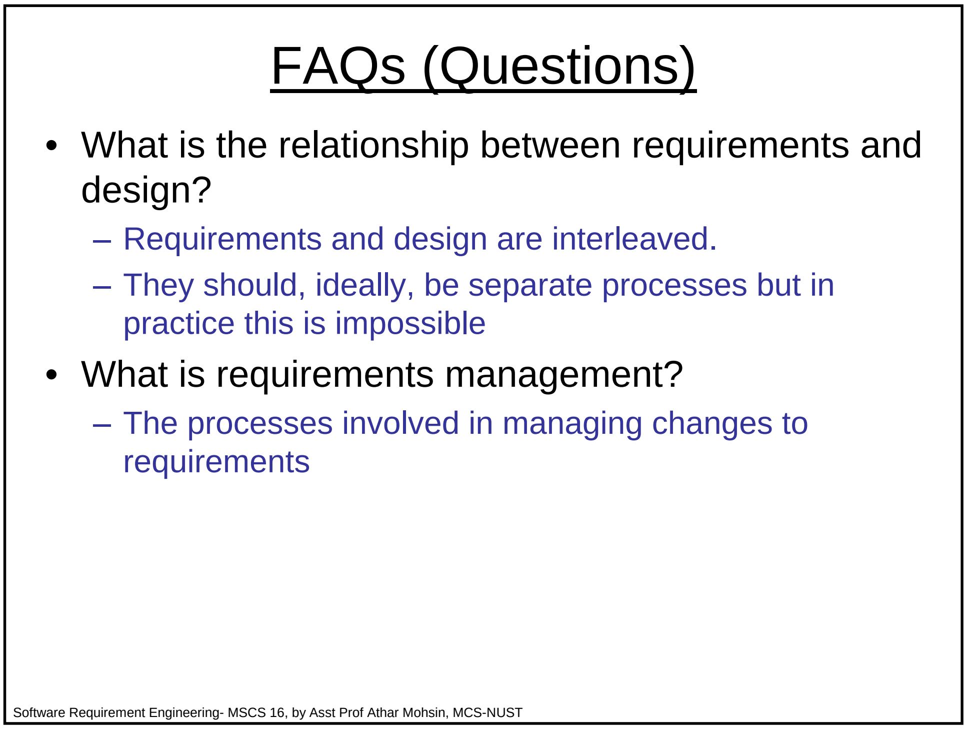
Criteria of Good Requirement

- Each requirement should be:
  - Necessary
  - Feasible
  - Correct
  - Concise
  - Unambiguous
  - Complete
  - Consistent
  - Verifiable
  - Traceable
  - Allocated
  - Design independent
  - Assigned a unique identifier



FAQs (Questions)

- What happens when the requirements are wrong?
  - Systems are late, unreliable and don’t meet customers needs
- Is there an ideal requirements engineering process?
  - No - processes must be tailored to organizational needs
- What is a requirements document?
  - The formal statement of the system requirements
- What are system stakeholders?
  - Anyone affected in some way by the system



FAQs (Questions)

- What is the relationship between requirements and design?
  - Requirements and design are interleaved.
  - They should, ideally, be separate processes but in practice this is impossible
- What is requirements management?
  - The processes involved in managing changes to requirements

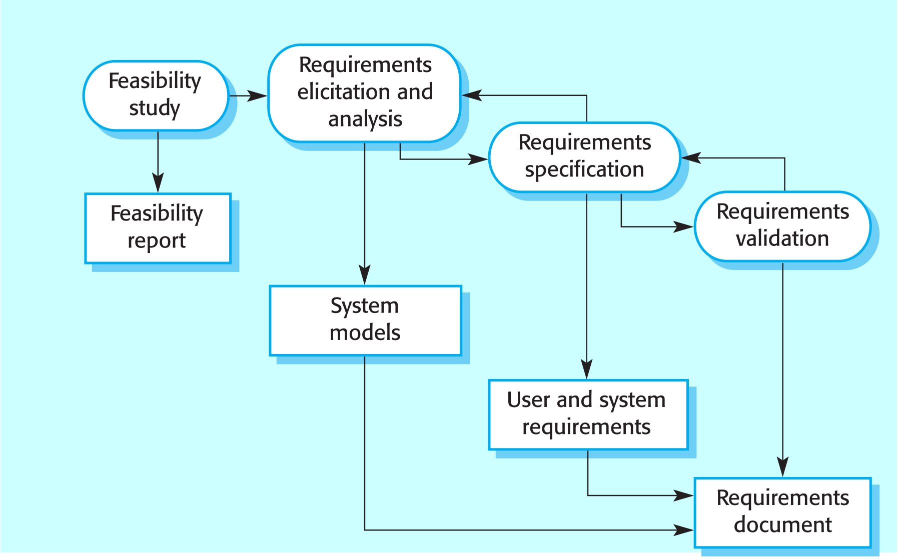
Requirement Engineering Process

- What is a requirements engineering process?
  - – The structured set of activities involved in developing system requirements
  - – RE – two main groups of activities
    - Requirements development
      - – Activates related to discovering, analyzing, documenting and validating requirements
    - Requirement Management
      - – Activities related to maintenance, identification, traceability and change management of requirement

Requirements engineering processes

- The processes used for RE vary widely depending on:
  - – The application domain,
  - – The people involved and
  - – The organisation developing the requirements.
- However, there are a number of generic activities common to all processes
  - – Requirements elicitation;
  - – Requirements analysis;
  - – Requirements validation;
  - – Requirements management.

The requirements engineering process



Feasibility studies

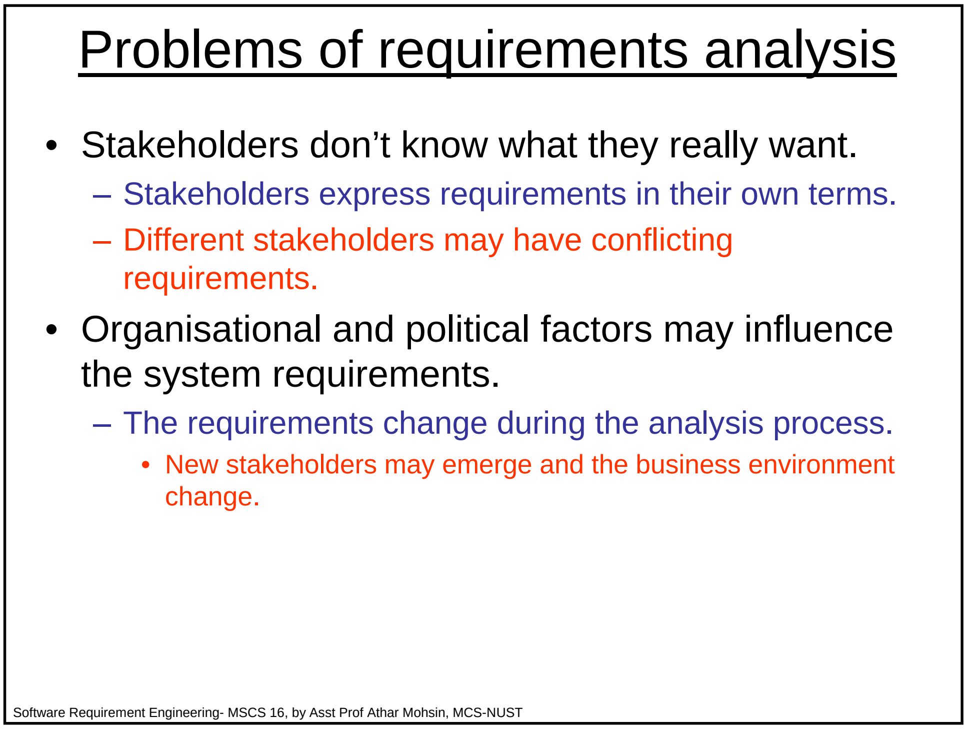
- A feasibility study decides whether or not the proposed system is worthwhile.
- A short focused study that checks
  - If the system contributes to organisational objectives;
  - If the system can be engineered using current technology and within budget;
  - If the system can be integrated with other systems that are used.

Feasibility study implementation

- Based on information assessment (what is required), information collection and report writing.
- Questions for people in the organisation
  - What if the system wasn't implemented?
  - What are current process problems?
  - How will the proposed system help?
  - What will be the integration problems?
  - Is new technology needed? What skills?
  - What facilities must be supported by the proposed system?

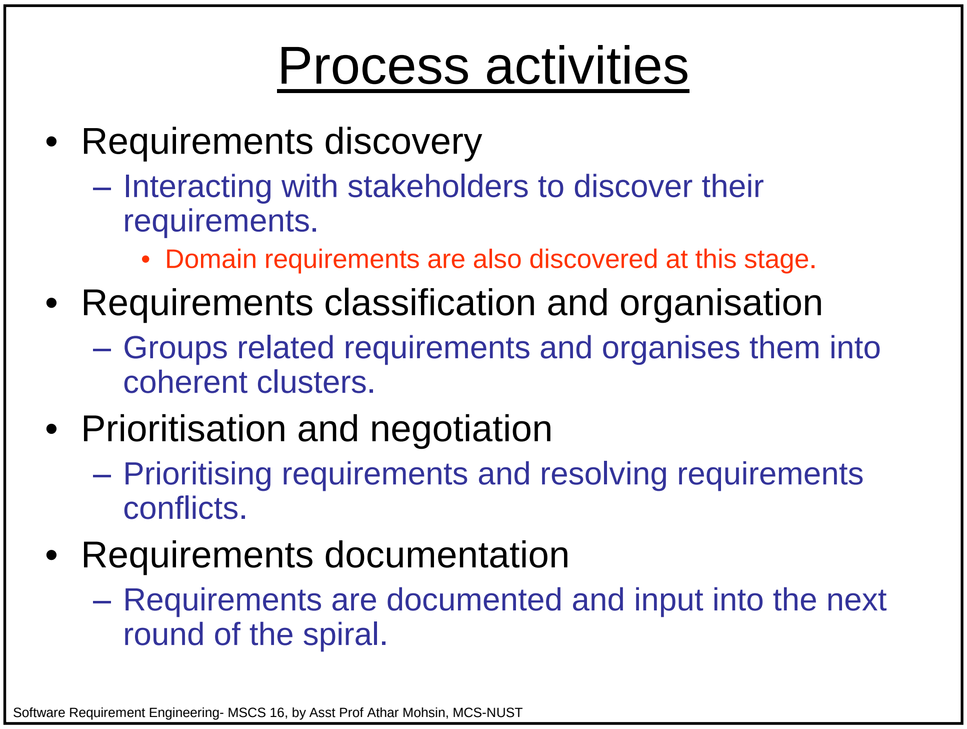
Elicitation and analysis

- Sometimes called requirements elicitation or requirements discovery.
  - Involves technical staff working with customers to find out about the application domain, the services that the system should provide and the system's operational constraints.
- May involve end-users, managers, engineers involved in maintenance, domain experts, trade unions, etc. These are called *stakeholders.*



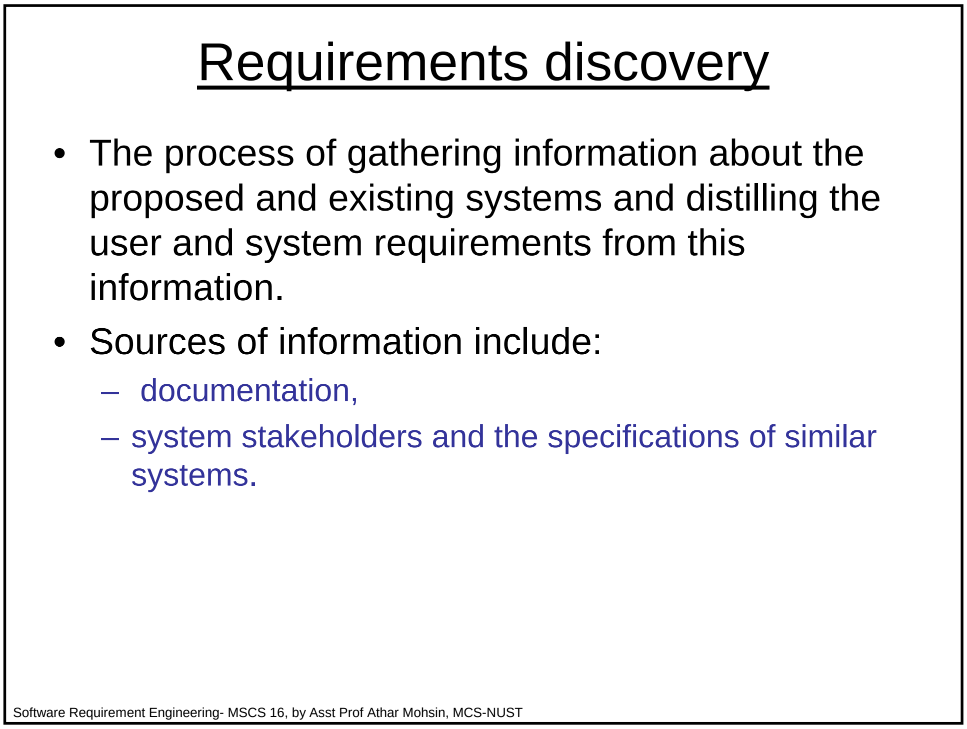
Problems of requirements analysis

- Stakeholders don’t know what they really want.
  - Stakeholders express requirements in their own terms.
  - Different stakeholders may have conflicting requirements.
- Organisational and political factors may influence the system requirements.
  - The requirements change during the analysis process.
    - New stakeholders may emerge and the business environment change.



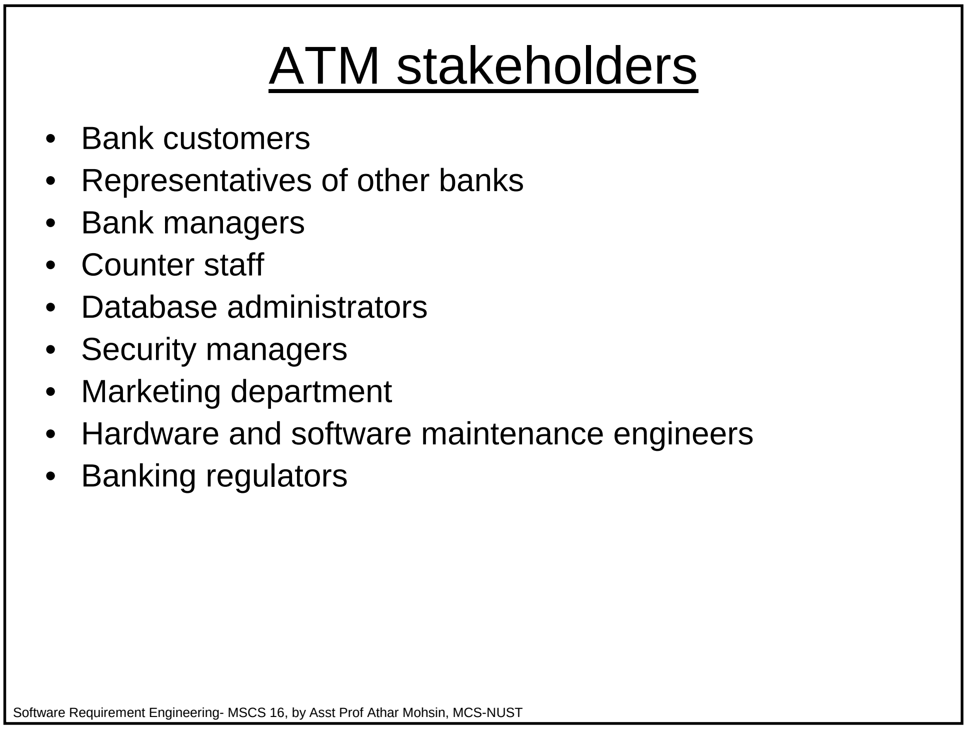
Process activities

- Requirements discovery
  - Interacting with stakeholders to discover their requirements.
    - Domain requirements are also discovered at this stage.
- Requirements classification and organisation
  - Groups related requirements and organises them into coherent clusters.
- Prioritisation and negotiation
  - Prioritising requirements and resolving requirements conflicts.
- Requirements documentation
  - Requirements are documented and input into the next round of the spiral.



Requirements discovery

- The process of gathering information about the proposed and existing systems and distilling the user and system requirements from this information.
- Sources of information include:
  - documentation,
  - system stakeholders and the specifications of similar systems.



ATM stakeholders

- Bank customers
- Representatives of other banks
- Bank managers
- Counter staff
- Database administrators
- Security managers
- Marketing department
- Hardware and software maintenance engineers
- Banking regulators

Viewpoints

- Viewpoints are a way of structuring the requirements to represent the perspectives of different stakeholders.
  - Stakeholders may be classified under different viewpoints.
- This multi-perspective analysis is important as there is no single correct way to analyse system requirements.

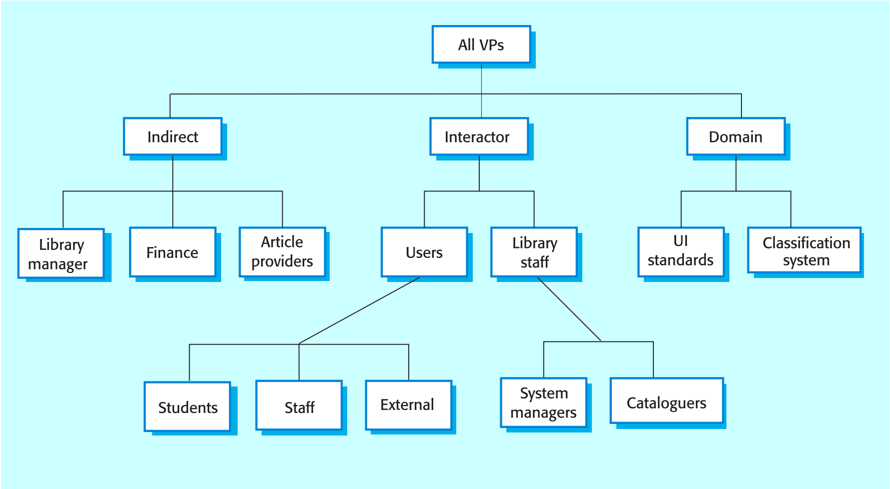
Types of viewpoint

- Interactor viewpoints
  - People or other systems that interact directly with the system.
    - In an ATM, the customer’s and the account database are interactor VPs.
- Indirect viewpoints
  - Stakeholders who do not use the system themselves but who influence the requirements.
    - In an ATM, management and security staff are indirect viewpoints.
- Domain viewpoints
  - Domain characteristics and constraints that influence the requirements.
    - In an ATM, an example would be standards for inter-bank communications.

Viewpoint identification

- Identify viewpoints using
  - Providers and receivers of system services;
  - Systems that interact directly with the system being specified;
  - Regulations and standards;
  - Sources of business and non-functional requirements.
  - Engineers who have to develop and maintain the system;
  - Marketing and other business viewpoints.

LIBSYS viewpoint hierarchy



Interviewing

- In formal or informal interviewing, the RE team puts questions to stakeholders about the system that they use and the system to be developed.
- There are two types of interview
  - Closed interviews where a pre-defined set of questions are answered.
  - Open interviews where there is no pre-defined agenda and a range of issues are explored with stakeholders.

Interviews in practice

- Normally a mix of closed and open-ended interviewing.
  - Interviews are good for getting an overall understanding of what stakeholders do and how they might interact with the system.
- Interviews are not good for understanding domain requirements
  - Requirements engineers cannot understand specific domain terminology;
  - Some domain knowledge is so familiar that people find it hard to articulate or think that it isn’t worth articulating.

Effective interviewers

- Interviewers should be:
  - open-minded,
  - willing to listen to stakeholders and
  - should not have pre-conceived ideas about the requirements.
- They should prompt the interviewee with a question or a proposal and should not simply expect them to respond to a question such as ‘what do you want’.

Next Class Activity

Can You walk on the water?

Walking on water and developing software from a specification are easy if they are frozen

Next Class Agenda:

- **Assignment 1 due**
- **Discussion & Presentation**
  - **IEEE Recommended Practice for Software Requirements Specifications**
  - **The What, why, who Paper**
  - **Communication Problems in RE**

Nominate “ Master of the Day” to

Organize the class for Presentation + Discussion on the given papers and assignment

Master of the day MUST prepare Schedule for next class and discuss by 1st Jul