Lecture-13

Requirement Engineering

Dronacharya College of Engineering

– Software Requirements –

Descriptions and specifications of a system

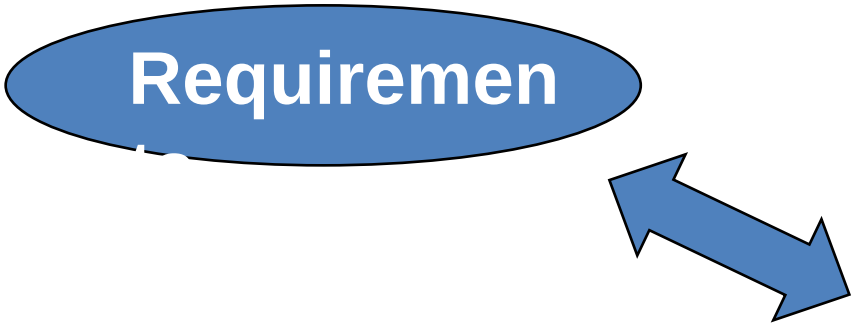
Objectives:

- To introduce the concepts of **user and system requirements**
- To describe **functional / non-functional requirements**
- To explain **how software requirements may be organised** in a requirements document

Requirements engineering

Requirements engineering is the process of establishing

- the services that the customer requires from a system
- the constraints under which it operates and is developed



The descriptions of the system services and constraints

that are generated during the requirements engineering process

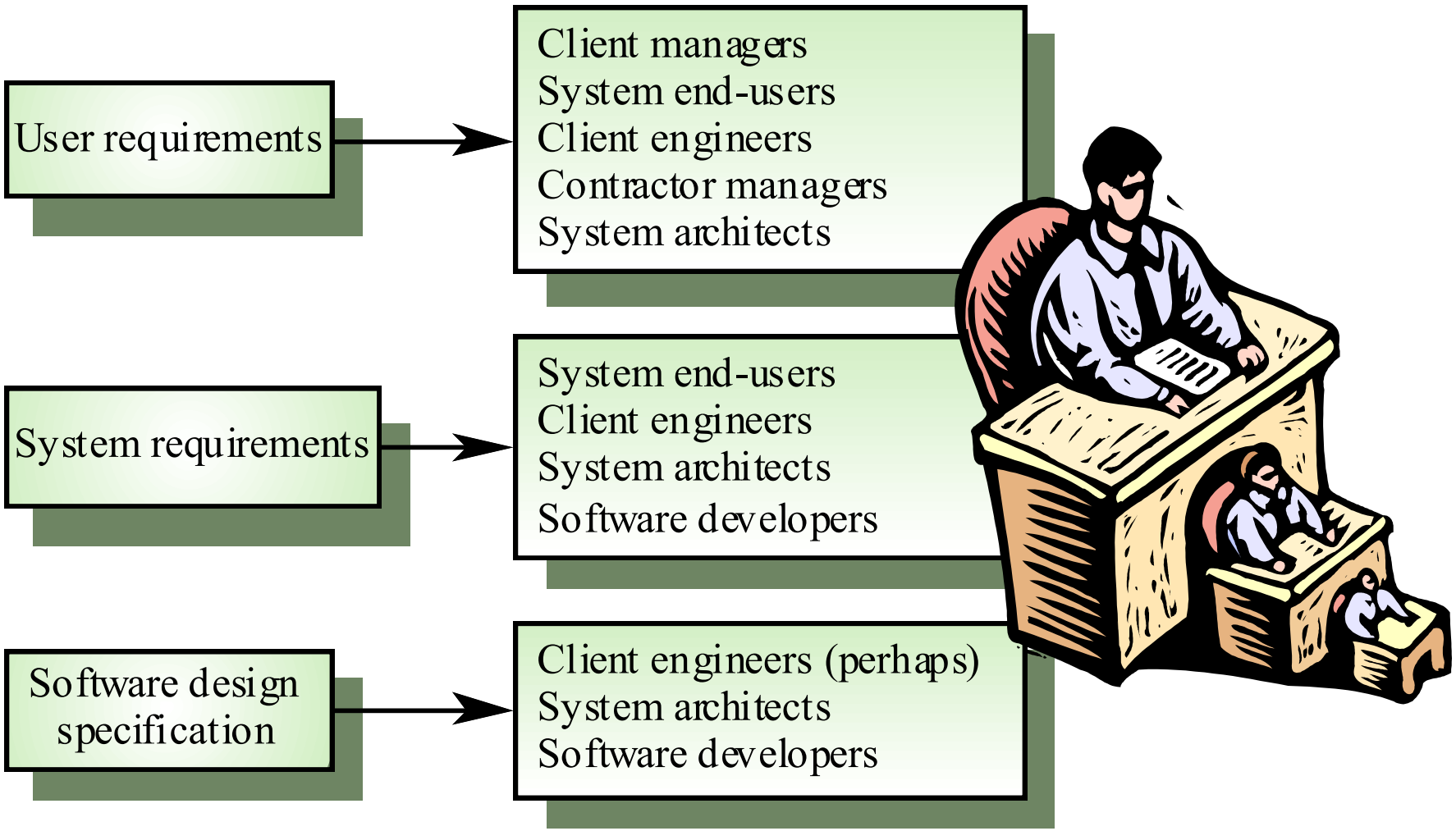
What is a requirement?

- It may range from a **high-level** abstract statement of a service or of a system constraint to a **detailed** mathematical functional specification
- This is inevitable as requirements may serve a **dual function**
  - May be the basis for a bid for a contract - therefore must be open to interpretation
  - May be the basis for the contract itself - therefore must be defined in detail
  - Both these statements may be called requirements

Types Of Requirement

- **User requirements**
  - Statements in natural language plus diagrams of the services the system provides and its operational constraints. Written for customers
- **System requirements**
  - A structured document setting out detailed descriptions of the system services. Written as a contract between client and contractor
- **Software specification**
  - A detailed software description which can serve as a basis for a design or implementation. Written for developers

Requirements readers



Functional and non-functional requirements

- **Functional requirements**
  - Statements of services the system should provide, how the system should react to particular inputs and how the system should behave in particular situations.
- **Non-functional requirements**
  - constraints on the services or functions offered by the system such as timing constraints, constraints on the development process, standards, etc.
- **Domain requirements**
  - Requirements that come from the application domain of the system and that reflect characteristics of that domain

Requirement Engineering

- RE helps software engineer to better understand the problem they will work to solve
- Participant : Software Engineers, managers, customers and end users
- RE is a software engineering action that begin during the communication activity and continues into the modeling activity

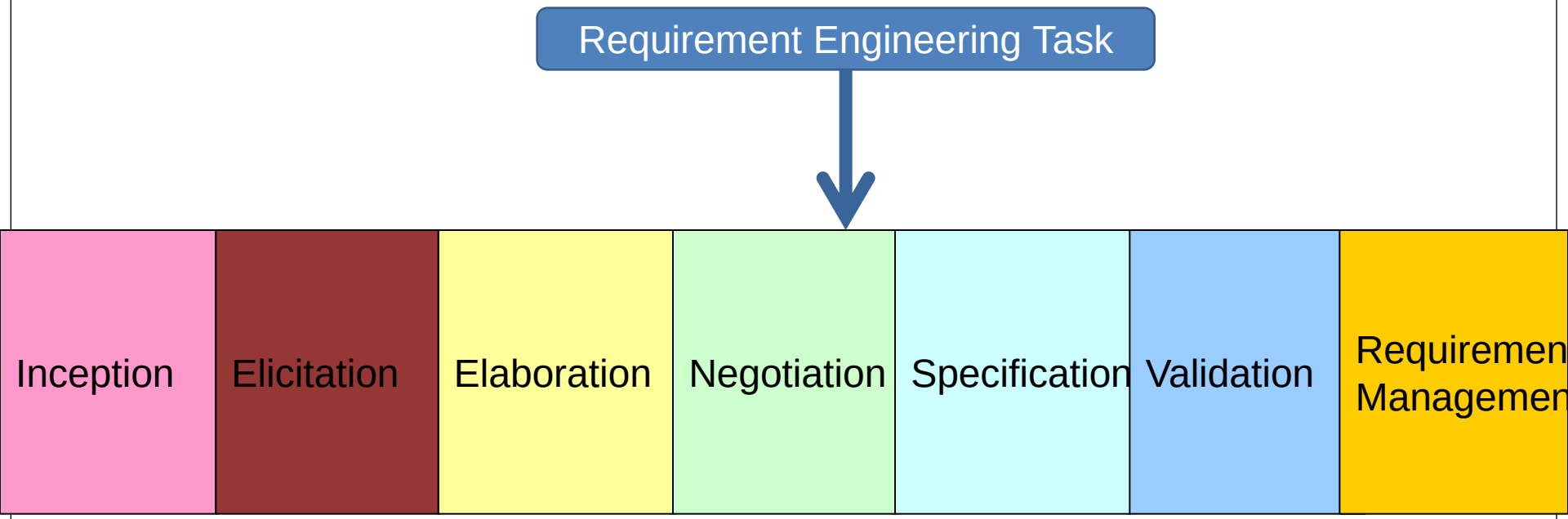
Requirement Engineering

- Provides the appropriate mechanism for :
  - Understanding what the customer want
  - Analyzing need
  - Assessing feasibility
  - Negotiating a reasonable solution
  - Specifying a solution unambiguously
  - Validating the specification
  - Managing the requirement as they are transformed into an operational system

Requirement Engineering Task

- *Inception*
- *Elicitation*
- *Elaboration*
- *Negotiation*
- *Specification*
- *Validation*
- *Management*

Requirement Engineering Task



RE Task :1. Inception

- ask a set of questions that establish …
  - basic understanding of the problem
  - the people who want a solution
  - the nature of the solution that is desired, and
  - the effectiveness of preliminary communication and collaboration between the customer and the developer

RE Task : Inception (cont.)

- Inception process :
  - Identify stakeholders
    - “who else do you think I should talk to?”
  - Recognize multiple points of view
  - Work toward collaboration
    - the effectiveness of preliminary communication and collaboration between the customer and the developer
  - Asking The first questions
    - Who is behind the request for this work?
    - Who will use the solution?
    - What will be the economic benefit of a successful solution
    - Is there another source for the solution that you need?

RE Task : 2.Elicitation

- It certainly simple enough, but…
- Why difficult :
  - Problem of Scope
    - The boundary of the system is ill-defined
  - Problem of Understanding
    - The customer/users are not completely sure of what is needed
  - Problem of volatility
    - The requirement change over time
- To help overcame these problem, requirement engineers ,must approach the requirement gathering activity in an organized manner

RE Task : Elicitation (cont.)

- Elicitation Process Guideline:
  - meetings are conducted and attended by both software engineers and customers
  - rules for preparation and participation are established
  - an agenda is suggested
  - a "facilitator" (can be a customer, a developer, or an outsider) controls the meeting
  - a "definition mechanism" (can be work sheets, flip charts, or wall stickers or an electronic bulletin board, chat room or virtual forum) is used
  - the goal is
    - to identify the problem
    - propose elements of the solution
    - negotiate different approaches, and
    - specify a preliminary set of solution requirements

Quality Function Deployment

- Is a technique that translate the need of the customer into technical requirement for software.
- QFD emphasize an understanding of what is valuable to the customer and then deploys these values throughout the engineering process
- QFD identifies three types of requirement :
  - Normal Requirement
  - Expected requirement
  - Exciting requirement

Elicitation Work Products

- a statement of need and feasibility.
- a bounded statement of scope for the system or product.
- a list of customers, users, and other stakeholders who participated in requirements elicitation
- a description of the system's technical environment.
- a list of requirements (preferably organized by function) and the domain constraints that apply to each.
- a set of usage scenarios that provide insight into the use of the system or product under different operating conditions.
- any prototypes developed to better define