MALLA REDDY UNIVERSITY

MR22-1CS0146: OBJECT ORIENTED SOFTWARE ENGINEERING

II YEAR B.TECH. CSE I - SEM

(MRU-R22)

UNIT-II

Software Engineering Principles:

Software Requirements: Functional and non-functional requirements, user requirements, system requirements, interface specification, the software requirements document.

Requirements Engineering:

Feasibility studies, requirements elicitation and analysis, requirements validation, requirements management.

SOFTWARE REQUIREMENTS

Software requirements are necessary

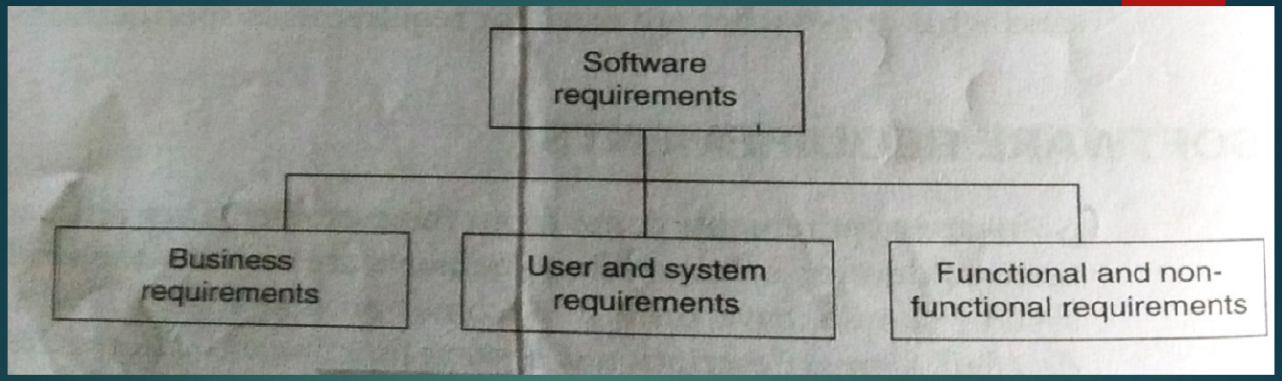
- To introduce the concepts of user and system requirements
- To describe functional and non-functional requirements
- To explain how software requirements may be organized

What is a requirement?

- the description of the services provided by the system and its operational constraints
- It may range from a high-level abstract statement of a service or of a system constraint to a detailed mathematical functional specification

Requirements engineering:

- The process of finding out, analysing documenting and checking these services and constraints.
- The requirements themselves are the descriptions of the system services and constraints that are generated during the requirements engineering process



Business Requirements: It defines the project goal and the expected business needs and opportunities for the product. Business needs also describe the product domain and product demand.

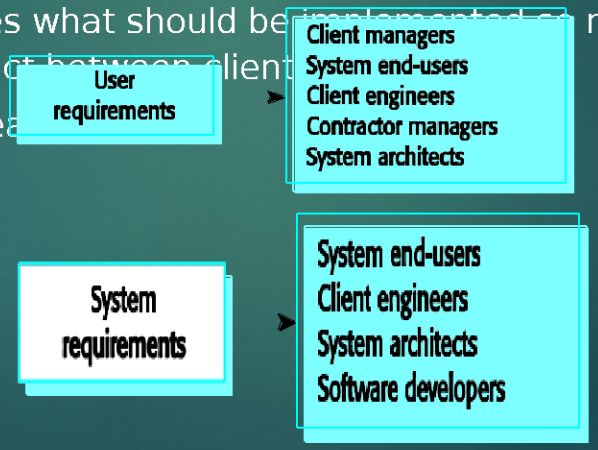
User requirements: are the high level abstract stalemates supplied by the customer, end users. These requirements are the functionalities that the system is expected to provide within the certain constraints or environment.

System requirements: are the detailed and technical functionalities written in a systematic manner that are implemented in the business process to achieve the goal of user 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's functions, services and operational constraints.
  - Defines what should be [illegible] may be part of a contract between client

Requirements rea[illegible]

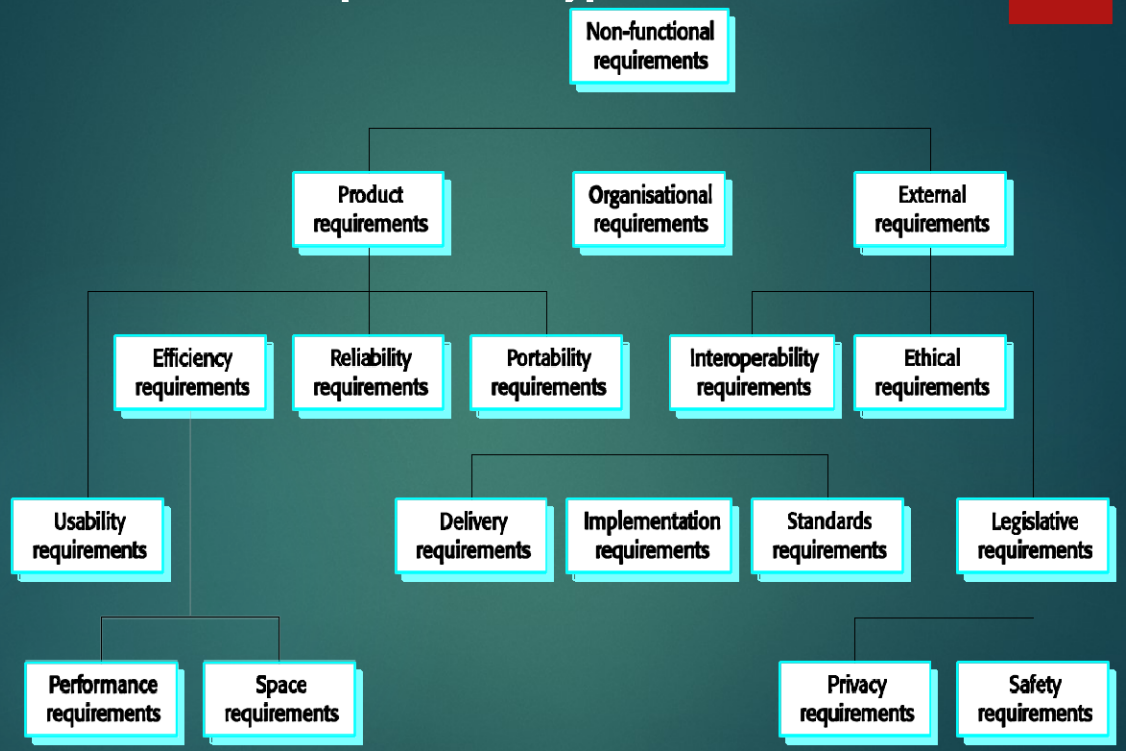


Functional and non-functional requirements:

- **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.

  Examples of functional requirements
  - The user shall be able to search either all of the initial set of databases or select a subset from it.
  - The system shall provide appropriate viewers for the user to read documents in the document store.
  - Every order shall be allocated a unique identifier (ORDER_ID) which the user shall be able to copy to the account's permanent storage area.
- **Non-functional requirements**
  - Constraints on the services or functions offered by the system such as timing constraints, constraints on the development process, standards, etc.
  - Non-functional requirements may be more critical than functional requirements. If these are not met, the system is

Non-functional requirement types:



Requirements measures:

Property	Measure
Speed	Processed transactions/second User/Event response time Screen
Size	M Bytes Number of ROM chips
Ease of use	Training time Number of help frames
Reliability	Mean time to failure Probability of unavailability Rate of failure occurrence Availability
Robustness	Time to restart after failure Percentage of events causing failure Probability of data corruption on
Portability	Percentage of target dependent statements Number of target systems

- **Domain requirements**
  - Requirements that come from the application domain of the system and that reflect characteristics of that domain.

USER REQUIREMENTS

User requirements are defined using natural language, tables and diagrams as these can be understood by all users.

Problems with natural language

- Lack of clarity
  - Precision is difficult without making the document difficult to read.
- Requirements confusion
  - Functional and non-functional requirements tend to be mixed-up.
- Requirements amalgamation
  - Several different requirements may be expressed together.

- **Guidelines for writing requirements**
  - Invent a standard format and use it for all requirements.
  - Use language in a consistent way. Use shall for

SYSTEM REQUIREMENTS

- More detailed specifications of system functions, services and constraints than user requirements.
- They are intended to be a basis for designing the system.
- They may be incorporated into the system contract.
- System requirements may be defined or illustrated using system models.

Problems with NL (natural language) specification

- Ambiguity
  - The readers and writers of the requirement must interpret the same words in the same way. NL is naturally ambiguous so this is very difficult.
- Over-flexibility
  - The same thing may be said in a number of different ways in the specification.
- Lack of modularization.
  - NL structures are inadequate to structure system

Alternatives to NL specification:

- **Structured language specifications**
  - All requirements are written in a standard way.
  - The terminology used in the description may be limited.
  - The advantage is that the most of the expressiveness of natural language is maintained but a degree of uniformity is imposed on the specification.

- **Form-based specifications**
  - Definition of the function or entity.
  - Description of inputs and where they come from.
  - Description of outputs and where they go to.
  - Indication of other entities required.
  - Pre and post conditions (if appropriate).
  - The side effects (if any) of the function.

- **Tabular specification**
  - Used to supplement natural language.
  - Particularly useful when you have to define a number of possible alternative courses of action.
- **Graphical models**
  - Graphical models are most useful when you need to show how state changes or where you need to describe a sequence of actions.
- **Sequence diagrams**
    - These show the sequence of events that take place during some user interaction with a system.
    - You read them from top to bottom to see the order of the actions that take place.
    - Cash withdrawal from an ATM
      - Validate card;
      - Handle request;
      - Complete transaction.

- Sequence diagram of ATM withdrawal

INTERFACE SPECIFICATION

- Most systems must operate with other systems and the operating interfaces must be specified as part of the requirements.

- Three types of interface may have to be defined
  - ***Procedural interfaces*** where existing programs or sub-systems offer a range of services that are accessed by calling interface procedures. These interfaces are sometimes called Application Programming Interfaces (APIs)
  - ***Data structures that are exchanged t***hat are passed from one sub-system to another. Graphical data models are the best notations for this type of description
  - ***Data representations*** that have been established for an existing sub-system
- Formal notations are an effective technique for interface specification.

THE SOFTWARE REQUIREMENTS DOCUMENT:

- **IEEE requirements standard** defines a generic structure for a requirements document that must be instantiated for each specific system.

1. Introduction.

i) Purpose of the requirements document ii) Scope of the project

iii) Definitions, acronyms and abbreviations iv) References

v) Overview of the remainder of the document

2. General description.

i) Product perspective ii) Product functions

iii) User characteristics iv) General constraints

v) Assumptions and dependencies

3. Specific requirements cover functional, non-functional and interface requirements. The requirements may document external interfaces, describe system functionality and performance, specify logical database requirements, design constraints, emergent system properties and quality characteristics.

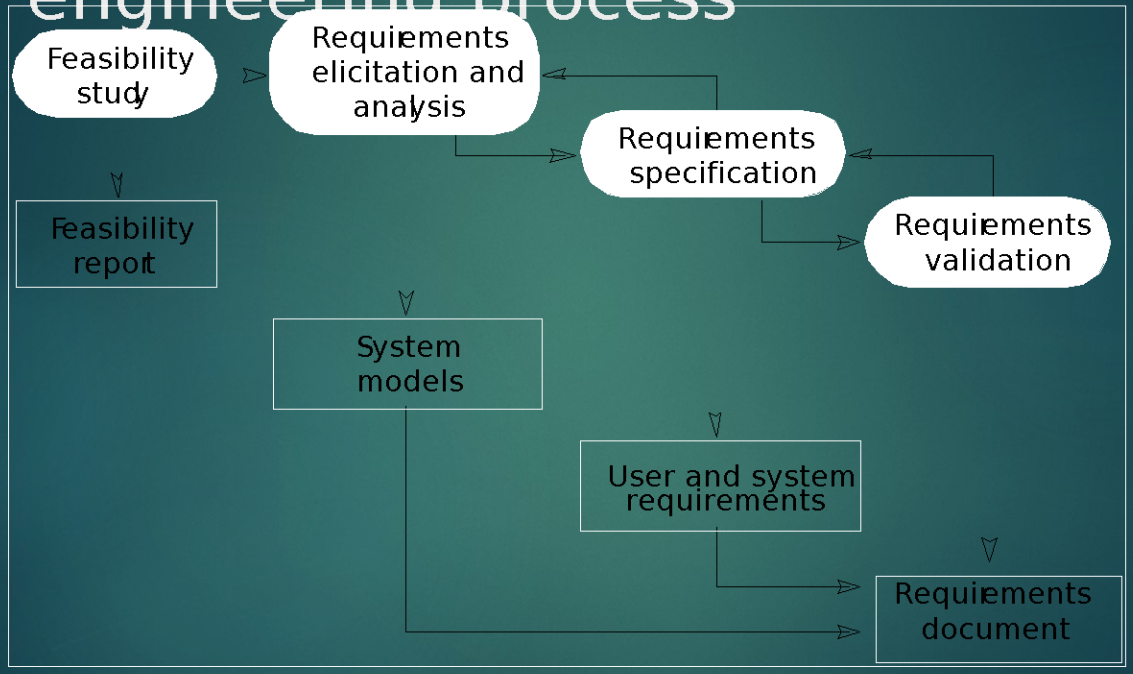
4. Appendices.

. Index.

REQUIREMENTS ENGINEERING PROCESS

- To create and maintain a system requirement document
- The overall process includes four high level requirements engineering sub-processes:
- 1.Feasibility study
  --Concerned with assessing whether the system is useful to the business
- 2.Elicitation and analysis
  --Discovering requirements
- 3.Specifications
  --Converting the requirements into a standard form
- 4.Validation
  -- Checking that the requirements actually define the system that the customer wants

The requirements engineering process

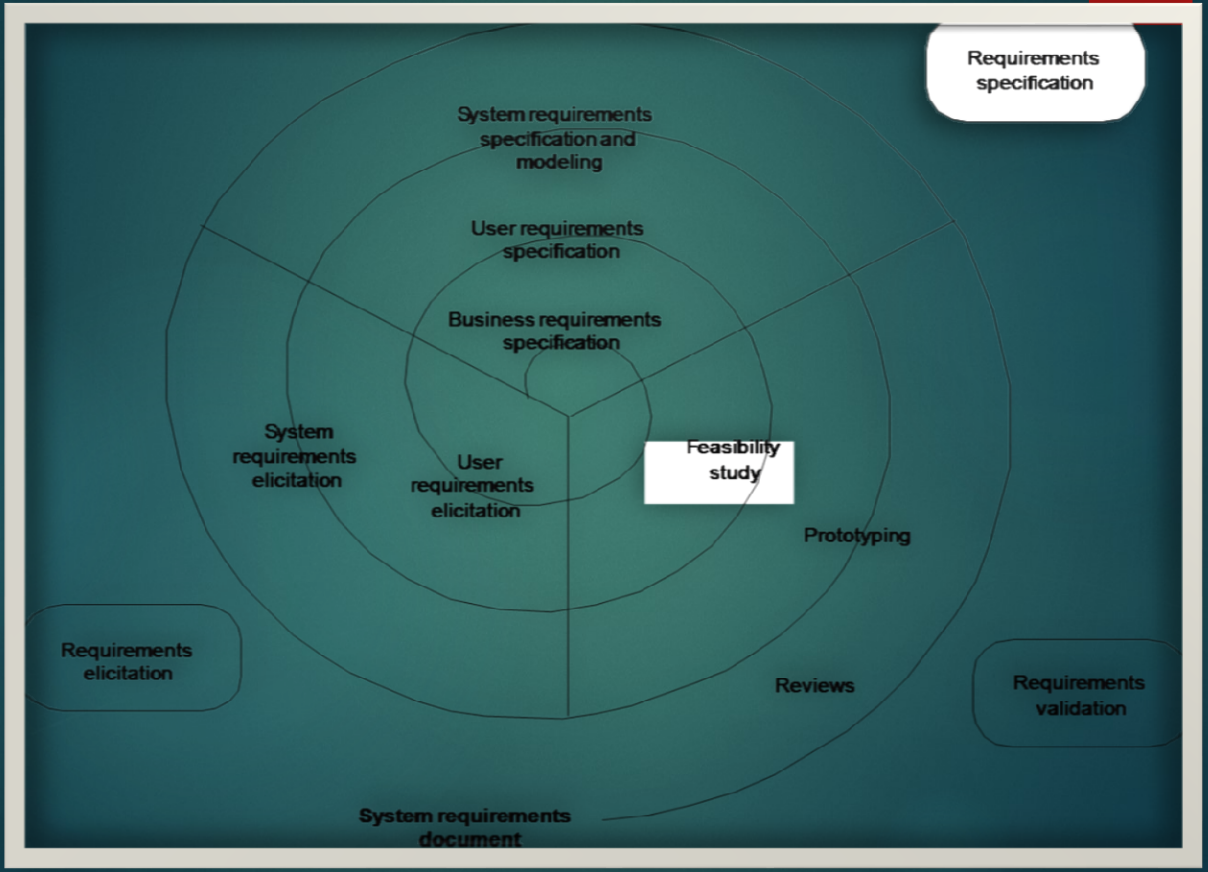


SPIRAL REPRESENTATION OF REQUIREMENTS ENGINEERING PROCESS

- Process represented as three stage activity
- Activities are organized as an iterative process around a spiral.
- Early in the process, most effort will be spent on understanding high-level business and the user requirement.
- Later in the outer rings, more effort will be devoted to system requirements engineering and system modeling

- Three level process consists of:
  - 1.Requirements elicitation
  - 2. Requirements specification
  - 3. Requirements validation

Requirements engineering



FEASIBILITY STUDIES

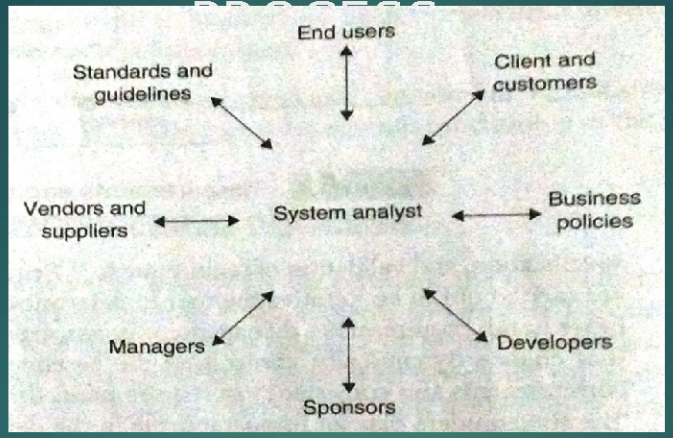
- Starting point of the requirements engineering process
- Input: Set of preliminary business requirements, an outine description of the system and how the system is intended to support business processes
- Output: Feasibility report that recommends whether or not it is worth carrying out further
- Feasibility report answers a number of questions:
  - 1.Does the system contribute to the overall objective?
  - 2.Can the system be implemented using the current technology and within given cost and schedule?
  - 3.Can the system be integrated with other system which are already in place?

REQUIREMENTS ELICITATION PROCESS

Process activities

- 1.Requirement Discovery
  - Interaction with stakeholder to collect their requirements including domain and documentation
- 2.Requirements classification and organization
  - Coherent clustering of requirements from unstructured collection of requirements
- 3. Requirements prioritization and negotiation
  - Assigning priority to requirements
  - Resolves conflicting requirements through negotiation
- 4. Requirements documentation
  - Requirements be documented and placed in the next round of spiral

REQUIREMENTS ELICITATION



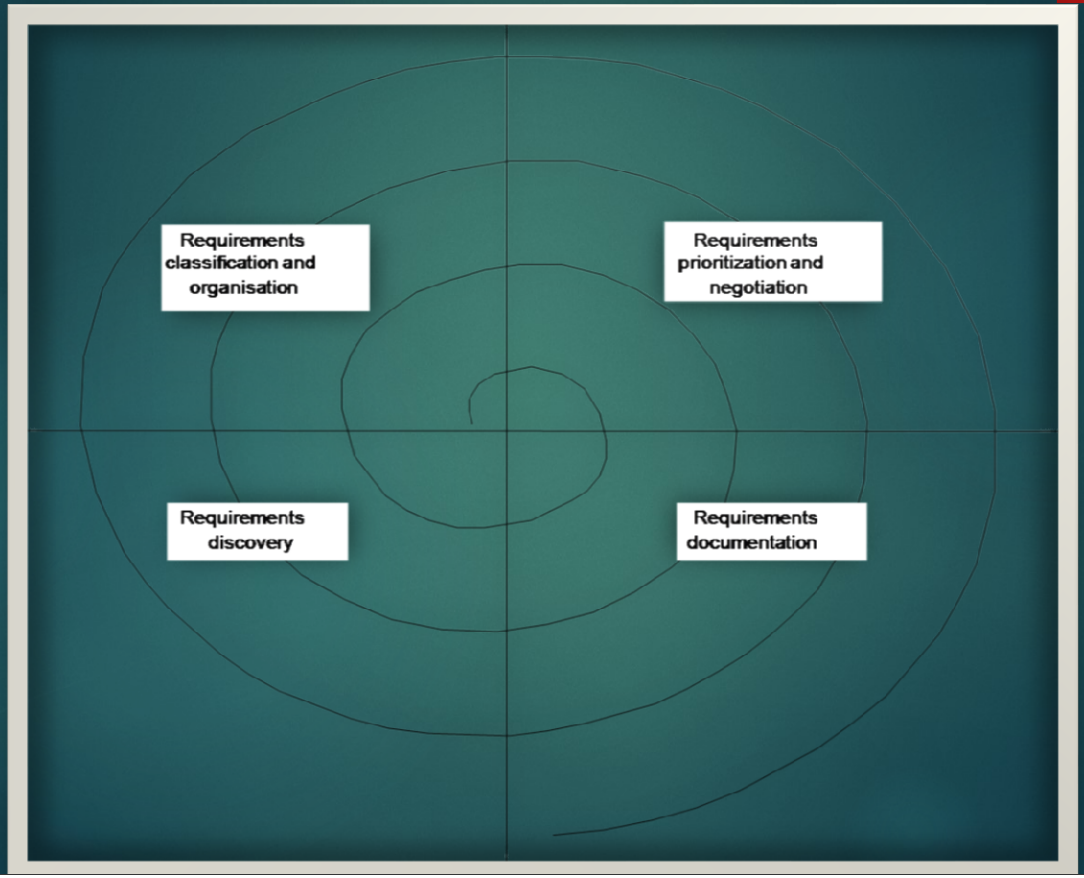
Requirements Elicitation is the process to find out the requirements for an intended software system by communicating with client, end users, system users and others who have a stake in the software system development.

Requirement Elicitation Techniques
Interviews
Surveys
Questionnaires

REQUIREMENTS ELICITATION ANALYSIS

- Involves a number of people in an organization
- Stakeholder definition
  Refers to any person or group who will be affected by the system directly or indirectly i.e. End users, Engineers, business managers, domain experts.
- Reasons why eliciting is difficult
  - 1.Stakeholder often don’t know what they want from the computer system.
  - 2.Stakeholder expression of requirements in natural language is sometimes difficult to understand.
  - 3.Different stakeholders express requirements differently
  - 4.Influences of political factors
- Change in requirements due to dynamic environments.

The spiral representation of Requirements Engineering



REQUIREMENTS DISCOVERY TECHNIQUES

1. View points
   --Based on the viewpoints expressed by the stake holder
      --Recognizes multiple perspectives and provides a framework for discovering conflicts in the requirements proposed by different stakeholders
Three Generic types of viewpoints
1.Interactor viewpoint
  --Represents people or other system that interact directly with the system
2.Indirect viewpoint
  --Stakeholders who influence the requirements, but don’t use the system
3.Domain viewpoint
  --Requirements domain characteristics and constraints that influence the requirements.

REQUIREMENTS DISCOVERY TECHNIQUES

2. Interviewing

--Puts questions to stakeholders about the system that they use and the system to be developed.

--Requirements are derived from the answers

Two types of interview

- Closed interviews where the stakeholders answer a pre-defined set of questions.
- Open interviews discuss a range of issues with the stakeholders for better understanding their needs.

Effective interviewers

a) Open-minded: no pre-conceived ideas

b) Prompter: prompt the interviewee to start discussion with a question or a proposal

REQUIREMENTS DISCOVERY TECHNIQUES

3. Scenarios

--Easier to relate to real life examples than to abstract description

--Starts with an outline of the interaction and during elicitation, details are added to create a complete description of that interaction

--Scenario includes:

1. Description at the start of the scenario
2. Description of normal flow of the event
3. Description of what can go wrong and how this is handled
4. Information about other activities parallel to the scenario
5. Description of the system state when the scenario finishes

LIBSYS scenario

- **Initial assumption**: The user has logged on to the LIBSYS system and has located the journal containing the copy of the article.
- **Normal**: The user selects the article to be copied. He or she is then prompted by the system to either provide subscriber information for the journal or to indicate how they will pay for the article. Alternative payment methods are by credit card or by quoting an organisational account number.
- The user is then asked to fill in a copyright form that maintains details of the transaction and they then submit this to the LIBSYS system.
- The copyright form is checked and, if OK, the PDF version of the article is downloaded to the LIBSYS working area on the user's computer and the user is informed that it is

LIBSYS scenario

- **What can go wrong**: The user may fail to fill in the copyright form correctly. In this case, the form should be re-presented to the user for correction. If the resubmitted form is still incorrect then the user's request for the article is rejected.
- The payment may be rejected by the system. The user's request for the article is rejected.
- The article download may fail. Retry until successful or the user terminates the session..
- **Other activities**: Simultaneous downloads of other articles.
- **System state on completion**: User is logged on. The downloaded article has been deleted from LIBSYS workspace if it has been flagged as print-only.

REQUIREMENTS DISCOVERY TECHNIQUES

4. Use cases

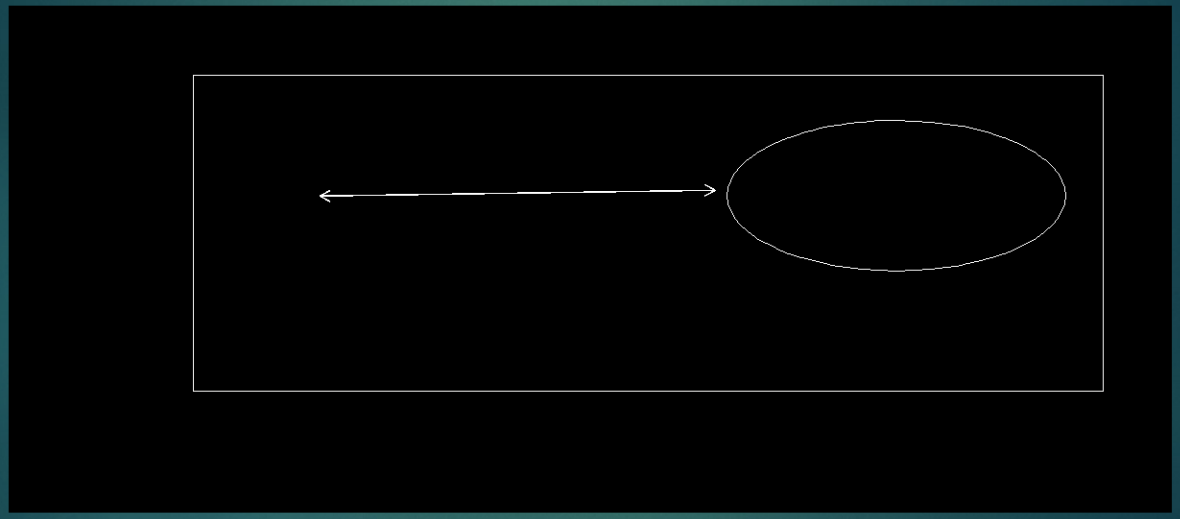
-- scenario based technique for requirement elicitation

-- A fundamental feature of UML, notation for describing object-oriented system models

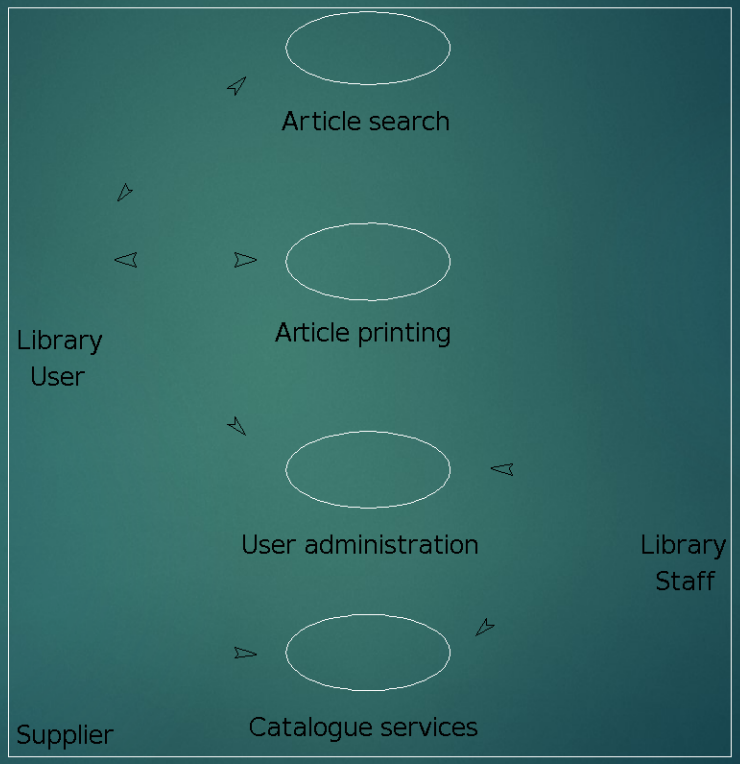
-- Identifies a type of interaction and the actors involved

-- Sequence diagrams are used to add information to a Use case

Article printing use-case



LIBSYS use cases



REQUIREMENTS VALIDATION

- Concerned with showing that the requirements define the system that the customer wants.
- Important because errors in requirements can lead to extensive rework cost
- Validation checks

1.Validity checks

--Verification that the system performs the intended function by the user

2.Consistency check

--Requirements should not conflict

3. Completeness checks

--Includes requirements which define all functions and constraints intended by the system user

4. Realism checks

--Ensures that the requirements can be actually implemented

5. Verifiability

-- Testable to avoid disputes between customer and developer.

VALIDATION TECHNIQUES

1.REQUIREMENTS REVIEWS
--Reviewers check the following:
(a) Verifiability: Testable
(b) Comprehensibility
(c) Traceability
(d) Adaptability
2.PROTOTYPING
3.TEST-CASE GENERATION

Requirements management

Requirements are likely to change for large software systems and as such requirements management process is required to handle changes.

Reasons for requirements changes

(a) Diverse Users community where users have different requirements and priorities

(b) System customers and end users are different

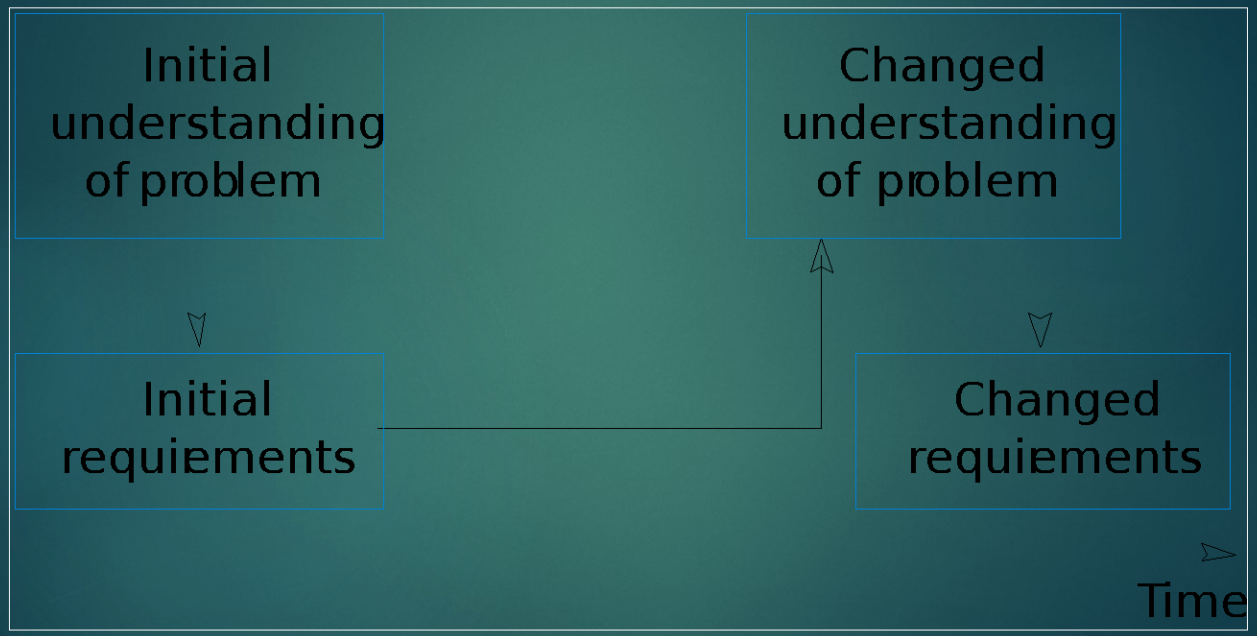
(c) Change in the business and technical environment after installation

Two classes of requirements

(a) Enduring requirements: Relatively stable requirements

(b) Volatile requirements: Likely to change during system development process or during operation

Requirements evolution



Requirements management planning

An essential first stage in requirement management process
Planning process consists of the following
1.Requirements identification
-- Each requirement must have unique tag for cross reference and traceability
2.Change management process
-- Set of activities that assess the impact and cost of changes
3.Traceability policy
-- A matrix showing links between requirements and other elements of software development
4.CASE tool support
--Automatic tool to improve efficiency of change management process. Automated tools are required for requirements storage, change management and traceability management

Traceability

Maintains three types of traceability information.

1.Source traceability

--Links the requirements to the stakeholders

2. Requirements traceability

--Links dependent requirements within the requirements document

3. Design traceability

-- Links from the requirements to the design module

A traceability matrix

Req. id	1.1	1.2	1.3	2.1	2.2	2.3	3.1	3.2
1.1		D	R					
1.2			D			D		D
1.3	R			R				
2.1			R		D			D
2.2								D
2.3		R		D				
3.1								R
3.2							R	

Requirements change management

Consists of three principal stages:

1. Problem analysis and change specification

-- Process starts with a specific change proposal and analysed to verify that it is valid

2. Change analysis and costing

--Impact analysis in terms of cost, time and risks

3. Change implementation

--Carrying out the changes in requirements document, system design and its implementation

Change management