Requirements Analysis and Specification

Requirements Analysis and Specification

- Goals of requirements analysis and specification phase:
  - Fully understand the user requirements.
  - Remove inconsistencies, anomalies, etc. from requirements.
  - Document requirements properly in an SRS document.

Requirements Analysis and Specification

- Consists of two distinct activities:
  - Requirements Analysis: Requirements Gathering and Analysis
  - Specification

Who Carries Out Requirements Analysis and Specification?

- The person who undertakes requirements analysis and specification: Systems analyst
  - Collects data pertaining to the product
  - Analyses collected data:
    - To understand what exactly needs to be done.
  - Writes the Software Requirements Specification (SRS) document.
  - Reviewed SRS document forms the basis of all future development activities

Requirements Analysis

- Analyst gathers requirements through:
  - Observation of existing systems,
  - Studying existing procedures,
  - Discussion with the customer and end-users,
  - Analysis of what needs to be done, etc.

Requirements Gathering Activities

. 1. Studying the existing documentation
. 2. Interview
. 3. Task analysis
. 4. Scenario analysis
. 5. Form analysis

Requirements Gathering (CONT.)

- In the absence of a working system,
  - Lot of imagination and creativity are required.
- Interacting with the customer to gather relevant data:
  - Requires a lot of experience.

Case Study: Automation of Office Work at CSE Dept.

- The academic, inventory, and financial information at the CSE department:
  - Being carried though manual processing by two office clerks, a store keeper, and two attendants.
- Considering the low budget he had at his
- Disposal:
  - The HoD entrusted the work to a team of student volunteers.

Case Study: Automation of Office Work at CSE Dept.

- The team was first briefed by the HoD about the specific activities to be automated.
- The analyst first discussed with the two clerks:
  - Regarding their specific responsibilities (tasks) that were to be automated.
- The analyst also interviewed student and faculty representatives who would also use the software.

Case Study: Automation of Office Work at CSE Dept.

- For each task, they asked:
  - About the steps through which these are performed.
  - They also discussed various scenarios that might arise for each task.
  - The analyst collected all types of forms that were being used.

Analysis of the Gathered Requirements

- Main purpose of requirements analysis:
  - Clearly understand the user requirements,
  - Detect inconsistencies, ambiguities, and incompleteness.
- Incompleteness and inconsistencies:
  - Resolved through further discussions with the end-users and the customers.

Inconsistent Requirement

- **Some part of the requirement:**
  - **contradicts with some other part.**
- **Example in Chemistry Lab:**
  - One customer says turn off heater and open water shower when temperature > 100 C
  - Another customer says turn off heater and turn ON cooler when temperature > 100 C

Incomplete Requirement

- Some requirements have been omitted:
  - Possibly due to oversight.
- Example:
  - The analyst has not recorded: when temperature falls below 90 C
    - heater should be turned ON
    - water shower turned OFF.

Software Requirements Specification

- Main aim of requirements specification:
  - Systematically organize the requirements arrived during requirements analysis.
  - Document requirements properly.

Software Requirements Specification

- The SRS document is useful in various contexts:
  - Statement of user needs
  - Contract document
  - Reference document
  - Definition for implementation

SRS Document

- The SRS document is known as **black-box specification:**
  - The system is considered as a black box whose internal details are not known.
  - Only its visible external (i.e. input/output) behavior is documented.

SRS Document (CONT.)

- SRS document concentrates on:
  - What needs to be done
  - Carefully avoids the solution ("how to do") aspects.
- The SRS document serves as a contract
  - Between development team and the customer.
  - Should be carefully written

SRS Document (CONT.)

- The requirements at initial stage
  - Written using end-user terminology.
- If necessary:
  - Later a formal requirement specification may be developed from it.

Properties of a Good SRS Document

- It should be concise
  - and at the same time should not be ambiguous.
- It should specify what the system must do
  - and not say how to do it.
- Easy to change.,
  - i.e. it should be well-structured.
- It should be consistent.
- It should be complete.

Properties of a Good SRS Document (cont...)

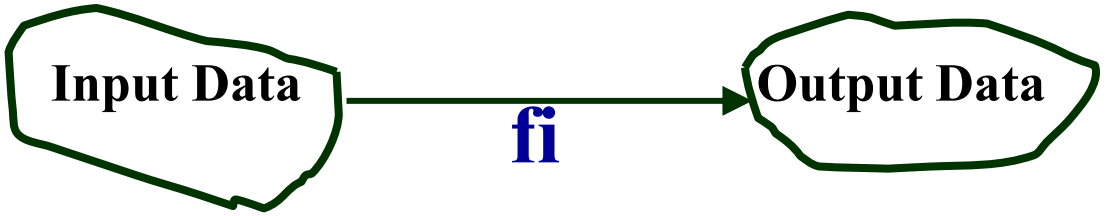
- It should be traceable
  - You should be able to trace which part of the specification corresponds to which part of the design, code, etc and vice versa.
- It should be verifiable
  - e.g. "system should be user friendly" is not verifiable

SRS Document

- SRS document, normally contains three important parts:
  - Functional requirements,
  - Non-functional requirements,
  - Goals of Implementation.

SRS Document (CONT.)

- It is desirable to consider every system:
  - Performing a set of functions {fi}.
  - Each function fi considered as:
  - Transforming a set of input data to corresponding output data.



Functional Requirements

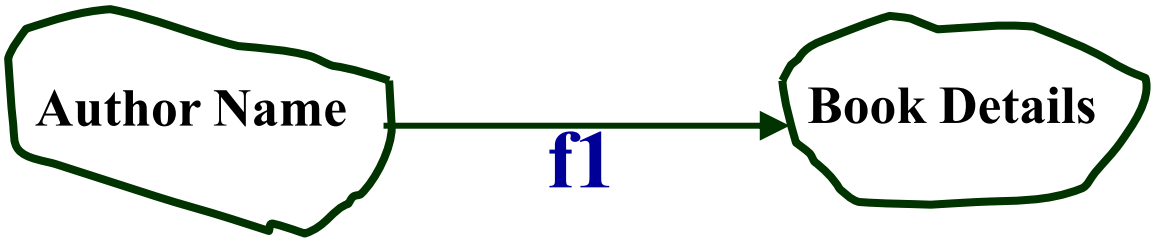
- Functional requirements describe:
  - A set of high-level requirements
  - Each high-level requirement:
    - takes in some data from the user
    - outputs some data to the user
  - Each high-level requirement:
    - might consist of a set of identifiable functions

High-Level Function

- A high-level function:
  - Usually involves a series of interactions between the system and one or more users.
- Even for the same high-level function,
  - There can be different interaction sequences (or scenarios)
  - Due to users selecting different options or entering different data items.

Example: Functional Requirement

- **F1: Search Book**
  - Input:
    - an author's name:
  - Output:
    - details of the author's books and the locations of these books in the library.



Example Functional Requirements

- List all functional requirements
  - with proper numbering.
- Req. 1:
  - Once the user selects the "search" option,
    - he is asked to enter the key words.
  - The system should output details of all books
    - whose title or author name matches any of the key words entered.
    - Details include: Title, Author Name, Publisher name, Year of Publication, ISBN Number, Catalog Number, Location in the Library.

Example Functional Requirements

- Req. 2:
  - When the "renew" option is selected,
    - The user is asked to enter his membership number and password.
  - After password validation,
    - The list of the books borrowed by him are displayed.
  - The user can renew any of the books:
    - By clicking in the corresponding renew box.

Req. 1:

- **R.1.1:**
  - **Input:** "search" option,
  - **Output:** user prompted to enter the key words.
- **R1.2:**
  - **Input:** key words
  - **Output:** Details of all books whose title or author name matches any of the key words.
    - Details include: Title, Author Name, Publisher name, Year of Publication, ISBN Number, Catalog Number, Location in the Library.
  - **Processing:** Search the book list for the keywords

Req. 2:

- **R2.1:**
  - **Input:** "renew" option selected,
  - **Output:** user prompted to enter his membership number and password.
- **R2.2:**
  - **Input:** membership number and password
  - **Output:**
    - list of the books borrowed by user are displayed. User prompted to enter books to be renewed or
    - user informed about bad password
  - **Processing:** Password validation, search books issued to the user from borrower list and display.

Req. 2:

- **R2.3:**
  - **Input:** user choice for renewal of the books issued to him through mouse clicks in the corresponding renew box.
  - **Output:** Confirmation of the books renewed
  - **Processing:** Renew the books selected by the in the borrower list.

Nonfunctional Requirements

- Nonfunctional requirements include:
  - Reliability issues,
  - Performance issues:
    - Example: How fast the system can produce results
      - so that it does not overload another system to which it supplies data, etc.
  - Human-computer interface issues,
  - Interface with other external systems,
  - Security, maintainability, etc.

Goals of Implementation

- Goals describe things that are desirable of the system:
  - But, would not be checked for compliance.
  - For example,
    - Reusability issues
    - Functionalities to be developed in future

Organization of the SRS Document

- Introduction.
- Functional Requirements
- Nonfunctional Requirements
  - External interface requirements
  - Performance requirements
- Goals of implementation