Requirements Analysis and Specification

- Many projects fail:
  - because they start implementing the system without determining whether they are building what the customer really wants.

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 Gathering and Analysis
    - Specification

Requirements Analysis and Specification

- The person who undertakes requirements analysis and specification:
  - known as systems analyst:
  - collects data pertaining to the product
  - analyzes collected data:
    - to understand what exactly needs to be done.
  - writes the Software Requirements Specification (SRS) document.

Requirements Analysis and Specification

- Final output of this phase:
  - Software Requirements Specification (SRS) Document.
- The SRS document is reviewed by the customer.
  - reviewed SRS document forms the basis of all future development activities.

Requirements Gathering

- 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 (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.

Requirements Gathering
(CONT.)

- Some desirable attributes of a good system analyst:
  - Good interaction skills,
  - imagination and creativity,
  - experience.

Analysis of the Gathered Requirements

- After gathering all the requirements:
  - analyze it:
    - 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.
- <u>Example:</u>
  - 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:
  - due to oversight.
- <u>Example:</u>
  - The analyst has not recorded: when temperature falls below 90 C
    - heater should be turned ON
    - water shower turned OFF.

Analysis of the Gathered Requirements (CONT.)

- Requirements analysis involves:
  - obtaining a clear, in-depth understanding of the product to be developed,
  - remove all ambiguities and inconsistencies.

Analysis of the Gathered Requirements(CONT.)

- Several things about the project should be clearly understood by the analyst:
  - What is the problem?
  - Why is it important to solve the problem?
  - What are the possible solutions to the problem?
  - What complexities might arise while solving the problem?

Analysis of the Gathered Requirements(CONT.)

- After collecting all data regarding the system to be developed,
  - remove all inconsistencies and anomalies from the requirements,
  - systematically organize requirements into a Software Requirements Specification (SRS) document.

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

Software Requirements Specification: A Contract Document

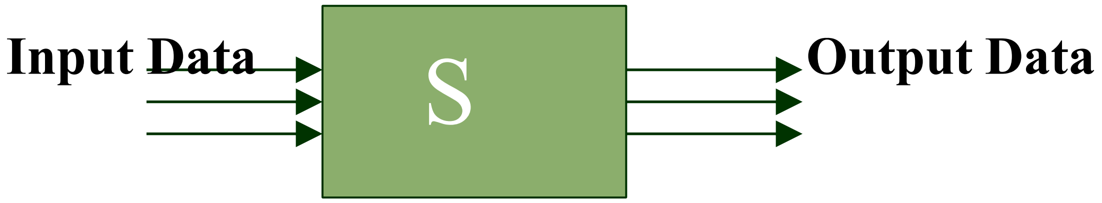
- Requirements document is a reference document.
- SRS document is a contract between the development team and the customer.
  - Once the SRS document is approved by the customer,
    - any subsequent controversies are settled by referring the SRS document.

Software Requirements Specification: A Contract Document

- Once customer agrees to the SRS document:
  - development team starts to develop the product according to the requirements recorded in the SRS document.
- The final product will be acceptable to the customer:
  - as long as it satisfies all the requirements recorded in the SRS document.

SRS Document (CONT.)

- 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) behaviour 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 this stage:
  - written using end-user terminology.
- 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...)

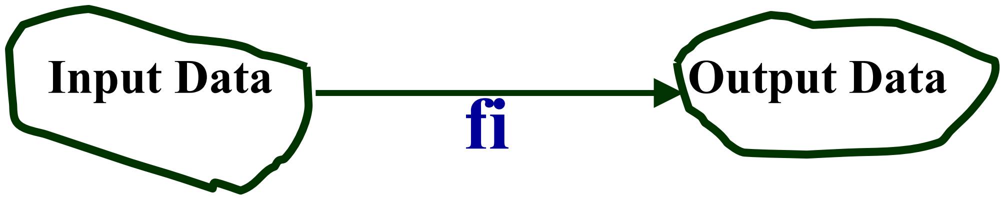
- It should be traceable
  - you should be able to trace which part of the specification corresponds to which part of the design and code, etc and vice versa.
- It should be verifiable
  - e.g. “system should be user friendly” is not verifiable

SRS Document (CONT.)

- SRS document, normally contains three important parts:
  - functional requirements,
  - Non functional requirements,
  - constraints on the system.

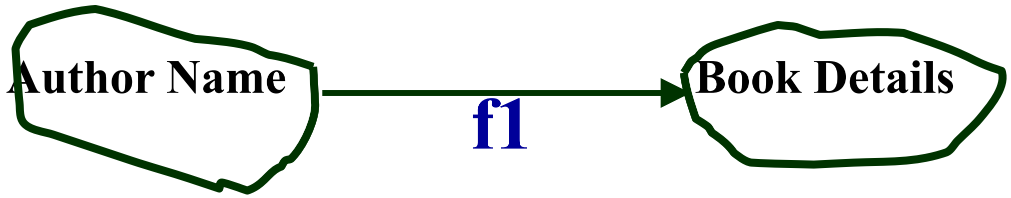
SRS Document (CONT.)

- 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.



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.



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

Functional Requirements

- For each high-level requirement:
  - every function is described in terms of
    - input data set
    - output data set
    - processing required to obtain the output data set from the input data set

Nonfunctional Requirements

- Characteristics of the system which can not be expressed as functions:
  - maintainability,
  - portability,
  - usability, etc.

Nonfunctional Requirements

- Nonfunctional requirements include:
  - reliability issues,
  - performance issues,
  - human-computer interface issues,
  - Interface with other external systems,
  - security, maintainability, etc.

Constraints

- Constraints describe things that the system should or should not do.
  - For example,
    - standards compliance
    - how fast the system can produce results
      - so that it does not overload another system to which it supplies data, etc.

Examples of constraints

- Hardware to be used,
- Operating system
  - or DBMS to be used
- Capabilities of I/O devices
- Standards compliance
- Data representations
  - by the interfaced system

Organization of the SRS Document

- Introduction.
- Functional Requirements
- Nonfunctional Requirements
  - External interface requirements
  - Performance requirements
- Constraints

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.

Examples of Bad SRS Documents

- Unstructured Specifications:
  - Narrative essay --- one of the worst types of specification document:
    - Difficult to change,
    - difficult to be precise,
    - difficult to be unambiguous,
    - scope for contradictions, etc.
- Forward References:
  - References to aspects of problem
    - defined only later on in the text.

Examples of Bad SRS Documents

- Overspecification:
  - Addressing "how to" aspects
  - For example, "Library member names should be stored in a sorted descending order"
  - Overspecification restricts the solution space for the designer.
- Contradictions:
  - Contradictions might arise
    - if the same thing described at several places in different ways.

Summary

- Requirements analysis and specification
  - an important phase of software development:
  - any error in this phase would affect all subsequent phases of development.
- Consists of two different activities:
  - Requirements gathering and analysis
  - Requirements specification

Summary

- The aims of requirements analysis:
  - Gather all user requirements
  - Clearly understand exact user requirements
  - Remove inconsistencies and incompleteness.
- The goal of specification:
  - systematically organize requirements
  - document the requirements in an SRS document.

Summary

- Main components of SRS document:
  - functional requirements
  - Non functional requirements
  - constraints