Slide 04 : Requirements Analysis and Specifications

Made by Md. Mehedi Hasan Rafy

SRS Document Structure

An SRS or Software Requirements Specification typically contains:

1. ***Functional Requirements***

   The functional requirements part discusses the functionalities required from the system.

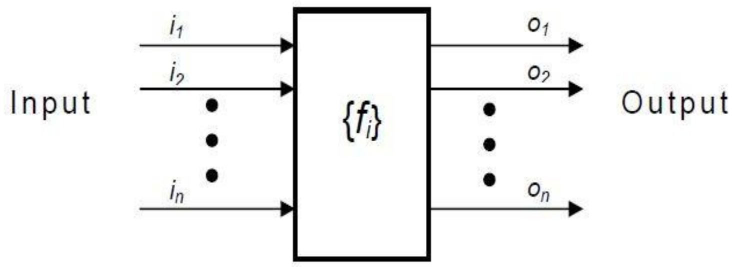

Fig: The functional view of the system

The system is considered to perform a set of high-level functions f_i . Each function f_i of the system can be considered as a transformation of a set of input data i_i to the corresponding set of output data o_i . The user can get some meaningful piece of work done using a high-level function.

***Example:* Withdraw Cash from an ATM**

***R1:* Withdraw Cash**

Description: The system first determines the type of account and the account number from which the user wishes to withdraw cash. It then checks whether the requested amount is available. If sufficient funds exist, it dispenses the cash; otherwise, it displays an error message.

- ***R1.1: Select Withdraw Amount Option***
  Input: User selects “Withdraw Amount” option.
  Output: System prompts the user to enter the account type.

- ***R1.2: Select Account Type***
  Input: User selects the account type.
  Output: System prompts the user to enter the withdrawal amount.

- ***R1.3: Enter Required Amount***

  Input: Withdrawal amount in integer values greater than 100 and less than 10,000, in multiples of 100.

  Output: Cash dispensed and transaction statement printed.

- ***R1.4:* Processing the Transaction**

  Description: The system debits the amount from the user's account if sufficient balance is available; otherwise, it displays an error message.

2. ***Non-functional Requirements***

   Non-functional requirements deal with the characteristics of the system which can not be expressed as functions such as the maintainability, portability, or usability of the system etc.

   In the withdrawal of cash from ATM, transaction process must be completed within 5 seconds or screen instruction should be clear for the first time users are good examples of non-functional requirements of the ATM system.

3. ***Goals of Implementation***

   Records general development guidelines to help balance design trade-offs. May include plans for future functionality revisions, support for new devices, and re-usability considerations, ensuring the system can meet future needs.

   For example, for future biometric authentication can be a goal of implementation for the ATM system.

Properties of a Good SRS

1. **Concise –** A SRS document should be clear, consistent, complete, and unambiguous at the same time. Also it should avoid unnecessary details.
2. **Structured –** Well-organized for easy understanding and modification as requirements evolve over time.
3. **Black-box View –** State *what* the system should do, not *how* it would do. It should specify external behavior.
4. **Conceptual Integrity –** Consistent in content and style for easy understanding.
5. **Response to Undesired Events –** Define how the system should response when exceptional situations arises.
6. **Verifiable –** Each requirement must be verifiable to confirm it has been met.

Problems Without a Proper SRS

- System may not be implemented according to customer needs.
- Developers may work without a clear target.
- Maintenance engineers may struggle to understand functionality of the system.
- Writers can’t create accurate user manuals without understanding SRS document.

Problems With an Unstructured SRS

- Hard to understand the document.
- Difficult to modify it.
- Lacks conceptual integrity in the doucment.
- Prone to ambiguity and inconsistency.