Topics: Software Requirement Specification

Software Requirement Specifications

The production of the requirements stage of the software development process is Software Requirements Specifications (SRS) (also called a requirements document).

Characteristics of good SRS

Requirements Analysis

Requirement analysis is significant and essential activity. We analyze, refine, and study the gathered requirements to make consistent and unambiguous requirements. This activity reviews all requirements and may provide a graphical view of the entire system.

The various steps of requirement analysis are shown in fig:

Steps of Requirements Analysis
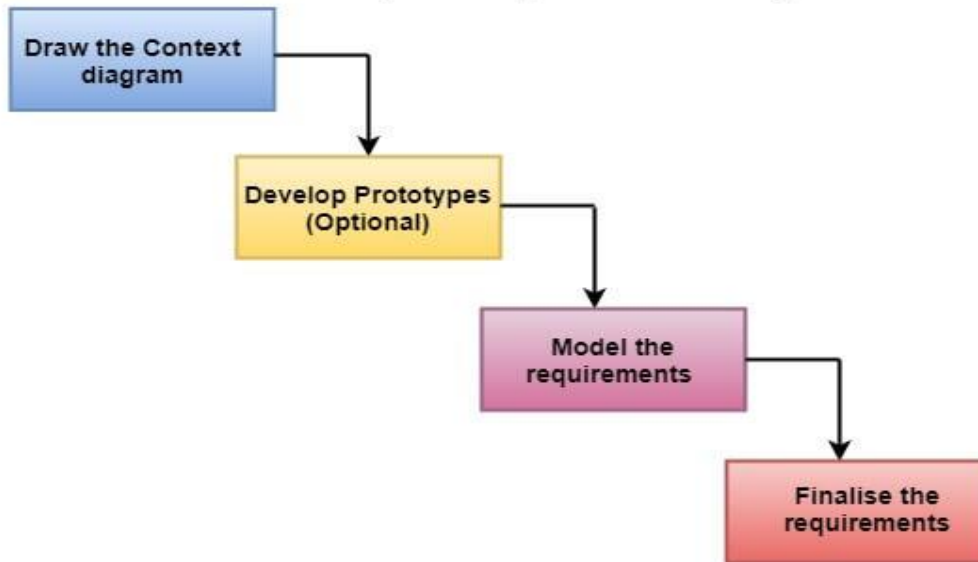
1. ***Draw the context diagram:*** The context diagram is a simple model that defines the boundaries and interfaces of the proposed systems with the external world. It identifies the entities outside the proposed system that interact with the system.
The context diagram of student result management system is given below:
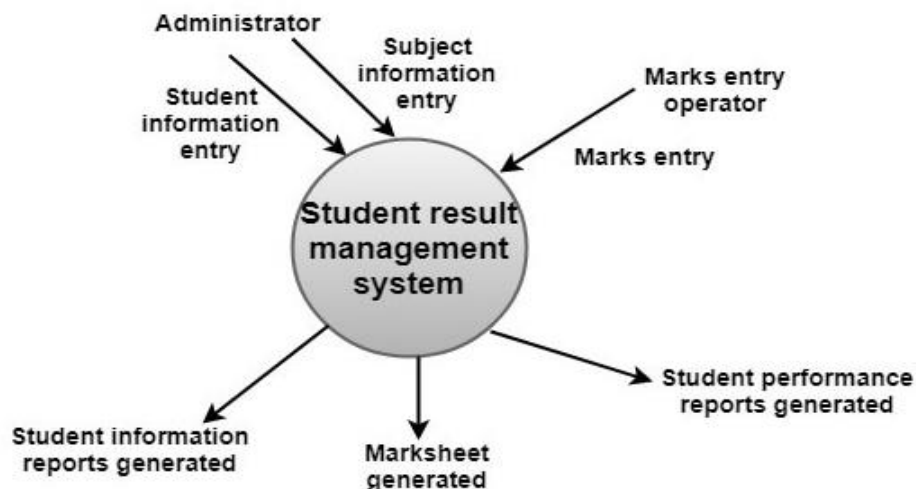
2. ***Development of a Prototype (optional):*** The prototype should be built quickly and at a relatively low cost. Hence it will always have limitations and would not be acceptable in the final system. This is an optional activity.

3. ***Model the requirements:*** This process usually consists of various graphical representations of the functions, data entities, external entities, and the relationships between them. The graphical view may help to find incorrect, inconsistent, missing, and superfluous requirements. Such models include the Data Flow diagram, Entity-Relationship diagram, Data Dictionaries, State-transition diagrams, etc.

4. *Finalize the requirements:* After modeling the requirements, we will have a better understanding of the system behavior. The inconsistencies and ambiguities have been identified and corrected.

Data Flow Diagrams

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It can be manual, automated, or a combination of both.
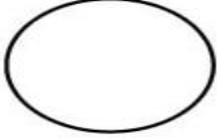
It shows how data centers and leaves the system, what changes the information, and where data is stored.

The objective of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communication tool between a system analyst and any person who plays a part in the order that acts as a starting point for redesigning a system. The DFD is also called as a data flow graph or bubble chart.

The following observations about DFDs are essential:

1. All names should be unique. This makes it easier to refer to elements in the DFD.

2. Remember that DFD is not a flow chart. Arrows is a flow chart that represents the order of events; arrows in DFD represents flowing data. A DFD does not involve any order of events.

3. Suppress logical decisions. If we ever have the urge to draw a diamond-shaped box in a DFD, suppress that urge! A diamond-shaped box is used in flow charts to represents decision points with multiple exists paths of which the only one is taken. This implies an ordering of events, which makes no sense in a DFD.

4. Do not become bogged down with details. Defer error conditions and error handling until the end of the analysis.

Standard symbols for DFDs are shown in below fig:

Symbols for Data Flow Diagrams

© **Circle:** A circle (bubble) shows a process that transforms data inputs into data outputs.
© **Data Flow:** A curved line shows the flow of data into or out of a process or data store.
© **Data Store:** A set of parallel lines shows a place for the collection of data items. A data store indicates that the data is stored which can be used at a later stage or by the other processes in a different order. The data store can have an element or group of elements.
© **Source or Sink:** Source or Sink is an external entity and acts as a source of system inputs or sink of system outputs.

DFD rules:

© Each process should have at least one input and an output.
© Each data store should have at least one data flow in and one data flow out.
© Data stored in a system must go through a process.
© All processes in a DFD go to another process or a data store.