

Lecture 8

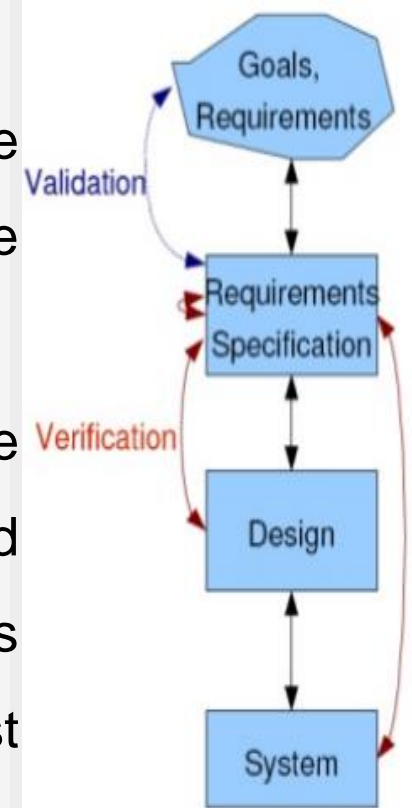
Writing Software Requirements

Requirements Validation

- Check that the **right product is being built**
- Ensures that the software being developed (or changed) will satisfy its stakeholders
- Checks that software requirements specification against stakeholders goals are requirements

Requirements Verification

- Check that the **product is being built right**
- Ensures that each step followed in the process of building the software yields the right products.
- Checks consistency of the software requirements specification artifacts and other software development products (design, implementation,....) against specification.



Software Requirements Specification

A software requirements specification (SRS) is a description of a software system to be developed. It lays out functional and non-functional requirements, and may include a set of use cases that describe user interactions that the software must provide.

Qualities of SRS

- **Correct:** Every requirement given in SRS is a requirement of software.
- **Unambiguous:** Every requirement has exact one interpretation.
- **Complete:** Include all functional, performance, design, external, interface requirements.
- **Consistent:** Internal consistency
- **Ranked for importance and/or stability:** Essential vs desirable

Qualities of SRS

- **Verifiable:** A requirement is verifiable if and only if there exists some finite process with which a person or machine can check that the SW meets the requirement.
- **Modifiable:** SRS must be structured to permit effective modifications (e.g. don't be redundant, keep requirement separate)
- **Traceable:** Origin of each requirement is clear.

Types of Specifications

- Informal

- Free form, natural language
- Ambiguity and lack of organization can lead to incompleteness, inconsistency, and misunderstandings

- Formatted

- Standardized syntax (e.g., UML)
- Basic consistency and completeness checks Imprecise semantics implies other sources of error may still be present.

Types of Specifications

- Formal

- Syntax and semantics rigorously defined.
- Precise form, perhaps mathematical.
- Eliminate imprecision and ambiguity.
- Provide basis for mathematically verifying equivalence between specification and implementation.

Writing Requirements

1. Functional Requirement

1. Module 1

1. Description as Stakeholder explained.

2. Module 2

1. Description as Stakeholder explained.

3. Module 3

4.

2. Non-Functional Requirement

1. Module 1

1. Description as Stakeholder explained.

2. Module 2

3.

Case Study: A Library Information System

- As the size and capacity of the institute is increasing with the time, it has been proposed to develop a Library Information System (LIS) for the benefit of students and employees of the institute. LIS will enable the members to borrow a book (or return it) with ease while sitting at his desk/chamber. The system also enables a member to extend the date of his borrowing if no other booking for that particular book has been made. For the library staff, this system aids them to easily handle day-to-day book transactions. The librarian, who has administrative privileges and complete control over the system, can enter a new record into the system when a new book has been purchased, or remove a record in case any book is taken off the shelf. Any non-member is free to use this system to browse/search books online. However, issuing or returning books is restricted to valid users (members) of LIS only.
- The final deliverable would a web application (using HTML 5), which should run only within the institute LAN. Although this reduces security risk of the software to a large extent, care should be taken no confidential information (e.g. passwords) is stored in plain text. Lot of members should be able to use the system.

Case Study: Program Manager (PM):

- When the PM login to system he should be able perform following tasks. add new counselors, update the counselor information, remove.
- PM should be able to add new course, update the course information, and remove any course. PM should be able to add, remove or update the Database records. PM should be able to add new record for student, update the student information and remove student record from the system. If the PM wants to empty all the tables he should be able to do it by just one click.