# Unit-2
## Software Requirement

Chamundeswari Arumugam
Professor
SSN College of Engineering, Chennai

February 2017

## Outline

- Requirement Engineering
- Functional requirements
- Non-functional requirement
- User requirements
- System requirements
- Software Requirements Document

# Requirement Engineering

- Tasks and techniques that lead to an understanding of requirements is called requirements engineering.
- Requirements engineering is a major software engineering action that begins during the communication activity and continues into the modeling activity.
- It builds a bridge to design and construction.
- It provides the appropriate mechanism for understanding what the customer wants, analyzing need, assessing feasibility, negotiating a reasonable solution, specifying the solution unambiguously, validating the specification, and managing the requirements as they are transformed into an operational system

### Seven distinct tasks of requirement engineering

- Inception, elicitation, elaboration, negotiation, specification, validation, and management.

# Requirement Engineering(contd..)

## Inception

- Identify the business needs, or potential market, or discover service.
- Stakeholders (e.g., business managers, marketing people, product managers) define idea, try to identify the market need, do feasibility analysis, and identify the projects scope.
- Understand the problem, desired solution, and communicate and collaborate between the other stakeholders and the software team.
- Query : How does a software project get started? Does the need evolve over time?

## Elicitation

- Question the customer, the users, and others about the objectives for the system or product ?
- How the system or product fits into the needs of the business?
- How the system or product used on a day-to-day basis?
- Number of problems that are encountered in elicitation are : (1) Problems of scope (2) Problems of understanding (3) Problems of volatility.

## Elaboration

- Inception and elicitation information are expanded and refined.
- Focuses on software function, behavior, and information.
- User scenario method identifies actors, classes, services for each class, relationship between classes.

## Negotiation

- Prioritize requirements
- Assesses the requirement cost and risk
- Modify or eliminate and combine the conflicting requirements

## Specification

- A specification can be a written document, a set of graphical models, a formal mathematical model, a collection of usage scenarios, a prototype, or any combination of these.

# Requirement Engineering(contd..)

## Validation

- Work products are assessed for quality
- Specification are examined for completeness, consistencies, unambiguous, conflicting requirement, unrealistic requirements.
- Sample validation checklist :Does the requirement violate any system domain constraints? Are requirements stated clearly? Can they be misinterpreted?

## Requirements management

- A set of activities that help the project team to identify, control, and track requirements and changes to requirements at any time as the project proceeds.
- Requirements engineering tools assist in requirements gathering, requirements modeling, requirements management, and requirements validation.
- RE tools : IBM Rational RequisitePro, EasyRM, CaliberRM, etc.

# Functional Requirement

- Introduction to FR and NFR

**FR**

- Statements of services the system should provide.
- How the system should react to particular inputs?
- How the system should behave in particular situations?
- It may also explicitly state what the system should not do.

**NFR**

- Constraints on the services or functions offered by the system.
- Timing constraints, constraints on the development process, and constraints imposed by standards.
- Applied to the system as a whole, rather than individual system features or services.

# Functional Requirement(Contd..)

## FR depends on :

- Type of software being developed
- Expected users of the software
- Organization approach in collecting requirement
- User requirements are described in an abstract way for system users
- System requirements describe the system functions, its inputs and outputs, exceptions, etc., in detail.

## Sample FR : Mental health problem

- A user shall be able to search the appointments lists for all clinics.
- The system shall generate each day, for each clinic, a list of patients who are expected to attend appointments that day.
- Each staff member using the system shall be uniquely identified by his or her eight-digit employee number.

## Complete and consistent FR

- Completeness means that all services required by the user should be defined.
- Consistency means that requirements should not have contradictory definitions.

# Functional Requirement(Contd..)

**Imprecision ( Incomplete, inconsistencies and ambiguous) FR may lead to:**

- Many software engineering problems.
- Requirement can be interpreted in a different way
- May delay system delivery
- Increases costs.

**Sample ambiguous FR**

- A user shall be able to search the appointments lists for all clinics.
  **Consequences** : Mental health problems are sometimes confused. They may have an appointment at one clinic but actually go to a different clinic.

**Large and complex FR**

- Easy to make mistakes and omissions when writing specifications for complex systems.
- Due to many stakeholders in a large system, inconsistent requirement are included.

# Non-Functional Requirement

## What are NFRs ?

- Requirements that are not directly concerned with the specific services delivered to its users.

- Relate to properties such as reliability, safety, confidentiality, response time, performance, security, availability and store occupancy.

- May define constraints on the system implementation.
  **Example :** Capabilities of I/O devices, data representations used in interfaces with other systems.

- Often more critical than individual functional requirements.

## Importance of NFR

- May affect the overall architecture of a system rather than the individual components

- Failing to meet a non-functional requirement can mean that the whole system is unusable.

- **Example 1:** if an aircraft system does not meet its reliability requirements, it will not be certified as safe for operation;

- **Example 2:** if an embedded control system fails to meet its performance requirements, the control functions will not operate correctly.

## How NFR arise ?

- User needs
- Budget constraints
- Organizational policies
- Interoperability with other software or hardware systems
- Safety regulations or privacy legislation.

## Usability requirements:

- The system should be easy to use by medical staff and should be organized in such a way that user errors are minimized.

## Testability requirements:

- Medical staff shall be able to use all the system functions after four hours of training.

## How to specify NFR in requirement docuemnt?

- Non-functional requirements often conflict and interact with other functional or non-functional requirements.
- To aviod conflict, functional and non-functional requirements are stated separately
- But explicitly highlight requirements that are clearly related, such as performance or reliability.

Table : NFR Metrics

| Property | Measure |
|---|---|
| Speed | Processed transactions/second<br>User/event response time<br>Screen refresh time |
| Size | Mbytes<br>Number of ROM chips |
| Ease of use | Training time<br>Number of help frames |
| Reliability | Mean time to failure<br>Probability of unavailability<br>Rate of failure occurrence<br>Availability |
| Robustness | Time to restart after failure<br>Percentage of events causing failure<br>Probability of data corruption on failure |
| Portability | Percentage of target dependent statements<br>Number of target systems |

## Types of NFR

- **Product requirements:** Specify or constrain the behavior of the software. **Examples** : Memory, speed, failure rate, security requirements, and usability requirements.

- **Organizational requirements :** Policies and procedures in the customers and developers organization. **Examples:** operational process requirements, development process requirements, development environment, process standards

### Fig. Types of NFR
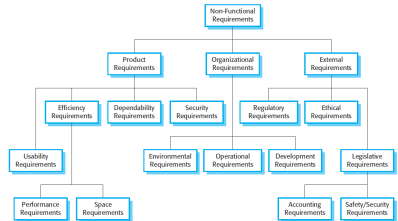


- **External requirements:** Factors external to the system and its development process. **Examples:** regulatory requirements for central bank; legislative requirements for law, and ethical requirements

# User and System Requirement

## User requirement

- High-level abstract requirements
- What services the system is expected to provide to system users and the constraints under which it must operate.
- For example, Security - Limiting access to authorized users

## Diagram
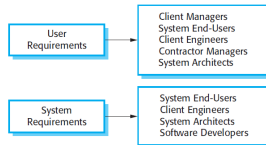


## System requirement

- Detailed description what the system should do
- Provide more specific information about the services, functions and operational constraints of the system that is to be implemented
- Example : Security - Need to include user authentication facilities in the system.

# Software Requirements Document(SRD) or SRS

## What does SRD / SRS contain ?

- Official statement of what the system developers should implement.
- User requirements for a system and a detailed specification of the system requirements.

## Disadvantages of SRS?

- Requirements are unstable
- Requirements keep changing and more effort is required to change the document.
- System evolution

## Advantages of SRS

- Essential to face legal issues
- Requirements are stable it is best
- Supporting document that defines the business and dependability requirements for the system
- Used by many stakeholders
- Compromise between communicating the requirements to customers
- Defining the requirements in precise detail for developers and testers
- Help system designers avoid restrictive design decisions
- Help system maintenance engineers who have to adapt the system to new requirements.

# Software Requirements Document(SRD) or SRS(contd..)

## Level of detail

- Depends on the type of system that is being developed and the development process used.
- Critical systems need to have detailed requirements because safety and security have to be analyzed in detail.
- An inhouse, iterative development process is used, the requirements document can be much less detailed and any ambiguities can be resolved during development of the system.

## Standard

- IEEE standard for requirements documents (IEEE, 1998)
- A generic standard that can be adapted to specific uses.
- This information helps the maintainers of the system and allows designers to include support for future system features.

## Specification

- Define the user requirements and high-level, non-functional system requirements.
- Designers and programmers use their judgment to decide how to meet the outline user requirements for the system.
- Long requirements documents should include table of contents and document index for easily use.

## Software Requirement Specification (SRS)template

1. **Introduction**
   1. Purpose
   2. Document Conventions
   3. Intended Audience and Reading Suggestions
   4. Project Scope
   5. References

2. **Overall Description**
   1. Product Perspective
   2. Product Features
   3. User Classes and Characteristics
   4. Operating Environment
   5. Design and Implementation Constraints
   6. User Documentation
   7. Assumptions and Dependencies

3. **System Features**
   1. System Feature 1
   2. System Feature 2 (and so on)

4. **External Interface Requirements**
   1. 4.1 User Interfaces
   2. 4.2 Hardware Interfaces
   3. 4.3 Software Interfaces
   4. 4.4 Communications Interfaces

5. **Other Nonfunctional Requirements**
   1. Performance Requirements
   2. Safety Requirements
   3. Security Requirements
   4. Software Quality Attributes

6. **Other Requirements**
   1. Appendix A: Glossary
   2. Appendix B: Analysis Models
   3. Appendix C: Issues List

- Seven tasks of requirement engineering
- Difference between User and System requirement
- Differentiate between FR and NFR
- Write a few functional requirements in online exam software.
- Write a few Non-functional requirements in online exam software.
- Prepare a software document for online exam software - Use IEEE SRS template

## Assessment

- Name the seven tasks of requirement engineering.
- Difference between FR and NFR.
- Write few functional requirements of IRCTC.
- Write few non-functional requirements of IRCTC.
- Who are the stakeholders responsible to propose FR & NFR ?
- Give an example of requirement conflict.
- Prepare SRS for online examination.
- List out few user and system requirements.

[1] RogerS. Pressman.
"Software Engineering a Practitiner's Approach"".
*Seventh Edition*, McGraw Hill Higher Education, 2010.

[2] Ian Sommerville.
"Software Engineering"".
*Ninth Edition*, Addison Wesley, 2011.