CSE 333 – Software Engineering

Pranta Sarker

Lecturer

Dept. of CSE, NEUB

Lecture 10

Outline

- Software Requirement
- Requirement Engineering
- Requirement Engineering Process
- Classification of Requirements
- Flow of Requirement Engineering

Software Requirement

- **Abstract** statement of a service.
- Description of **features and functionalities** of the target system.
- It is a **detailed**, **formal definition** of a system function.
- It is said that the hardest part of building a software system is deciding what to build.

Requirement Engineering

- The broad spectrum of **tasks and techniques** that lead to an **understanding of requirements** is called *requirements engineering.*
- RE refers to the process of –
  - Defining, documenting and maintaining requirements in the engineering design process.
  - Establishing the services that the **customer requires** from a system and the **constraints** under which it operates and is developed.
- The goal of requirement engineering is to develop and maintain sophisticated and descriptive ‘System Requirements Specification’ (SRS) document.

Requirement Engineering Process

- Feasibility study
- Requirement Elicitation and Analysis
- Software Requirement Specification
- Software Requirement Validation
- Software Requirement Management

Feasibility Study

- When the client approaches the organization for getting the desired product developed, it comes up with rough idea about what all functions the software must perform and which all features are expected from the software.
- Types of feasibility
  - **Technical:** evaluates the current technologies, which are needed to accomplish customer requirements within the time and budget.
  - **Operational:** assesses the range in which the required software performs a series of levels to solve business problems and customer requirements.
  - **Economical:** decides whether the necessary software can generate financial profits for an organization.

Requirement Elicitation Process

- Requirements gathering
  - Analysts and engineers **communicate with the client and end-users** to know their ideas on what the software should provide, and which features they want the software to include.
- Organizing requirements
  - The developers prioritize and arrange the requirements in order of importance, urgency and convenience.
- Negotiation and discussion
  - If requirements are **ambiguous** or there are some **conflicts** in requirements of various stakeholders, if they are, it is then negotiated and discussed with stakeholders. Requirements may then be prioritized and reasonably compromised.
- Documentation
  - All formal & informal, functional and non-functional requirements are documented and made available for next phase processing.

Requirement Elicitation Techniques

- Interviews
- Surveys
  - Organization may conduct surveys among various stakeholders by querying about their expectation and requirements from the upcoming system.
- Questionnaires
  - A document with pre-defined set of objective questions and respective options is handed over to all stakeholders to answer, which are collected and compiled.
- Task Analysis
  - Team of engineers and developers may analyze the operation for which the new system is required. If the client already has some software to perform certain operation, it is studied, and requirements of proposed system are collected.

Requirement Elicitation Techniques

- Domain Analysis
  - Every software falls into some domain category. The expert people in the domain can be a great help to analyze general and specific requirements.
- Brainstorming
  - An informal debate is held among various stakeholders and all their inputs are recorded for further requirements analysis.
- Prototyping
- Observation
  - Team of experts visit the client's organization or workplace. They observe the actual working of the existing installed systems.
  - They observe the workflow at client's end and how execution problems are dealt.
  - The team itself draws some conclusions which aid to form requirements expected from the software.

Software Requirement Specification

- SRS is a document created by system analyst after the requirements are collected from various stakeholders.
- The requirements received from client are written in natural language.
- It is the responsibility of system analyst to document the requirements in technical language so that they can be comprehended and useful by the software development team.

Software Requirement Specification

- SRS defines how the intended software will interact with –
  - Hardware
  - External interfaces
  - Speed of operation
  - Response time of system
  - Portability of software across various platforms
  - Maintainability
  - Speed of recovery after crashing
  - Security
  - Quality
  - Limitations

Software Requirement Specification

- SRS should come up with following features:
  - **User Requirements** are expressed in natural language.
  - **Technical requirements** are expressed in structured language, which is used inside the organization.
  - **Design description** should be written in Pseudo code.
  - **Format of Forms and GUI** screen prints.
  - **Conditional and mathematical notations** for DFDs etc.

Software Requirement Specification

- A complete Software Requirement Specification must have:
  - Clear
  - Correct
  - Consistent
  - Coherent
  - Comprehensible
  - Modifiable
  - Verifiable
  - Prioritized
  - Unambiguous
  - Traceable

Software Requirement Validation

- After requirement specifications are developed, the requirements mentioned in this document are validated.
- User might ask for illegal, impractical solution or experts may interpret the requirements incorrectly.
- Requirements can be checked against following conditions –
  - If they can be **practically implemented**.
  - If they are valid and as **per functionality and domain** of software.
  - If there are any **ambiguities**.
  - If they are **complete**.
  - If they can be **demonstrated**.

Software Requirement Management

- Requirement management is the process of <u>managing changing requirements</u> during the <u>requirements engineering process and system development</u>.
- <u>New requirements emerge</u> during the process as business needs a change, and a better understanding of the system is developed.
- The <u>priority of requirements from different viewpoints changes</u> during development process.
- The <u>business and technical environment</u> of the system changes during the development.

Classification of Requirements Engineering

- Now, we should try to understand <u>what type of requirements may arise</u> in software requirement elicitation phase and <u>what kind of requirements are expected</u> from the software system.
- Broadly software requirements are classified into 2 categories:
  - Functional requirements and
  - Non-Functional requirements

Functional Requirements

- In functional requirements, a function is nothing but <u>inputs, its behavior, and outputs</u>.
- It can be a calculation, data manipulation, business process, user interaction, or any other specific functionality which defines what function a system is likely to perform.
- Functional software requirements help you to capture the intended <u>behavior of the system</u>.

Functional Requirements

- Some examples of functional software requirements:
  - The software automatically validates customers against the ABC Contact Management System.
  - The Sales system should allow users to record customers sales.
  - The background color for all windows in the application will be blue and have a hexadecimal RGB color value of 0x0000FF.
  - Only Managerial level employees have the right to view revenue data.
  - The software system should be integrated with banking API.
  - The software system should pass Section 508 accessibility requirement.

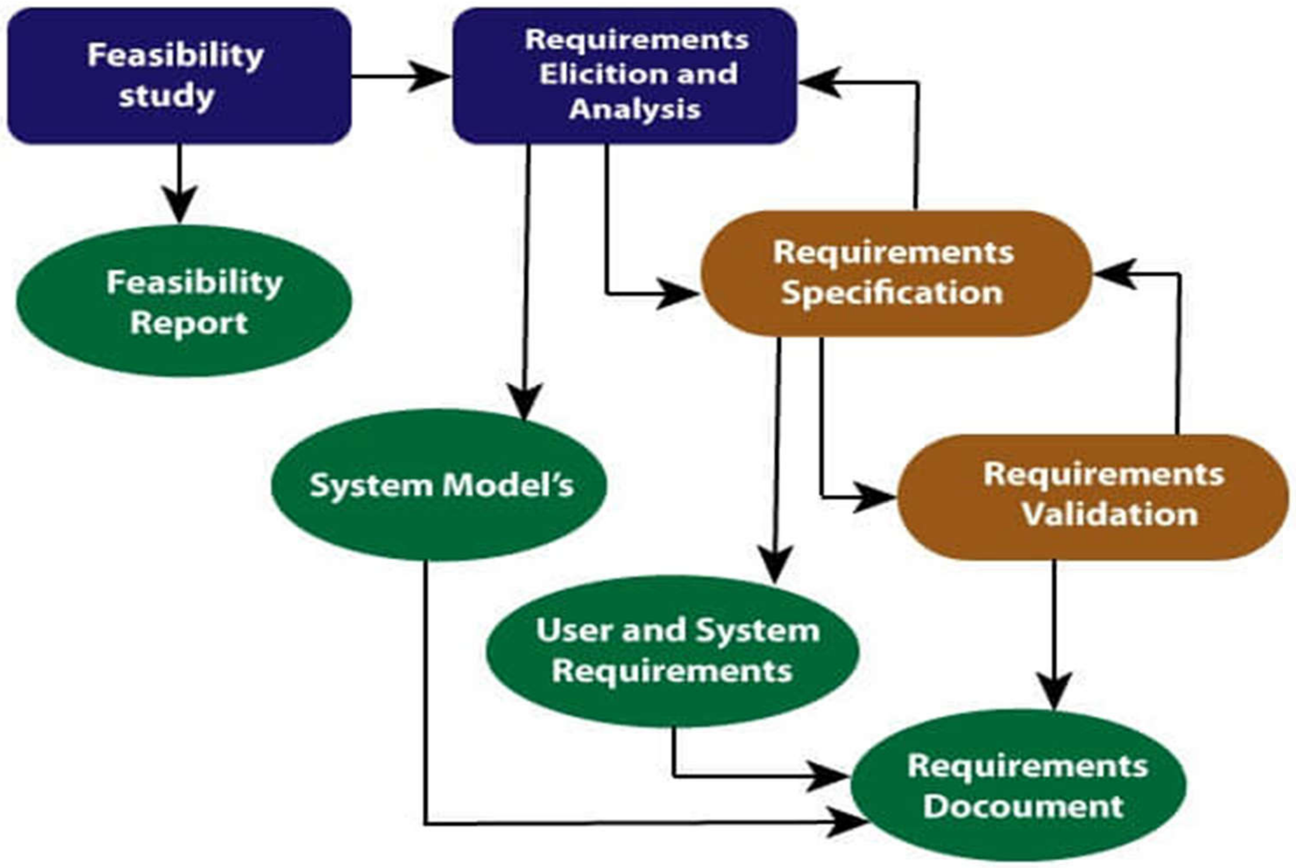
Non-Functional Requirements

- A non-functional requirement defines the quality attribute of a software system.
- They represent a set of standards used to judge the specific operation of a system. Example, how fast does the website load?
- These are the implicit expectations from the product.
- Since these are the expected features and are not specifically documented requirements, they are also called **Quality Attributes**.

Non-Functional Requirements

- Examples:
  - Performance
  - Reliability
  - Error handling
  - Ease of use
  - The software should be **portable**. So, moving from one OS to other OS does not create any problem.
  - **Privacy** of information, the export of restricted technologies, intellectual property rights, etc. should be audited.
  - Users must change the initially assigned login password immediately after the first successful login. Moreover, the initial should never be reused.
  - Employees never allowed to update their salary information. Such attempt should be reported to the security administrator.

Requirement Engineering Process



Requirement Engineering Process

Thank you!!