Software Engineering

Dr. Meisam Nazariani

Email: m_nazariani@aut.ac.ir

February 2022

Outline

1. Introduction
2. The Nature of Software
3. Software Engineering
4. The Software Process
5. Process Models
6. Agile Development
    1. XP
    2. Scrum
7. DevOps
8. **Requirement Engineering**
9. Software Modeling
10. Design Concepts
11. Umbrella Activities
12. Case Studies

Requirement Engineering

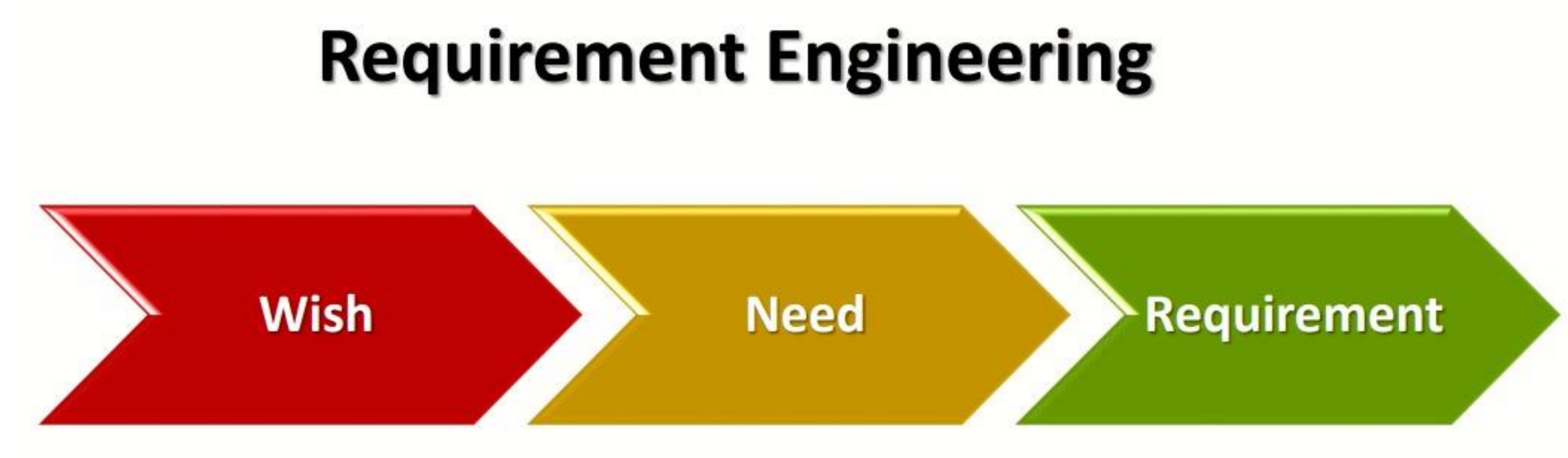
Requirement Engineering

What is it?

The broad spectrum of tasks and techniques that lead to an understanding of requirements is called requirements engineering.

- ❑ From a software process perspective, requirements engineering is a major software engineering action that begins during the communication activity and continues into the modeling activity.
- ❑ It must be adapted to the needs of the process, the project, the product, and the people doing the work.

Requirement Engineering



Requirement Engineering

Wish

Need

Requirement

Requirement Engineering

Requirements engineering builds a bridge to design and construction.

But where does the bridge originate?

Where:

- ✓ business need is defined
- ✓ user scenarios are described
- ✓ functions and features are delineated
- ✓ project constraints are identified

Functional and non-functional requirements

Functional requirements

- Statements of services the system should provide, how the system should react to particular inputs and how the system should behave in particular situations.
- May state what the system should not do

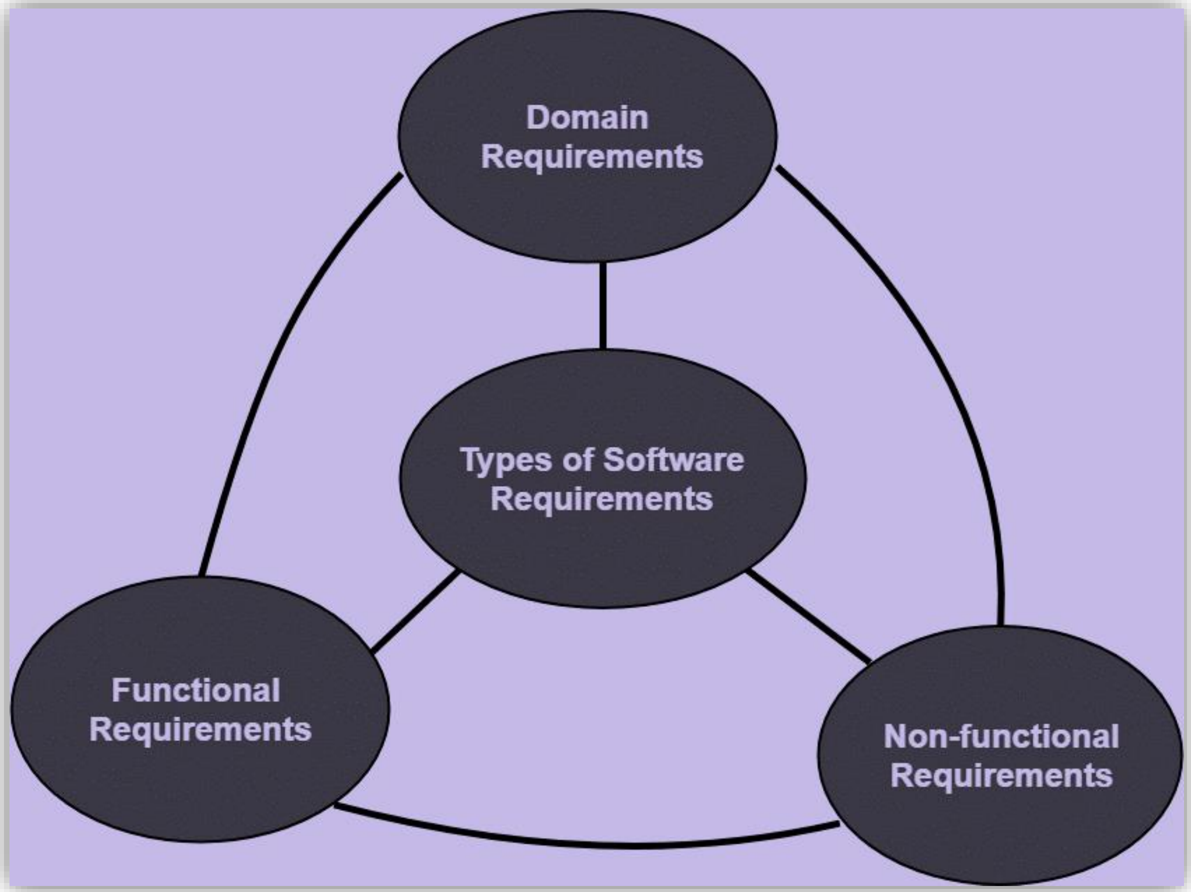
Non-functional requirements

- Constraints on the services or functions offered by the system such as timing constraints, constraints on the development process, standards, etc.
- Often apply to the system as a whole rather than individual features or services

Domain requirements

- Constraints on the system from the domain of operation

Functional and non-functional requirements



Requirement Engineering

Tasks

Requirements engineering encompasses seven distinct tasks:

1) Inception
2) Elicitation
3) Elaboration
4) Negotiation
5) Specification
6) Validation
7) management

It is important to note that some of these tasks occur in parallel and all are adapted to the needs of the project.

1.Inception

- In general, most projects begin when a business need is identified or a potential new market or service is discovered.

- Stakeholders from the business community define a business case for the idea, try to identify the breadth and depth of the market, do a rough feasibility analysis, and identify a working description of the project's scope.

❑ At **project Inception**

you establish a basic understanding of the problem the people who want a solution, the nature of the solution that is desired, and the effectiveness of preliminary communication and collaboration between the other stakeholders and the software team.

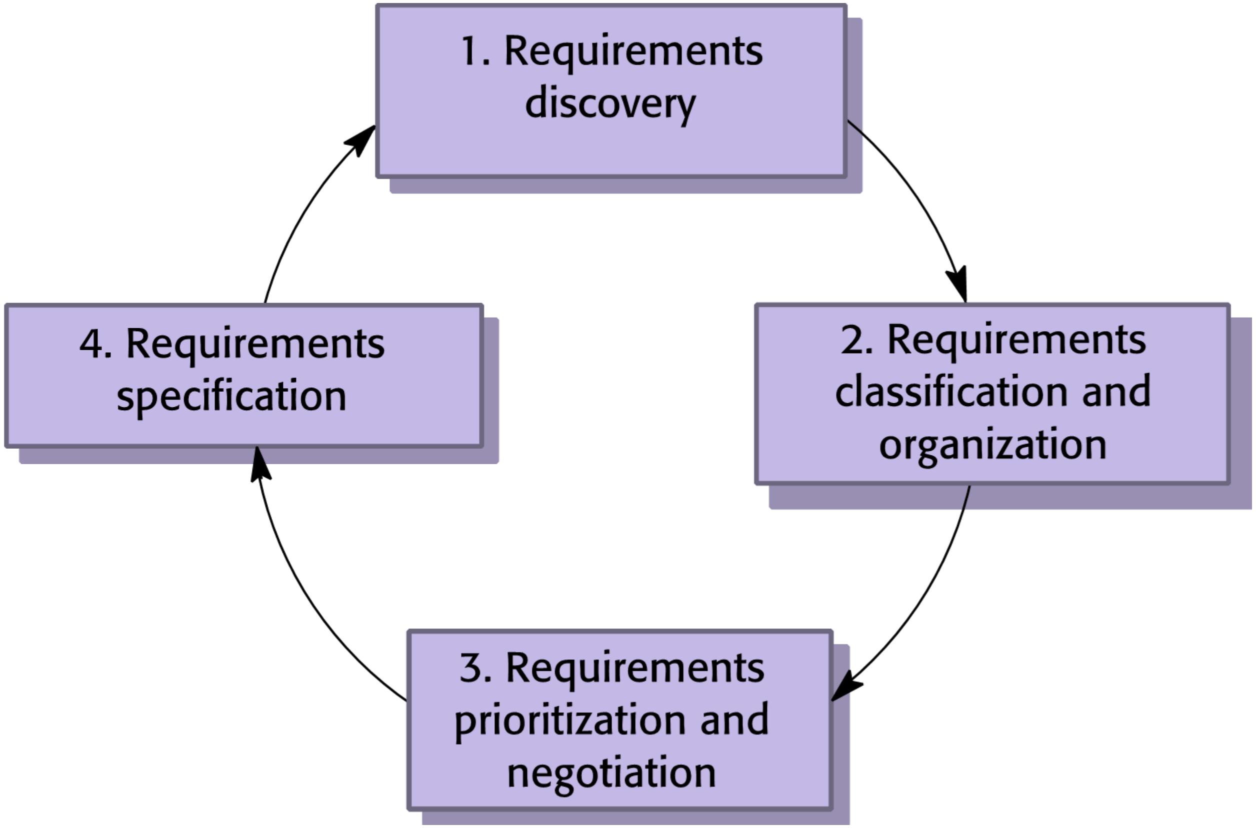
2.Elicitation

- It certainly seems simple enough ask the customer, the users, and others what the objectives for the system or product are, what is to be accomplished, But it isn’t simple — it’s very hard.

❑ At **project Elicitation**

An important part of elicitation is to establish business goals . Your job is to engage stakeholders and to encourage them to share their goals honestly. Once the goals have been captured, a prioritization mechanism should be established, and a design rationale for a potential architecture (that meets stakeholder goals) can be created.

The requirements elicitation process


1. Requirements discovery
2. Requirements classification and organization
3. Requirements prioritization and negotiation
4. Requirements specification

3.Elaboration

- The information obtained from the customer during inception and elicitation is expanded and refined during elaboration. This task focuses on developing a refined requirements model that identifies various aspects of software function, behavior, and information

❑ At project Elaboration

Elaboration is driven by the creation and refinement of user scenarios that describe how the end user will interact with the system. Each user scenario is parsed to extract analysis classes business domain entities that are visible to the end user. The attributes of each analysis class are defined, and the services that are required by each class are identified. The relationships and collaboration between classes are identified, and a variety of supplementary diagrams are produced.

4.Negotiation

- ○ It's relatively common for different customers or users to propose conflicting requirements, arguing that their version is "essential for our special needs." You have to reconcile these conflicts through a process of negotiation

❑ At project Negotiation

Customers, users, and other stakeholders are asked to rank requirements and then discuss conflicts in priority. Using an iterative approach that prioritizes requirements, assesses their cost and risk, and addresses internal conflicts, requirements are eliminated, combined, and or modified so that each party achieves some measure of satisfaction.

5. Specification

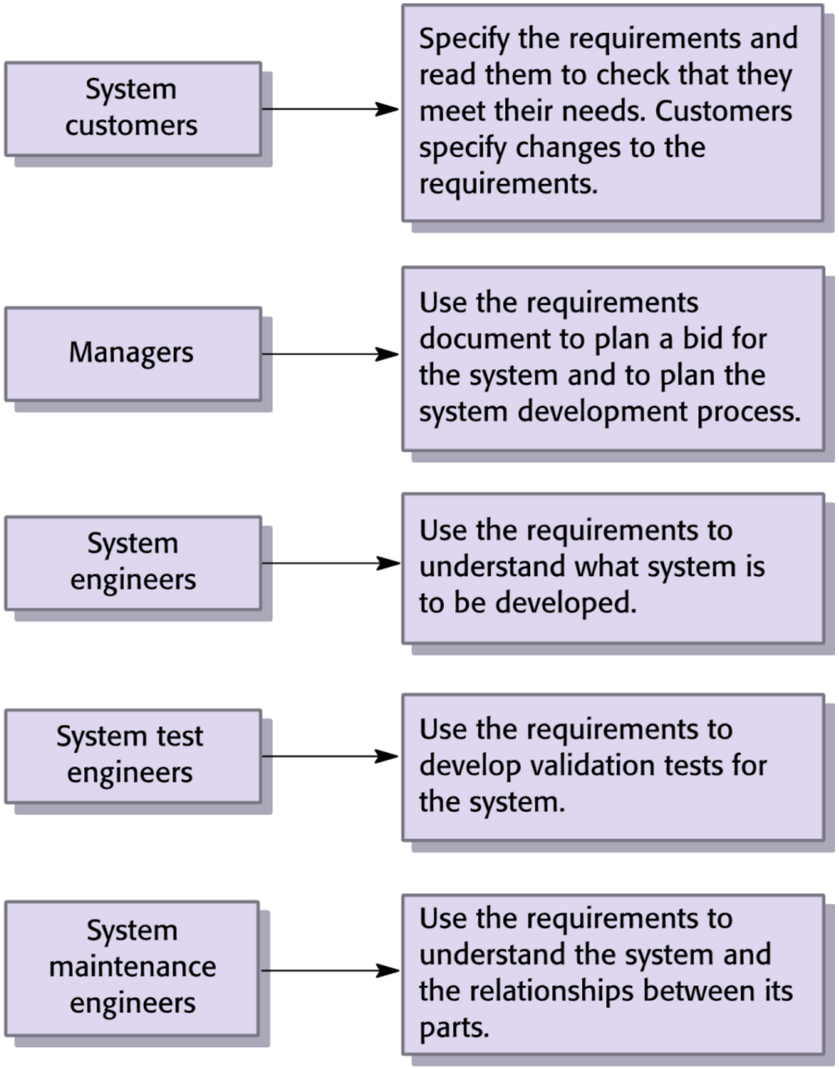
❑ At **project Specification**

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.

Some suggest that a “standard template” should be developed and used for a specification. However, it is sometimes necessary to remain flexible when a specification is to be developed.

For large systems, a written document, combining natural language descriptions and graphical models may be the best approach. However, usage scenarios may be all that are required for smaller products or systems that reside within well-understood technical environments.

Users of a requirements document



System customers → Specify the requirements and read them to check that they meet their needs. Customers specify changes to the requirements.

Managers → Use the requirements document to plan a bid for the system and to plan the system development process.

System engineers → Use the requirements to understand what system is to be developed.

System test engineers → Use the requirements to develop validation tests for the system.

System maintenance engineers → Use the requirements to understand the system and the relationships between its parts.

6. Validation

❑ At project Validation

The work products produced as a consequence of requirements engineering are assessed for quality during a validation step.

Requirements validation examines the specification to ensure that all software requirements have been stated unambiguously, that inconsistencies, omissions, and errors have been detected and corrected; and that the work products conform to the standards established for the process, the project, and the product.

The review team that validates requirements includes software engineers, customers, users, and other stakeholders who examine the specification looking for errors.

7. Requirements management

- ❑ At project Requirements management

Requirements for computer-based systems change, and the desire to change requirements persists throughout the life of the system. Requirements management is a set of activities that help the project team identify, control, and track requirements and changes to requirements at any time as the project proceeds.

Many of these activities are identical to the software configuration management (SCM) techniques.