Software Engineering

Lecture 3

Lectures	Topics
1	Introduction to Software Engineering
2	Software Development Process (SDLC Activities) - SDLC Activity: Specification or Requirement Engineering - SDLC Activity: System Modeling/Design - SDLC Activity: Implementation - SDLC Activity: Testing - SDLC Activity: Evolution - SDLC Activity: Deployment/Installation - SDLC Activity: Maintenance
3	SDLC Activity: Requirement Engineering - Requirement Elicitation - Requirement Analysis and Management - Requirement Validation
4, 5, 6	SDLC Activity: System Modeling/Design - Context Modeling - Data Modeling - Structural/Architectural Modeling - Process Modeling - UI/UX Modeling
7,8,9	SDLC Activity: Implementation (Coding, tools, GIT – Version management, IDE, RESTFUL architecture)
10	SDLC Activity: Testing
11	SDLC Activity: Deployment (tools to deploy, cloud computing)
12	SDLC Activity: Maintenance

Agenda

- To know about Requirement Engineering Phase
    - Requirement Elicitation
        - Fact Finding Techniques
    - Requirement Analysis and Management
        - Types of Requirements
    - Requirement Validation

3.0 Requirement Engineering

- + Expansion of (specification Phase – part of a SDLC phase)
- Requirements engineering helps software engineers to **better understand the problem** they are to solve.
- Requirements may range from a **high-level abstract statement of a service or of a system constraint to a detailed** specification

3.1 Requirement Engineering Processes

Requirement Engineering Processes Involves:

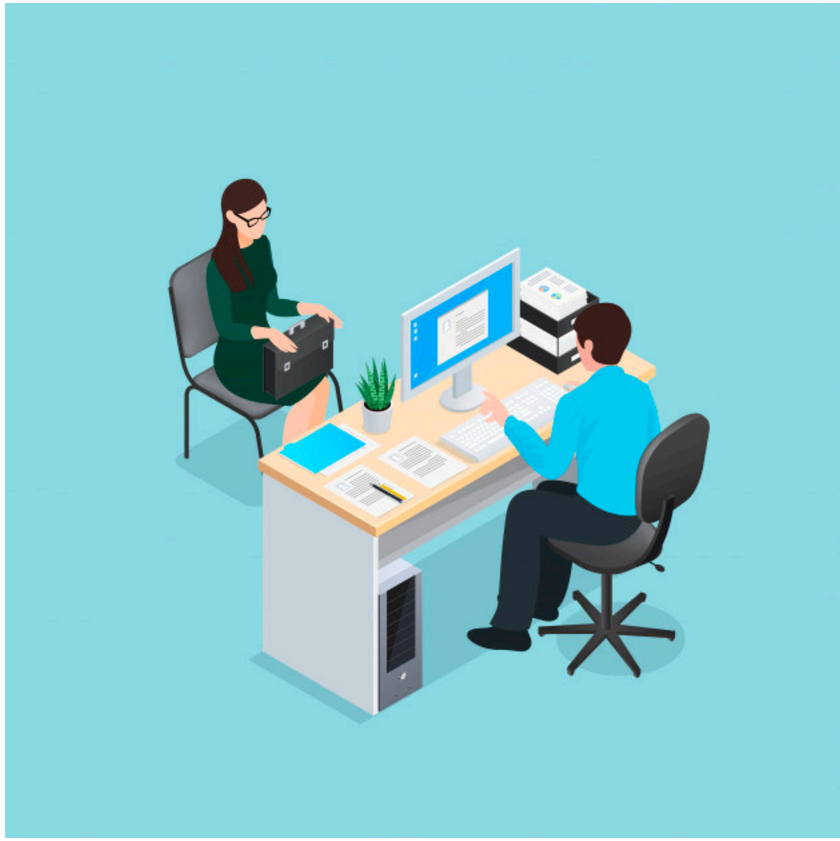
1. Requirement Elicitation
2. Requirement Analysis
3. Requirement Validation

3.1.1 Requirement Elicitation

- Specifying software functions
- Discovery of Requirements using the following techniques (Also called Fact-Finding Techniques)
  1. Interview
  2. Background Reading
  3. Document Sampling
  4. Questionnaire/Survey
  5. Observation
  6. Research

3.1.1.1 Fact Finding Technique: Interview

- One and one
- One to Many (JAD)
- Many to Many (JAD)
- A system analysis interview is a **STRUCTURED or UNSTRUCTURED** meeting between the analyst and a member of staff of the organization being investigated.
- The degree of structure may vary from a fixed set of questions to open-ended covering certain topics.

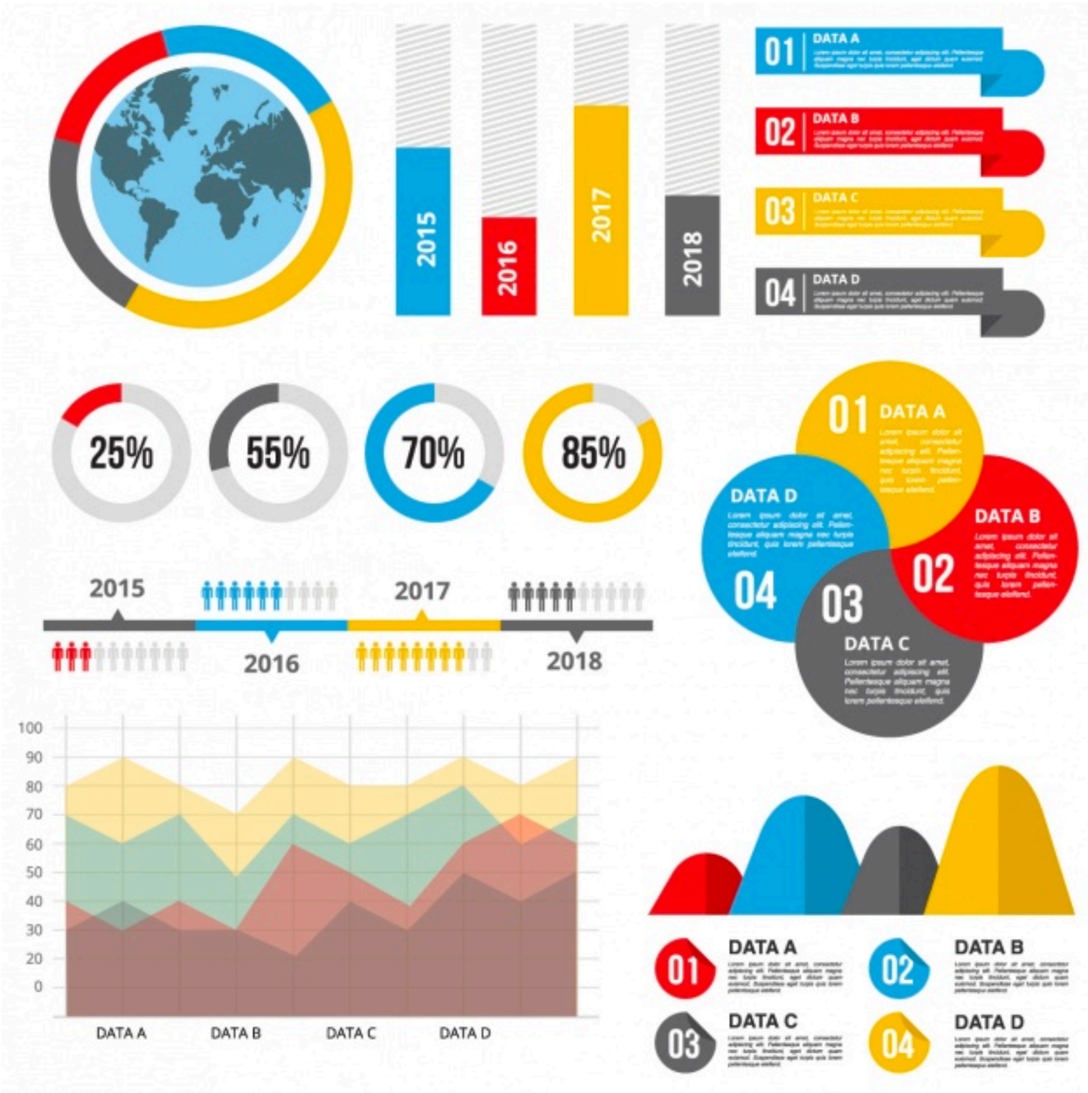


3.1.1.2 Fact Finding Technique: Background Reading

- Understanding of existing system or business operation via:
  - Company reports
  - Charts
  - Policy Manuals
  - Job Descriptions
  - Documentation of existing systems
- Helps you understand, **how the current operation is carried out.**

3.1.1.3 Fact Finding Technique: Document Sampling

- Copies of blank and **completed documents are used to determine the data and information used by staff**, input to and outputs from processes they carry out
- Statistical analysis of documents will help to **estimate volumes of data to be held**, to be input, and output (give the idea of how much the data size will be)
- Filled form (**Invoice, order slip, forms**)



3.1.1.4 Fact Finding Technique: Questionnaire/Survey

- They consist of a series of written questions. The range of replies is usually limited (e.g. Yes or No).
- Some questions do not have a fixed number of responses, and must be left **open-ended** for the respondents to enter what they like.

3.1.1.5 Fact Finding Technique: Observation

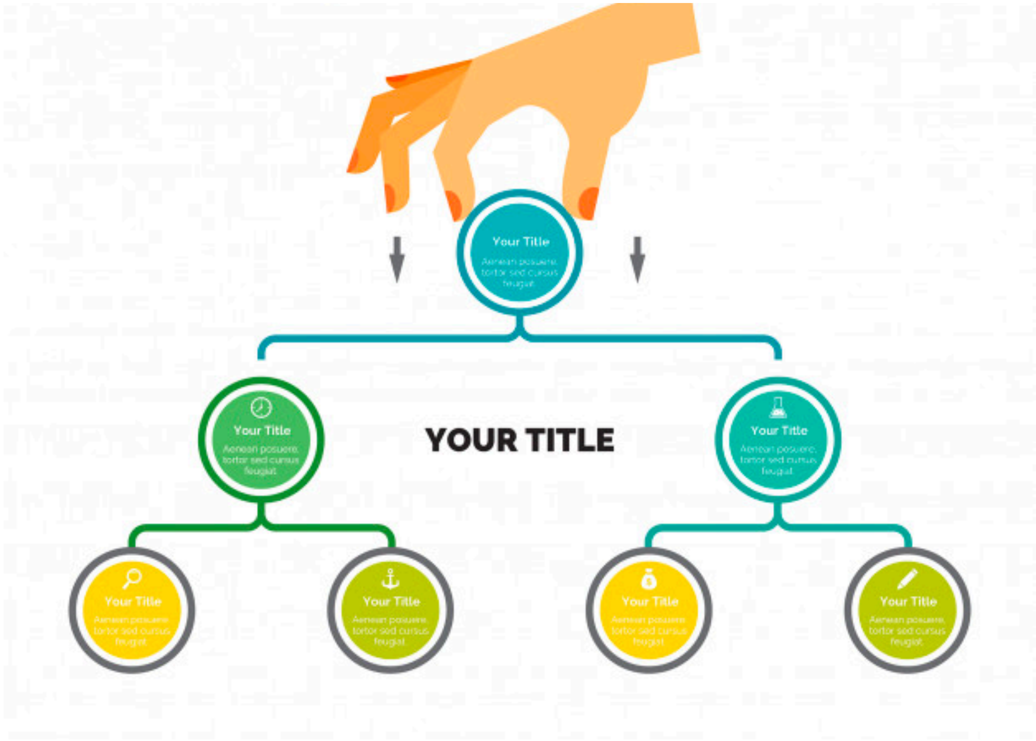
- **Watching people carrying their work in a natural setting** allows to observe the normal aspects of a job as well as exceptional situations (which the system will have to cope with!).
- Observation also allows the analyst to see what information people use to carry out their jobs (is all the necessary information available, at hand?)
- Observation can provide **quantitative data about typical times to perform a task, task duration etc**.
- NOTE: need to think about **Hawthorne effect** too during observation

3.1.1.6 Fact Finding Technique: Research

- External Research
- Research on **similar project**
- Research through: Inter**net, IT magazines, books**

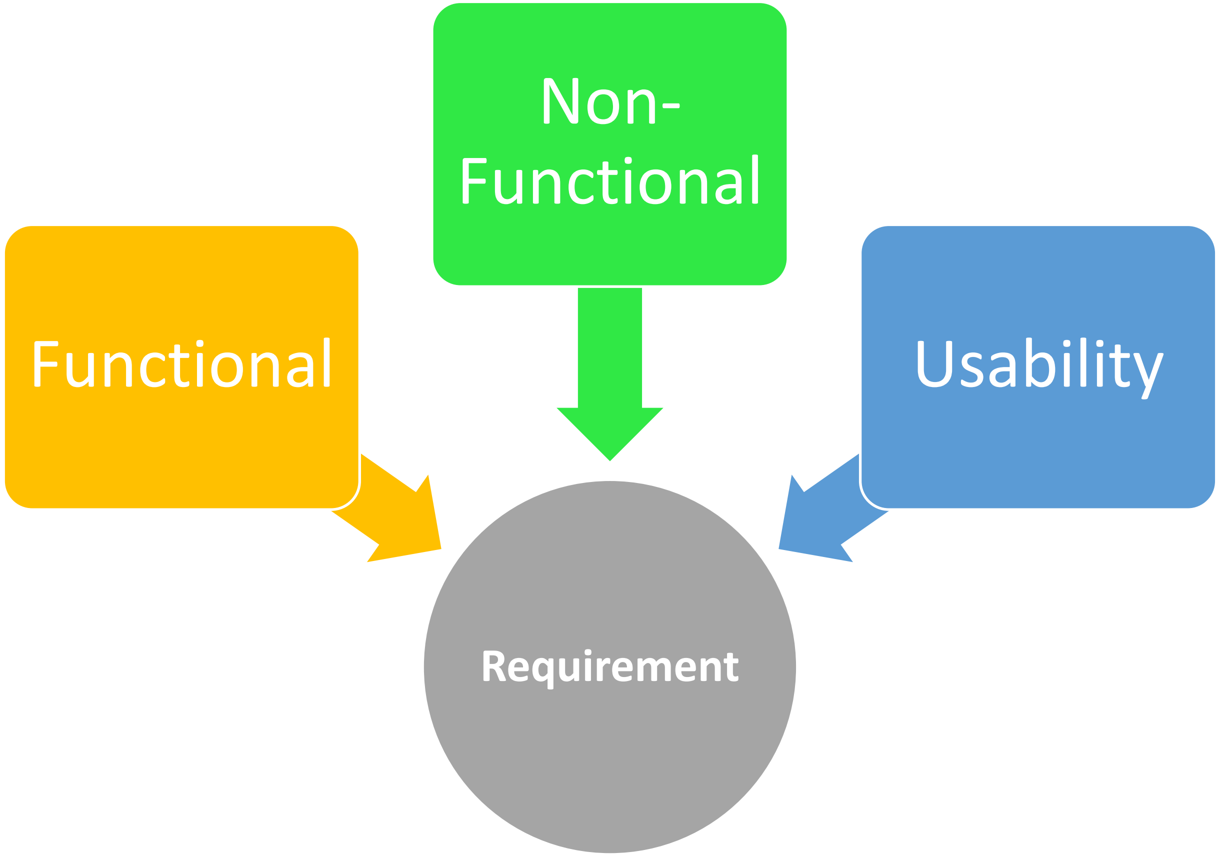
3.2 Requirement Analysis and Management

- Req. Feasibility test
- Requirement partition (Dividing main system in to sub-systems and assigning requirements to each sub-systems)
- Req. Prioritization and Negotiation


Your Title
Aenean posuere, tortor sed cursus feugiat
YOUR TITLE
Your Title
Aenean posuere, tortor sed cursus feugiat
Your Title
Aenean posuere, tortor sed cursus feugiat
Your Title
Aenean posuere, tortor sed cursus feugiat
Your Title
Aenean posuere, tortor sed cursus feugiat
Your Title
Aenean posuere, tortor sed cursus feugiat
Your Title
Aenean posuere, tortor sed cursus feugiat

3.2 Types of Requirements

Requirement classification is carried out during requirement analysis and management.



Non-Functional

Functional

Usability

Requirement

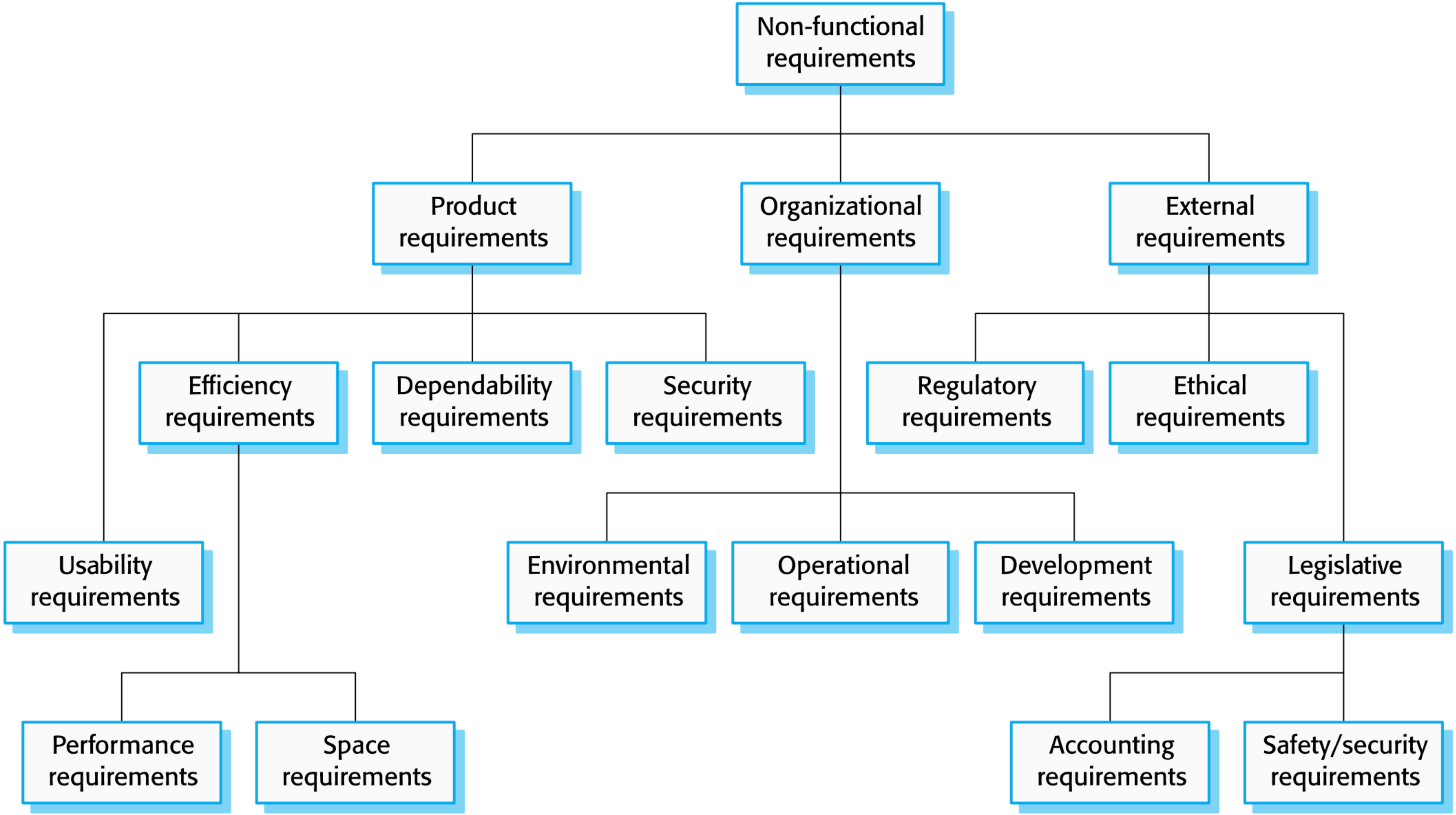
3.2.1 Functional Requirement

- **Describes high level functionality:** Statements of services the system should provide, how the system should react to particular inputs and how the system should behave in particular situations
- In summary functional requirements refer to **WHAT the system must do**.
- May state what the system should not do.
- Example:
  - System should allow user to upload image
  - System should display the uploaded image in image gallery

3.2.2 Non Functional Requirements

- Constraints on the services or functions offered by the system such as **timing constraints**, constraints on the development process, **standards**, etc.
- Concerned with **HOW well it provides the functional requirements**
- These define **system properties** and constraints e.g. **reliability, response time, execution speed, portability and storage requirements**. Constraints are I/O device capability, system representations.
- Non Functional requirements are directly associated with individual requirements or sometime apply to the system as a whole.
- **Example:**
  - System should support only .jpg and .png file format when uploading image.
  - System should display the image in the frame size of 60X60
  - Gallery page should be loaded in less than 3 seconds in an average. However, not exceeding 6 seconds.
- Complete tree structure of non-functional requirements (Figure)

3.2.2 Non Functional Requirements



Non-functional requirements

Product requirements

Organizational requirements

External requirements

Efficiency requirements

Dependability requirements

Security requirements

Regulatory requirements

Ethical requirements

Usability requirements

Environmental requirements

Operational requirements

Development requirements

Legislative requirements

Performance requirements

Space requirements

Accounting requirements

Safety/security requirements

3.2.2 Non-Functional Requirements

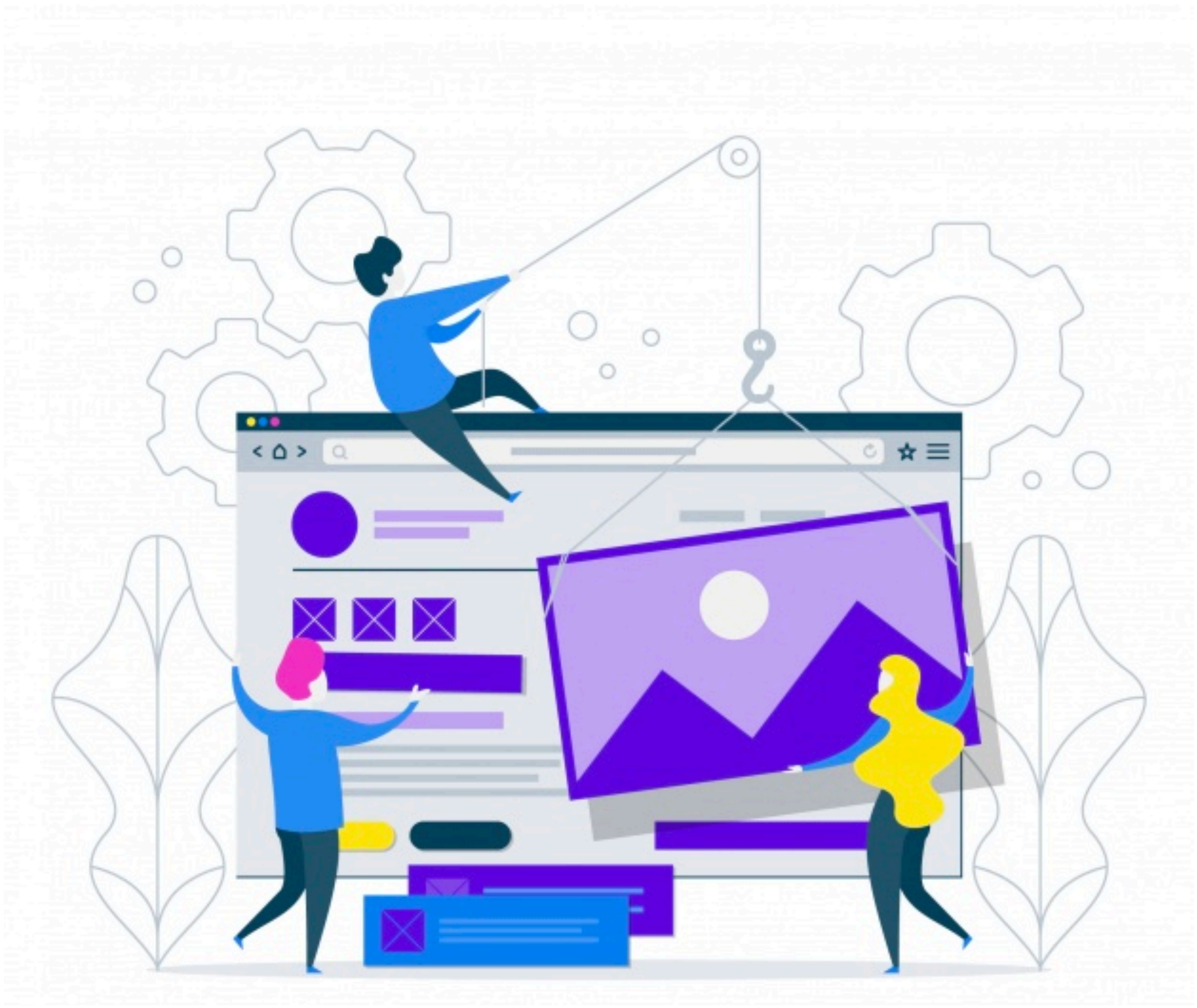
Categorization of Non-Functional Requirements

- **Product Requirements**
  - Requirements which specify that the delivered product must behave in a particular way e.g. **execution speed, reliability,** etc.
- **Organizational Requirements**
  - Requirements which are a **consequence of organizational policies** and procedures e.g. process standards used, implementation requirements, etc.
- **External Requirements**
  - Requirements which arise from factors which are external to the system and its development process e.g. **interoperability requirements (need in json response)** etc.

3.2.3 Usability Requirements

- To ensure that there is a good match between the system and both the users and the tasks they will undertake when using it.
- An attempt to qualify user-**friendliness**
- **Example**:
  - When displaying gallery, display 20 images at a time and display “view more” button at the bottom of the page to load more images.
  - Each image in the gallery should display image caption as an overlay while hovering over image.

3.2.3 Usability Requirements



3.3 Requirement Validation

- Requirements validation examines the specification to ensure that all requirements have been **stated unambiguously**, that **inconsistencies**, **omissions**, and **errors** have been detected and corrected, and that the work product conforms to the standards established for the process, the project, and the product
- **Req. Retrospective test with client**
- Req. Specs (SRS Document)
  - Structure: Glossary/Legend, Classification table of requirements, system architecture
- Spiral view of the requirement engineering process

Thank you