

LECTURE 1

Software requirements Engineering

Software requirements: What, why, and who

Instructor:

Dr. Natalia Chaudhry,

Assistant Professor, PUCIT, University of the Punjab, Lahore.

Recommended readings

- **Chapter 1**, Software Requirements, Wiegers K. & Beatty J., 3rd Ed. Microsoft Press, 2013
- Requirements Engineering, Elizabeth Hull, Ken Jackson and Jeremy Dick. 3rd Ed, Springer-Verlag London Limited, 2011.

What is a requirement?

- Requirements are the descriptions of the services provided by a system and its operational constraints
- It may range from a high level abstract statement to a detailed mathematical specification
- It may be as complex as a 500 pages of description
- It may serve as the basis for a bid for a contract or the basis for the contract itself

What is requirements engineering?

- It is the process of discovering, analyzing, documenting and validating the requirements of the system
- Each software development process goes through the phase of requirements engineering

Why requirements?

- What are the advantages of a complete set of documented requirements?
 - Ensures the user (not the developer) drives system functionalities
 - Helps avoiding confusion and arguments
 - Helps minimizing the changes
- Changes in requirements are expensive. Changing the requirements costs:
 - 3 x as much during the design phase
 - 5-10 x as much during implementation
 - 10-100 x as much after release

Why requirements?

- 2/3 of finished system errors are requirements and design errors
- A careful requirements process doesn't mean there will be no changes later
 - Average project experiences about 25% changes in the requirements
 - This accounts for 70-80% of the rework of the project
 - Important to plan for requirements changes

Levels and types of requirements

Functional requirement	A description of a behavior that a system will exhibit under specific conditions.
Nonfunctional requirement	A description of a property or characteristic that a system must exhibit or a constraint that it must respect.
Quality attribute	A kind of nonfunctional requirement that describes a service or performance characteristic of a product.
System requirement	A top-level requirement for a product that contains multiple subsystems, which could be all software or software and hardware.
User requirement	A goal or task that specific classes of users must be able to perform with a system, or a desired product attribute.
Business requirement	A high-level business objective of the organization that builds a product or of a customer who procures it.

User Requirements:

Example 1: The system should provide a user-friendly interface to ensure that users can easily access and utilize the features.

Example 2: Users require the ability to customize their profiles, including uploading a profile picture and setting privacy preferences.

System Requirements:

Example 1: The system must support at least 500 concurrent users without experiencing a degradation in performance.

Example 2: The software must be compatible with operating systems such as Windows 10, macOS 11, and Ubuntu 20.04.

Business Requirements:

Example 1: The system must comply with industry regulations and standards, ensuring data security and privacy for users.

Example 2: The software should be scalable to accommodate future business growth and increased user demand.

In summary, **user requirements** focus on what end-users need or desire from the system, **system requirements** outline the technical specifications and capabilities of the system, and **business requirements** articulate the high-level goals and objectives that the system needs to achieve in alignment with the overall business strategy.

Users need the ability to export data in CSV and PDF formats from the system's reporting module.

- a) This is a system requirement.
- b) This is a user requirement.**
- c) This is a business requirement.

The system must utilize a relational database management system (RDBMS) and support SQL queries for efficient data retrieval.

- a) This is a system requirement.**
- b) This is a user requirement.
- c) This is a business requirement.

The software should integrate with existing enterprise authentication systems using OAuth 2.0 for secure user access management.

- a) This is a system requirement.
- b) This is a user requirement.
- c) This is a business requirement.**

Users require a responsive and mobile-friendly design to access the system seamlessly on various devices and screen sizes.

- a) This is a system requirement.
- b) This is a user requirement.**
- c) This is a business requirement.

The system must implement encryption algorithms (AES-256) to ensure secure transmission and storage of sensitive user data.

- a) This is a system requirement.**
- b) This is a user requirement.
- c) This is a business requirement.

In a banking application, users have expressed the need for a more secure authentication process. Describe potential user requirements related to enhancing authentication, considering both security and user experience.

Solution: implementing two-factor authentication (2FA), biometric authentication (fingerprint, face recognition), secure password policies, and user-friendly error messages.

You are tasked with designing the system architecture for a high-traffic e-commerce website. Outline the key system requirements that ensure scalability, reliability, and performance under heavy loads.

Solution: Answers may include requirements such as implementing load balancing, horizontal scaling using cloud services, caching mechanisms, database sharding, and a content delivery network (CDN) for static assets.

Functional Requirements:

- 1. Definition:** Functional requirements describe the specific functionalities or features that a system must provide. They outline the actions the system must perform in response to various inputs.
- 2. Focus:** Functional requirements focus on what the system should do to meet the user's needs and achieve the intended goals.
- 3. Examples:**
 - 1.The system must allow users to log in with a valid username and password.
 - 2.The system must be able to process online payments for customer orders.
 - 3.The system must generate monthly financial reports for management.

Non-functional Requirements:

- 1. Definition:** Non-functional requirements are attributes that characterize the system's overall behavior, performance, and qualities. These requirements are not directly related to specific functionalities but rather describe how the system should behave or perform under certain conditions.
- 2. Focus:** Non-functional requirements focus on qualities such as performance, reliability, usability, security, and maintainability.
- 3. Examples:**
 - 1.The system must respond to user inputs within 2 seconds to ensure a responsive user interface.
 - 2.The system must be available 99.9% of the time to ensure high reliability.
 - 3.The system must be able to handle 1000 concurrent users to ensure scalability.
 - 4.The system must encrypt sensitive user data to ensure data security.

Non-functional requirements

They can be further categorized into:

Product requirements

Product behavior

Ex: Timing, performance, memory, reliability, portability, usability

Organizational requirements

Policies and procedures in the customer's and developer's organizations

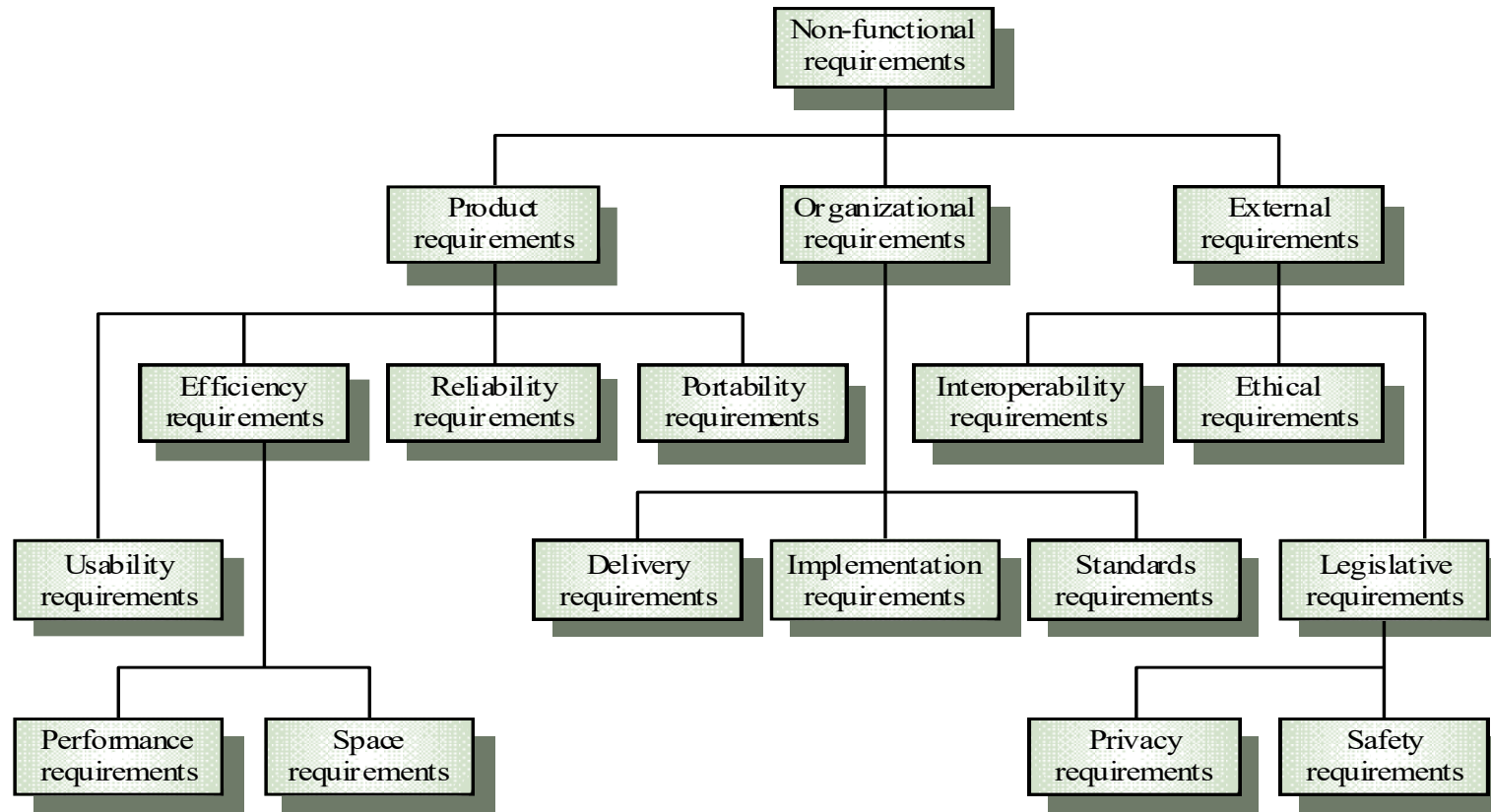
Example: Process requirements, implementation requirements, delivery requirements

External requirements

Factors external to the system and the development process

Example: Interoperability, legislation, ethics

Non-functional requirements



Non-functional requirements

Property	Measure
Speed	Processed transactions/second User/Event response time Screen refresh time
Size	K Bytes Number of RAM chips
Ease of use	Training time Number of help frames
Reliability	Mean time to failure Probability of unavailability Rate of failure occurrence Availability
Robustness	Time to restart after failure Percentage of events causing failure Probability of data corruption on failure
Portability	Percentage of target dependent statements Number of target systems

Levels and types of requirements (contd..)

Constraints: Constraints are limitations or restrictions on the system design. They define the conditions or criteria that must be satisfied, and they often arise from external factors, standards, regulations, or limitations imposed by the environment. Constraints may restrict the design choices and influence how the system should behave or be implemented.

Constraints are typically negative requirements, specifying what the system must not do or the conditions it must avoid. They focus on limitations or boundaries that must be observed during system development and operation.

Features: Features, on the other hand, represent the functionalities or characteristics that the system is expected to provide. They describe what the system is supposed to do or the services it should offer to meet the users' needs. Features are positive requirements that outline the desired behavior or capabilities of the system.

Features are positive requirements that highlight the desired behaviors, capabilities, or functionalities of the system. They describe what the system should accomplish or deliver.

Constraints:

1. Budget Constraint:

1. Example: The project budget is limited to \$500,000.

2. Time Constraint:

1. Example: The system must be developed and deployed within six months.

3. Regulatory Constraint:

1. Example: The system must comply with industry-specific data protection regulations.

4. Compatibility Constraint:

1. Example: The system must be compatible with Windows 10 and macOS 12.

5. Resource Constraint:

1. Example: The system must run on hardware with a minimum of 4 GB RAM.



Features:

1.User Authentication Feature:

- 1. Example:* Users must be able to log in using a username and password.

2.Data Encryption Feature:

- 1. Example:* All sensitive data must be encrypted using AES-256.

3.Reporting Feature:

- 1. Example:* The system should generate monthly performance reports for administrators.

4.Search Functionality Feature:

- 1. Example:* Users should be able to search for products by name, category, and price range.

5.Multi-language Support Feature:

- 1. Example:* The system must support English, Spanish, and French languages.


Software requirements include three distinct levels:
**business requirements, user requirements, and functional
requirements**

Business requirements

- Business requirements describe why the organization is implementing the system—the business benefits the organization hopes to achieve.
- The focus is on the business objectives of the organization or the customer who requests the system.
- E.g.; Suppose an airline wants to reduce airport counter staff costs by 25 percent. This goal might lead to the idea of building a kiosk that passengers can use to check in for their flights at the airport.
- Business requirements typically come from the funding sponsor for a project, the acquiring customer, the manager of the actual users, the marketing department, or a product visionary.

User requirements

- User requirements describe goals or tasks the users must be able to perform with the product that will provide value to someone.
- The domain of user requirements also includes descriptions of product attributes or characteristics that are important to user satisfaction.
- User requirements describe what the user will be able to do with the system.
- An example of a use case is “Check in for a flight” using an airline’s website at the airport. Written as a user story, the same user requirement might read: “As a passenger, I want to check in for a flight so I can board my airplane.”



Ways to represent user requirements include use cases, user stories, and event-response tables.

Ideally, actual user representatives will provide this information.

Functional requirements

- Functional requirements specify the behaviors the product will exhibit under specific conditions.
- They describe what the developers must implement to enable users to accomplish their tasks (**user requirements**), thereby satisfying the **business requirements**.
- This alignment among the three levels of requirements is essential for project success.
- Functional requirements often are written in the form of the traditional “shall” statements: “The Passenger shall be able to print boarding passes for all flight segments for which he has checked in” or “If the Passenger’s profile does not indicate a seating preference, the reservation system shall assign a seat.”

The business analyst (BA) documents functional requirements in a software requirements specification (SRS), which describes as fully as necessary the expected behavior of the software system.

The SRS is used in development, testing, quality assurance, project management, and related project functions.

People call this deliverable by many different names, including business requirements document, functional spec, requirements document, and others.

System requirements

- System requirements describe the requirements for a product that is composed of multiple components or subsystems
- A “system” in this sense is not just any information system. A system can be all software or it can include both software and hardware subsystems.
- People and processes are part of a system, too, so certain system functions might be allocated to human beings.

A good example of a “system” is the cashier’s workstation in a supermarket. There’s a bar code scanner integrated with a scale, as well as a hand-held bar code scanner.

The cashier has a keyboard, a display, and a cash drawer. You’ll see a card reader and PIN pad for your loyalty card and credit or debit card, and perhaps a change dispenser.

The requirements for the system or product as a whole, then, lead the business analyst to derive specific functionality that must be allocated to one or another of those component subsystems, as well as demanding an understanding of the interfaces between them.

Business rules

- Business rules include corporate policies, government regulations, industry standards, and computational algorithms.
- business rules are not themselves software requirements because they have an existence beyond the boundaries of any specific software application.
- However, they often dictate that the system must contain functionality to comply with the pertinent rules



Product vs. project requirements

- Projects certainly do have other expectations and deliverables that are not a part of the software the team implements, but that are necessary to the successful completion of the project as a whole.
- These are **project requirements** but **not product requirements**.
- An SRS houses the product requirements, but it should not include design or implementation details (other than known constraints), project plans, test plans, or similar information.
- Separate out such items so that requirements development activities can focus on understanding what the team intends to build.

Project requirements include:

- Physical resources the development team needs, such as workstations, special hardware devices, testing labs, testing tools and equipment, team rooms, and videoconferencing equipment.
- Staff training needs.
- User documentation, including training materials, tutorials, reference manuals, and release notes.
- Support documentation, such as help desk resources and field maintenance and service information for hardware devices.
- Infrastructure changes needed in the operating environment.
- Requirements and procedures for releasing the product, installing it in the operating environment, configuring it, and testing the installation.

- Requirements and procedures for transitioning from an old system to a new one, such as data migration and conversion requirements, security setup, production cutover, and training to close skills gaps; these are sometimes called transition requirements
- Product certification and compliance requirements.
- Revised policies, processes, organizational structures, and similar documents.
- Sourcing, acquisition, and licensing of third-party software and hardware components.
- Beta testing, manufacturing, packaging, marketing, and distribution requirements.
- Customer service-level agreements.
- Requirements for obtaining legal protection (patents, trademarks, or copyrights) for intellectual property related to the software.



That's it