Course: Software Requirements Engineering

Week 2: Types & Characteristics of Requirements

Lecturer: Yimer Amedie (MSc.)

Addis Ababa Science and Technology University

March, 2026

Contents

- Introduction
- Types of Requirements
- Characteristics of Requirements

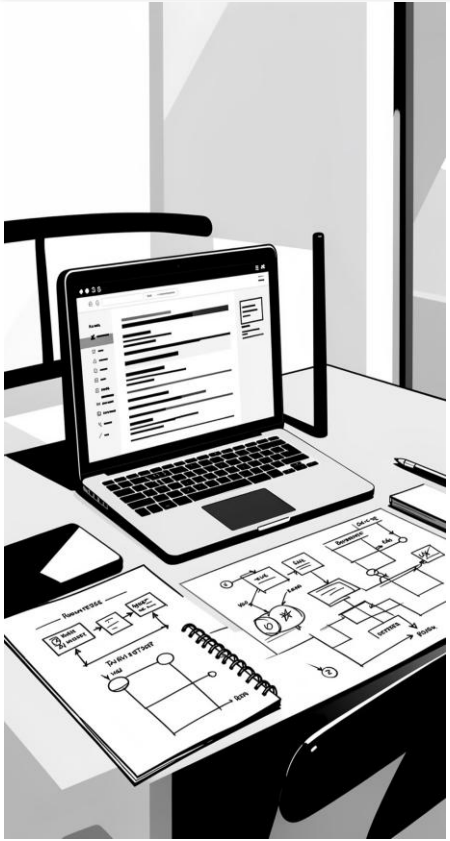
Figure 1. *The software requirements*
Note. Image generated using Sora by OpenAI (2026).

Learning Outcomes

After completing this lesson, you will be able to:

- Differentiate between business, user, and system requirements
- Identify and classify functional and non-functional requirements
- Explain interface requirements and their role in system integration
- Analyze requirements and determine their correctness and quality
- Apply requirement characteristics (INVEST & CUCCVT) to improve requirement quality

Introduction … (1/2)



Software requirements describe:

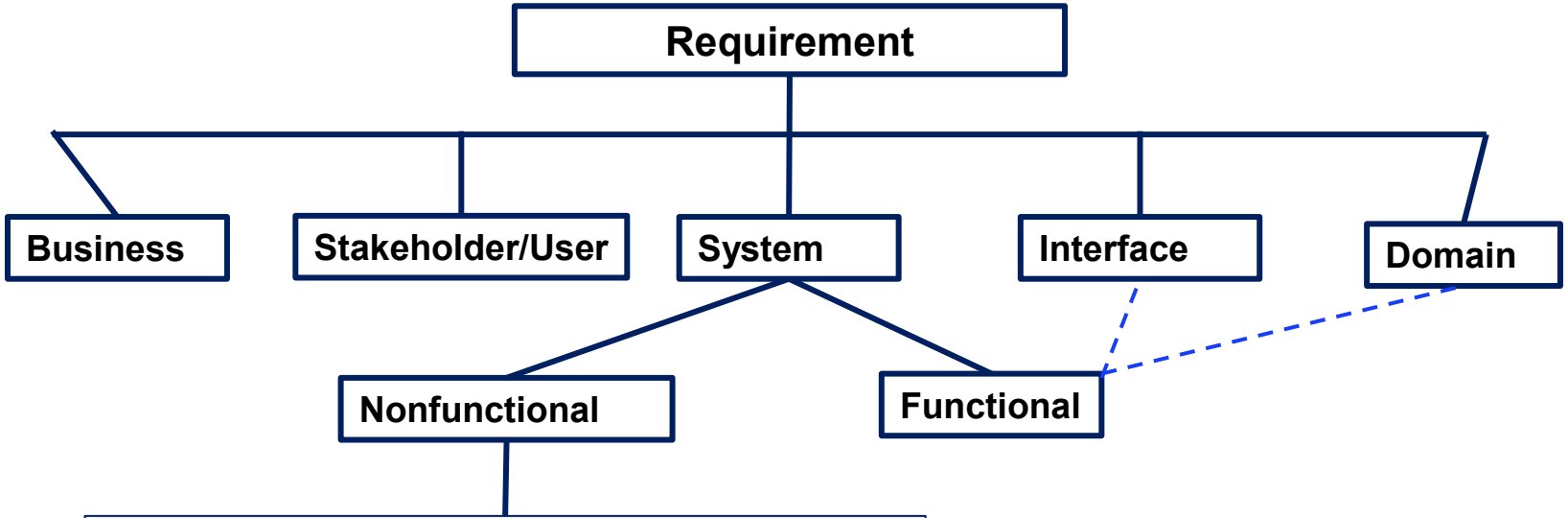
- ✓ **What the system should do**
- ✓ **How the system should behave**

Foundation for
- ✓ Design
- ✓ Development
- ✓ Testing

Introduction … (2/2)

Figure 2. *Software Requirements interaction*
Note. Image generated using Sora by OpenAI (2026).

Types of Requirements



- ✓ Quality Attributes (-ilities)
- ✓ Constraints / Implementation
- ✓ Compliance / Business Rules
- ✓ Other / Cross-Cutting Requirements

Business Requirements … (1/2)

- Describe ***why*** the organization is implementing the system
  - The business benefits the organization hopes to achieve.
- High-level organizational goals.
- **Focus:**
  - Business **objectives/goal** of organization or customer who requests the system.

- ✓ **Reduce fraud**
- ✓ **Ensure compliance**
- ✓ **Improve efficiency**

Business Requirements … (2/2)

Come from

- ✓ Funding sponsor for a project,
- ✓ The acquiring customer,
- ✓ The manager of the actual users,
- ✓ The marketing department, or
- ✓ A product visionary.

- ➢ Record in a ***vision*** *and* ***scope*** *document, or*
- ➢ A project charter, business case, and market requirements document.

Business Requirements: Example -1

Scenario 1: Suppose an airline wants to reduce airport counter staff costs.

Business Requirement:

What the business wants to achieve?

The Airline wants to reduce airport counter staff costs by 25% within one year.

Technical Solution:

How the goal might be achieved?

This goal might lead to the idea of building a kiosk that passengers can use to check in for their flights at the airport

Figure 3. *Example of BR*
Note. Image generated using Sora by OpenAI (2026).

Business Requirements: Example -2

Scenario 2: Suppose an Airline wants to reduce airport counter staff costs, decrease passenger waiting time at check-in counters, and increase adoption of digital check-in to improve efficiency and customer satisfaction

.

Business Requirement:

What the business wants to achieve?

1. Ethiopian Airline wants to reduce airport counter staff costs by 25% within one year.
2. Decrease passenger waiting time at check-in counters by 50%,
3. Increase adoption of digital check-in by 70% to improve efficiency and customer satisfaction

Business Requirements Document (BRD)

- ✓ BRD is a formal outline of a project's business objectives, expected outcomes, and business needs

1. Executive Summary
2. Project Scope
3. Business Objectives & Goals
4. Stakeholder Identification
5. Current vs. Future State (Business Context)
6. Functional Requirements (High Level)
7. Non-functional Requirements
8. Constraints, Dependencies, and Assumptions
9. Project Timeline & Milestones
10. Acceptance Criteria
11. Success Factors & KPIs
12. Risks & Mitigations
13. Approval & Sign-off

BRD components

Project Name: [Enter Project Name]
Document Version: [Version #]
Author(s): [Your Name/Team]
Date: [Date]

Document Revision History

Version	Date	Author	Description of Changes
1.0	[Date]	[Name]	Initial BRD creation

Stakeholder or User Requirements … (1/2)

Figure 4. *Stakeholder needs*
Note. Image generated using Sora by OpenAI (2026).

- ✓ Describe what users and other stakeholders need.
- ✓ *What* the user will be able to do with the system
  - ✓ **Define *who* needs what from the system.**
- ✓ Usually written as a user story in simple language.
  - ✓ Express requirements in plain language that real users can understand.

✓Derived from **business requirements**.

Stakeholder or User Requirements … (2/2)

Characteristics

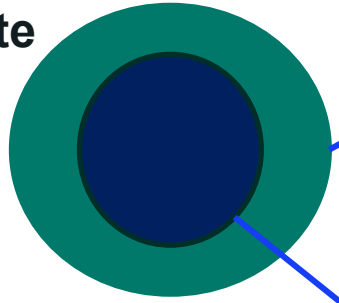
- ✓ High-level, abstract
- ✓ User-focused, outcome-oriented
- ✓ Often includes **“as a [role], I want [goal]”** style

Example

- ❑ As a student, I want to view my course schedule online.
- ❑ A student must be able to register courses online.
- ❑ Check in for a flight.

System Requirements … (1/2)

- Describes **what the complete system must achieve and under what conditions**, serving as the foundation for software, hardware, and overall system design.
  - People and processes



System Requirement
➔What the whole system must achieve.

Software Requirement
➔What the software must do inside that system.

System Requirements … (2/2)

- Specifies
  - **Functional requirements:**
    - What the system must do
  - **Non-functional requirements:**
    - How well the system must perform
  - **Constraints:**
    - Limitations or conditions the system must follow.

Functional Requirements (FRs) … (1/6)

- Specifies **what the system must do** to meet the stakeholder requirements.
- Written in **technical, precise language** suitable for developers.
  - They describe *what* **the developers must implement** to enable users to accomplish their tasks (user requirements), thereby satisfying the **business requirements** (Beatty, 2013).
- Focuses on **system behavior**, not user goals.

FRs … (2/6)

Characteristics:

- ✓ Detailed, implementable, measurable
- ✓ System-focused / feature specific
- ✓ Often starts with **“The system shall…”**

Example

- ❑ **FR-01:** The system shall allow a student to view their course schedule online.
- ❑ **FR-02:** The system shall allow a student to register for courses online.
- ❑ **FR-03:** The system shall allow administrators to create, update, and delete user accounts.

FRs … (3/6)

- A high-level FR can be broken into smaller FRs.

Example:

High-level FR:

“The system shall process online payments.”

Sub-FRs:

✓ “The system shall validate credit card details.”

✓ “The system shall calculate tax.”

✓ “The system shall generate payment receipt.”

FRs … (4/6)

Do’s	Don’ts
☑ Use clear, specific verbs	✗ Use vague terms
☑ Keep FRs about system behavior	✗ Mix in qualities/ constraints
☑ Make it testable	✗ Cannot verify
☑ Avoid redundancy	✗ Duplicate FRs
☑ Include preconditions / context	✗ Ignore conditions
☑ Trace FR to stakeholder need	✗ Write FRs without purpose
☑ Break complex FRs into sub-functions	✗ Overly broad FRs

FRs … (5/6)

Example / Correction
✗ “The system shall manage users properly.” ☑ “The system shall allow administrators to create, update, and delete user accounts.”
✗ “The system shall process payments quickly.” ☑ FR: “The system shall process credit card payments.” ☑ NFR: “Payments shall complete within 3 seconds.”
✗ “The system shall be easy to use.” ☑ “New users shall complete registration within 3 minutes without assistance.”
✗ “The system shall allow users to login” plus “The system shall allow registered users to enter credentials.” ☑ “The system shall allow registered users to login using username and password.”

FRs … (6/6)

Example / Correction … continued
✗ “The system shall allow course registration.” ☑ “Students can register for courses after logging in and before the deadline.”
✗ “The system shall generate reports.” ☑ “System shall generate monthly enrollment reports for the registrar to monitor course capacity.”
✗ “The system shall handle payments.” ☑ “System shall validate payment details, process payments, and generate confirmation receipts.”

FRs: Example

Scenario: Suppose an airline wants to reduce airport counter staff costs by 25 percent.

Functional Requirement:

What the system must do?

The kiosk shall print boarding passes

Non-Functional Requirements (NFRs)

- ✓ Define the quality attributes, constraints, and compliance conditions under which a system must operate.
- ✓ Describe **how well the system performs a function.**
- ➢ **Harder to elicit**
  - ➢ Stakeholders naturally describe functionality
  - ➢ Often assumed or invisible

NFRs

- **Cause of failure** (Ignored, Poorly specified, Discovered too late)
  - Example:
    - System crashes under load (performance ignored)
    - Data breach (security ignored)
    - Users abandon system (usability ignored)

Often apply system-wide (cross-cutting across modules).

NFRs: Characteristics … (1/2)

- **They Constrain Design and Architecture**
  - NFRs heavily influence:
    - Technology choices, System architecture, and Infrastructure decisions
  - **Example:**
    - High availability requirement ➔
      - Requires redundancy, failover design.
    - Security requirement ➔
      - Requires encryption, authentication mechanisms.

NFRs: Characteristics … (2/2)

- **Require special testing to verify**
  - Performance ➔ load, stress), security ➔penetration
  - Usability ➔UAT, reliability ➔ endurance
- **They are often trade-off based**, architectural decisions often balance it
  - Strong security ➔ slower performance
  - High availability ➔ higher cost,
  - Usability improvements ➔ increased complexity

NFRs: Quality Attributes (-ilities)

- Describing how well the system performs or behaves.
- **Example:**
  - **Performance:** “System shall respond to user requests within 2 seconds.”
  - **Reliability:** “System shall have 99.9% uptime.”
  - **Security:** “All user data shall be encrypted in transit and at rest.”
  - **Usability:** “System shall allow new users to complete registration within 3 minutes.”
  - **Maintainability:** “System shall allow updates without downtime.”

NFRs: Constraints / Implementation

- Restrictions on technology, design, or operational environment
- **Example:**
  - **Programming language:** “System shall be implemented in Java 17.”
  - **Hardware:** “System shall run on existing Linux servers.”
  - **Database:** “System shall use PostgreSQL 15.”
  - **Architectural constraints:** “System shall use microservices architecture.”

NFRs: Compliance / Business Rules

- Legal, regulatory, or organizational/business rules the system must obey.
- **Example:**
  - **Legal/Regulatory:** “System shall comply with GDPR for all user data.”
  - **Company Policy:** “Only administrators can delete student records.”
  - **Industry Standards:** “System shall follow ISO 27001 security standards.”
  - **Business Rules:** “Students cannot enroll in more than 6 credit hours per semester.”

NFRs: Categories

Categories

Product Requirements

Specify characteristics of the system itself.

Examples:

- ✓ Performance (response time < 2 seconds)
- ✓ Reliability (99.9% uptime)
- ✓ Security (data must be encrypted)
- ✓ Usability (new users can learn within 30 minutes)

The system shall process 1000 transactions per second.

Organizational Requirements

Come from company policies or standards.

Examples:

- ✓ Must be developed using Java/Oracle DB
- ✓ Must follow ISO 27001 security standards.
- ✓ Must use company-approved cloud infrastructure.

External Requirements

Come from outside the organization.

Examples:

- ✓ Must comply with GDPR.
- ✓ Must follow government banking regulations.
- ✓ Must integrate with external payment gateways.

NFRs: Example

Scenario: Suppose an airline wants to reduce airport counter staff costs by 25 percent.

Nonfunctional Requirement:
How well the system must perform?

The kiosk shall complete check-in within 2 minutes

FRs, NFRs: Activities

1: Identify FR or NFR

1. “The system shall allow users to reset passwords.” → **Functional**
2. “The system shall encrypt all stored user data.” → **NFR**
3. “The system shall process transactions within 3 seconds.” → **NFR**
4. “The system shall generate monthly sales reports.” → **Functional**
5. “The system shall be available 24/7.” → **NFR**

Interface Requirements (IRs) … (1/3)

- Describe how the system interacts with:
  - Users (UI requirements)
  - Other systems (API requirements)
  - Hardware devices, Databases

✓ They define communication behavior and interface expectations.

✓ **Example:** The system shall communicate with the backend via a **REST API.**

Figure 5. *Interface Requirements*
Note. Image generated using Sora by OpenAI (2026).

Interface Requirements (IRs) … (2/3)

✓ Many IRs have NFR constraints.

Example:

- **Interaction:** Use REST API
  - NFR: API response time < 1 second
  - NFR: API must use TLS 1.3 encryption

✓ Interaction requirement defines the connection.

✓ NFR defines the quality of that connection

Interface Requirements … (3/3)

Categories

User Interaction (UI/UX) Requirements

Define how users interact with the system

Examples:

- ✓ The system shall provide error messages in plain language.
- ✓ The user shall confirm before deleting data.
- ✓ The system shall support touch-screen interaction.

These overlap with usability NFRs but focus specifically on interaction behavior

System to System Interaction

Define communication between systems.

Examples:

- ✓ The system shall communicate with the payment gateway via REST API.
- ✓ The system shall use HTTPS protocol.
- ✓ Data exchange shall follow JSON format

Hardware Interaction Requirements

Define system to hardware communication.

Examples:

- ✓ The system shall read input from a barcode scanner.
- ✓ The system shall send print commands to a thermal printer

Activity

Assume: “Online Food Ordering System for university students” with the following requirements. Classify as FR,NFR, interaction. Justify your reasons.

1. The system shall allow users to track orders. ➔ **Functional**
2. The system shall respond within 3 seconds. ➔ **NFR (Performance)**
3. The system shall use HTTPS. ➔ **NFR (Security constraint)**
4. The system shall integrate with Telebirr. ➔ **Interaction requirement**

Characteristics of Requirements

- ✓ A well-written requirement is not just a statement;
  - ✓ it possesses specific attributes.

For Individual Requirements – INVEST

- ✓ Independent
- ✓ Negotiable
- ✓ Valuable
- ✓ Estimable
- ✓ Small
- ✓ Testable

For the Set of Requirements (Overall Specification) – CUCCVT

- ✓ Correct
- ✓ Unambiguous
- ✓ Complete
- ✓ Consistent
- ✓ Verifiable
- ✓ Traceable

Characteristics of Requirements: Activities

Note: **Requirements** must be **measurable**, **testable**, **Vague words** (**fast**, **secure**, **user-friendly**, **reliable**) are **not acceptable**

1: Rewrite each one so it becomes specific and measurable.

1. "The system shall be fast."
2. "The system shall be secure."
3. "The application shall be user-friendly."
4. "The system shall be reliable."
5. "The website shall handle many users."

2: Improve the NFR

"The system shall be scalable."

Activities – Answers

Note: Requirements must be **measurable**, **testable**, **Vague words** (**fast**, **secure**, **user-friendly**, **reliable**) are **not acceptable**

1: Rewrite each one so it becomes specific and measurable.

1. System shall respond to 95% of user requests within 2 seconds.
2. The system shall require multi-factor authentication for all administrative users
3. New users shall complete account registration within 3 minutes without external assistance
4. The system shall maintain 99.9% uptime measured monthly
5. The system shall support 10,000 concurrent users without performance degradation

2: Improve the NFR

The system shall support up to 20,000 concurrent users without increasing average response time beyond 3 seconds

Summary

Type	Focus	Source	Example
Business	business goal or objective	Business stakeholders	Increase online sales
Functional	What system does	Users	Transfer money
NFR	How well / constraints	Quality needs	Response < 2 sec
Interaction	Communication rules	Integration needs	Use REST API
Domain	Industry rules	External domain	Follow tax law

- ✓ There are different categories of requirements.
  - ➢ Business requirements ➔User requirements ➔System Requirements
- ✓ System requirements are the basis for design and implementation

References

1. Beatty, K. W. (2013). Software Requirements (3rd ed.). Washington: Microsoft Press.

Thank You!