

Process of collecting requirements

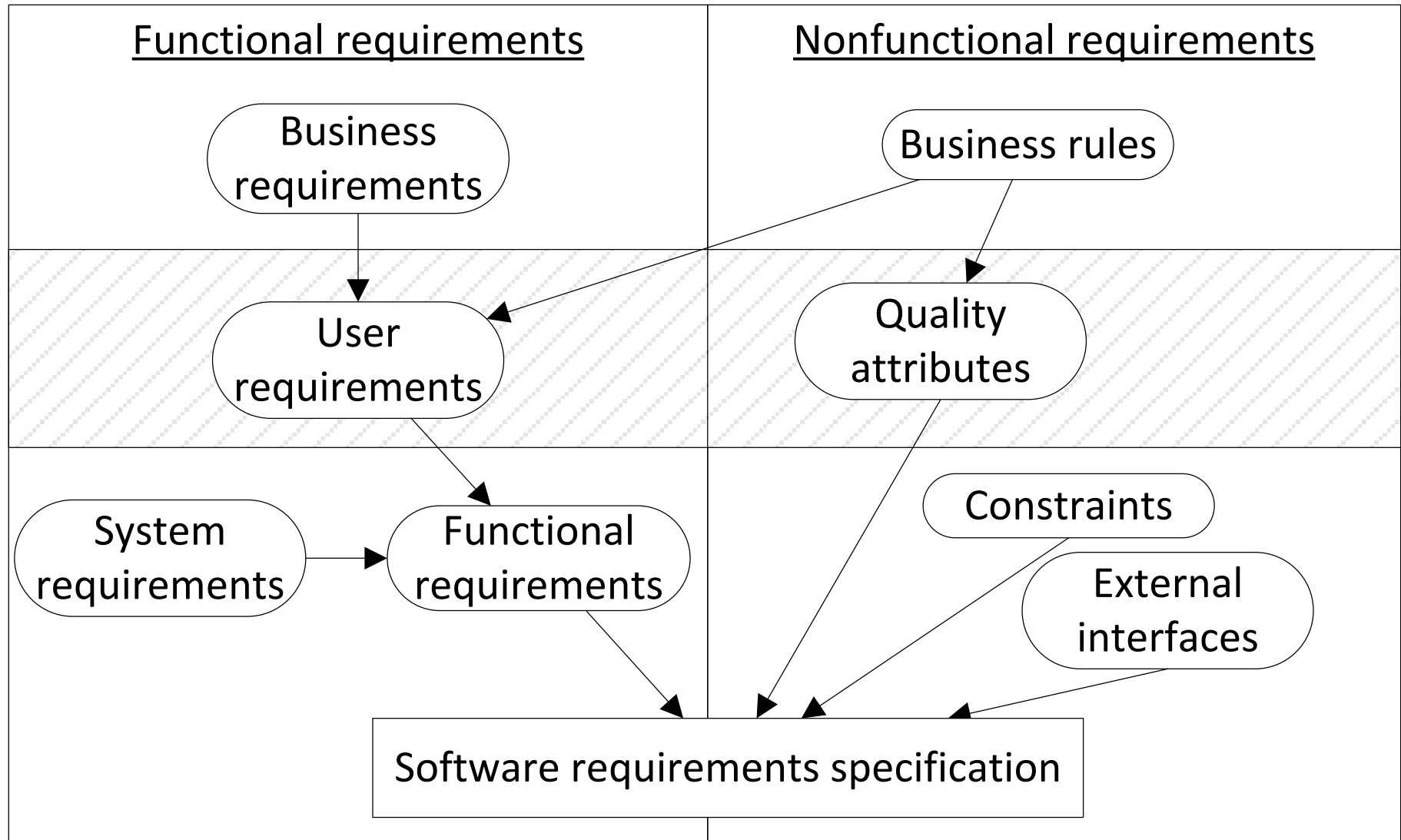
Collect Requirements is the process of determining, documenting, and managing stakeholder needs, requirements, and expectations to meet project objectives. These requirements need to be elicited analyzed, and recorded in enough detail to be measured once project execution begins.

Criteria of a Good Requirement

Criterion	Description
Complete	Each requirement must contain all the information necessary for the reader to understand it. This means providing the information the developer needs to be able to implement it correctly.
Correct	Each requirement must accurately describe a capability that will meet some stakeholder's need and must clearly describe the functionality to be built.
Feasible	It must be possible to implement each requirement within the known capabilities and limitations of the system and its operating environment, as well as within project constraints of time, budget, and staff.
Necessary	Each requirement should describe a capability that provides stakeholders with the anticipated business value, differentiates the product in the marketplace, or is required for conformance to an external standard, policy, or regulation. Each requirement should relate to a business objective that clearly indicates why it's necessary.

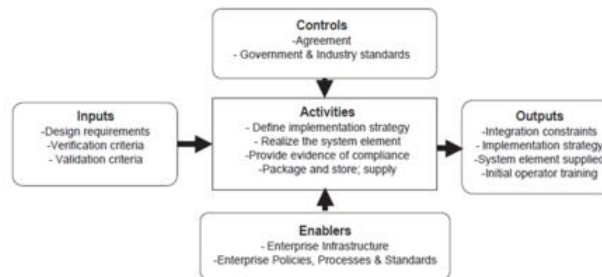
Prioritized	Prioritize business requirements according to which are most important to achieving the desired value. Assign an implementation priority to each requirement by importance and urgency to indicate how essential it is to a particular product release.
Unambiguous	The requirement should state simply and clearly. The requirement should have only one interpretation or meaning.
Verifiable	Each requirement should be specified in the way that the tester could verify whether it is implemented properly
Consistent	Consistent requirements don't conflict with other requirements of the same type or with higher-level business, user, or system requirements.
Modifiable	Modifiability dictates that each requirement be uniquely labeled and expressed separately from others requirements
Traceable	Each requirement should have a unique identifying number that assists in identification, maintaining change history, and providing traceability. It provides bi-directional traceability between various associated requirements, and it enables users to find the origin of each requirement and track every change that was made to this requirement.
Attainable	Each requirement should be met in the system under development
Allocated	The requirement should be allocated to an element of the system design where it can be implemented
Relevance	This requirement does not have to become obsolete over time.

Types of software requirements



Different ways to collect requirements

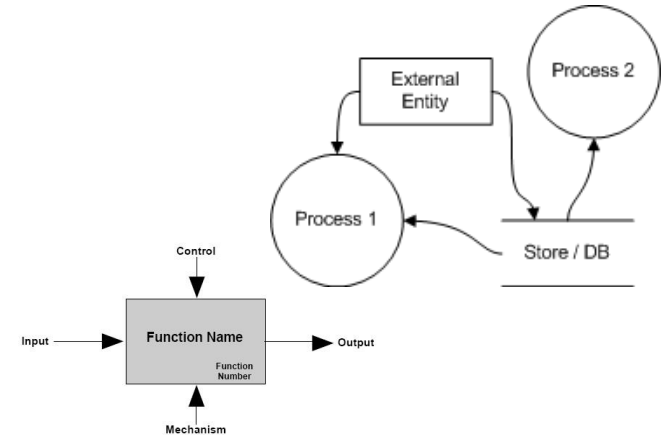
- 1 Interviews .
- 2 Focus groups .
- 3 Facilitated workshops .
- 4 Group creativity techniques .
- 5 Group decision-making techniques .
- 6 Questionnaires and surveys .
- 7 Observations .
- 8 Prototypes .
- 9 Benchmarking
- 10 Context diagrams
- 11 Document analysis



The formats for documenting software requirements



- documents in the natural language;
- graphic models and notations: IDEF0, IDEF3, DFD, UML, SysML;
- formal specifications: SRS.



A software requirements specification (SRS) is a comprehensive description of the intended purpose and environment for software under development. The SRS fully describes what the software will do and how it will be expected to perform.

2. OVERALL SYSTEM.....	
2.1 Product Perspective.....	
2.2 Product Features.....	
2.3 User Classes and Characteristics.....	
2.4 Operating Environment.....	
2.5 Design and Implementation Constraints.....	
2.6 User Documentation.....	
2.7 Assumptions and Dependencies.....	
3. System Features.....	
3.1 System Feature 1.....	
3.2 System Feature 2 (and so on).....	
4. External Interface Requirements.....	
4.1 User Interfaces.....	
4.2 Hardware Interfaces.....	
4.3 Software Interfaces.....	
4. Communications Interfaces.....	
Other Nonfunctional Requirements.....	
Performance Requirements.....	

Pattern for SRS according by standard IEEE 830

1. Introduction

- 1.1 Purpose
- 1.2 Document conventions
- 1.3 Intended audience
- 1.4 Additional information
- 1.5 Contact information/SRS team members
- 1.6 References

2. Overall Description

- 2.1 Product perspective
- 2.2 Product functions
- 2.3 User classes and characteristics
- 2.4 Operating environment
- 2.5 User environment
- 2.6 Design/implementation constraints
- 2.7 Assumptions and dependencies

3. External Interface Requirements

- 3.1 User interfaces
- 3.2 Hardware interfaces
- 3.3 Software interfaces
- 3.4 Communication protocols and

interfaces

4. System Features

- 4.1 System feature A
 - 4.1.1 Description and priority
 - 4.1.2 Action/result
 - 4.1.3 Functional requirements
- 4.2 System feature B

5. Other Nonfunctional Requirements

- 5.1 Performance requirements
- 5.2 Safety requirements
- 5.3 Security requirements
- 5.4 Software quality attributes
- 5.5 Project documentation
- 5.6 User documentation

6. Other Requirements

Appendix A: Terminology / Glossary / Definitions list

Appendix B: To be determined

Requirements Traceability Matrix

Requirements Traceability Matrix (RTM) is a table that links requirements to their origin and traces them throughout the project life cycle and making sure that the documented plan is being delivered at the end of the project. It links each requirement to the business and project objectives, thus ensuring that each requirement adds business value.

The RTM is usually developed in concurrence with the SRS.

Requirement traceability Matrix – Parameters (example 1):

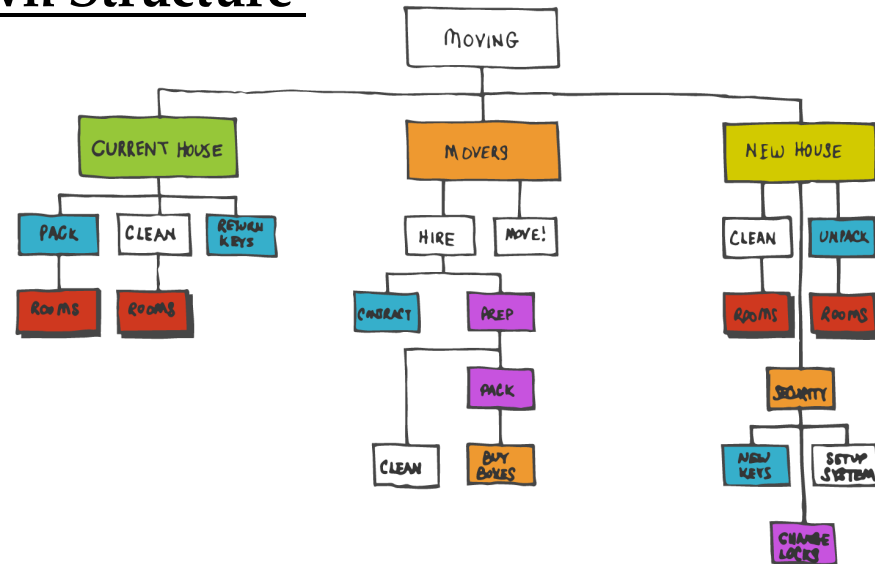
- Requirement ID
- Risks
- Requirement Type
- Requirement Description
- Trace to Design Specification
- Unit Test Cases
- Integration Test Cases
- System Test Cases
- User Acceptance Test Cases
- Trace to Test Script

Example 2

[illegible]

Work Breakdown Structure

Create **Work Breakdown Structure** is the process of subdividing project deliverables and project work into smaller, more manageable components.



Example 2

Example 1

