

UNIT -2
REQUIREMENT ANALYSIS AND SPECIFICATION

Goals —

- To fully understand the user's requirement
- To remove inconsistency and anomalies data from the requirements.
- Then document the data properly in a SRS.
- These SRS consist of 2 distinct activities such as requirement gathering and analysis and requirement specification.
- The person who undertakes requirement analysis and specification is known as system analyst.

Functions of a System analyst —

- To collect data related to the product
- Analyze the collected data.
- To understand what exactly needs to be done.
- Write a SRS document.
- At the end the SRS is reviewed by the customer, the reviewed SRS documents form the basic of all future development activities.

Requirement Analysis —

- It consists of two main activities i.e. requirement gathering and analyze the gathered requirement.
- The system analyst gather requirement through —
 - (i) Observation of existing systems
 - (ii) Studying existing procedure
 - (iii) Discussing the needs with the customer and end user.
 - (iv) At the end the system analyst analyze what needs to be done.

17/01/2026

Requirement Gathering —

- A system analyst can obtain input and output format and accurate details of the operational procedure.
- In the absence of a working system a lot of creativity and imagination are required and interacting with the customer to gather relevant data requires a lot of experience.

Characteristics of a good system analyst —

The system analyst should have a good interaction skill, imagination and creativity, and prior experience in related projects.

Analysis of gathered requirement —

- After gathering all the requirement, a system analyst analyse it clearly and understand the user requirement.
- The system analyst detect inconsistencies, ambiguities and incompleteness within the requirement.
- The incompleteness and inconsistency are solved by further discussion with the end-user and the customer.

SRS —

The main aim of SRS is to systematically organize the requirements arrived during requirement analysis and document the requirement properly.

Uses of SRS —

- It works as a statement of user needs.
- It works as a contract document.
- It works as reference document.
- It defines the definition of implementation.

Purpose of SRS document —

- SRS establish the basic of agreement between the user and the supplier.
- The user needs have to be satisfied but user may not understand the software.
- The developer will develop the system but may not know about the problem domain.

Characteristics of SRS —

- 1) correctness — Each requirement accurately represent some desired features of the final system.
- 2) completeness — The SRS should have all the desired features specified. It should be hardest to satisfy the needs of the customer.
- 3) Unambiguous — Each requirement has exactly one meaning. The SRS should be error free. Important as natural language often used to be.
- 4) Verifiability — There must exist a cost effective way of checking if the software satisfy the requirement or not.
- 5) consistent — Two requirements should not have to contradict each other.
- 6) Traceable — The origin of the requirement and how the requirement related to that software element can be determined.
- 7) Rank for importance or stability — Needed for prioritizing in construction. To reduce risk due to changing requirement.

Structure of SRS —

→ Introduction

- Purpose, the basic objective of the system
- Scope of what the system is to do, not to do
- Definition
- Overview

→ Overall description

- User and system interfaces
- Product perspective
- Product functions
- User characteristics
- Assumptions
- Constraints

specific requirements

- external interfaces

- functional requirements

- performance requirements

- design constraints

Acceptable criteria

- desirable to specify this up-front.

This standardization of the SRS was done by IEEE.

19/01/2026

SRS : A Contract Document (2marks)

- A SRS document is a contract between the development team and the customer
- Once the SRS is approved by the customer the subsequent controversies are settled by referring the SRS document. So, once the customer agrees to the SRS document the development team starts to develop the product according to the requirements recorded in the SRS document.
- The final product will be acceptable to the customer as long as it satisfies the requirements recorded in the SRS document.
- The SRS document is considered as blackbox whose internal details are not known.
- It is only visible as an external (input/output) behaviour that is documented.
- Finally, the SRS document concentrates on what needs to be done, it also serves as a contract between the development team and the customer and should be carefully written.

Properties of a good SRS document —

- It should be concise and at the same time it should not be ambiguous.
- It should specify what the system must do and not say how to do.
- The SRS should be easy to change and should be well structured.
- It should be consistent and complete.
- The SRS should be traceable i.e. a developer should able to know which part of specification corresponds to which part of the design and code and vice-versa.

3 parts of SRS

- 1) Functional requirements
- 2) Non-functional requirements
- 3) Goals of implementation

Requirement Engineering — (marks Q)

- It is described as the functional and non-functional requirements that can be easily understandable by the system user who don't have detailed technical knowledge.
- User requirements are generally defined using natural language, tables and diagrams.
- It is often referred as the user need, describe what the user does with the system such as what activities the user should be able to perform.

Requirement —

- The software requirements are the description of the features and functionalities of the targeted system.
- Requirements convey the expectations of users from the software product.
- The requirements can be obvious, hidden, known or unknown, expected or unexpected from client's point of view.

Functional Requirements

- In ^{software} engineering, a functional requirement defines a system on its document. It describes that a function is nothing but an input, its behaviour and its output. It can be a calculation, data manipulation, business process, user interaction or any other specific functionalities which defines what function a system is likely to perform.
- Functional software requirement help you to capture the actual behaviour of the system. This behaviour may be expressed as functions, services or tasks on which it required to perform.
- Examples of functional requirements
 - Authentication of user whenever he/she logs into the system.
 - System shutdown in case of a cyber attack.
 - Verification email is sent to user whenever he/she registers for the first time on some software system.
 - Business Rules
 - Transaction corrections, adjustments and cancellations
 - Administrative functions
 - Authorization levels
 - Audit tracking
 - External Interfaces
 - Certification Requirements
 - Reporting Requirements
 - Historical data
 - Legal or Regulatory Requirements

Non-functional Requirements

- A non-functional requirement defines the quality attribute of the software. They represent a set of standards that is used to judge the specific operation of a system.
- It is essential to ensure the usability and effectiveness of the entire software system. Failing the system or requirement can result in system that can fail to satisfy user needs.
- It allows to impose constraints or restriction on the design of the system across various agile backlog.

- Eg: 1) Performance: Response time, throughput, utilisation,
- 2) Capacity
- 3) Availability
- 4) Reliability
- 5) Maintainability
- 6) Security
- 7) Environmentable
- 8) Data Integrity
- 9) Usability

Goals of Implementation —

- It is a part of document and some general suggestion regarding the development of the software.
- These suggestions guide trade-off among design tools.
- These section might document issues such as revision to system functionalities that may be required in the future, new devices to be supported in future, reusability issues, etc.
- These are the items which the developer might keep in their mind during development so that the developed system may meet some aspects that are not required immediately.

Q) Difference between functional and non-functional requirement.

Functional requirement

- They define a system or its component.
- It specifies what the system should do.
- The user specifies the requirement.
- It is mandatory to meet these requirements.
- It is captured in use case.

Non-functional requirement

- They define the quality attributes of a system.
- It specifies how should the system fulfil the functional requirement.
- It is specified by technical people, architect, technical leaders and software developer.
- It is not mandatory to meet the requirement.
- It is captured as quality attributes.

- Defined at component level.
- It helps you to verify the functionality of the software.
- Functional testing like system, integration, end to end testing, API testing, etc. are done.
- Functional requirements are usually easy to define.
- Applied to a whole system.
- Helps you to verify the performance of software.
- Non-functional testing like performance, stress, usability, security, testing are done.
- Non-functional requirements are more difficult to define.

User Requirement —

- It is an official document that records the specific needs, features of a project or a software to meet the user's requirement. It is considered as a roadmap that guides the entire project team towards a common goal.
- It is described as the functional and non-functional requirement so that it can be easily understandable by the system users who don't have detailed technical knowledge.
- User requirements is generally defined using natural language, tables and diagram.
- It is often referred to as user needs, describe what the user does with the system such as what activities that the user must be able to perform.

24/01/2026

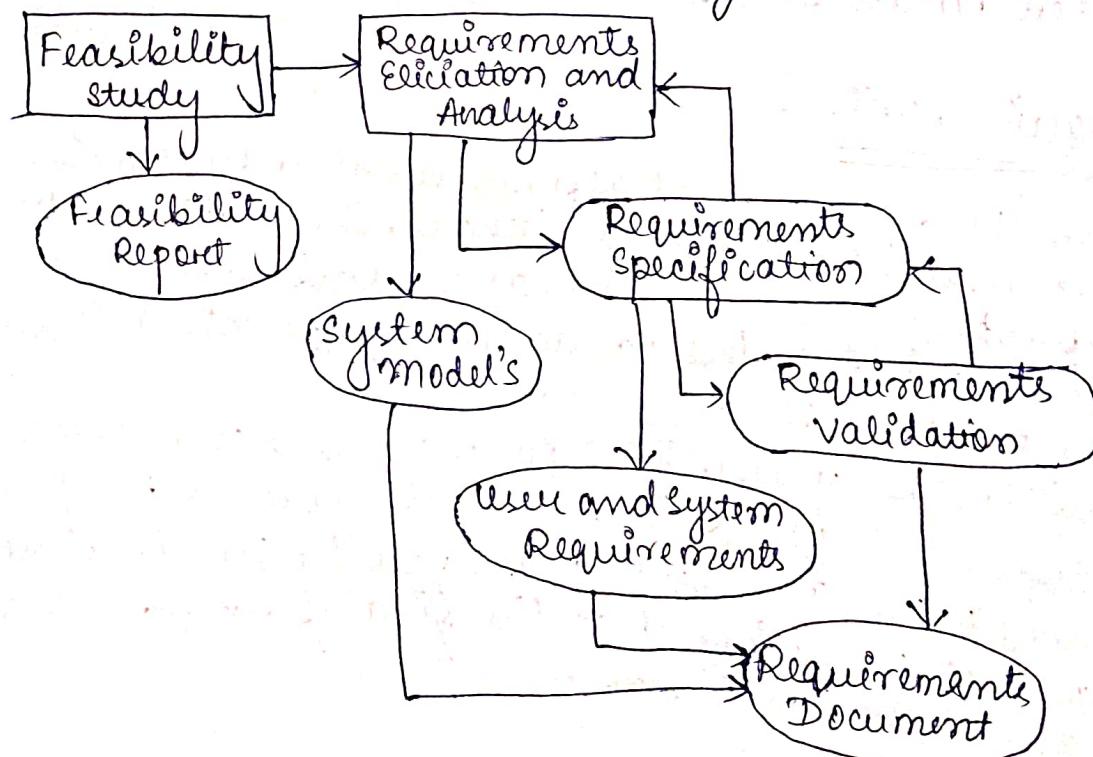
System Requirement —

- It is a set of specification that defines what a system is expected to do and how it should behave. They outline the necessary properties of the system as a whole and are crucial for guiding the design development and verification process.
- System requirement specify the capabilities and characteristics that a system must possess to fulfil its required purpose.
- These document act as a blueprint for the system's design and development ensuring that the final product meets the needs of its user and stakeholders.

→ These requirements are often documented in software requirement specification or system Requirements Specification (SyRS) document.

Requirement Engineering Process

- Requirement Engineering refers to the process of defining, documenting and maintaining requirement in the engineering design process.
- It provides the appropriate mechanism to understand what the customer desires, analysing the need and assessing the feasibility, negotiating a reasonable solution, specifying the solution clearly, validating the specification and managing the requirements as they are transformed into a working system.
- Therefore, requirement engineering is a disciplined application of proven principles, methods, tools and different notation to describe a proposed system and its behaviour, into its associated constraint.
- It consists of 4 steps —
- (i) Feasibility study.
 - (ii) Requirement Elicitation and Analysis.
 - (iii) Software Requirement Specification.
 - (iv) Software Requirement Validation.
 - (v) Software Requirement Management.



Requirement Elicitation —

- Requirement elicitation is a process of collecting requirements from users, customers and stakeholders.
- It involves techniques such as interviews, questionnaires, observation and discussion to understand user's needs fully.
- Communicating with bank staff and customers to understand ATM features.

Feasibility Study —

- Feasibility study evaluates whether the proposed system is practical and achievable.
- It examines technical, economic, operational and time viability before the development begins.
- Eg: Checking whether a company can afford a cloud infrastructure within a given budget.

Requirement Validation —

- It ensures that the documented requirements correctly represent user needs.
- It checks for correctness, completeness, consistency and feasibility of all the requirement.
- Eg: Reviewing the SRS with the customer to confirm all required features that are included.

Requirement Management —

- It is the process of systematically handling the requirements through out the software development life-cycle.
- It involves tracking changes, maintaining version and ensuring that all the requirements are implemented correctly despite changes.
- Eg: updating the SRS when a customer requests an additional feature during the development.

27/01/2026

Privacy and Data Protection Law —

- Data protection and privacy laws establish the legal framework for how organizations handle individuals' personal information. These laws define the rights of individuals concerning their data and the responsibilities of those who collect, process and store it.
- India recently enacted the Digital Personal Data Protection Act (DPDPA) 2023, marking a significant step in its data protection landscape.
- Data protection and privacy laws include:
 - Individual Rights: These laws typically grant individuals rights like access to their data, the ability to correct inaccuracies, and the right to request deletion of their data.

Data controller Responsibilities: Organizations that collect and determine the purpose of processing personal data are designated as data controllers and have specific obligations, including ensuring data security and transparency.

Lawful Basis for Processing: Laws define the conditions under which personal data can be processed, often requiring consent or other lawful grounds like contractual necessity or legitimate interests.

Data Breach Notification: Organizations are typically required to notify relevant authorities and individuals in the event of a data breach.

Enforcement and Penalties: Data protection laws often establish mechanisms for enforcement, including the appointment of a data protection authority and the imposition of penalties for non-compliance.

Indian Digital Personal Data Protection Act (DPPDA) 2023

→ India's first comprehensive data protection law, aimed at safeguarding the privacy of individuals in the digital realm.

Explain GDPR law.

- The General Data Protection Regulation (GDPR) is a European Union law focused on data protection and privacy. It aims to give individuals more control over their personal data and standardize data protection across the EU.
- GDPR applies to any organization, regardless of location, that handles the personal data of individuals in the EU.
- Applies to any entity (inside or outside the EU) that processes or monitors personal data of individuals within the EU.
- Covers all types of personal data — not just sensitive or health data — including names, emails, IP addresses, biometric data, etc.

Key Principles of GDPR

- 1) Lawfulness, Fairness and Transparency : Individuals must be informed about data collection and its purposes.
- 2) Purpose Limitation : Personal data collected for specific purposes must not be used beyond those intentions.
- 3) Data Minimization : Only collect data necessary for the specified purpose.
- 4) Accuracy : Data must be correct and kept up-to-date.
- 5) Storage Limitation : Data shouldn't be held longer than necessary.
- 6) Integrity and Confidentiality : Data must be protected from unauthorized access, loss or damage.
- 7) Accountability : Organisations are responsible for compliance and must document their data processing activities.

Rights Granted by GDPR

- 1) The Right to be informed
- 2) The Right to access
- 3) The Right to Rectification
- 4) The Right to be forgotten
- 5) The Right to restrict processing
- 6) The Right to Data Portability
- 7) The Right to Object
- 8) Rights in Relation to Automated Decision Making and Profiling.

30/01/2026

HIPAA (Health Insurance Portability and Accountability Act)

- HIPAA is a United States law, enacted in 1996, which focuses primarily on protecting the privacy and security of individual health information.
- Applies to covered entities in the US healthcare sector, such as providers, health plans, clearinghouses, and their business associates.
- Only covers Protected Health Information (PHI) — individually identifiable health data related to medical history, treatment, billing, etc.

Main Components of HIPAA —

- 1) Privacy Rule : Governs the use and disclosure of PHI in all forms (electronic, written, oral). Only those who need access for care or billing may see PHI, and patient consent is generally required for disclosure.
- 2) Security Rule : Sets standards to protect electronic PHI (ePHI) with administrative, physical, and technical safeguards.
- 3) Transaction and code set Rule : Standardizes electronic healthcare transactions.
- 4) Unique Identifiers Rule : Introduces unique IDs for healthcare entities.

5) Enforcement Rule : Provides penalties for violations.

Key Requirements —

- Entities must maintain safeguards to ensure confidentiality, integrity, and availability of PHI.
- Patients have the right to access their health records.
- PHI may only be disclosed without patient consent for specific lawful reasons (like public health or as required by law).
- Data breach notifications are mandatory, and workforce training on privacy is required.

Enforcement and Penalties —

- Enforced by the Department of Health and Human Services' Office for Civil Rights (OCR). Violations can result in hefty fines and corrective action mandates.
- Aims to reduce healthcare fraud, assure information probability, and protect patient data.

Ethical Testing and Debugging —

- Ethical testing and debugging involve ensuring that software is tested thoroughly and errors or bugs are identified and fixed responsibility to —
 - Protect users from harm or data loss.
 - Ensure software behaves as expected and does not violate privacy or security.
 - Avoid intentionally hiding known issues by providing accurate and transparent information about software limitations.
 - Follow professional standards and best practices to maintain integrity in software development.

31/01/2026

Software licensing —

- Software licensing defines the legal terms under which software can be used, modified, and distributed. Ethical software licensing means:
 - Respecting the license terms (e.g., proprietary, open-source).
 - Not using software beyond the granted permissions.
 - Providing proper credit to original authors.
 - Avoiding unauthorized copying/distribution, or commercialization.
 - Licensing promotes fairness by clearly defining rights and restrictions to prevent misuse and infringement.

Common types include:

- Proprietary licenses: restrict use and distribution; often require payment.
- Open-source licenses: allow use, modification, and redistribution under specified terms (GPL, MIT, Apache, etc.).

Open Source Ethics —

→ Open source ethics embody principles that guide contributors and users of open source software, including —

- Transparency: Code should be openly available for inspection.
 - Collaboration: Encouraging shared development and knowledge exchange.
 - Respect: Giving credit where due and acknowledging contributions.
 - Freedom: Ensuring users can use, modify, and distribute software freely within license terms.
 - Responsibility: Fixing bugs and addressing security vulnerabilities promptly.
- Avoiding misuse for unethical purposes.
- Open source ethics promote community trust, innovation, and equitable access to technology.

Plagiarism —

Plagiarism in software development refers to —

- Copying code, designs, or documentation from others without proper acknowledgement.
- Presenting someone else's work as your own.
- Using proprietary or copyrighted work without permission.

Ethically, developers must —

- Cite sources and give credit.
- Obtain permission where necessary.
- Use original code or clearly state modifications.
- Avoid dishonest practices to maintain professional integrity and respect intellectual property rights.

Misrepresentation —

Misrepresentation involves —

- Providing false or misleading information about software capabilities, performance, or origin.
- Claiming credit for work not done.
- Concealing defects or risks.

Ethically, developers and companies must —

- Be transparent about software limitations and issues.
- Communicate — skills, experiences, and software origins.
- Avoid deceptive marketing or overselling features.
- Maintain trust with users, clients and the community.

Whistleblowing —

Whistleblowing is the ethical act of exposing wrongdoing or unethical practices within an organisation, including —

- Reporting security vulnerabilities ignored by the management
- Disclosing illegal or harmful actions affecting users or the public.
- Revealing unethical data handling or privacy breaches

Ethical whistleblowing involves —

- Acting in good faith with evidence.
- Protecting user and public interests
- Following internal reporting channels first when safe
- Accepting personal risks for greater good of transparency and accountability.