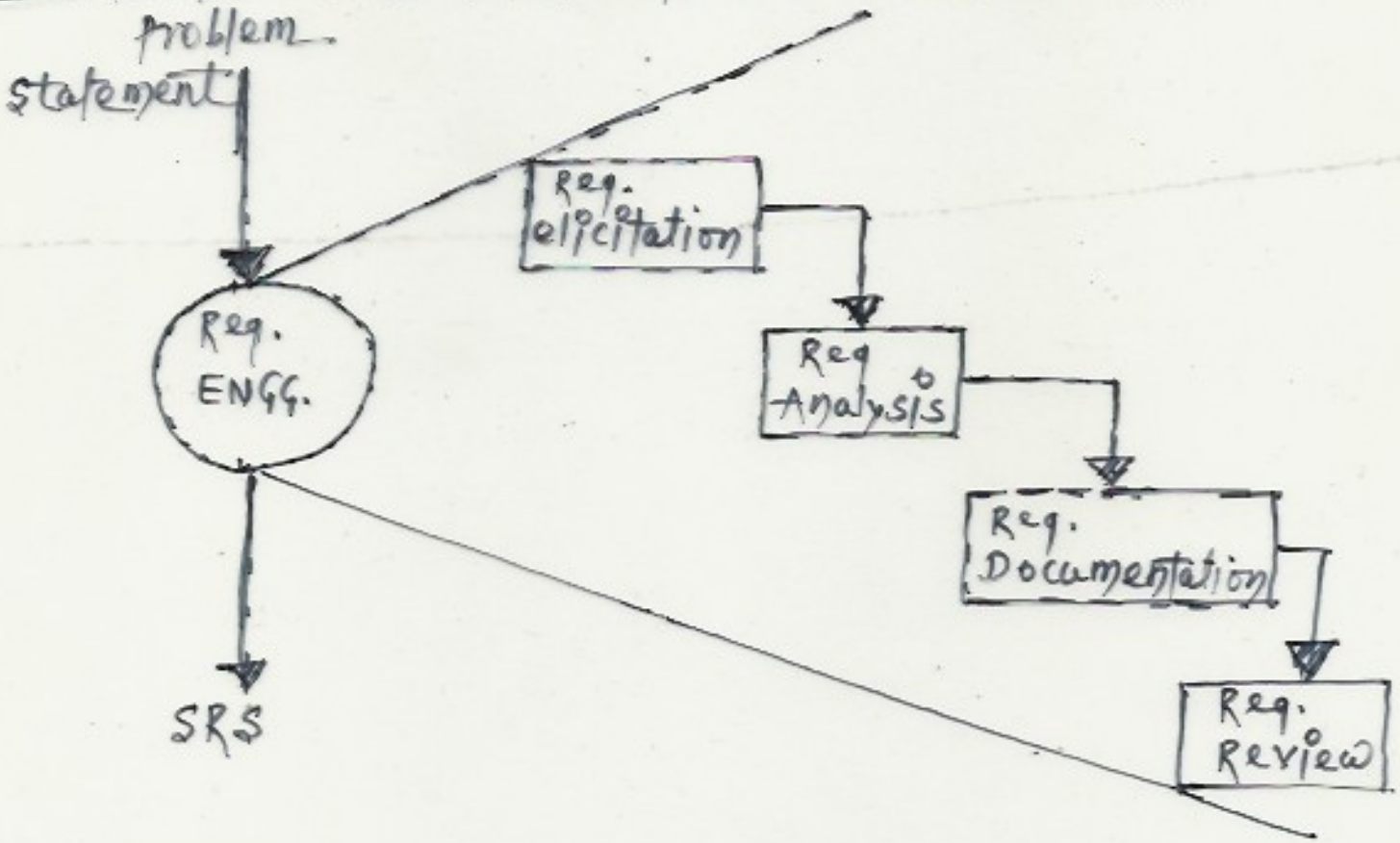
Requirements Elicitation

* A requirement is a feature of the system or a description of something the system is capable of doing in order to fulfill the system's purpose.
* Requirements describe the "What" of a system, not the "how". Meaning What the sys. will do not how it will do.
* Requirements engineering produces one large document, written in a natural language.
* The input to requirements engineering is the problem statement prepared by the customer.
* Crucial process steps of requirement engineering



* Requirement engineering is the disciplined application of proven principles, methods, tools and notations to describe a proposed system's intended behaviour and its associated constraints.

four steps →

(i) Requirements elicitation: ↳ Known as gathering of requirements.

↳ Requirements are identified with the help of customer and existing system processes

(ii) Requirements analysis: ↳ Analysis of requirements starts with requirement elicitation

↳ By Analysis, we identify inconsistencies, defects, omissions, etc.

(iii) Requirements documentation: ↳ This is the end product of requirement elicitation and analysis.

↳ The documentation is very important as it will be the foundation for the design of the software.

↳ The documentation is known as software requirements specification (SRS)

(iv) Requirements review → ↳ The review process is carried out to improve the quality of the SRS.

↳ It may also be called as requirement verification.

* The primary output of requirement engineering is requirement specifications.
* If it describes both hardware and software, it is system requirement specification.
* If it describes only software, it is software requirement specification.

Goal of Requirement elicitation: ↳ To find out what user really need.

↳ user need can be identified only if we understand the expectations of the users from the desired software.

↳ Requirements are gathered by asking questions, writing down the answers, asking other questions etc.

↳ Requirements gathering is the most communication intensive activity of software development.

↳ Requirement elicitation method -

* selection of a particular methodology based on some reasons →
  (i) It is the only method that we know
  (ii) It is our favorite method for all situation
  (iii) We understand intuitively that the method is effective in the present circumstances.

↳ interview - * After receiving the problem statement from the customer, the interview is arranged.
* Specialised developer, normally called 'Requirement engineers' interact with the customer.
* objective of interview is to understand the customers' expectations from the software.
* interview may be open-ended or structured.
* There is no pre-set agenda in open-ended interview

↳ Selection of Stakeholder → * Representatives from groups must be selected based on their technical expertise, domain knowledge, credibility and accessibility.

(i) Entry level personnel: * They may not have sufficient domain knowledge and experience, but may be very useful for fresh ideas and different views.

(ii) Mid level Stakeholder — * They may have better domain knowledge and experience of the project.
* They know the sensitive, complex and critical areas of the project.

(iii) Managers and/or other Stakeholder →
* Higher level management officers like Vice presidents, General Managers, Managing Directors should also be interviewed.
* It provides the rich info. for the S/W development

(iv) Users of the S/W — * Users spend more time with the S/W.
* Their info. may be eye opener and may be original at times.

② Brainstorming sessions → * It is a group technique that may be used during requirement elicitation to understand the requirements.

* Requirements in the long list can be categorized prioritized and pruned.

* It is being used by most of the companies.

* It promotes creative thinking, generates new idea and provides platform to share view, apprehensions expectations and difficulties of implementation.

* This group technique may be carried out with specialised group like actual users, middle level managers etc. or with total stakeholders.

* Every idea will be documented in such a way that everyone can see it.

* After the session, a detailed report will be prepared and facilitator will review the report.

* every idea will be written in simple english so that it conveys same meaning to every stakeholder

* finally, a document will be prepared which will have list of requirements and their priority.

③ FAST : facilitated Application specification technique

* similar to brainstorming sessions
* objective is to bridge the expectation gap.
* expectation gap: What developer think they are supposed to build and what customers think they are going to get.
* In other to reduce expectation gap, a team oriented approach is developed for requirements gathering and is called facilitated Application specification Technique (FAST)
* Basic guidelines for FAST - * Arrange a meeting at a neutral site for developers and customers.
* Establishment of rules for preparation and participation
* prepare an informal agenda that encourages free flow of ideas.
* Appoint a facilitator to control the meeting. A facilitator may be a developer, a customer, or an outside expert.
* prepare a definition mechanism board, flipcharts worksheets, wall stickies etc.
* participants should not criticize or debate.

Use Case - Approach. * Initially, use cases were designed for object oriented software development world.

* They can be applied to any project that follow any development approach because the user does not care how we develop the S/W.

* This approach uses a combination of text and pictures in order to improve the the understanding the requirement.

* The usecases describe "what of a system and not how".

* They only give functional view of the system.

* Differences among use case, use case scenario, and use case diagram.

↳ usecases → are structured outline or templates for the discription of user requirements, moded in structured language like english.

↳ usecase scenarios → usecase scenarios are unstructured description of user requirements.

↳ usecase diagram - are graphical representation that may be decomposed into further levels of abstraction

components for use case approach

Actor - * lies outside the system model, but interact with the system in some way.

* It may be a person, machine, or an info. system that is external to the system model
* An Actor is represented as stick figure and not in part of the system itself

use cases → * A use case is initiated by a user with a particular goal in mind, and completes successfully when that goal is satisfied

* It describes the sequence of interactions b/w actors and the system necessary to deliver the services that satisfies the goal.
* use cases capture who (actor) does what (interaction) with the system, for what purpose (goal), without dealing with system internals.
* A complete set of use cases specifies all the different ways to use the system, and therefore defines all behaviour required of the system.
* use cases are written in any easy to understand structured narrative.

Use Case template

1.	Brief Description
2.	Actors
3.	flow of events
	1. Basic flow
	2 Alternative flow
4	Special Requirements
5	pre-condition
6	post Condition
7	extension points

Use Case diagram. * A use case diagram visually represents what happens when an actor interacts with a system

* use case dia. captures the functional aspects of a system.

* The system is shown as a rectangle with the name of the system inside.

* Actors are shown as sticks figures

* The use cases are shown as solid bordered oval labled with the name of the use case

* the relationships are lines or arrows between actors and use cases and/or between the use cases themselves.