4 REQUIREMENTS ANALYSIS AND TECHNIQUES

Requirement Analysis Techniques

Requirement Analysis, also known as Requirement Engineering, is the process of defining user expectations for a new software being built or modified. In software engineering, it is sometimes referred to loosely by names such as requirements gathering or requirements capturing. Requirements analysis encompasses those tasks that go into determining the needs or conditions to meet for a new or altered product or project, taking account of the possibly conflicting requirements of the various stakeholders, analyzing, documenting, validating, and managing software or system requirements. Here are the objectives for performing requirement analysis in the early stage of a software project:

From What to How: Software engineering task bridging the gap between system requirements engineering and software design.

- 3 Orthogonal Views: Provides software designer with a model of:
- system information (static view)
- function (functional view)
- behavior (dynamic view)
- Software Architecture: Model can be translated to data, architectural, and component-level designs.
- Iterative and Incremental Process: Expect to do a little bit of design during analysis and a little bit of analysis during design.

What is the Requirement?

A software requirement is a capability needed by the user to solve a problem or to achieve an objective. In other words, a requirement is a software capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed documentation. Ultimately, what we want to achieve is to develop quality software that meets customers' real needs on time and within budget.

Perhaps the greatest challenge being faced by software developers is to share the vision of the final product with the customer. All stakeholders in a project - developers, end-users, software managers, customer managers - must achieve a common understanding of what the product will be and do, or someone will be surprised when it is delivered. Surprises in software are rarely good news.

Therefore, we need ways to accurately capture, interpret, and represent the voice of customers when specifying the requirements for a software product.

Activities for Requirement Analysis

Requirements analysis is critical to the success or failure of a systems or software project. The requirements should be documented, actionable, measurable, testable, traceable, related to identified business needs or opportunities, and defined to a level of detail sufficient for system design. Conceptually, requirements analysis includes four types of activity:

1. Eliciting requirements: the task of communicating with customers and users to determine what their requirements are. This is sometimes also called requirements gathering.
2. Analyzing requirements: determining whether the stated requirements are unclear, incomplete, ambiguous, or contradictory, and then resolving these issues.

3. Requirements modeling: Requirements might be documented in various forms, such as natural-language documents, use cases, user stories, or process specifications.
4. Review and retrospective: Team members reflect on what happened in the iteration and identify actions for improvement going forward.

Requirements analysis is a team effort that demands a combination of hardware, software, and human factors engineering expertise as well as skills in dealing with people. Here are the main activities involve in requirement analysis:

- Identify customers' needs.
- Evaluate the system for feasibility.
- Perform economic and technical analysis.
- Allocate functions to system elements.
- Establish a schedule and constraints.
- Create system definitions.

<u>Requirement Analysis Techniques</u>

A requirement analysis has a

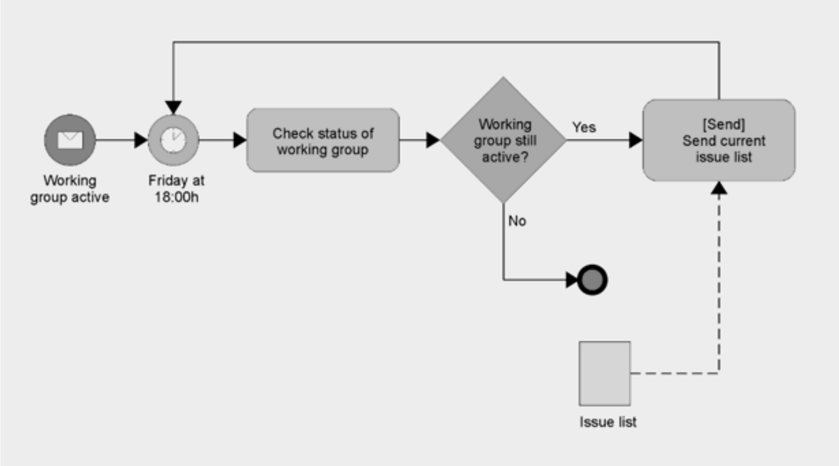
- Specific Goal
- Specific Input
- Specific Output
- Uses resources
- Has several activities to be performed in some order
- It May affect more than one organization unit
- Creates value of some kind for the customer

1. Business process modeling notation (BPMN) BPMN (Business Process Modeling & Notation) is a graphical representation of your business process using simple objects, which helps the organization to communicate in a standard manner. Various objects used in BPMN include
   - Flow objects
   - Connecting objects
   - Swim lanes
   - Artifacts

A good design BPMN model should be able to give the detail about the activities carried out during the process like,

- Who is performing these activities?
- What data elements are required for these activities?

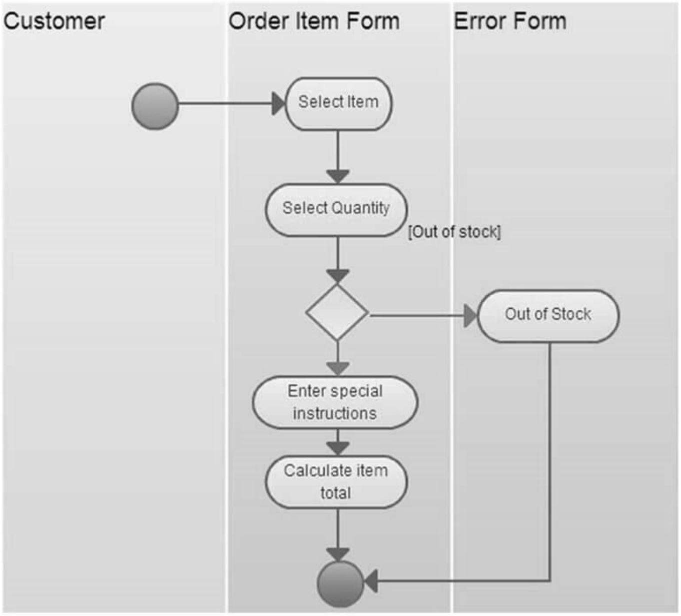
The biggest benefit of using BPMN is that it is easier to share, and most modeling tools support BPMN.


Working group active
Friday at 18:00h
Check status of working group
Working group still active?
Yes
No
[Send] Send current issue list
Issue list

2. UML is a modeling standard primarily used for the specification, development, visualization, and documenting of a software system. To capture important business processes and artifacts UML provides objects like

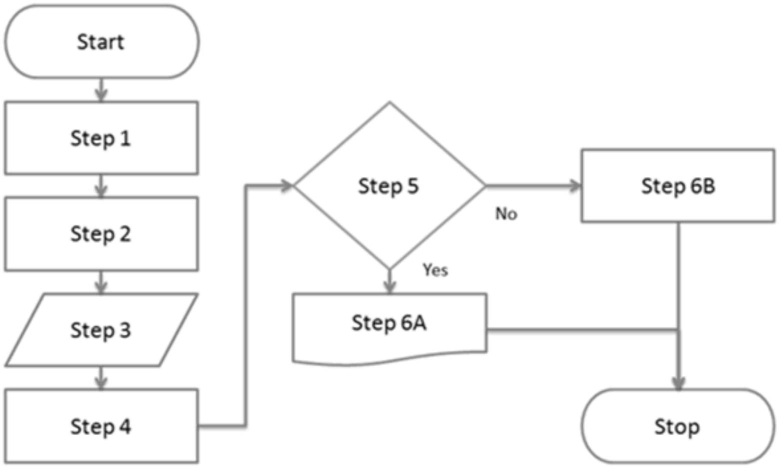
- State
- Object
- Activity
- Class diagram

14 UML diagrams help with modeling like the use case diagram, interaction diagram, class diagram, component diagram, sequence diagram, etc. UML models are important in the IT segment as it becomes the medium of communication between all stakeholders. A UML-based business model can be a direct input to a requirements tool. A UML diagram can be of two type's Behavioral model and the Structural model. A behavioral model tries to give information about what the system does while a structural model will give what the system consists of.



3. Flow chart technique

A flowchart is a visual representation of the sequential flow and control logic of a set of related activities or actions. There are different formats for flowcharts which include Linear, Top-down, and cross-functional (swim lanes). A flow chart can be used for different activities like representing data flows, system interactions, etc. The advantage of using a Flowchart is that it can be easy to read and write even for non-technical team members, and can show the parallel process by function, critical attributes of a process, etc.



4. Data flow diagram

Data flow diagrams show how data is processed by a system in terms of inputs and outputs. Components of the data flow diagram include

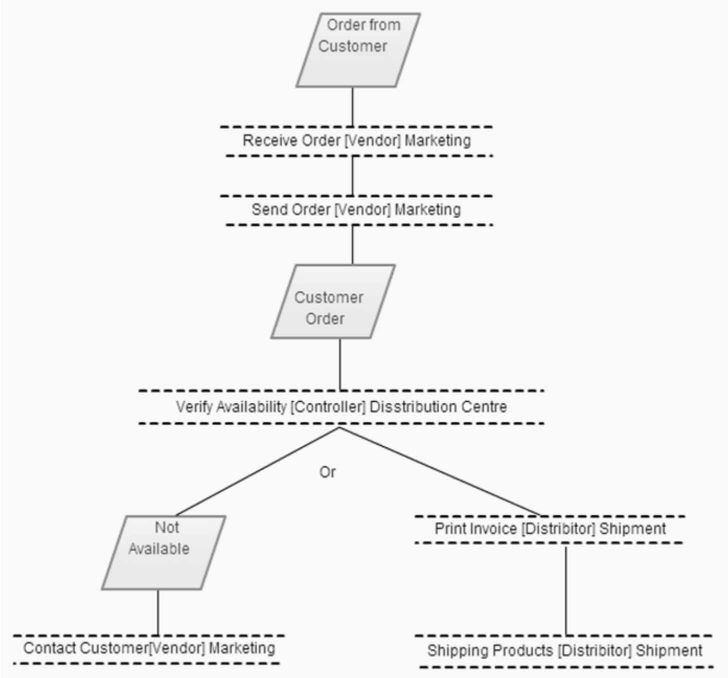
- Process
- Flow
- Store
- Terminator

A logical data flow diagram shows a system's activities while a physical data flow diagram shows a system's infrastructure. A data flow diagram can be designed early in the requirement elicitation process of the analysis phase within the SDLC (System Development Life Cycle) to define the project scope. For easy analysis, a data flow diagram can be drilled down into its sub-processes known as "leveled DFD".

5. Role Activity Diagrams- (RAD)

The role activity diagram is similar to flowchart type notation. In the Role Activity Diagram, role instances are process participants, which have start and end states. RAD requires a deep knowledge of processes or organizations to identify roles. The components of RAD include

- Activities
- External events
- States


Order from Customer
Receive Order [Vendor] Marketing
Send Order [Vendor] Marketing
Customer Order
Verify Availability [Controller] Disstribution Centre
Or
Not Available
Print Invoice [Distribitor] Shipment
Contact Customer[Vendor] Marketing
Shipping Products [Distribitor] Shipment

Roles group activities together into units of responsibility, according to the set of responsibilities they are carrying out. An activity can be carried out in isolation from a role, or it may require coordination with activities in other roles.

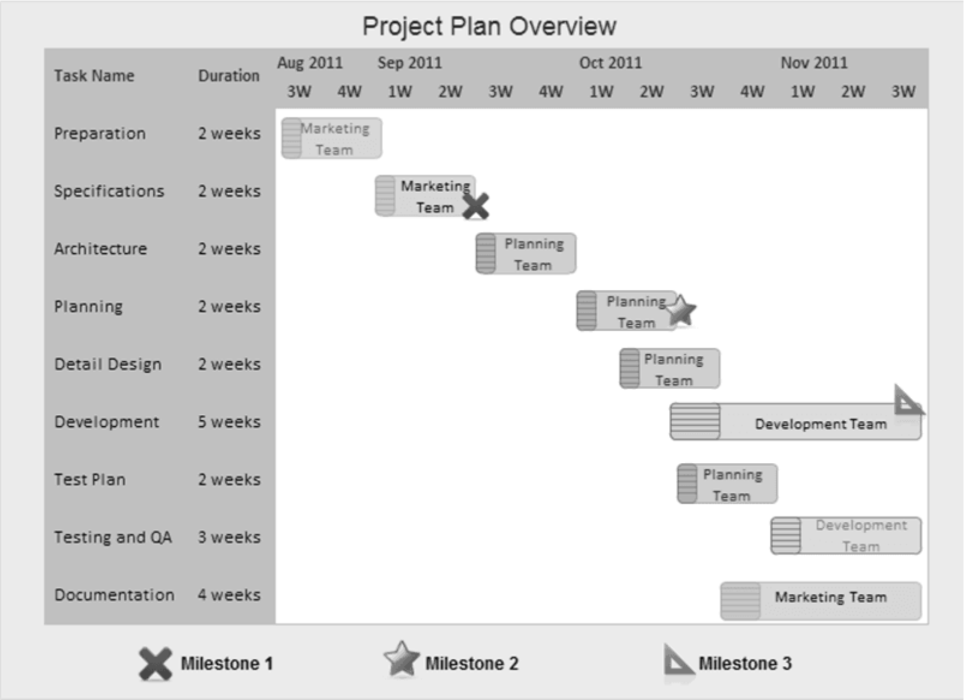
External events are the points at which state changes occur.

States are useful to map activities of a role as it progresses from state to state. When a particular state is reached, it indicates that a certain goal has been achieved.

RAD helps support communication as it is easy to read and presents a detailed view of the process and permitting activities in parallel.

6. Gantt Charts

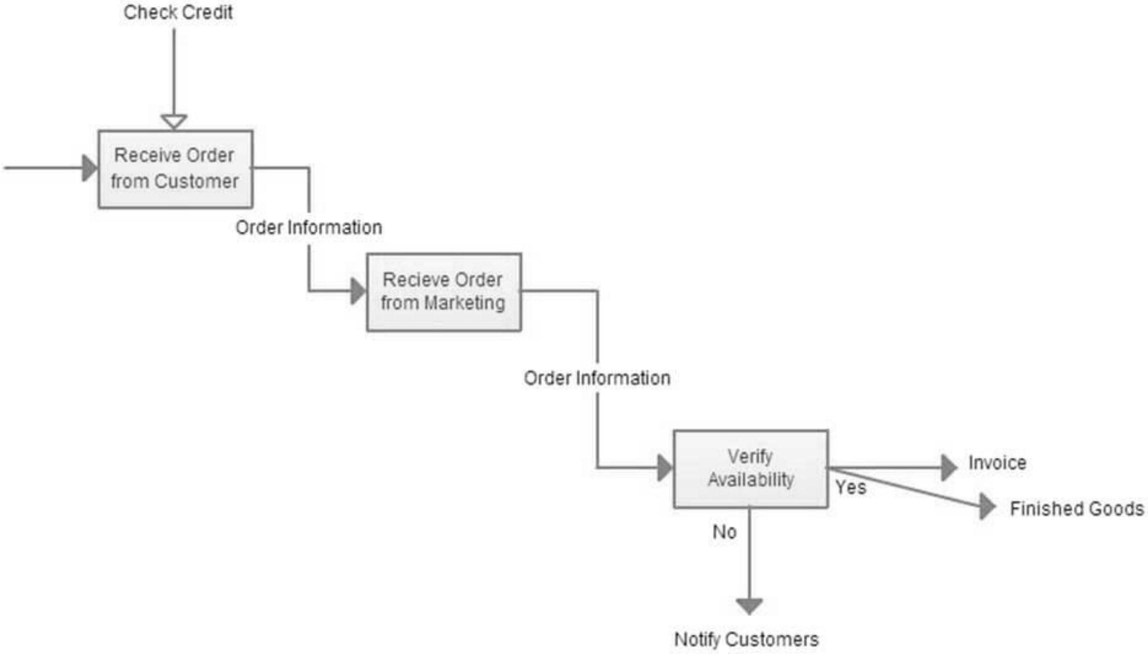
A Gantt chart is a graphical representation of a schedule that helps to coordinate, plan and track specific tasks in a project. It represents the total period of the object, broken down into increments. A Gantt chart represents the list of all tasks to be performed on the vertical axis while, on the horizontal axis, it lists the estimated activity duration or the name of the person allocated to the activity. One chart can demonstrate many activities.



Task Name	Duration
Preparation	2 weeks
Specifications	2 weeks
Architecture	2 weeks
Planning	2 weeks
Detail Design	2 weeks
Development	5 weeks
Test Plan	2 weeks
Testing and QA	3 weeks
Documentation	4 weeks

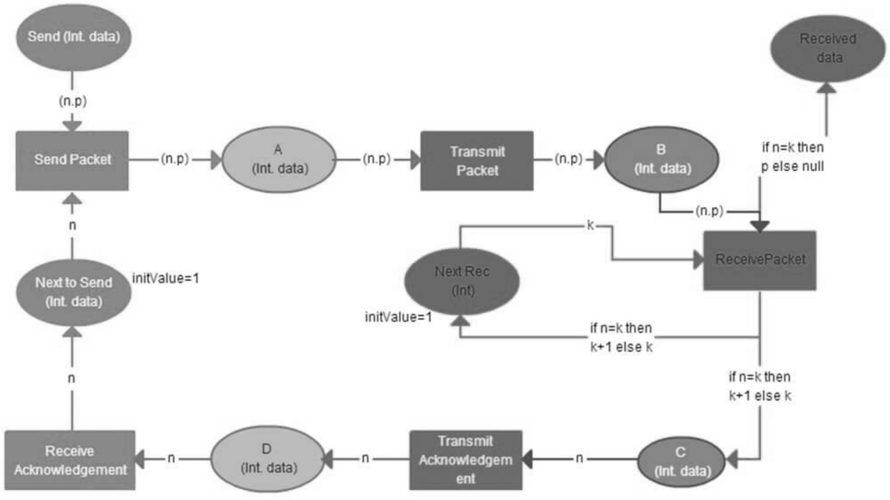
7. IDEF (Integrated Definition for Function Modeling)

IDEF or Integrated Definition for Function Modeling is a common name referred to classes of enterprise modeling languages. It is used for modeling activities necessary to support system analysis, design, or integration. There are about 16 methods for IDEF, the most useful versions of IDEF are IDEF3 and IDEF0.



8. Colored Petri Nets (CPN)

CPN or colored Petri nets are graphically oriented language for specification, verification, design, and simulation of systems. Colored Petri Nets is a combination of graphics and text. Its main components are Places, Transitions, and Arcs.

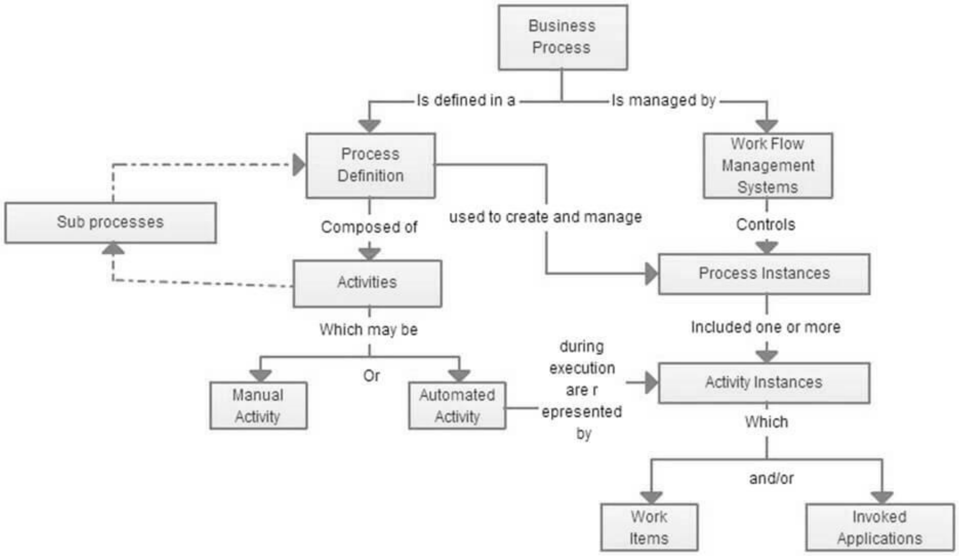


Petri nets objects have specific inscriptions like for

- Places: It has an inscription. Name, Color Set, Initial marking, etc. While
- Transition: It has inscription like.Name (for identification) and.Guard (Boolean expression consists of some of the variables)
- Arcs: It has an inscription. Arc. When the arc expression is assessed, it yields a multi-set of token colors.

9. Workflow Technique

The workflow technique is a visual diagram that represents one or more business processes to clarify understanding of the processor to make process improvement recommendations. Just like other diagrams like flowcharting, UML activity, and process map, the workflow technique is the oldest and most popular technique. It is even used by BA for taking notes during requirements elicitation. The process comprises four stages



- Information Gathering
- Workflow Modeling
- Business process Modeling
- Implementation, Verification & Execution

10. Object-oriented methods

The object-oriented modeling method uses an object-oriented paradigm and modeling language for designing a system. Its emphasis is on finding and describing the object in the problem domain. The purpose of the object-oriented method is

- To help characterize the system
- To know what are the different relevant objects
- How do they relate to each other
- How to specify or model a problem to create an effective design
- To analyze requirements and their implications

This method applies to the system which has dynamic requirements (changes frequently). It is a process of deriving use cases, activity flow, and events flow for the system. Object-oriented analysis can be done through textual needs, communication with system stakeholders, and vision documents.

The object has a state, and state changes are represented by behavior. So, when the object receives a message, the state changes through behavior.

11. Gap Analysis

Gap Analysis is the technique used to determine the difference between the proposed state and the current state of any business and its functionalities. Does it answer questions like what is the current state of the project? Where do we want to be? etc. Various stages of Gap Analysis include