System:

A system is an orderly grouping of interdependent components linked together according to a plan to achieve a specific objective.

The study of system concepts has three basic implications:

1. A system must be designed to achieve a predetermined objective.
2. Interrelationships and interdependence must exist among the components.
3. The objectives of the organization as a whole have a higher priority than the objectives of its subsystems.

Characteristics of a system:

1. **Organization:**

   It implies structure and order. It is the arrangement of components that helps to achieve objectives.

2. **Interaction:**

   It refers to the manner in which each component functions with other components of the system.

3. **Interdependence:**

   It means that parts of the organization or computer system depend on one another. They are coordinated and linked together according to a plan. One subsystem depends on the output of another subsystem for proper functioning.

4. **Integration:**

   It refers to the holism of systems. It is concerned with how a system is tied together.

5. **Central Objective:**

   A system should have a central objective. Objectives may be real or stated. Although a stated objective may be the real objective, it is not uncommon for an organization to state one objective and operate to achieve another. The important point is that users must know the central objective of a computer application early in the analysis for a successful design and conversion.

Elements of a System:

1. **Outputs and inputs:**

   A major objective of a system is to produce an output that has value to its user. In order to get a good output, inputs to system must be appropriate. It is important to point out here that determining the output is a first step in specifying the nature, amount and regularity of the input needed to operate a system.

2. **Processors:**

   It is the element of a system that involves the actual transformation of input into output. It is the operational component of a system. Processors may modify the input totally or partially, depending on the specifications of the output. In some cases, input is also modified to enable the processor to handle the transformation.

3. **Control:**

   The control elements guide the system. It is the decision-making subsystem that controls the pattern of activities governing input, processing, and output.

4. **Feedback:**

   Feedback measures output against a standard in some form of cybernetic procedure that includes communication and control.

   Feedback may be positive or negative, routine or informational. Positive feedback reinforces the performance of the system. It is routine in nature. Negative feedback generally provides the controller with information for action.

5. **Environment:**

   The environment is the “supra-system” within which an organization operates. It is the source of external elements that impinge on the system. In fact, it often determines how a system must function.

6. **Boundaries and Interfaces:**

   A system should be defined by its boundaries- the limits that identify its components, processes, and interrelationships when it interfaces with another system.

Types of System

1. **Physical or Abstract Systems:**

   Physical systems are tangible entities that may be static or dynamic in operation.

   Abstract systems are conceptual or nonphysical entities. They may be formulas of relationships among sets of variables or models – the abstract conceptualization of physical situations.

2. **Open or Closed Systems:**

   An open system has many interfaces with its environment. It permits interaction across its boundaries; it receives inputs from and delivers outputs to the outside.

   A closed system is isolated from environment influences.

3. **Man-made Information Systems:**

   An information system is the basis for interaction between the user and the analyst. It provides instructions, commands, and feedback. It determines the nature of relationships among decision makers. From this basis, an information system may be defined as a set of devices, procedures, and operating systems designed around user-based criteria to produce information and communicate it to the user for planning, control and performance.

System Development Life Cycle

Stage	Key Question	Result
1. Recognition of need Preliminary survey/ Initial investigation	What is the problem or opportunity?	Statement of scope and objectives Performance criteria
2. Feasiblility Study Evaluation of existing system and procedures Analysis of alternative candidate systems Cost estimates	What are the user's demonstrable needs? Is the problem worth solving? How can the problem be redefined?	Technical / behavioral feasibility Cost / benefit analysis System scope and objectives Statement f new scope and objectives
3. Analysis Detailed evaluation of present system Data collection	What must be done to solve the problem? What are the facts?	Logical model of system – e.g. data dictionary, data flow diagrams Pertinent data
4. Design General design specifications Detailed design specifications Output, Input, Files Procedures Program construction Testing Unit Testing Combined module Testing User acceptance Testing	In general, how must the problem be solved? Specifically, how must the problem be solved? What is the system (processing) flow? Does the user approve the system? How well do individual programs / modules test out? How ready are programs for acceptance test?	Design of alternative solutions Final cost/ benefit analysis Hardware specifications Cost estimates Implementation specifications Implementation schedule Approval of systems by user Programs, Test plans Security, audit and operating procedures Actual hardware use Formal system test
5. Implementation User training File / system Conversion	What is the actual operation? Are user manuals ready? Are these delays in loading files?	Training program User-friendly documentation
6. Post-implementation and maintenance Evaluation Maintenance Enhancements	Is the key system running? Should the system be modified?	User requirements met User standards met Satisfied user

STAGE I :INITIAL INVESTIGATION

The objective is to determine whether request is valid and feasible before a recommendation is reached to do nothing, improve or modify the existing system or build a new one.

The user's request form specifies the following:

1. User assigned title of work requested.
2. Nature of work requested (problem definition).
3. Date request was submitted.
4. Date job should be completed.
5. Job objective (s) – purpose of job requested.
6. Expected benefits to be derived from proposed change.
7. Input/ Output description – quantity and frequency of inputs and outputs of proposed change
8. Requester' signature, title, department, and phone number.
9. Signature, title, department and phone number approving, the request.

1. NEEDS IDENTIFICATION

User need identification and analysis are concerned with what the user needs rather than what he/she wants. This step is intended to help the user and the analyst understand the real problem rather than its symptom.

2. DETERMING THE USER'S INFORMATION REQUIREMENTS

There are three strategies for gathering information regarding the user's requirements. They are:

1. Asking: This strategy obtains information from users by simply asking them about the requirements. It assumes a stable system where users are well informed and can overcome biases in defining their problem. There are three key asking methods:

1. **Questionnaire:** Questions may be open-ended or closed. An open-ended question allows the respondent to formulate a response. It is used when feelings or opinions are important. In contrast, a closed question requests one answer from a specific set of responses. It is used when factual responses are known.
2. **Brainstorming:** It is technique used for generating new ideas and obtaining general information requirements. This method is appropriate for gathering nonconventional solutions to problems. A guided approach to brainstorming asks each participant to define ideal solutions and then select the best feasible one. It works well for users who have system's knowledge but have difficulty accepting new ideas.
3. **Group Consensus:** It asks participants for their expectations regarding specific variables. In a Delphi inquiry, for example, each participant fills out a questionnaire. The results are summarized and given to participants along with a

follow-up questionnaire. Participants are invited to change their responses. The results are again summarized and fed back to the participants. This debate by questionnaire continues until participants responses have converged enough. This method has an advantage over brainstorming in that participants are not subjected to psychological pressure from others with presumed authority or influence.

2. **Getting information from the existing information system:** Determining information from an existing application has been called the data analysis approach. It simply asks the user what information is currently received and what other information is required. It relies heavily on the user to articulate information needs. The analyst examines all reports, discusses with the user each piece of information examined, and determines unfulfilled information needs by interviewing the user. The analyst is primarily involved in improving the existing flow of data to the user.

   The data analysis method is ideal for making structured decisions, although it requires that users articulate their information requirements. A major drawback is a lack of established rules for obtaining and validating information needs that are not linked to organizational objectives.

3. **Prototyping:** This strategy for determining user information requirements is used when the user cannot establish information needs accurately before the information system is built. The reason could be the lack of an existing model on which to base requirements or a difficulty in visualizing candidate systems. In this case, the user needs to anchor on real-life systems from which adjustments can be made. Therefore, the iterative discovery approach captures an initial set of information requirements and builds a system to meet these requirements. As users gain experience in its use, they request additional requirements or modifications, in the system. In essence, information requirements are discovered by using the system. Prototyping is suitable in environments where it is difficult to formulate a concrete model for defining information requirements and where the information needs of the user are evolving, such as in Decision Support System (DSS).

3. PROBLEM DEFINITION AND PROJECT INITIATION:

The first step in an initial investigation is to define the problem that led to the user request. The problem must be stated clearly, understand, and agreed upon by the user and the analyst. It must state the objectives the user is trying to achieve and the results the user wants to see.

4. BACKGROUND ANALYSIS:

Once the project is initiated, the analyst begins to learn about the setting, the existing system, and the physical processes related to the revised system.

5. **FACT-FINDING:**

After obtaining this background knowledge, the analyst begins to collect data on the existing system's outputs, inputs, and costs. The tools used for information gathering are:

1. **Review of Written Documents:** The first step to gather information of a system is to search and review the various forms of written documents such as professional references, procedure manuals, textbooks, company studies, government publications, consultant studies, etc. of that system.
2. **On-site observations:** The major objective of on-site observation is to get close to the real system being studied. The methods used may be natural or contrived, obtrusive or unobtrusive, direct or indirect, and structured or unstructured. The main limitation of observation is the difficulty of observing attitudes and motivation and the many unproductive hours that often are spent in observing one-time activities.
3. **Interviews:** It is a face-to-face interpersonal meeting designed to identify relations and capture information as it exists. It is flexible tool, offering a better opportunity than the questionnaire to evaluate the validity of the information gathered. The major drawback is preparation time. Interviewing is an art that requires experience in arranging the interview, setting the stage, establishing rapport, phrasing questions clearly, avoiding arguments, and evaluating the outcome.
4. **Questionnaires:** It is a self-administered tool that is more economical and requires less skill to administer than the interview. It examines a large number of respondents at the same time, provides standardized wording and instructions, and places less pressure on subjects for immediate response. The main drawback is the low percentage of returns.

   In constructing a questionnaire, the analyst must focus on question content, wording, and format. These are considered with validity and that stem from respondent's failure to remember specific details, reluctant to report the true impressions of what occurred, or inability to communicate information.

   An interview or a questionnaire may be structured or unstructured. The unstructured approach allows respondents to answer questions freely in their own words, whereas the structured approach requires specific response to open-ended or closed questions.

   There are five major varieties of closed questions:

   a. Fill-in-the-blanks questions request specific information.

   b. Dichotomous questions offer a two-answer choice.

   c. Ranking scales questions ask the respondent to rank a list of items in order of importance or preference.

   d. Multiple-choice questions ask for specific answer choices.

   e. Rating scales questions ask the respondent to rank various items along a single dimension (scale).

6. FACT ANALYSIS:

As data are collected, they must be organized and evaluated and conclusions drawn for preparing a report to the user for final review and approval. Many tools are used for data organization and analysis. The tools are known as the tools of structured analysis.

Structured Analysis:

It is a set of techniques and graphical tools that allow the analyst to develop a new kind of system specifications that are easily understandable to the user.

The tools used for structured analysis are:

1. Data flow diagram (DFD).
2. Data Dictionary.
3. Structured English.
4. Decision trees.
5. Decision tables.

1. Data flow diagram:

A DFD, also known as a "bubble chart", has the purpose of clarifying system requirements and identifying major transformations that will become programs in system design. So, it is starting point of the design phase that functionally decomposes the requirements specifications down to the lowest level of detail. A DFD consists of a series of bubbles joined by lines. The bubbles represent data transformations and the lines represent data flows in the system.

DFD Symbols

1. A square defines a source or destination of system data.
2. An arrow identifies data flow – data in motion
3. A circle or a "bubble" represents a process that transforms incoming data flow(s) into outgoing data flow(s).
4. An open rectangle is a data store – data at rest, or a temporary repository of data.

Context Diagram

A context diagram is a structured graphical tool for identifying the organization’s functional areas and the processes that are performed within and between the organization and the outside world.

Context diagrams serve three important purposes whereby you can determine preliminary requirements:

1. Context diagrams support a data-oriented approach to system design,
2. help you investigate the output and process requirements of the organization, and
3. help you define the boundaries of the proposed system

Levels of Context Diagrams

There are three levels of context diagrams:

1. A user-level diagram describes one functional area’s operational activity. This is Level 1 context diagram.
2. A combined user-level diagram provides an overall view of the activities of related user groups. This leads to Level 2 context diagram.
3. An organization-level diagram reflects a consolidated view of the activities of the organization. This is the final Level 3 context diagram.

Example

The company, CAT Logistics is into warehousing business. The consultant from your organization has been invited to do the initial study and evolve the scope of the system. The system proposed is an "Order Tracking System “. The departments involved are Order Processing, Warehouse Shipping, Warehouse Receiving and Purchase departments. The company has furnished from each department a list of the players who fall into the scope of the study.

Players

Srinivasan – Consultant

Lata - Order Processing Department

Roshmi - Warehouse Shipping

Bhushan - Purchasing

Debie - Warehouse Receiving Discussion

During Srinivasan's first interview, Lata described how the Order Processing department should interact with the customer (an external entity) and Warehouse Shipping (an internal entity) to process an order. But first, the customer places an order with the Order Processing department.

The Order Processing department then requests the Warehouse Shipping to ship the product to the customer and notify Order Processing department about the shipment. The Order Processing department then prepares an invoice and sends it to the customer.

Further, Srinivasan met Roshmi from the Warehouse Shipping department. She confirmed that the Warehouse Shipping department was responsible for inventory management. According to her, when stock is low, Shipping notifies Order Processing, which keeps a copy and sends another copy to Purchasing. Purchasing places and order with the product supplier. When the product is received, Warehouse Receiving confirms receipt to Purchasing.

Thereafter Srinivasan met Bhushan of Purchasing, who explained that Purchasing notifies Order Processing when a product is ordered and again when the product is received. Purchasing awaits confirmation of product receipt before processing and paying invoices from the supplier.

Finally Srinivasan met Debie of Warehouse Receiving, and confirmed the details he had collected.

Level 1 (Step 1)

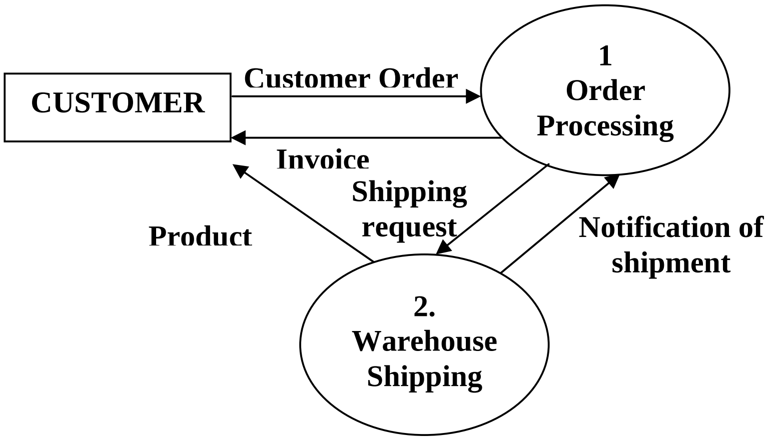


Fig: User-level context diagrams after interview with Lata

Level 1 (Step 2)

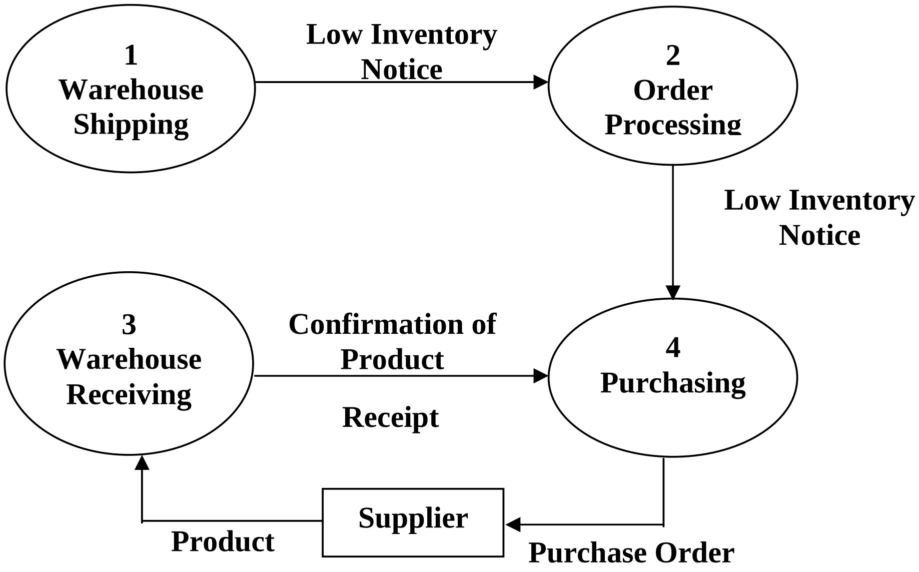


Fig: User-level context diagrams after interview with Roshmi

Level 1 (Step 3)

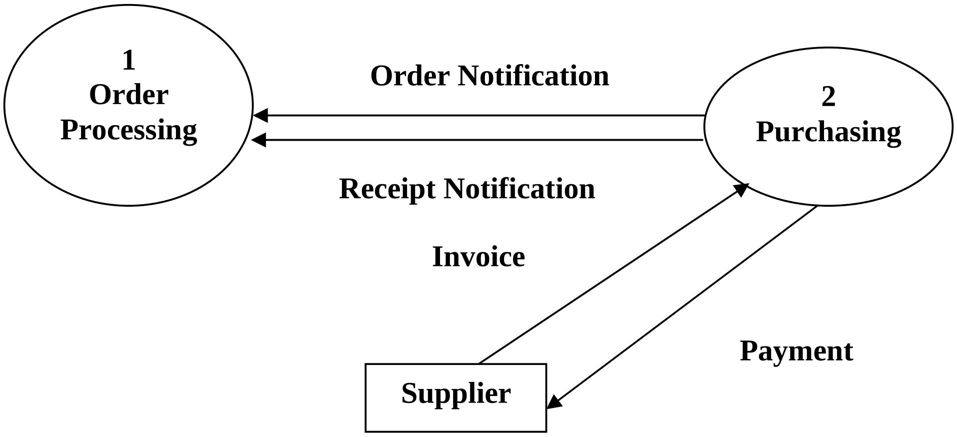


Fig: User-level context diagrams after interview with Bhushan

Level 2

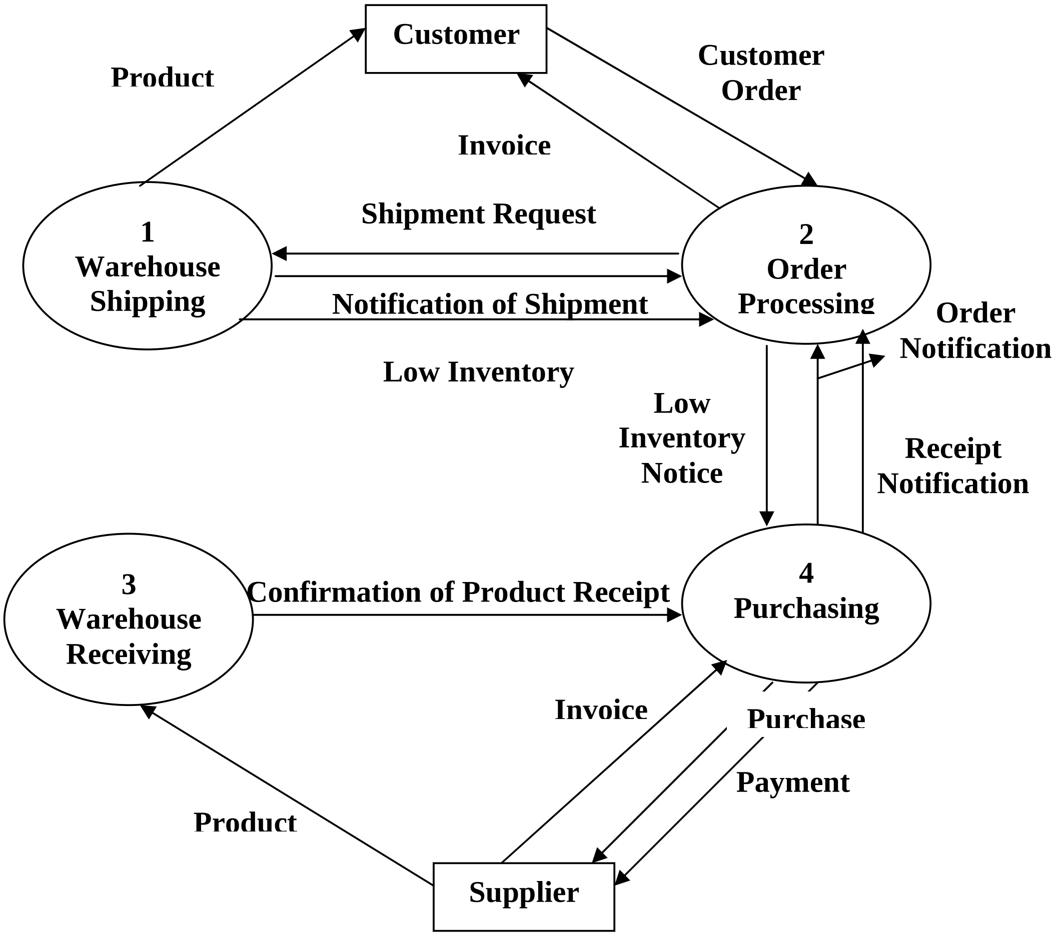


Fig Combined user-level context diagram for Order Tracking System

Level 3

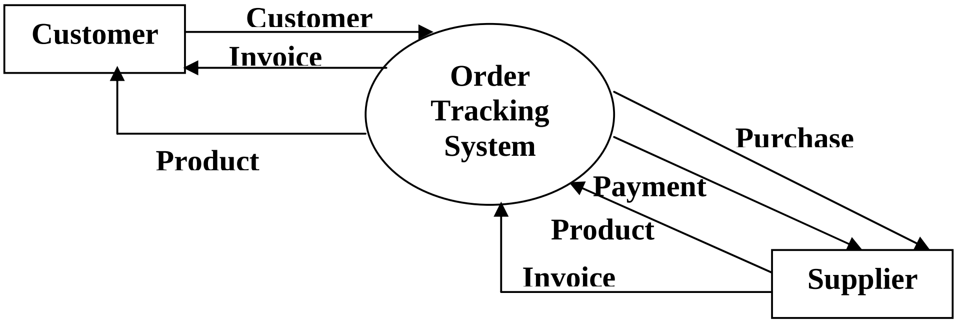


Fig: Organizational-level Context Diagram

Data Flow Diagram (DFD)

- DFD is one of several notations that are called **structured analysis technique**
- It is a graphical tool that allows analyst-users to depict the flow of data in an IS
- It concentrates on the **movement** of data between **processes**
- DFD evolves four symbols for process modelling
  - **Data flows**
  - **Data stores**
  - **Process**
  - **Source / destinations**

Levels of Data Flow Diagram

The context diagram can explode into different levels of DFDs.

Level-1: It is a data flow diagram that represents a system's **major processes**, **data flows**, and **data stores** at a high level of detail

- Shows all the processes that comprise the overall system
- Shows how information moves from and to each process
- Adds data stores

Level-2:

- Shows all the processes that comprise a single process on the level 1diagram
- Shows how information moves from and to each of these processes
- Shows in more detail the content of higher level process
- Level 2 diagrams may not be needed for all level 1 processes

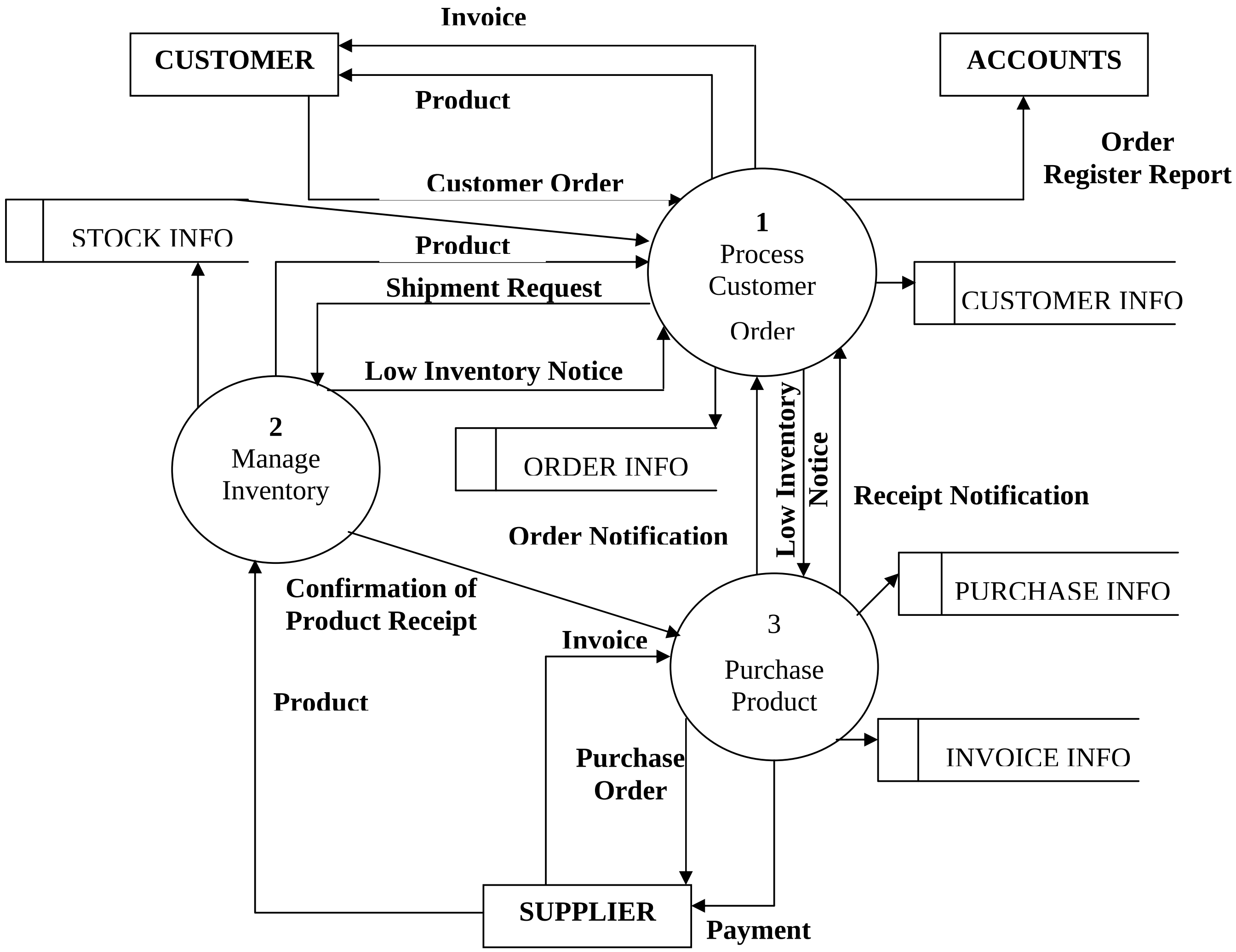


Fig: Level-1 DFD for Order Tracking System.

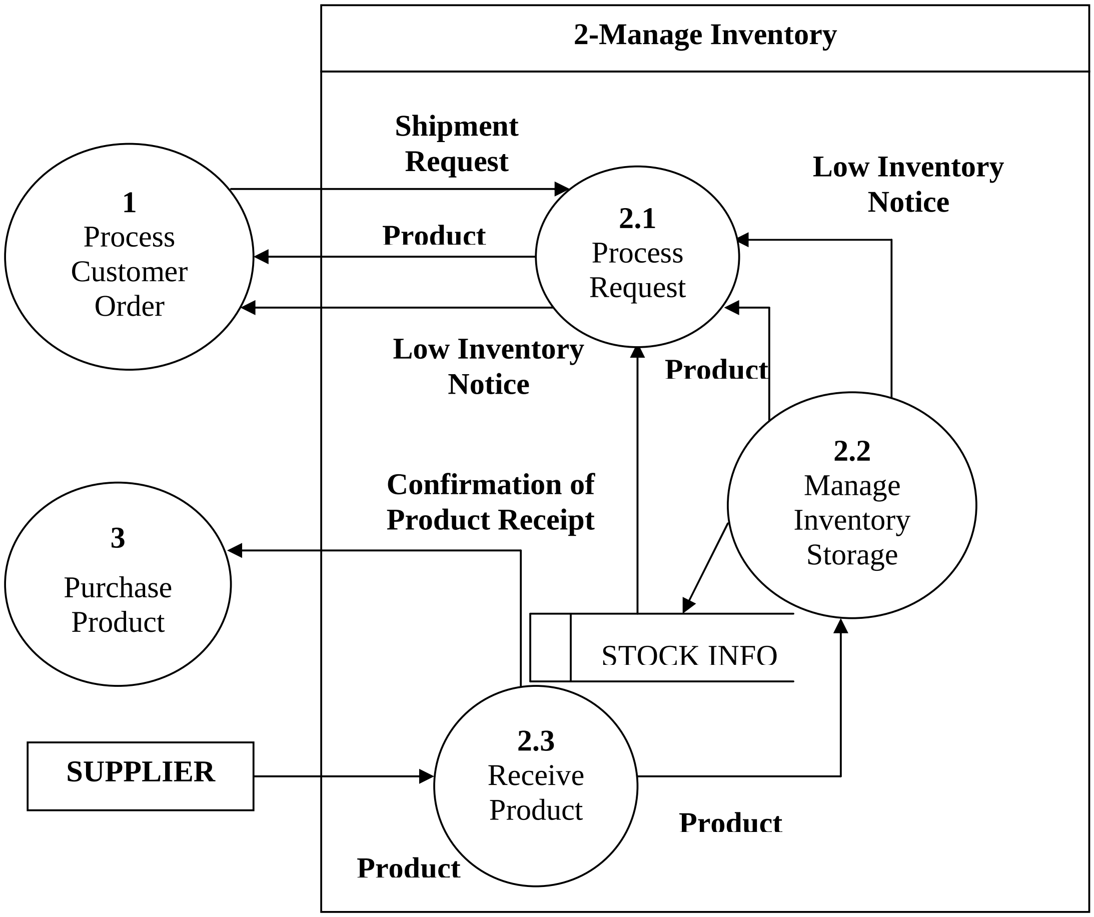


Fig: Level-2 DFD, breaking up of process “Manage Inventory”.

<u>Example:</u> If the volume of sales is greater than Rs. 10000 and advance collected is 50% or more, then the commission is 16%. If the advance collected is less than 50% then it is 14%. For sales equal to Rs. 10000 irrespective of the advance collected, commission is 10%. For sales less than Rs. 10000 commission is 9% or 8% based on whether advance collected is 50% or more, or less than 50% respectively.

Write a Structured English specification for the above statement.

```
IF      VOLUME OF SALES > 10000
        AND- IF ADVANCE COLLECTED >=50%
                THEN: COMMISSION IS 16%
        ELSE  (ADVANCE COLLECTED <50%)
                SO: COMMISSION IS 14%
ELSE – IF VOLUME OF SALES = 10000
        THEN: COMMISSION IS 10%
ELSE  (VOLUME OF SALES < 10000)
        SO – IF ADVANCE COLLECTED >=50%
                THEN : COMMISSION IS 9%
        ELSE  (ADVANCE COLLECTED < 50%)
                SO: COMMISSION IS 8%
```

Entity Relationship Model Concepts

Entity Relationship Model (E-R Model)

It describes data as entities, relationships and attributes.

Entity

It is a thing in the real world with an independent existence.

Attributes

Each entity has particular properties called attributes that describe it.

Types of Attributes

- ✓ Composite
- ✓ Single-valued Vs Multi-valued
- ✓ Stored Vs Derived

Relationship types

A relationship type R among n entity types E_1, E_2, E_3,-------, E_n defines a set of associations among entities from these types.

Cardinality Ratio

It specifies the number of relationship instances that an entity can participate in.

Cardinality for Binary Relationship

1:1, 1:N & M:N

Key Attribute of an Entity type

An entity type usually has an attribute whose values are distinct for each individual entity. Such an attribute is called a key attribute, and its value can be used to identify each entity uniquely.

Weak Entity

Some entity types may not have any key attributes of their own; those are called weak entity types.

Relational Model Concepts

Domain

It is a set of atomic values. By atomic we mean that each value in the domain is indivisible as far as the relational model is concerned.

Relational Model Constraints

- ✓ **Domain Constraint**

  It specify that the value of each attribute A must be an atomic value from the domain dom(A) for that attribute.

✓ **Key Constraint**

All elements of a set are distinct hence; all tuples in a relation must also be distinct. This means that no two tuples can have the same combination of values for all their attributes.

✓ **Entity Integrity Constraint**

It states that no primary key value can be null.

✓ **Referential Integrity Constraint**

It states that a tuple in one relation that refers to another relation must refer to an existing tuple in that relation.

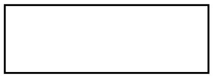
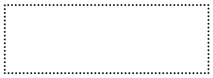
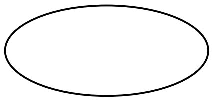
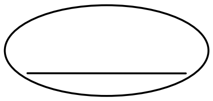
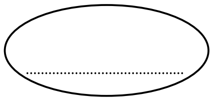
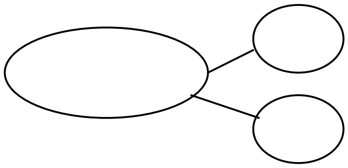
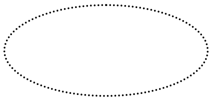
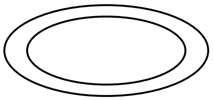
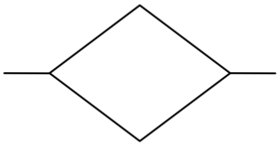
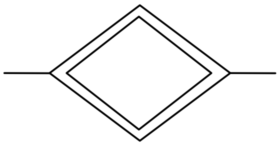
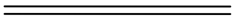
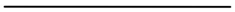
Foreign Key

A set of attributes F_k in a relation schema R_1 is a foreign key of $R_{1,}$ if it satisfies the following two rules:

1. The attributes in F_k have the same domain as the primary key attributes P_k of another relation schema R_2; the attributes F_k are said to refer to the relation R_2.
2. A value of F_k in a tuple t_1 of R_1 either occurs as a value of P_k for some tuple t_2 in R_2 or is null.

$$t_1 [F_k] = t_2 [P_k]$$

Notations used in ER Diagram

SYMBOL	**MEANING**
	Regular Entity
	Weak Entity
	Attribute
	Key Attribute
	Partial Key
	Composite Attribute
	Derived Attribute
	Multivalued Attribute
	Relationship
	Identifying Relationship
	Total Participation
	Partial Participation
1:1 , 1:M , M:N	Cardinality Ratio

ER to Relational Mapping Algorithm

Step 1: For each regular entity E in the ER diagram, create a relation R that includes all the simple attributes of E.

Step 2: For each weak entity W with owner entity E, create a relation R and include all simple attributes of W. In addition, include as foreign key attributes of R the primary key attributes of their owner E. The primary key of R is the combination of primary key of E and the partial key of W.

Step 3: For each binary 1:1 relationship type R relating to relations S and T, choose S or T and include the primary key of S or T as foreign key of T or S.

Step 4: For each binary 1:N relationship type R relating to relations S and T, include the primary key of 1-side as foreign key in the N-side.

Step 5: For each binary M:N relationship type R, create a new relation S whose primary key is a combination of the two foreign keys coming from the two relations associated with R. Also include as attributes of R the relationalship attributes.

Step 6: For each multivalued attribute A, create a new relation R that include and attribute corresponding to A plus the primary key attribute k (as a foreign key in R) as composite primary key.

Example Draw an ER Diagram from the following information.

(i) Workers (identified by WORK-NO), work on machines (identified by MACHINE-NO) to knit and produce garments.

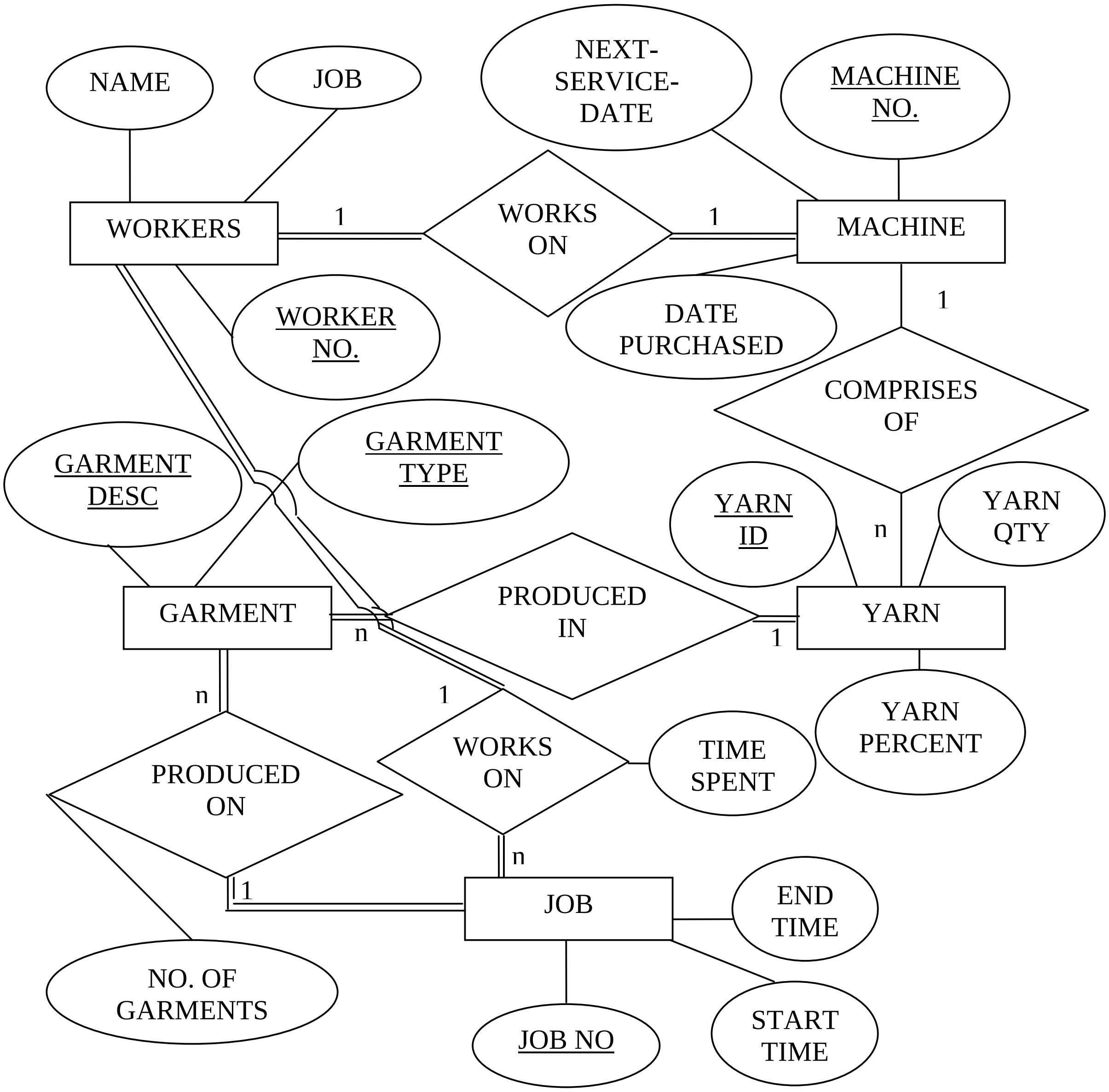
(ii) Various types of Garments (GARMENT-TYPES) are produced in the factory. Each GARMENT-TYPE has a description (GARMENT- DESCRIPTION) and is made up of different yams (YARN-ID) and has a fixed composition of yam (YARN-PER CENT) and quantity (YARN-QUANTITY).

(iii) The production of each garment is recorded as a job identified by JOB-NO. Each JOB-NO has a START-TIME and END-TIME and is performed by one worker on one machine. A number of garments of different kinds are produced on one job.

(iv) Other information required is:

- NAME and JOB of workers
- DATE-PURCHASED and NEXT-SERVICE-DATE for machines.
- TIME-SPENT by each worker on a job
- NO-OF-GARMENTS produced on one Job.

Also draw a Relational Schema Diagram from the ER Diagram.



. Draw a E-R diagram from the following information. (10)

(a) Departments are identified by DEPT-NO and have a budget allocated for them. A department can handle and manage many projects, but only one department manages each project. Projects are identified by Project-No and have a START- DATE.
(b) An order is prepared for each item. The order with a unique ORDER-NO and ORDER-DATE is made for any number of Parts (identified by ITEM-NO) and QUANTITY-ORDERED. Each order is made to one supplier. Supplier has a unique SUPPLIER-CODE, NAME and ADDRESS.
(c) A fault occurs on one item of equipment. A log-book contains FAULT-NO, FAULT-DATE and FAULT-DESCRIPTION. Each item of equipment and TYPE. Each such item is located in one building which has a unique BUILING-NAME and ADDRESS.

Input-Output Design

Form Design

Requirements of Form Design

Forms design follows analyzing forms, evaluating present documents, and creating new or improved forms. Since the purpose of a form is to communicate effectively through forms design, there are several major requirements:

1. *Identification and* wording. The form title must clearly identify its purpose. Columns and rows should be labeled to avoid confusion. The form should also be identified by firm name or code number to make it easy to reorder.

2. Maximum *readability and use.* The form must be easy to use and fill out. It should be legible, intelligible and uncomplicated. Ample writing space must be provided for inserting data. This means analyzing for adequate space and balancing the overall forms layout administration, and use.

3. *Physical factors*. The form's composition, color, layout (margins, space. etc,), and paper stock should lend themselves to easy reading. Pages should be numbered when multipage reports are being generated for the user.

4. *Order of data items.* The data requested should reflect a logical sequence. Related data should be in adjacent positions. Data copied from source documents should be in the same sequence on both forms. Much of this design takes place in the forms analysis phase.

5. *Ease of data entry*. If used for data entry, the form should have field positions indicated under each column of data and should have some indication on of where decimal points are (use broken vertical lines).

6. Size and arrangement. The form must be easily stored and filed. It should provide for signatures. Important items must be in a prominent caption on the form.

7. *Use of instructions.* The instructions that accompany a form should early show how it is used and handled.

8. *Efficiency considerations.* The form must be cost effective. This means eliminating unnecessary data and facilitating reading lines across the form.

9. *Type of report*. Forms design should also consider whether the content is executive summary, intermediate managerial information, or supporting-data. The user requirements for each type often determine the final form design.

Menu Design

The human-computer dialogue defines how the interaction between the user and the computer takes place. There are two common types of dialogues:

1. Menu
2. Question and answers

1. Menu: With a menu dialogue, a menu displays a list of alternate selections. The user makes a selection by choosing the number or letter of the desired alternative. In menu selection, the users read a list of items and select the one most appropriate to their task, usually by highlighting the selection and pressing the return key, or by keying the menu item number.
2. Questions and answers: With a question and answer, dialogue, questions and alternative answers are presented. The user selects the alternative that best answers the question.

Output Design

Output is information delivered to users through information system. Output can take many forms:

- Traditional hard copy of printed reports
- Soft copy such as display screens
- Microforms
- Audio/Video output

There are five objectives for output:

1. Design output to serve the intended purpose.
2. Design output to fit the user.
3. Deliver the appropriate quantity of output.
4. Assure that the output is where it is needed.
5. Provide the output on time.

System Testing & Quality Assurance

Testing

Testing is vital to the success of the system. System testing makes a logical assumption that if all the parts of the system are correct, the goal will be successfully achieved.

Another reason for system testing is its utility as a user-oriented vehicle for implementation.

Techniques used for system testing

1. *Online response.* Online systems must have a response time that will not cause a hardship to the user. One way to test this is 10 input transactions on as many CRT screens as would normally be used in peak hours and time the response to each online function to establish a true preference level

2. *Volume*. In this test, we create as many records as would normally be produced to verify that the hardware and Software will function correctly. The user is usually asked to provide test data for volume testing

3. *Stress testing*. The purpose of stress testing is to prove that the candidate system does not malfunction under peak loads. Unlike volume testing, where time is not a factor, we subject the system to a high volume of data for a short time period. This simulates an online environment where a high volume of activities occurs in spurts.

4. *Recovery and security.* A forced system failure is induced to test a backup recovery procedure for file integrity. Inaccurate data are entered to see how the system responds in terms of error detection and protection. Related to file integrity is a test to demonstrate that data and programs are secure from unauthorized access.

5. *Usability documentation and procedure*. The usability test verifies the user-friendly nature of the system. This relates to normal operating and error-handling procedures, for example. One aspect of user-friendliness is accurate and complete documentation. The user is asked to use only the documentation and procedures as a guide to determine whether the system can be run smoothly

The Nature of Test Data

The proper choice of test data is as important as the test itself. If test data as input are not valid or representative of the data to be provided by the user, then the reliability of the output is suspect. Test data may be artificial or live.

Activity Network for System Testing

A test plan includes the following activities:

1. Prepare test plan.
2. Specify conditions for user acceptance testing.
3. Prepare test data for program testing.
4. Prepare test data for transaction path testing.
5. Plan user training.
6. Compile /assemble programs.
7. Prepare job performance aids.
8. Prepare operational documents.

System Testing

The purpose of system testing is to identify and correct errors in the candidate system. In system testing, performance and acceptance standards are developed. Substandard performance or service interruptions that result in system failure are checked during the test. The following performance criteria are used for system testing:

1. ***Turnaround time*** is the elapsed time between the receipt of the input and the availability of the output in online systems; high-priority processing is handled during peak hours, while low-priority processing is done later in the day or during the night shift. The objective is to decide on and evaluate all the factors that might have a bearing on the turnaround time for handling all applications.

2. ***Backup*** relates to procedures to be used when the system is down. Backup plans might call for the use of another computer. The software for the candidate system must be tested for compatibility with a backup computer.

In case of a parallel system breakdown, provisions must be made for dynamic reconfiguration of the system. For example, in an online environment, when the printer breaks down, a provisional plan might call for automatically "dumping" the output on tape until the service is restored.

3. ***File protection*** pertains to storing files in a separate area for protection against fire, flood or natural disaster. Plans should also be established for reconstructing files damaged through a hardware malfunction.

4. ***The human factor*** applies to the personnel of the candidate system. During system testing, Lighting, air conditioning, noise and other environmental factors are evaluated with people's desks, chairs, CRTs, etc. Hardware should be designed to

match human comfort. This is referenced to as *ergonomics.* It is becoming an extremely important issue in system development.

Types of System Tests

System testing consists of the following steps:

1. Program(s) testing.

2.String testing

3. System testing.

4. System documentation.

5. User acceptance testing.

QUALITY ASSURANCE

Quality assurance defines the objectives of the project and reviews the overall activities so that errors are corrected early in the development process. Steps are taken in each phase to ensure that there are no errors in the final software.

Quality Assurance Goals in the Systems Life Cycle

The software life cycle includes various stages of development and each stage has the goal of quality assurance. The goals and their relevance to the quality assurance of the system are summarized next.

Quality Factors Specifications

The goal of these stages is to define the factors that contribute to the quality of the candidate system. Several factors determine the quality of a system:

1. *Correctness*-the extent to which a program meets system specifications and user objectives.

2. *Reliability*-the degree to which the system performs its intended functions over a time.

3. *Efficiency*-the amount of computer resources required by a program to perform a function.

4. *Usability*-the effort required to learn and operate a system.

5. *Maintainability*-the ease with which program errors are located and corrected.

6. *Testability*-the effort required to test a program to ensure its correct' performance.

7. *Portability*-the ease of transporting a program from one hardware configuration to another.

8. *Accuracy*-the required precision in input editing, computations, and output.

9. *Error tolerance*-error detection and correction versus error avoidance.

10. *Expandability*-ease of adding or expanding the existing database.

11. *Access control and audit*-control of access to the system and extent to which that access can be audited.

12. *Communicativeness*-how descriptive or useful the inputs and outputs of the system are.

Levels of Quality Assurance

Quality assurance specialists use three levels of quality assurance.

a. *Testing* a system to eliminate errors.

b. *Validation* to check the quality of software in both simulated and live environments.

c. *Certification* that the program *o*r software package is correct and conforms to standards.

Implementation

Implementation is the process of converting a new or revised system design into an operational one. Conversion is one aspect of implementation. The other aspects are the post implementation review and software maintenance.

There are three types of implementation:

1. Implementation of a computer system to replace a manual system.
2. Implementation of a new computer system to replace an existing one.
3. Implementation of a modified application to replace an existing one.

Conversion

Conversion means changing from one system to another. The objective is to put the tested system into operation while holding costs, risks, and personnel irritation to a minimum. It involves

(a) creating computer-compatible files,

(b) training the operating staff, and

(c) installing terminals and hardware.

Activity Network for Conversion

The activities of conversion are as follows.

1. Conversion begins with a review of the project plan, the system test documentation, and the implementation plan. The parties involved are the user, the project team, programmers, and operators.
2. The conversion portion of the implementation plan is finalized and approved.
3. Files are converted.
4. Parallel processing between the existing and the new systems is initiated.
5. Results of computer runs and operations for the new system are logged on a special form.
6. Assuming no problems, parallel processing is discontinued. Implementation results are documented for reference.
7. Conversion is completed. Plans for the post-implementation review are prepared. Following the review, the new system is officially operational.

Post-Implementation Review

It has the objective of evaluating the system is terms of how well performance meets stated objectives. The study begins with the review team, which gathers requests for evaluation. The team prepares a review around the type of evaluation to be done and the time frame for its completion. An overall plan covers the following areas.

1.***Administrative Plan*** – Review area objectives, operating costs, actual operating costs, actual operating performance and benefits.

2. ***Personal requirement plan*** – Review performance objectives and training performance to date.

3***. Hardware plan*** - Review of performance specifications.

4. ***Documentation review plan*** – Review the system development effort.

Maintenance

Maintenance is actually the implementation of the post- implementation review plan. It can be classified as follows.

1. **Corrective maintenance** means repairing processing or performance failures or making changes because of previously uncorrected problems or false assumptions.

2. **Adaptive maintenance** means changing the program function.

3. **Perfective maintenance** means enhancing the performance or modifying the program(s) to respond to the user’s additional or changing needs.

Out of these three types of maintenance, more time and money are spent on perfective than on corrective and adaptive maintenance together.