UNIT:- 3

System Engineering and Quality Assurance

(most Imp)

Design of Software: - (Software Design concepts and principles)

Software Design Concepts and Principles

Ans: - **1. Modularity and Portioning:-**

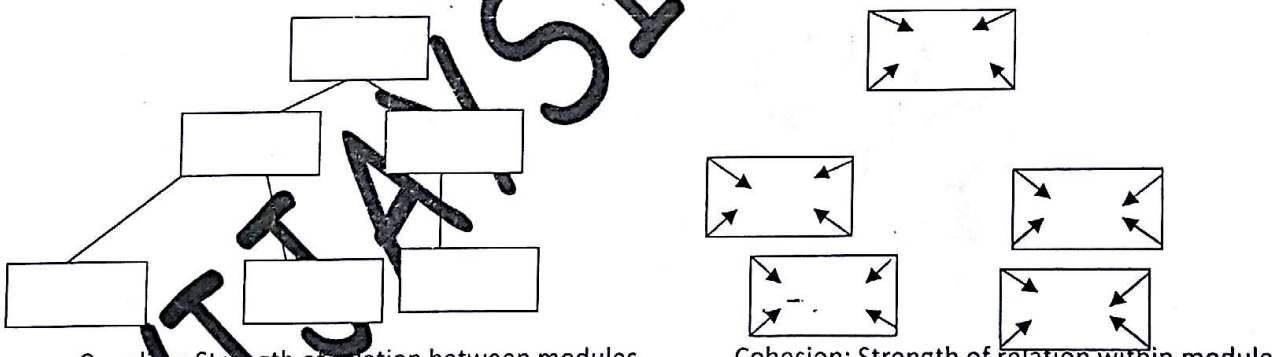
Each system should consist of a hierarchy of modules lower level modules are generally smaller in scope and size compared to higher level modules and serve to partition processes into separate function.

2. Coupling:-

Coupling refers to the strength of a relationship between modules in a system. Modules should have little dependence on other modules in a system.

Loose coupling minimize the interdependence between module we can achieve this in the following way.

- Control the number of parameters passed between modules.
- Avoid passing unnecessary data to called modules.
- Pass data (Whether upward or downward) only when needed.



Coupling: Strength of relation between modules Cohesion: Strength of relation within module

Fig: - Coupling & Cohesion in software

3. **Cohesion:-**

Cohesion defines the strength of relations within a module. The content of the modules should be so designed that they perform a specific function.

i.e.:- Modules should carry out a single processing system.

There are four general type of module contains which are discuss below in order of list to most to most desirable.

1. Modules contents determined by function performed:-

All activities in a module have the same single purpose. I.e. perform a single function.

Ex: - Calculate finance charge on an account balance edits a transaction.

2. Module contents determined by data used:-

   All elements in the module refer to the same data or files.

   Ex: - Printing, displaying and copying data from a common file.

3. Module contents determined by logic of processing:-

   All steps are performed together or handle the same functions.

   Ex: - All input operation all output operation and all initialization activities.

4. Module contents not closely related:-

   Modules are developed by size or number or instruction.

   Ex: - All modules are no more than 50 statements in length: all modules must fit on a single page.

5 Span of control:--

Span of control refers to the number of subordinate modules control by a calling module. Modules should interact with and manage the functions of a limited number of lower level modules.

5 Size:--

Size implies the number of instructions composing a module the number of instructions contained in a module should be limited so that module size is generally small.

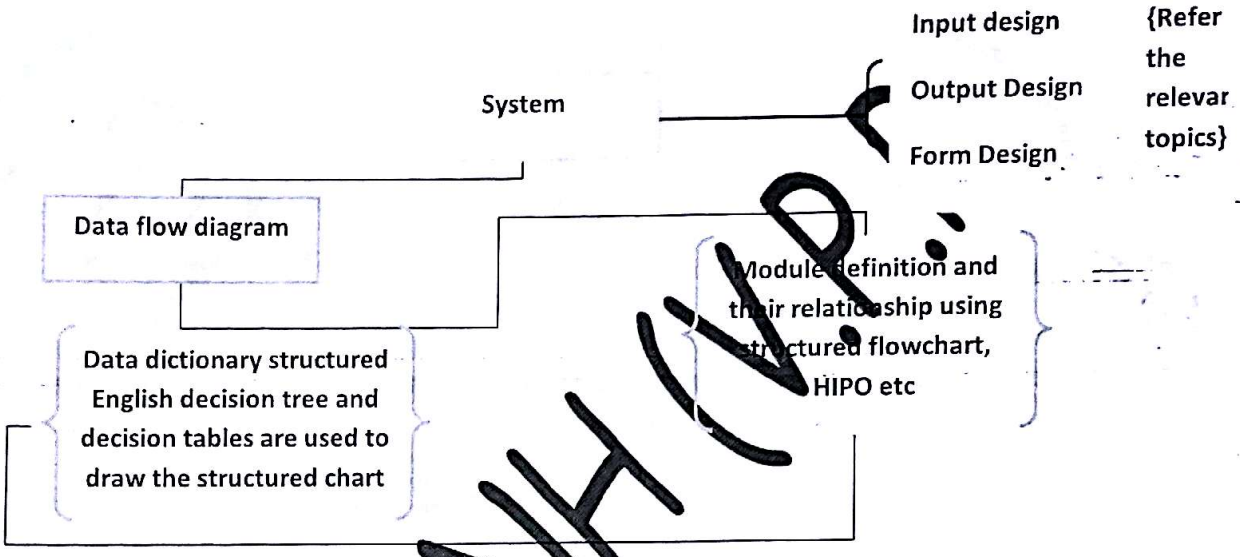
6 Shared Use:--

Shared use means use of modules by other calling modules. Function should not be duplicated in separate modules but establish in a single module that can be invoke by any other module when needed.

Software Design and Documentation tools

Structured Chart Techniques

Structured design partitions a program or a system into small independent modules in the structured design method, the approach as given below.



1. Structured flow charts.

2. Hierarchical input process output (HIPO) chart including visual table of contents (VTOC) and input process output charts (IPO).

3. Warnier - Orr Diagrams.

There are important tools in the system and software design. They are graphical and/or visual displays having a top down approach.

Structure Flowchart

Structured Flow chart also called Nassi Schneider man chart, are graphic tools that force the designer to structure software that is both modular and top down. They provide the structure that can be retained by programmer who develops the application software. Organization responsibilities very. In some Organizations analysts are responsible for developing module logic while in other that responsibility is delegated to the Programmer

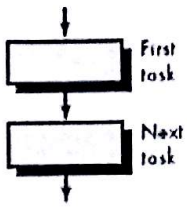
Basic Element:-

1. Process
2. Decision
3. Iteration

Process:-

Simple processer or steps in a program symbol. This symbol represents initialization of value, Input and output activates and call to execute other procedures.

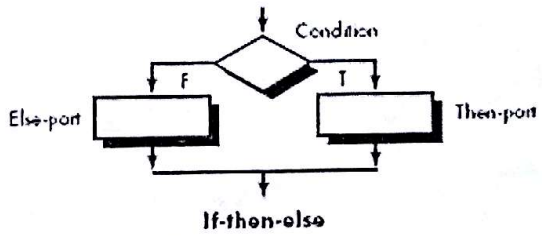
A name of brief description written in the box states the purpose of the process the succession of step shown using several process



most imp

Decision:-

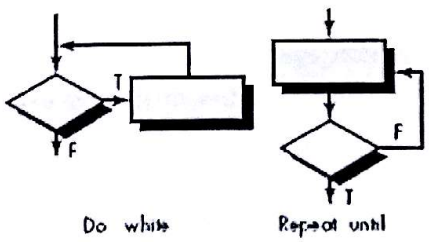
The decision symbol represents alternative condition that can occur and that program must have a manner of handling they show the equivalent of the If then else structure as discussed previously under structured English and common in many programming language. As Example will show the decision symbol may show action for more than two alternatives at the same time.



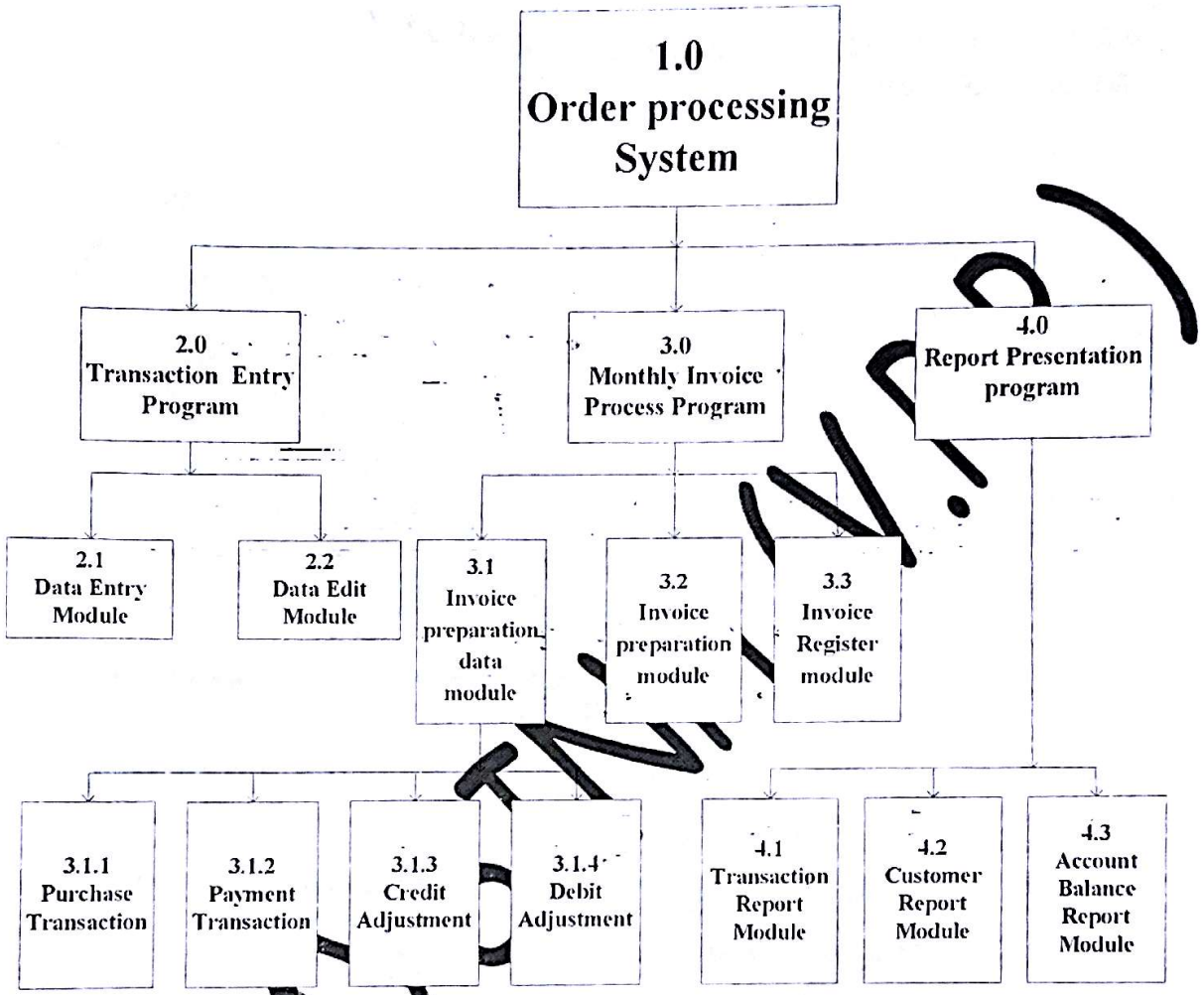
If-then-else

Iteration:-

The iteration symbol represents looping and repetition of operation while a certain condition exists or until A condition exist the form of the iteration symbol clearly show the scope of the iteration including all processes and decisions that are contained within the loop. The soft hand portion of the symbol shows path of repetition to follow. Until the condition controlling the iteration is satisfied.



HIERARCHICAL INPUT PROCESS OUTPUT CHART [HIPO]



This is commonly used tool for developing system software. They are used in documenting information. It was first developed by IBM as a design aid and for large complex operating system. They are graphic rather than narrative. HIPO emphasizes on functions of the system rather than structure, logic or organization.

The components of a HIPO package are:

1) Visual tables of Contents.
2) Functional diagrams.

- Functional Diagrams:-

The input – process – output portion of a HIPO chart identifies for each program module what inputs, processing takes place and outputs are involved. The functional diagrams represent this IPO charts in traditional flowchart symbols showing movement of data and control points.

They assist the analysts in answering three guiding Question

1. What does the system or Module do?
2. How does the system or Module do? (How does It do it/)
3. What are the input and output?

HIPO charts have the following components:- most Imp

Visual Table of Contents

- Visual Table of Contents: - The visual table of content (VTOC) showed the relation between each of the Documents Making up a HIPO Packages.
- It consists of a hierarchy chart that identifies the modules in a system by number and in relation to each other and gives a brief description of each module.

Advantages of HIPO:-

- HIPO allows a program or a system to be easily understood.
- HIPO is design, development and documentation tool.
- HIPO packages provide a common, visual base for education and communication.
- HIPO has less duplication on information and more information can be obtained in a glance.
- It is a top-down approach with successive levels going for greater detail.
- In HIPO, errors can be detected and isolated on a functional basis.
- HIPO indicates clearly what input, processing and outputs are involved in each programming module.
- HIPO is designs computer programs in modules that can be individually assigned, designed and tested.
- HIPO is a management tool because the functional approach allows planning and scheduling to be made accurately and early in the system development life cycle.

Disadvantages of HIPO:-

- HIPO concentrates more on what is to be accomplished rather than how it is to be accomplished.
- Requires extra documentation beyond flowcharts or coding.
- Necessitates some training for programmer before use.

WARNIER-ORR DIAGRAM:-

W-O diagrams were first developed by jean-Dominique Warnier, a Belgian mathematician and later extended by Kenneth Orr; an American W-O diagram technique is a system design and program design technique. Known as Logical construction of programs / Logical construction of system.

This method aids the design of program structures by identifying the output and processing result and then working backwards to determine the steps and combination of input needed to produce them

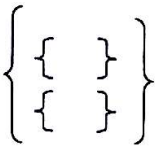
It's approach is a structured, hierarchical methods using brackets that flow left to right .like all structured design methodologies, the basic building blocks are:-

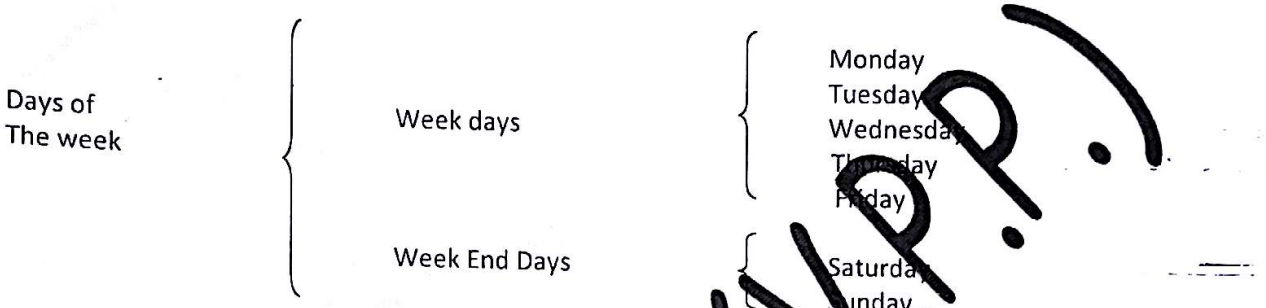
1).Process
2).Decision
3).Iteration

The W-O techniques consist of several steps that end in the programming of the system. The first step in W-O technique is to identify system. Output (result) There are then developed in to data or functions in a backwards hierarchical manner.

Brackets

Denote sets and subsets





The set of day in the week has weekdays and weekend as subset.
An illustration of W-O diagram for processing taxes in a payroll program.

❖ **Managing Quality Assurance:-**

Quality assurance is the review of software product and related documentation for completeness, Correctness, Reliability, and Maintainability and of course, it includes assurance that the system meets the specification and the requirements for its intended use and performance.

Levels of Assurance:-

(1) Testing
(2) Verification & Validation
(3) Certification.

Verification and Validation:-

Verification:-

Verification refers to the set of activities that ensure that software correctly implements a specific function.

Validation:-

Validation refers to a different set of activities that ensure that the software that has been built is traceable to customer requirement

Verification:-"Are we building the product right?"

Validation:-"Are we building the right product?"

❖ **Quality Control V/S Quality Assurance**

Quality Control	Quality Assurance
1. Correction (Reactive)	1. Prevention (Proactive)
2. Product	2. Process
3. Confident to Producer	3. Confident to Customer
4. Line Function	4. Staff Function
5. Find defects	5. Prevent defects
6. Ex: - Testing, Check Point, Reviews	6. Quality, Audit, Training.

❖ **TESTING:-**

"Testing is the process which is used to find errors not to solve errors."

SYSTEM TESTTING:-

It tests to find discrepancies between the system and its original objective, current specifications, and system documentation. The primary concern is the compatibility of individual modules; System testing must verify that file sizes are sufficient and that indices have been built properly. Sorting and rendering procedures assumed to be present in lower- level modules must be tested at the system level.

PEAK LOAD TEST:-

Determine whether the system will handle the volume of activities that occur when the system is at the peak of its processing dim and time when the user want to work with web application and at the same time the page will load slowly so at that time it also test the loading time of page.

STORAGE TESTING:-

Determine the capacity of the system to store transaction data or in other files.

PREFORMANCE TESTING:-

Determine the length of time system used by the system to process transaction data

E. g: - Responses time form server when system is fully loaded with operating data.

RECOVERY TESTING:-

Determine ability of user to recover data or restart system after failure.

PROCEDURE TESTING:-

Determine clarity of documentation on operation and use of system by having done exactly what manuals request.

HUMAN FACTORS TESTING:-

Determine how users will use the system when processing data or preparing reports.

❖ DOCUMAENTATION:-

Anything that is written about how a system is designed or functions is documentation. An exhaustive documentation list as given Fit is listed here:

SYSTEM DOCUMENTATION:-

It describes the overall system design and includes system flowchart input / output formats, file descriptions, control requirements, report specifications etc.

For example it will contain,

<1>

- Name of the system.
- Why the system was developed.
- Purpose and objective.
- Who are the users?
- Where the system fits in the company.
- Description of design methodology (T/S).

<2>

<A>Narrative description:-

<B>Structured documentation:-

- Context DFD.
- Logical DFD.
- Level 0 of Physical DFD.
- Lower Level DFDS.
- Data Structure and Data Access Diagrams.
- Data Dictionary.
- Mini Specifications for Processes Including Logic.
  - Decision Table.
  - Decision Trees.
- System Structured Charts.

<C> Traditional documentation:-

- System Flowchart.
- Documented flowchart and Documentation section.
- System Requirement specification.
- Input / Output sheets.
- Equipment sheets.
- Personnel sheets.
- File sheets.
- Lay out charts and cost sheets.

<D>Control's Matrix and Auditor's Report

PROGRAMMING DOCUMENTATION:-

This includes programming specifications, descriptions of program logic including graphic aids such as program flowcharts. Design and program documentation are for the use by technical personnel. Amore detailed list id give here:

Program listing.

- Comments and note statements.
- Structured charts such as nassi- shneiderman charts.
- One program flowchart for each program and an overall flowchart showing how the various programs interact. (system structure charts)
- Detailed narrative descriptions and decision trees or decision tables for complicated logic or calculations within each program (mini specifications).
- Pictorial layouts of files, the outputs and input (data structure diagrams, data access diagrams).

OPERATIONS DOCUMENTATION:-

It is for those who will keep the system running from day to day. It contains:-

- Operating instructions for normal operations.
- Directions for handling problems and breakdowns.

TRAINING DOCMENTATION:-

Includes the user training manuals and materials to be used in the conversion and installation of new system. Users must know to fill out forms, how to correct errors and how to interpret output.

IMPLEMENTATION PLANS:-

Implementation plans and the results of implementation must be documented.

APPENDIX:-

This contains all documentation, for example,

- Feasibility study report.
- Problem definition report.
- Study plan.
- Normalization documents.

- Summary of general information understandings existing system, new systems requirements, economic cost comparisons.
- Final written report.
- List of controls.

Thus it becomes evident that documentation

- Should be adequate.
- Should be clear and,
- Should be updated regularly.

❖ **NEED FOR DOCUMENTATION:-**

- Without proper documentation communication of who, what, where, when and of system is difficult.
- Audit ability and control are difficult to achieve without documentation.
- If changes are made then it will not be clear to those who try to comprehend the system later on.
- While converting from one system to another system if there is no proper documentation, there will be chaos.

❖ **SYSTEM IMPLEMENTATION**

"Implementation includes all those activities that take place to convert the old system into a new one from the old system to the new"

TRAINING:-

These who will be associated with or affected by the system must know in detail what their roles will be, how they system, and what the system will or will not do. Both system operators and users need training.

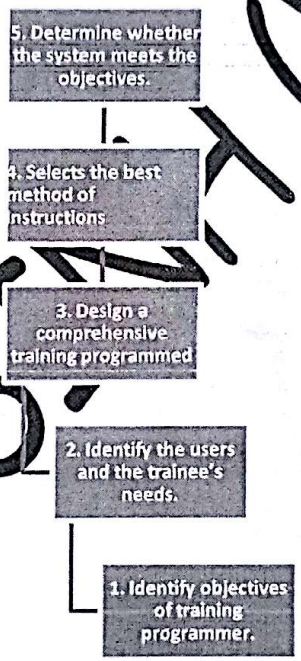
Training must ensure that they are able to handle all possible operations, both routine and extraordinary, operator training must also involve the data entry personal.

Training operates should be given both a troubleshooting list that identifies possible problems and remedies for them, as well as names and telephone number of individuals to contact when unexpected problems arise.

- Training may be for system operators and users. This is done with a view to providing hands-on experience with the new system. With interactive system, user can try out software directly.
- In fact training should include an overview of how the system functions. How it will affect their jobs and. How it will relate to current manual procedures.
- This is the point where structured system analysis pays off. Since many users will have already been intimately involved in the creation of the system from the start. Every employee needs to be reassured that the system isn't a threat to his job and he also has something to gain from it.

The training programmers should be carefully planned and organized.

The five important steps to such programmers are:-



Further, the training should specifically take care of:-

- User involvement in the equipment use.
- Instruction to individuals in troubleshooting the system and coming out unscathed from troubles.
- Data handing.
- System maintenance.

Managing System Implementation

Implementation includes all those activities that take place to convert the old system into a new one from the old system to the new

Replacing an existing manual or automated system or it may be a major modification to an existing system.

- Training
- Conversion

Training

These who will be associated with or affected by the system must know in detail what their roles will be, How they system and what the system will or will not do both system operators and users need training

System Operators Training:-

Running of the system successfully depend on the personnel working computer center. They are responsible for providing the necessary support there must ensure that they are able to handle all possible operation, both routine ordinary in nature.

User Training:-

User may be trained on use of equipment, particularly in the case where for example micro computer is in used and the individual involved is both operator and user cases user must be given training on How to operate the system also.

Training Method:-

Training of operators and users can be organized in several different ways most important are:-

(1) Vendor and in Service Training
(2) In House Training

- **CONVERSION:-**

Conversion is the process of changing from the old system to the new one. There are four methods of handing a system conversion.

1).THE PARALLEL-SYSTEM METHOD: -

In the parallel-systems method, the new system is set to work alongside the one. Data are input to both simultaneously until the new system has demonstrated that it functions effectively. This method is safer

because we need not convert to new system unless it performs satisfactorily. However, the cost conversion is one disadvantage. But, the switch over will be natural and the user can see the benefits of the new system.

2).THE DUAL SYSTEM METHOD OR PHASE-IN METHOD:-

Here system is gradually phased out while the new one is being phased in. the cost of conversion is lower and there is little duplication of work or data .in addition direct comparisons between new and old system need not be made. However, long phase-in periods may create difficulties for analysts. Also, if there are problem in the early phases of implementation rum or about difficulties may assumes greater proportions and the remaining phases. In certain system dual conversion may not be possible at all.

3).THE DRIECT CONVERSION METHOD:-

Here the conversion takes place abruptly. The old system work till a particular day when it is replaced by the new system. There are no parallel activities. There is no falling back to old system. But this method requires careful planning and training sessions must be scheduled and maintained in the case of hotel reservation, airline reservation etc this method is more suitable.

4).PILOT APPROACH:-

In this method, a working version of the system is implementing in one part of the organization such as in a particular department. The user knows that it is pilot testing and hence they can experiment to improve the system. When the system is deemed complete it can be installed throughout the organization either by direct cutover method or by phase-in method