UNIT-5

System Implementation

System Implementation is the way of carrying out a developed system into working condition. System implementation involves activities like coding, testing, installation, documenting system & training and supporting users.

⊛. The process of Coding, Testing and Installation:

→ Coding is the process by which the physical design specifications created by analysis team are turned into working computer code by the programming team.

→ Once coding has begun, the testing process can begin and proceed in parallel. As each program module is produced, it can be tested individually, then as a part of larger program, and then as a part of larger system.

→ Installation is the process during which the current system is replaced by the new system. This includes conversion of existing data, software, documentation, and work procedures to those consistent with the new system.

⊛. The process of Documenting the system, Training Users and Supporting Users:

→ Although the process of documentation proceeds throughout the life cycle, it receives, formal attention during the implementation phase because the end of implementation largely marks the end of the analysis team's involvement in systems development.

→ Larger organizations also tend to provide training and support to computer users throughout the organization. Some of the training and support is very specific to particular application systems, whereas the rest is general to particular operating systems.

⊛. Software Application Testing:

Software testing is a method to judge the functionality of a software program. It is the process of verifying and validating that a software or application is bug free and meets the technical requirements. It also measures efficiency, accuracy and usability of software or application.

Types of testing:

i> **Inspections:** A testing technique in which participants examine program code for predictable language specific errors. Syntax, grammar, and some other routine errors can be checked by automated inspection software.

ii> **Desk checking:** A testing technique in which the programmer understands the logic of the program works through the code with a paper and pencil. The reviewer acts as the computer, mentally checking each step and its results for the entire set of computer instructions.

iii> **Unit testing:** Automated technique where each module is tested alone in an attempt to discover any errors that may exist in the module's code.

iv> **Integration testing:** The process of bringing together more than one modules that a program comprises for testing purposes.

v> **System testing:** The process of bringing together of all of the programs that a system comprises for testing purposes. The system can be tested by black box testing or white box testing.

vi> **Stub testing:** It is a technique used in testing modules, especially where modules are written and tested in a top down fashin.

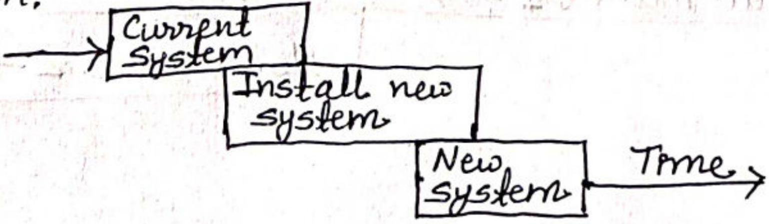
⊛. Differences between Black Box Testing and White Box Testing:

Black Box Testing	White Box Testing
i) Black box testing is the software testing method which is used to test the software without knowing the internal structure of code or program.	i) White box testing is the software testing method in which internal structure is being known to tester who is going to test the software.
ii) This type of testing is carried out by testers.	ii) This type of testing is carried out by developers.
iii) Black box testing means functional test or external testing.	iii) White box testing means structural test or interior testing.
iv) Programming knowledge is not required to carry out black box testing.	iv) Programming knowledge is required to carry out white box testing.
v) Implementation knowledge is not required for black box testing.	v) Implementation knowledge is required for white box testing.

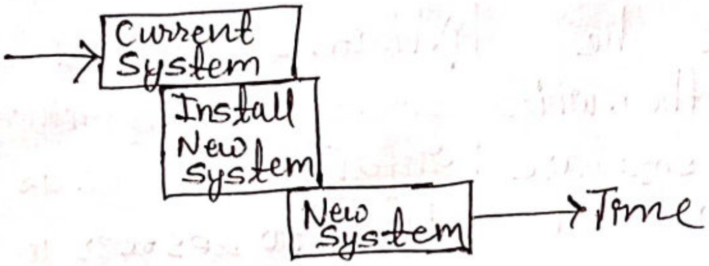
⊛. Installation and its types/approaches: [Imp]

Installation is the process during which the current system is replaced by the new system. This includes conversion of existing data, software, documentation, and work procedures to the new system.

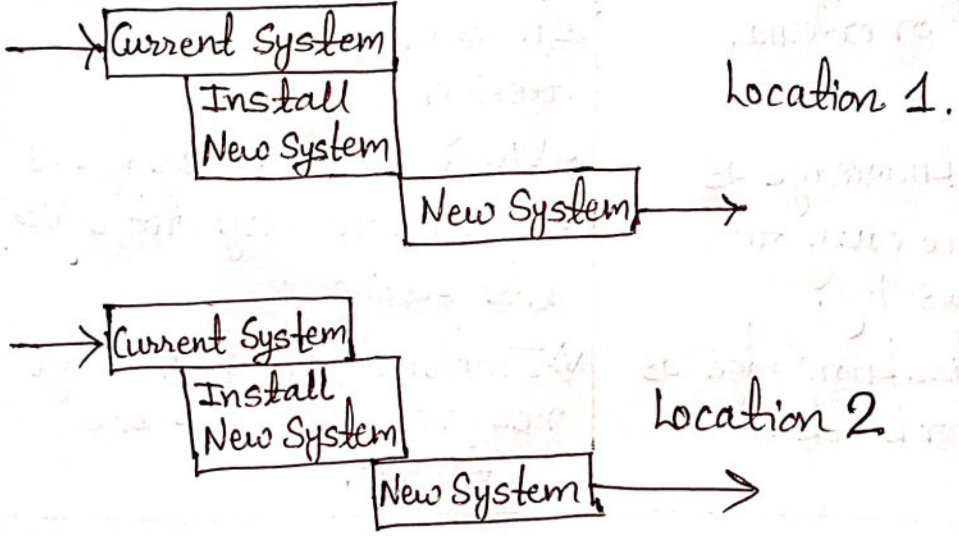
i) Direct installation: Changing over from the old system to a new one by turning off the old system when the new system is turned on.



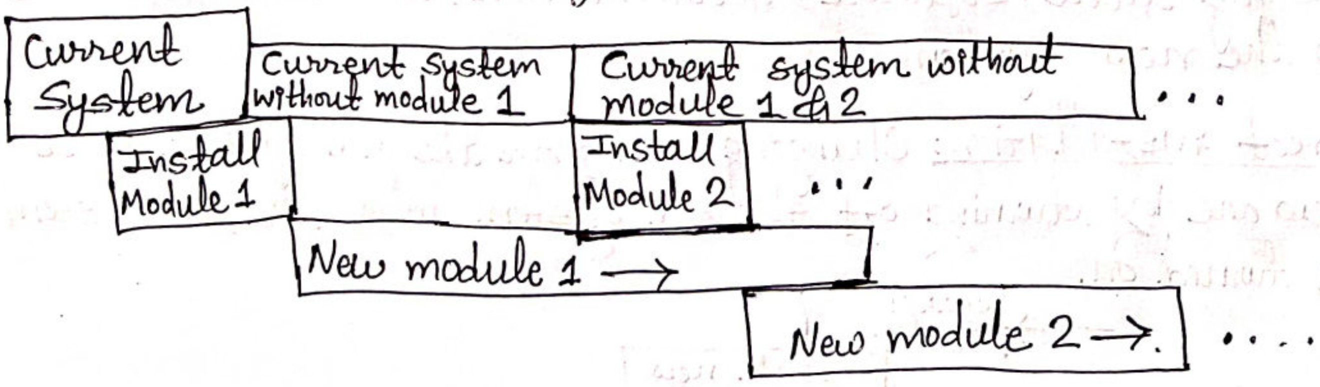
ii) Parallel installation: Running the old information system and the new one at the same time until management decides the old system can be turned off.



iii) Single-location installation: Trying out a new information system at one site and using the experience to decide if and how the the new system should be deployed throughout the organization. Also known as location or pilot installation.



iv) Phased Installation: Changing from the old information system to the new one incrementally, starting with one or a few functional components and then gradually extending the installation to cover the whole new system.



⊛. Documenting the system: [Imp.]

Documentation is the process of collecting, organizing, storing and maintaining a complete record of system and other documents used or prepared during the different phases of the life cycle of the system. System cannot be considered to be complete, until it is properly documented.

Types of documentation:

1) System documentation: It records detailed information about a system's design specification, its internal workings. and its functionality. System documention is intended primarily for maintenance programmers. It contains following information:

→ Detailed diagram of system flowchart and program flowchart.
→ Problem definition and the objective of developing the program.
→ Output and test report of program.
→ Upgrade or maintenance history, if modification of program is made.

There are two types of system documentation: Internal documentation and external documentation. Internal documentation is part of the program source code or is generated at compile time. External documentation includes the outcome of structured diagramming techniques such as data flow and entity-relationship diagrams.

2) User documentation: User documentation consists of written or other visual information about an application system, how it works and how to use it. User documentation is intended primarily for users. It contains the following information:

→ Set up an operational details of each system.
→ Loading and unloading procedures.
→ Special checks and security measures.
→ Problems which could arise, their meaning reply and operation action.

⊛ Training and Supporting users:-

The type of training needed will vary by system type and user expertise. Types of training methods are:-

→ Resident expert.
→ E-learning / distance learning.
→ Blended learning
→ Software help components
→ External sources such as vendors.

Computing supports for users has been provided in one of a few forums:

Automating support: online support forums provide users access to information on new releases, bugs and tips for more effective usage.

Providing support through a help desk: The help desk is the first place users should call when they need assistance with an information system. The help desk staff members either deal with the users questions or refer the users to the most appropriate person.

⊛. Maintaining Information Systems: [Imp]

Correcting and upgrading process of the system is called System maintenance. Maintenance is necessary to eliminate errors in the working system during its working life. Four major activities occur within maintenance:

i) Obtaining Maintenance requests: In this step a formal process is established where users can submit system change requests.

ii) Transforming request into changes: Once a request is received, analysis must be performed to identify the scope of request. It must be determined how request will affect the current system.

iii) Designing changes: A change request can be transformed into a formal design change.

iv) Implementing changes: In this activity proposed changes are implemented in respective components of system.

Types of system maintenance:

i) Corrective maintenance: It refers to changes made to repair defects in the design, coding or implementation of the system.

ii) Adaptive maintenance: It involves making changes to an information system to evolve it's functionality to changing business needs.

iii) Perfective maintenance: It involves making enhancements to improve processing performance or interface usability or to add desired system features.

iv) Preventive maintenance: It involves changes made to a system to reduce the chance of future system failure.

⊛. The cost of maintenance: [Imp]

Information systems maintenance cost are a significant expendicture. For some organizations, as much as 60 to 80 percent of their information systems budget is allocated to maintenance activities. On average, 52 percent of a company's programmers are assigned to maintain existing software. Only 3 percent are assigned to new application development.

Numerous factors influence the maintainability of a system. The most significant factors are: no. of latent defects, the number of customers, and documentation quality. The others are personnel, tools and software structure.

⊛. Managing Maintenance:

Maintenance activities consume more and more of the systems development budget, maintenance management has become increasingly important. It consists of following two things:

Managing Maintenance Personnel: Many organizations has a "maintenance group" that is seperate from the "development group". With the increased number of maintenance personnel, changing organizational forms, end-user computing etc. have rethought the organization of maintenance and development personnel.

Measuring Maintenance Effectiveness: To measure effectiveness we must measure the following factors:

→ Number of failures.

→ Time between each failure.

→ Type of failure.

⊛. Factors influencing maintenance cost:

i) Latent defects: This is the number of unknown errors existing in the system after it is installed. The no. of latent defects in the system influences most of the costs associated with maintaining a system.

ii) Number of customers: In general, greater the number of customers, greater the maintenance costs.

iii) System documentation quality: Without quality documentation, maintenance efforts can increase exponentially. High-quality documentation leads reduction in the system maintenance.

iv) Maintenance personnel: In some organizations, the best programmers are assigned for maintenance. Maintenance needs highly skilled programmers who can understand quickly and change software.

v) Tools: Tools that can automatically produce system documentation lowers maintenance costs.

vi) Well-structured programs: Well-designed system are caiser to understand and fix.