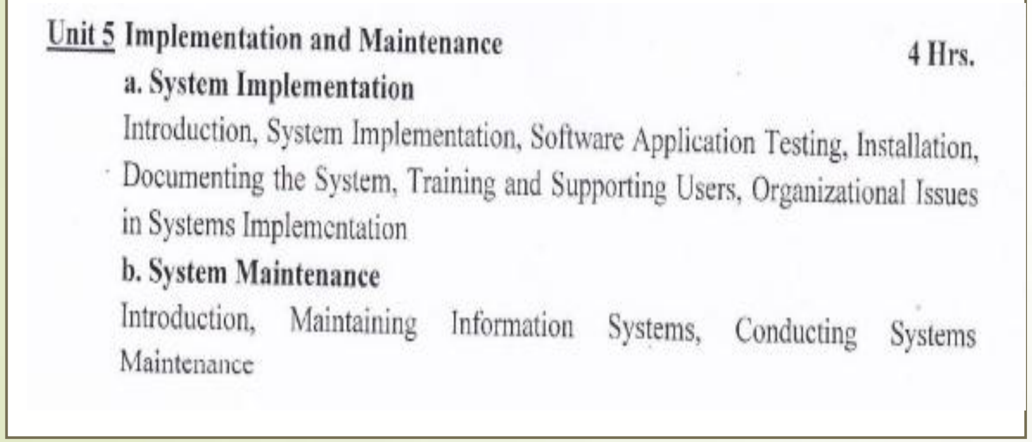
SYSTEM ANALYSIS AND DESIGN(SAD)

Chapter 5: Implementation & Maintenance



Unit 5 Implementation and Maintenance **4 Hrs.**

a. System Implementation

Introduction, System Implementation, Software Application Testing, Installation, Documenting the System, Training and Supporting Users, Organizational Issues in Systems Implementation

b. System Maintenance

Introduction, Maintaining Information Systems, Conducting Systems Maintenance

Compiled By : Er. Omkar Nath Gupta

Introduction

System implementation refers to the process of transitioning a newly developed or modified system into a functioning state, including ***Coding, Testing, installation,Doccumenation,Training and Suppor***t. It's a crucial phase in the systems development lifecycle, ensuring the system meets requirements and is effectively used by users.

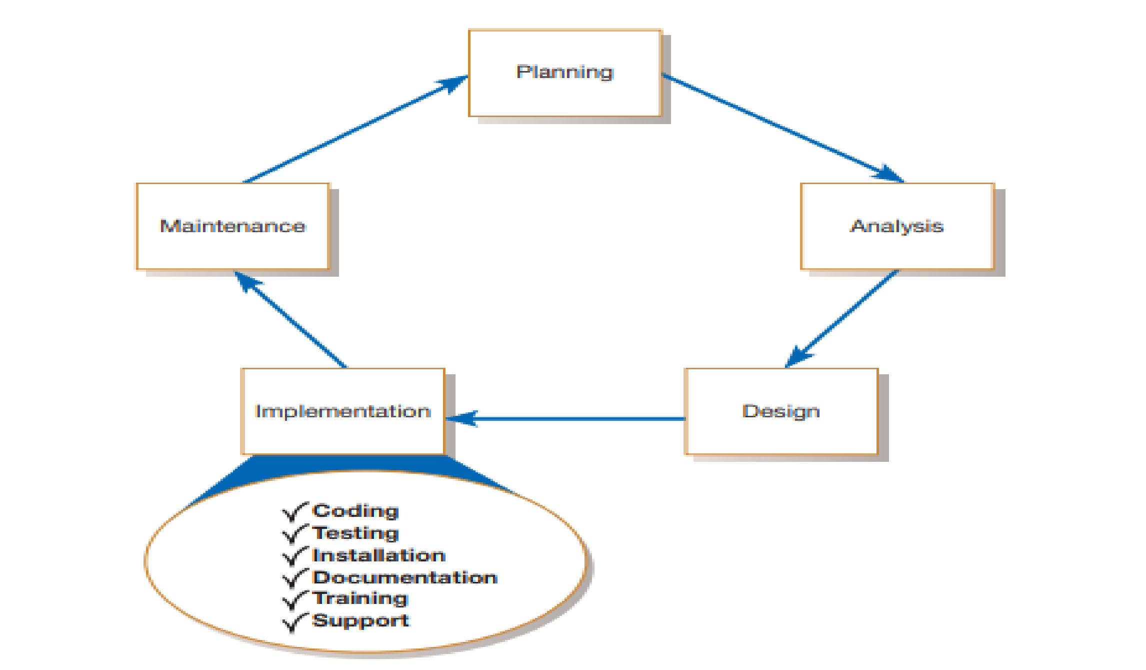


Fig: System development life cycle with the implementation phase highlighted

The process of Coding, Testing and Installation

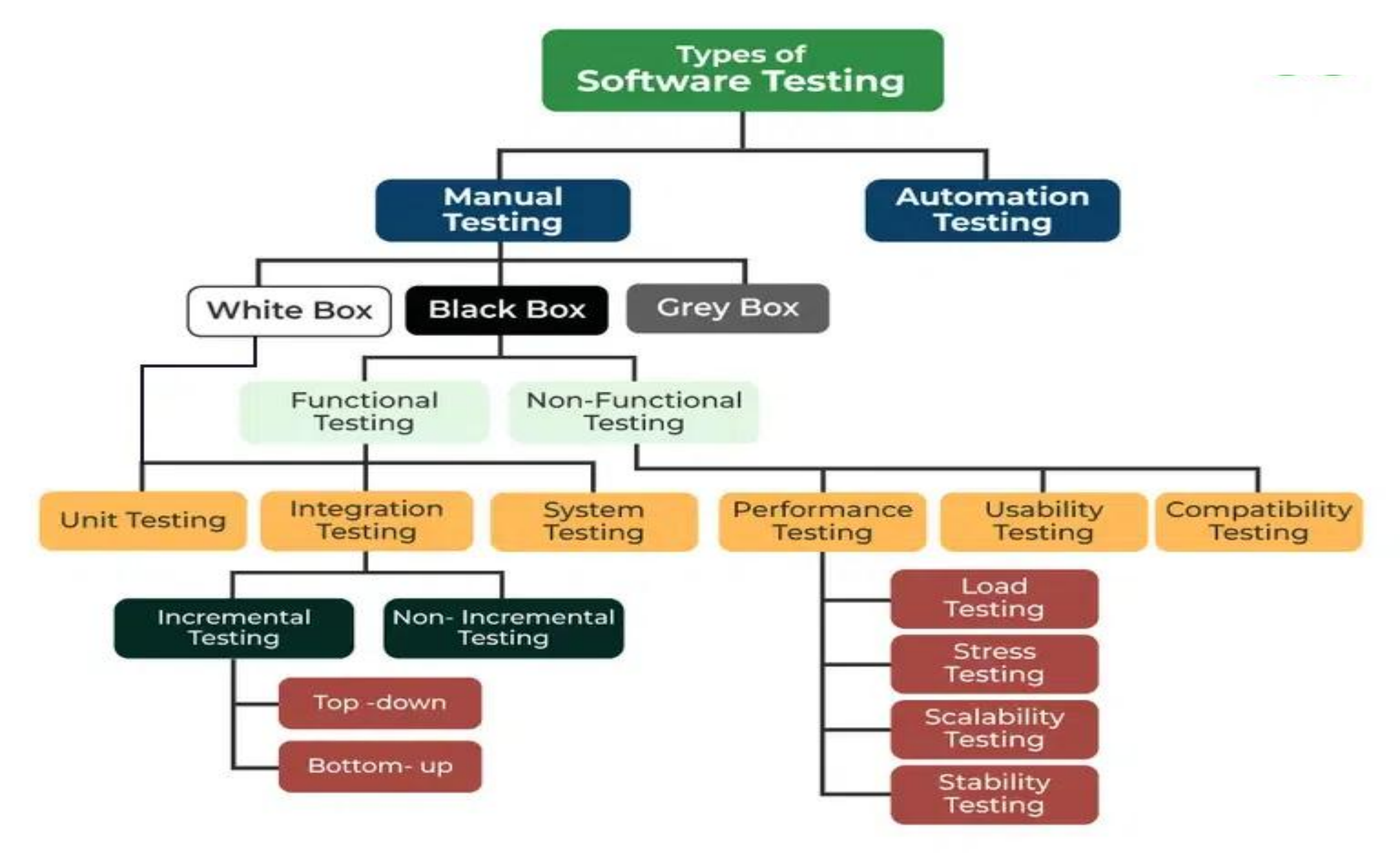
- **Coding** is the process whereby the physical design specifications created by the analysis team are turned into working computer code by the programming team. Depending on the size and complexity of the system, coding can be an involved, intensive activity.
- Once coding has begun, the **testing** process can begin and proceed in parallel. As each program module is produced, it can be tested individually, then as part of a larger program, and then as part of a larger system. Although testing is done during implementation, we must begin planning for testing earlier in the project. Planning involves determining what needs to be tested and collecting test data. This is often done during the analysis phase because testing requirements are related to system requirements.
- **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 Testing

Software testing is an important process in the Software Development Lifecycle(SDLC). It involves verifying and validating that a Software Application is free of bugs, meets the technical requirements set by its Design and Development, and satisfies user requirements efficiently and effectively.



Manual Testing : Manual testing is carried out by a tester manually without the use of any kind of automation tool. It can identify visible and hidden defects. This kind of software testing often involves the testers trying the software from the perspective of the end-users. It is among the most fundamental testing processes, and the strategies can range from being fully scripted test cases to high-level guides for exploratory testing sessions.

Automation Testing: Automation testing involves the use of special automation tools and requires a huge investment of money and resources. It is among the types of testing that aim to minimize manual test cases. Here, testers handle test scripts and return the result automatically. Test suites are recorded by using tools, and these can be played again by the testers as per the requirement. No human intervention is required for automation testing.

Manual Testing vs Automation Testing

Parameters	Manual Testing	Automation Testing
Definition	In manual testing, the test cases are executed by the human tester.	In automated testing, the test cases are executed by the software tools.
Processing Time	Manual testing is time-consuming.	Automation testing is faster than manual testing.
Resources requirement	Manual testing takes up human resources.	Automation testing takes up automation tools and trained employees.
Exploratory testing	Exploratory testing is possible in manual testing.	Exploratory testing is not possible in automation testing.
Framework requirement	Manual testing doesn’t use frameworks.	Automation testing uses frameworks like Data Drive, Keyword, etc.

White-box Testing: White-box testing is also known as structural testing, glass-box testing, transparent-box testing, or clear-box testing. This kind of testing is executed at the unit level and is based on the internal code structure. The test cases are designed based on the internal perspective of the system and programming skills such as code statements, conditions, branches, paths, etc.

Black-box Testing: Black-box testing is also known as behavioral testing, specification-based testing, or input-output testing. It is a testing method where testers, without looking at the internal code structure, test the functionality of the software. The tests are based on the requirements and functionality alone. The internal system design is not taken into consideration.

Grey-box testing: is the combination of both white-box and black-box testing. The testers will have access to design documents and internal coding for the purpose of designing test cases. The testing process here is executed at a functionality level. Grey-box testing is good at diagnosing context-specific errors of web systems and pays attention to all the layers of any complex system. This helps increase testing coverage. Grey-box testing is primarily implemented in penetration testing and integration testing.

Black Box Testing vs Gray Box Testing vs White Box Testing

S. No.	Black Box Testing	Gray Box Testing	White Box Testing
1.	This testing has Low granularity.	This testing has a medium level of granularity.	This testing has high-level granularity.
2.	It is done by end-users and also by the tester and developers.	It is done by end-users (called user acceptance testing) and also by testers and developers.	It is generally done by testers and developers.
3.	Here, Internals are not required to be known.	Here, Internals relevant to the testing are known.	Here, the Internal code of the application and database is known.
4.	It is likely to be less exhaustive than the other two.	It is kind of in-between.	Most exhaustive among all three.
5.	It is based on requirements, and test cases on the functional specifications, as the internals are not known.	It provides better variety/depth in test cases on account of high-level knowledge of the internals.	It can exercise code with a relevant variety of data.

Functional Testing: Functional Testing is a type of Software Testing in which the system is tested against the functional requirements and specifications. Functional testing ensures that the requirements or specifications are properly satisfied by the application.

Non-Functional Testing: Non-Functional Testing is a type of Software Testing that is performed to verify the non-functional requirements of the application. It verifies whether the behavior of the system is as per the requirement or not. It tests all the aspects that are not tested in functional testing.

Unit Testing is a method of testing individual units or components of a software application. It is typically done by developers and is used to ensure that the individual units of the software are working as intended.

Integration Testing is a method of testing how different units or components of a software application interact with each other. It is used to identify and resolve any issues that may arise when different units of the software are combined.

System Testing is a type of software testing that evaluates the overall functionality and performance of a complete and fully integrated software solution. It tests if the system meets the specified requirements and if it is suitable for delivery to the end-users.

Performance Testing is a type of software testing that ensures software applications perform properly under their expected workload. It is a testing technique carried out to determine system performance in terms of sensitivity, reactivity, and stability under a particular workload.

Usability Testing is a type of testing, that is done from an end user's perspective to determine if the system is easily usable. Usability testing is generally the practice of testing how easy a design is to use on a group of representative users.

Compatibility Testing is software testing that comes under the **non functional testing** category, and it is performed on an application to check its compatibility (running capability) on different platforms/environments. This testing is done only when the application becomes stable.

Other Testing

Static Testing: Static testing or verification is the testing method of checking files and documents to ensure the development of the right product according to the specified design, test cases, test plans, etc. Static testing involves activities such as inspection, static analysis, peer review, walk-throughs, etc.

It is crucial to execute static testing because the errors diagnosed during this phase make it cost-effective from the project's perspective.

Dynamic Testing: Dynamic testing or validation is the process of testing the actual product through the tests done on the software application. Various parameters, including CPU and memory usage, response time, and the overall performance of the software, are tested and reviewed. The dynamic behavior of the code is examined through this testing technique.
Inspection: a testing technique in which participants examine program code for predictable language-specific errors

Walkthrough: a peer group review of any product created during the systems development process, including code

Desk checking: a testing technique in which the program code is sequentially executed manually by the reviewer

Stub testing :A technique used in testing modules, especially where modules are written and tested in a top-down fashion, where a few lines of code are used to substitute for subordinate modules.

Installation

Installation, in the context of system implementation, refers to the process of replacing a current system with a new one. This process involves the conversion of existing data, software, documentation, and work procedures to the new system. There are different approaches to installation, including:

Direct installation: Also known as abrupt cut-over installation, this method involves switching from the old system to the new system by turning off the old system and turning on the new one.

Parallel installation: This approach involves running both the old and new systems simultaneously for a period of time to ensure the new system is working correctly before completely switching over.

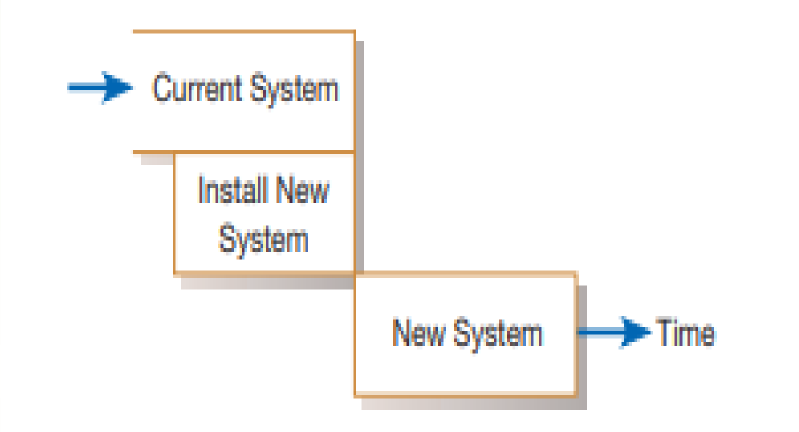


Fig: Direct installation

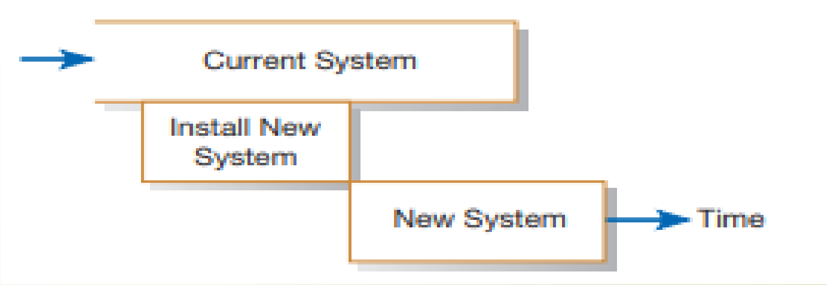


Fig:Parallel installation

Single-location installation: This method involves implementing the new system in a single location or department within an organization to test its effectiveness before rolling it out to the entire organization.

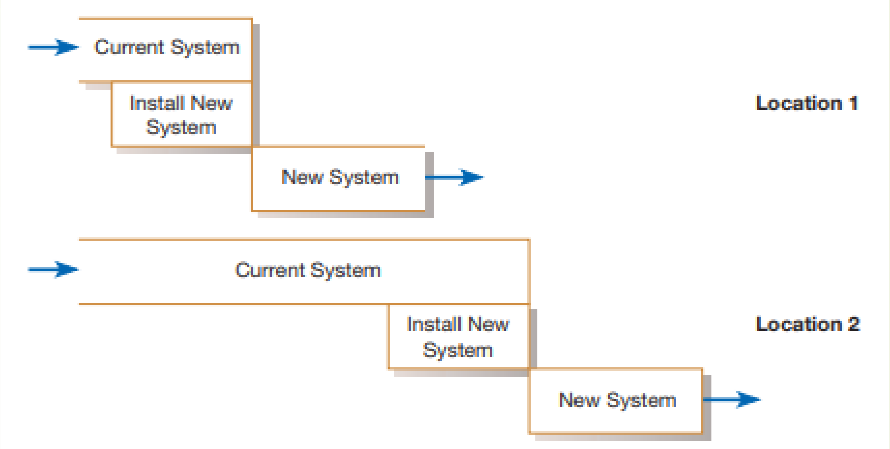


Fig:Single-location installation

Phased installation: is a method of changing from an old information system to a new one incrementally. It involves starting with one or a few functional components of the new system and then gradually extending the installation to cover the entire system.

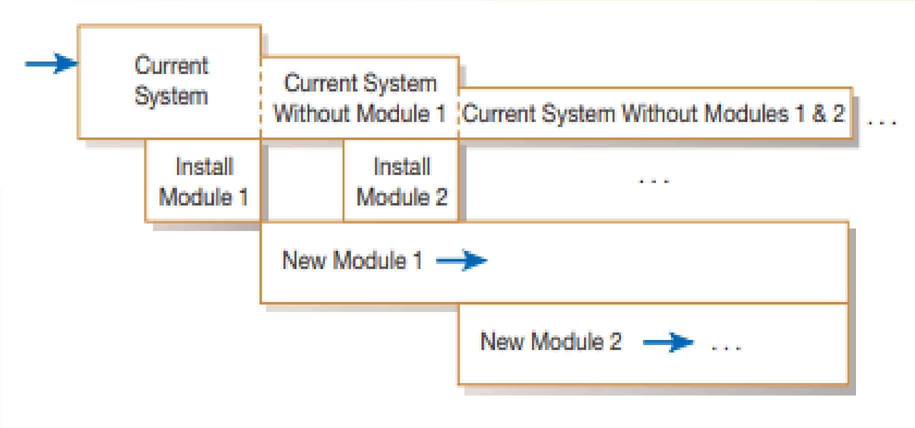


Fig: Phased installation

Documenting the system:

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 documentation is intended primarily for maintenance programmers. It contains following information:

- ✓ A description of the system specifying the scope of the problem, the environment in which it functions, its limitation, its input requirements, and form and types of output required.
- ✓ Detailed diagram of system flowchart and program flowchart.
- ✓ A source listing of all the full details of any modifications made since its development.
- ✓ Specification of all input and output media required for the operation of the system.
- ✓ Problem definition and the objective of developing the programs.
- ✓ Output and test report of the program.
- ✓ Upgrade or maintenance history, if modification of the program is made.

There are two types of system documentation. They are:

i) ***Internal documentation:*** Internal documentation is part of the program source code or is generated at compile time.

ii) ***External documentation:*** External documentation includes the outcome of structured diagramming technique such as dataflow an 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 and operational details of each system.
- ✓ Loading and unloading procedures.
- ✓ Problems which could arise, their meaning reply and operation action.
- ✓ Special checks and security measures.
- ✓ Quick reference guides about operating a system in a short, concise format.

Training And Supporting Users

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

- ✓ Resident expert (to fellow users for training).
- ✓ Traditional instructor-led classroom training.
- ✓ E-learning/ distance learning.
- ✓ Blended learning (combination of instructor-led and e-learning)
- ✓ External sources, such as vendors

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

- ✓ **Automating support:** Online support forums provides users access to information on new releases, bugs and tips form more effective usage. Forums are offered over the internet or over company intranets.
- ✓ **Providing support through a help desk:** A help desk is an information systems department function and is staffed by IS personnel. 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.

Organizational Issues In Systems Implementation

- Organizational issues in systems implementation can significantly hinder the success of a project, arising from factors like **poor planning, user resistance, and communication breakdowns**.
- These issues can manifest as misaligned expectations, inadequate training, data migration problems, and difficulty in gaining user adoption.
- Addressing these challenges requires careful planning, effective change management, and a focus on user engagement throughout the implementation process.

Common Organizational Issues:

Resistance to Change: Employees may resist adopting new systems due to perceived threats to their jobs or lack of training.

Data Migration and Quality: Migrating data from legacy systems can be complex, and data quality issues can negatively impact system performance.

User Adoption and Training: Ensuring employees effectively use the new system requires adequate training and change management strategies.

Communication Breakdown : Lack of clear communication and information sharing can lead to confusion and delays.

Data Integrity : Ensuring the accuracy and consistency of data during migration and throughout the system's lifecycle is crucial.

Internal Politics : Internal politics and competing priorities can hinder the successful implementation of a new system.

Addressing Organizational Issues:

Develop a comprehensive project plan: This should include clear objectives, timelines, and a budget.

Communicate effectively with all stakeholders: Keep everyone informed of the project's progress and address any concerns.

Provide adequate training and support: Ensure that users are prepared to use the new system effectively.

Implement a strong change management strategy: Address the human impact of the new system and help users to embrace the change.

Engage with users early and often: Involve users in the planning and implementation phases to ensure their buy-in.

Address any data quality issues: Ensure that the data being migrated is accurate and consistent.

Manage expectations: Set realistic expectations for the new system and its impact.

Maintaining Information Systems:

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

- ✓ ***Obtaining Maintenance requests:*** In this step a formal process is established where users can submit system change requests.
- ✓ ***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.
- ✓ ***Designing changes:*** A change request can be transformed into a formal design change.
- ✓ ***Implementing changes:*** In this activity proposed changes are implemented in respective components of system.

Types of system maintenance:

- ✓ ***Corrective maintenance:*** It refers to changes made to repair defects in the design, coding or implementation of the system.
- ✓ ***Adaptive maintenance:*** It involves making changes to an Information system to evolve this functionality to Changing business needs.
- ✓ ***Perfective maintenance:*** It involves making enhancements to d Improve processing performance or interface usability desired system features.
- ✓ ***Preventive maintenance:*** It Involves changes made to a system to reduce the chance of future system failure.

The cost of maintenance:

- Information systems maintenance cost are a significant expenditure. For some organizaions, as much as 60 to 80 percent of their information systems budget is allocated to maintenance activities. On average, 52 are percent of a company's programmers 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 separate 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 perssonel.

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

- ✓ Numbers of failure
- ✓ Time between each failure
- ✓ Type of failure

Factors influencing maintenance cost:

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.

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

System documentation quality: With quality documentation, maintenance efforts can increase exponentially. High-quality documentation leads reduction in the system maintenance.

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

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

Well-structured programs: Well-designed system are easier to understand and fix.