
UNIT 11 IMPLEMENTATION AND MAINTENANCE OF SYSTEMS

Structure

- 11.0 Introduction
 - 11.1 Objectives
 - 11.2 Implementation of Systems
 - 11.2.1 Conducting System Tests
 - 11.2.2 Preparing Conversion Plan
 - 11.2.3 Installing Database
 - 11.2.4 Training the End Users
 - 11.2.5 Preparing User Manual
 - 11.2.6 Converting to the New System
 - 11.3 Maintenance of Systems
 - 11.3.1 Different Maintenance Activities
 - 11.3.2 Issues Involved in Maintenance
 - 11.4 Summary
 - 11.5 Solutions/Answers
 - 11.6 Further Readings
-

11.0 INTRODUCTION

Implementation and Maintenance of System brings the SDLC life cycle to an end. After the design phase is over, actual writing of computer program as per the design specification and testing the system as a single entity called system testing which actually differs from the unit and module testing carried out during program development stage is carried out. A detailed conversion plan is the design to change over from the existing system to the new system. A properly designed conversion plan ensures a smooth transition to the new system. Database is designed, created and installed by using the existing data from the old system or by creating data manually. Appropriate training of the end user is important as the success of any system depends on the involvement of end user who is actually going to use the system. A comprehensive user manual is prepared outlining the procedural things to use the system. Actual conversion to the new system can be either in phased manner or in just one instance depending on the situation. Maintenance of the system after implementation is a major activity considering the total life of a software product. Maintenance is important to make the system current and relevant in a changing organizational environment.

11.1 OBJECTIVES

After going through this unit, you should be able to:

- conduct System tests;
 - prepare Conversion plans;
 - install Databases;
 - train the End Users;
 - prepare User Manual;
 - move to the New System; and
 - perform Various Maintenance Activities.
-

11.2 IMPLEMENTATION OF SYSTEMS

Implementation of system involves developing working computer software from the design specification through coding by a team of programmers. Many times, the user requirements are either not built-into the design specification or compromised to make the design simple and manageable. Implementation of the system is done by

coding, testing and creating the necessary hardware and network environment, and imparts training to the end users. Of course, apart from Coding and Testing, the running implementation activities differ from project to project. This phase of the software development requires intensive user involvement.

11.2.1 Conducting System Tests

No system can be perfect. Testing is of vital importance as all information systems are designed by a team of Software Engineers and end users have little or no knowledge of system development. Testing is done to bridge the gap between the perceived outcomes desired by the user to that of systems analysts and programming team. The design specifications are requirements of the user and are translated to working software by the programmer. Hence, it is the ability of the programmer to code exactly as per the design specification that is to be judged by testing the software module.

The objective of any testing mechanism is to discover and fix bugs before the product is delivered to the customer. A good testing scheme has a high probability of discovering an undiscovered error. The objective of any good testing scheme is to find and fix bugs with minimum time and resources. Besides, bugs and errors systems are tested for response time, volume of transactions that can be handled, stress under which it can function, security and usability. For an Online Transaction Processing System, testing of the system for response time could be quite vital.

System testing assumes that all parts of the system are correct and error-free. Even though the system has been tested for individual components and modules, there is no guarantee that the system after integration will work as per the desired specification. System test involves a holistic approach for testing the working of the application in totality.

The following are various types of System Testing:

Recovery Testing: Test the ability of the system to recover from errors. Errors or any other processing faults must not cause overall system to fail. The recovery time of the system after failure must be within a specific period and tolerance limit. System failures are forced during this phase of testing by introducing exceptions to see how the system responds to the case.

Security Testing: System used for processing sensitive information are prone to high security risks. Individual often tries to access unauthorized data for various reasons. Threats could be external or internal. Hacking of passwords is a common problem. Individual can use software to generate random passwords to gain access of the system. Security testing takes care of these aspects of the system security.

Stress Testing: Stress test is designed to test the system as to how the system behaves in abnormal situation. The aim of the stress test is to find the limit of quantity or frequency of input after which the system fails. Stress test cases are designed which require maximum memory and other resources; in excess of what a normal situation demands.

Performance Testing: Performance testing is specifically important to embedded and real time systems. It checks the run time performance of the system. It is often coupled with stress testing.

Response Testing: Testing of response time is of special importance in OLTP (on-line transaction processing systems like railway reservation system, points of sale, etc.). Testing is done to measure the response time. The same is compared with desired maximum response time.

Usability and Documentation Testing: Testing is done to review the usability and user friendliness of the software. Most often, systems are provided with on-line help

screen to help the end user. This also includes whether proper care had been taken to document the development stage of the project. User friendliness of the system is often compromised, which may lead to problem during implementation and maintenance of the system.

The following are the various activities involved during system testing:

Preparation of Test Plan: The first step in system testing is to prepare a document called a Test Plan. Test plan is a document which outlines the aspect of the system to be tested. A workable Test Plan is prepared in accordance with the design specification such as –

- Expected output from the system;
- Criteria for evaluating the output;
- Nature and Volume of test data; and
- Procedure for using the test data.

Specifications of User Acceptance Test: User is involved to prepare test cases. These can be derived from the test plan. Other parameters are test schedule, test duration and the person delegated to carry out the user acceptance test.

Preparation of Test data for Testing: Test data are often generated during testing of program. The test data must be true representative of the live data to be actually used by the end users after installation. Care should be taken to select the nature and volume of data.

Although enough care is taken to test the system as per the documented specification, it is almost always a confusion regarding how the user will use the end product. In case there is one customer (a specific application designed for a specific use), a series of acceptance tests are carried out to validate all the user requirements. But this is not possible if the software is to be used by many customers (general purpose software like word processor, etc.). An alternate approach is application of Alpha and Beta testing techniques.

Alpha Testing: Alpha testing is carried out by the customer at the developers' site. The customer uses the software and records the errors/bugs and usage problem. Alpha testing is carried out in a controlled environment.

Beta Testing: Beta testing is carried out at one or more customer sites by the end users. It is live testing of the software product and not controlled by the developer. The customer tests the software using her/his own data records and reports the bugs or problems in regular intervals to the developer.

Many organizations deploy specially trained personnel for system testing. The problems of bugs uncovered in alpha and beta testing are fixed before the product is shipped and installed in customer's premises. Testing of complex software can be time consuming and frustrating also. The aim of system testing is to uncover every possible error that may come up at the user end. The role of Data Processing Auditor (EDP auditors)/Information System Auditor is quite involved during all stages of system development especially during testing. Auditors can provide useful independent inputs to minimize complications during maintenance.

Check Your Progress 1

1. The first step in system testing is to prepare a document called
2. is designed to test the system as to how the system behaves in abnormal situations.
3. In live data is used in the customer's real working environment.

11.2.2 Preparing Conversion Plan

A conversion plan is a document which spells out detailed requirements for a successful conversion from existing system to proposed system. The complexity of conversion is directly proportional to the complexity of the system in question. An important role of Systems Analyst is to see that the newly designed system is implemented to the set specification. Conversion is just one aspect of implementation, other being software maintenance and system review.

A proper conversion plan ensures that conversion from old system to new system is smooth without affecting the normal business operation. The conversion process can be tedious and disrupt normal functioning of system and also involves financial and human resources. A well designed conversion plan facilitates a smooth switch over to the new system while keeping the cost and human involvement to the minimum.

A typical conversion plan is a document that consists of the following information:

- Guidelines regarding Conversion processes involved and the roles of end user.
- Planning conversion of files, creation of computer compatible files.
- Types of conversion to be undertaken depending on the existing types of system. It could be from an existing manual system to a newly designed system or from an existing old computerized system to a newly designed enhanced system.
- Types of conversion may be parallel, phased or direct.
- Evaluation of hardware, software and related services.

11.2.3 Installing Database

Installation of database is nothing but creating computer readable files from the existing systems/documents. Each installation involves data. The new system is going to use data created either manually or data that has been obtained from the old system. If the current system is using computer readable data, it must be made error free and compatible for use in the new system. The data must be converted to the new format supported by the current technology on which the system is being developed.

Usually, there will be upward compatibility between various versions of software. The data conversion process can be tedious depending on the format supported by the new system. Special software are designed to facilitate the installation of Database.

11.2.4 Training the End User

Training the user is one of the vital activities. The project team must make sure that the end users are trained to operate the new system. Many systems fail to get implemented or deliver the desired result because the end users are not trained.

Managers and the users must be trained on fundamentals of information technology in addition to knowing the operation of the new system. Training and support form the two crucial issues involving success of any information system. While training is imparted in a fixed schedule, support is an ongoing process. In support activity, the user is provided continual operational and technical support to carry out the work. Support materials are developed to facilitate this task. The goal of any training and support activity is to achieve highest possible productivity with lowest cost. Training may involve the following activities:

- Entering the data into the system. Generating the required reports.
- Basic training of computers not specific to the application program like copying a file, starting and shutting down system, etc.
- Briefing about Hardware and Software concepts.
- Reporting non compliance and bugs in the program? Process of taking backup of daily work.

There is no exhaustive list of training requirement of the end user and can vary depending on the nature of application. The training must be scheduled in logical sequence depending on the pre-requisite for the next module of the training. A dependency chart could be useful for this purpose.

Training can be imparted in different ways:

- Computer-aided training
- Classroom tutorial
- Interactive training manual
- Resident technical expert
- One to one training
- External sources
- Information center / help desk

Many organizations have a well-developed automated support mechanism for end user support. To make a system success, following issues should be taken care of:

- Develop a satisfactory support base for providing support to the user;
- Obtain user participation and commitments;
- Institutionalize the system of training; and
- Insist on mandatory use of the information system.

Regular training should become part of the organization's policy as information system changes as per the requirement of the organization and new features are added.

11.2.5 Preparing User Manual

All information systems are unique and different from one another. Documentation starts from the day one of system development lifecycle, but preparation of end user documentation is of specific importance as the end user does not understand the intricacies of system development and hence operational problems are bound to occur. Documentation of any information system is generally of two types. System Documentation and User Documentation. System documentation contains detailed information about systems design specification, its internal structure and related technical details. The system documents are primarily for the programmer for maintenance purpose. The user documentation on the other hand is for the end user. The document should be structured and self-contained.

A user manual generally contains written as well as pictorial representation of the information system about its working and application. A well-designed user manual can reduce the overall cost of training and support. On-line help system with hyperlinks and context sensitive help systems are slowly replacing bulky and non-interactive documents.

The following are the components of a User Manual:

- Title and Version of software release
- Table of contents
- Salient features of the product
- Installation Guide and System requirements
- Getting started
- Frequently asked questions
- Sample scenario
- Glossary of terms used in the manual
- Known bugs in the applications

With changing technology, user documentation is often bundled to the information system. It is separate document. On-line documentations are being extensively

utilized in user environments due to their convenience. Context sensitive helps are making the users' life easy by reducing the time to browse the bulky documents.

11.2.6 Converting to the New System

Actual conversion process involves equipments, personnel, data and financial resources. The process of conversion from the existing system (manual or computerized) to the newly developed system can be performed in several ways depending on the criticality of the system and other related issues.

- **Direct Conversion:** This is abrupt approach. The old system is shutdown and the new system starts. This kind of conversion although economical, the users are at the mercy of the new system, hence direct installation can be very risky. Some times due to procedural reasons where two systems can't be run parallel, this kind of conversion is the only option. When the new system fails, there is no way to start the old system as a backup as it has been shutdown. This kind of conversion plan is often the least preferred for critical business applications.
- **Pilot Conversion:** This is the middle path approach. Instead of converting all at once throughout the organization, this kind of pilot installation involves conversion/installation of system at a single pre-decided location. The location may be a branch office of the organization. Proper selection of the pilot site is important as it should be able to perform a true conversion process to test all functionalities of the new system. The advantage of the pilot conversion is that the potential risk in case of failure of the system is limited to a single location. Once the user is ascertained that the implementation of the system has been successful in a particular location, it is proposed to replicate the system in other locations. Although this kind of pilot conversion plan is beneficial for the user, it places a substantial burden on the implementation team as it has to maintain two systems in parallel.
- **Parallel Conversion:** is least risk prone. Under this kind of conversion, the old system is allowed to run alongside the new system until the management and the end user are satisfied with the result of the new system. It is compared with the new system to test whether the functionalities covered by the old system are thoroughly covered in the new system by comparing the outputs. Errors and bugs identified with the new system are not detrimental for normal functioning of the organization as the new system is replaced and normal functions are resumed by the old system. Parallel conversion is costly as two systems are run in parallel, but results of only one system are used for business operations.
- **Phased Conversion:** is an incremental approach to switch over to the new system. Different sub-systems of the new system is used in conjunction until the whole new system is converted. This kind of approach for conversion limits the potential risk of failure of the new system. In a phased installation as a sub-system is made functional, actual results are visible before the whole new system is made functional.

Each conversion strategy not only involves data and software, but also other resources like personnel, hardware, etc. Hardware and software selection is an important issue to be considered before actually carrying the conversion.

Check Your Progress 2

1. is an incremental approach to switch over to the new system.
2. is nothing but creating computer readable files from the existing systems/documents.
3. A is a document, which spells out detailed requirements for a successful conversion.

11.3 MAINTENANCE OF SYSTEMS

Once the information system is successfully installed and started showing result, the next issue is to maintain the system. System maintenance involves more than 80% of the total life of a software product; this shows the importance of maintenance. System maintenance is the task of monitoring, evaluating and modifying the information system to make necessary desirable changes during the total life cycle of the software. Organizational requirements as perceived during the analysis phase changes, the system has to accommodate all such changes to make the system current and useful for the organization. Maintenance of system also takes care of the failure and shortcomings that arise during the operation of the information system by the end user. During the implementation phase, one person from the system maintenance group is nominated to collect information from the user for maintenances. Maintenance activity involves collecting requests for changes, transforming these requests to changes, designing the changes to be incorporated and implementing the changes in the system.

Any maintenance activity comprises the following four key stages:

- **Help Desk:** The problem is received from the user through a formal change request. A preliminary analysis of the change request will be done, and if the problem is sensible, it is accepted.
- **Analysis:** Managerial and technical analyses of the problems are undertaken to investigate the cost factors and other alternative solutions. Feasibility Analysis is done to assess the impact of the modification, to investigate alternative solutions, to assess short and long term costs, and to compute the benefit of making the change.
- **Implementation:** The chosen change/solution is implemented and tested by the maintenance team. All infected components are to be identified and brought in to the scope of the change. Unit test, integration test, user-oriented functional acceptance tests and regression test strategies are provided.
- **Release:** The changes are released to the customer, with a release note and appropriate documentation giving details of the changes.

11.3.1 Different Maintenance Activities

Once the system is fully implemented and starts operating, the maintenance phase begins. When the user starts operating the system, initial difficulty diminishes as the user learns to operate the system. The maintenance may include modification of system due to changes in business environment, government regulations, new business ventures and enhancement of functionalities.

The majority of Software Maintenance activity is concerned with evolution derived from user requested changes. A program that is used in a real world environment necessarily must change or become progressively less useful in that environment. As an evolving program changes, its structure tends to become more complex. Extra resources must be devoted to preserving the semantics and simplifying the structure.

The following are the different types of maintenance activities:

- **Corrective Maintenance:** This type of maintenance is to rectify design, coding and implementation problems detected after the implementation of the System. This kind of problem generally surfaces immediately after the system is implemented. This type of problem needs immediate attention as it hampers the day to day work of the end user. Proper planning and interaction with the end user during system development process can minimize Corrective Maintenance. In spite of the all these kinds of maintenance, these constitute more than 60 percent of total maintenance effort. Corrective maintenance is very much

undesirable. It does not do any value addition to the software. Care should be taken to see that normal business operations are not disturbed because of it.

- **Adaptive Maintenance:** Changes are needed as a consequence of upgraded versions or changes in operation system, hardware, or DBMS. Adaptive maintenance is required because business operates on a social environment and need of the organization changes as organization ventures in to new areas, or as government regulation policy changes, etc. Maintenance of the software to adapt to this kind of changes is called adaptive maintenance. Unlike corrective maintenance, this kind of activity adds value to the information system and affects a small part of the organization. This activity is not as urgent as corrective maintenance as these changes are gradual and allow sufficient time to the system group to make changes to the software.
- **Perfective Maintenance:** This kind of maintenance activity involves adding new functionalities and features to the software to make it more versatile and user oriented. Some times, changes are made to improve performance of the software. In some sense, this maintenance can be thought of as a new development activity. This adds value to the information system and is required to stay ahead of the competition.

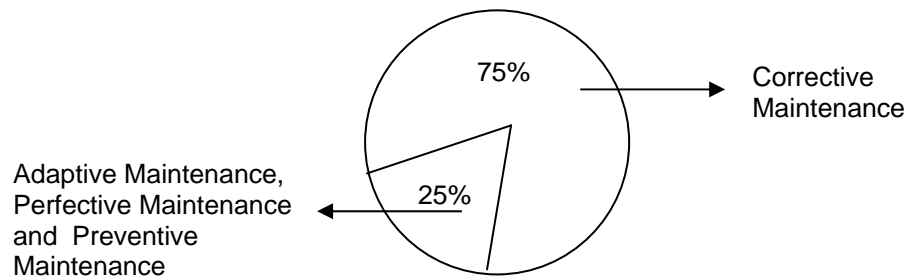


Figure 11.1: Comparative figures of maintenance effort.

- **Preventive Maintenance:** Changes are made to software to make it easily maintainable and to prevent any kind of system failure in future. This reduces the need of corrective maintenance. As corrective maintenance could lead to hamper normal functioning, preventive maintenance is done periodically to ensure that the probability of system failure is minimized. Preventive maintenance could increase the volume of transactions that can be handled by the system. Preventive maintenance is done when the system is least used or not used at all. This does not add value to the system, but certainly lowers the cost of corrective maintenance.

Figure 11.1 depicts the maintenance efforts that are to be put during each maintenance activity.

11.3.2 Issues Involved in Maintenance

The responsibility of the software development team and clients does not end once the product is released for implementation and installed. If software is not properly maintained, a well-documented and cleanly designed system can decay into a poorly documented and ill-maintained system. Additional vulnerability may get introduced during the activity of maintenance. In a network environment, a bug has ramification beyond just poor performance or functionalities. A bug can open up avenue for a hostile intruder.

It is very important that the Software should be easily maintainable. Factors like availability of source code, availability of system manuals, etc., are very important for maintainability. One of the most important issues is the cost factor for maintenance of software. There are a number of factors that influence the cost of maintenance. Maintenance activity may some times introduce new bugs while rectifying it.

The following are various factors which affect the ease of maintenance:

- **Volume of Defects:** The inherent errors / bugs that are found in the system after installation. Cost of maintenance increases with the increase in volume of defects.
- **Number of Customers:** More number of customers means more requests for changes in the system after installation.
- **Availability of System Documentation:** The quality and availability of system documentation is vital to carry out the maintenance. Poorly written system documentation increases the cost of maintenance. Most often, the programmers for development are different than the team of programmers for maintenance and the later often finds it difficult to understand a program written by the former. Structured programming and program documentations are very useful in maintaining the system

The following are various issues in Software Maintenance:

1. Organizational Issues

- The maintenance activity must align with organizational objectives.
- Most of the Software Maintenance activity is resource consuming and it has no clear quantifiable benefit for the organization.
- Outsourcing the job of Software Maintenance

2. Process Issues

- Software Maintenance requires a number of additional activities not performed at development stage. Impact analysis and Regression tests on the software changes are crucial issues

3. Technical Issues

How to construct software that it is easy to comprehend is a major issue and the technology to do this is still not available. Still, the following are some guidelines for the same:

- Translate the problem into software terms to decide if it is viable or not.
- Determine the origin of the change request and suggest solutions.
- All solutions are investigated to determine that they are applied to all software components affected.
- Make a decision on the best implementation route or to make no change.
- Ripple effect propagation is a phenomenon by which changes made to a software component along the software life cycle have a tendency to be felt in other components

Legacy System

A legacy system is typically a very old and large system which has been modified heavily since it started operation. Legacy systems are based on old technology with very little or no documentation. Dealing with a legacy system can be very hard.

Solutions for the problems mentioned above relating to a Legacy System:

- Explore possibility of subcontracting the maintenance
- Replace software with a package
- Re-implement from scratch
- Discard software and discontinue
- Freeze maintenance and phase in new system
- Reverse engineer the legacy system and develop a new software suite.

Check Your Progress 3

1. At, the problem is received from the user through a formal change request, a preliminary analysis is done, and if the request is sensible, it is accepted.
2. is performed to rectify problems in design, coding etc. detected after the implementation of the System.
3. A, is typically a very old and large system which has been modified heavily since it started operation.

11.4 SUMMARY

Implementation of system involves coding, testing installation and user training. System design specifications are converted to computer programs and database structures are created. The programs are tested using a code walk through and by creating different test scenarios. System testing is testing of the software in its totality after individual modules had been tested. Different conversion plans are discussed like software and hardware installation etc. Installation of a system is usually moving from old system to a new system. Different methodology is adopted for conversion/ installation like direct conversion, parallel conversion, phased conversion, single location conversion. User documentation is a written document of visual and textual information about the application and how to use it. Well-designed user documentation can reduce training cost of the organization. Training of user is vital for success of any system. Training should be conducted after any significant changes are made to the system

Software maintenance is the activity of modifying the software once it is delivered to the customer depending of the requirement of the customer or to add additional functionalities to the software. Software maintenance activity in general does not provide any quantifiable benefit to the organization. Different maintenance activities are adaptive maintenance, corrective maintenance, perfective maintenance and preventive maintenance. Out of all maintenance activity, corrective maintenance constitutes more than 60 percent of the total maintenance activity. Different issues related to software maintenance are technical, organizational and procedural.

11.5 SOLUTIONS/ANSWERS

Check Your Progress 1

1. Test Plan
2. Stress Test
3. Beta Testing

Check Your Progress 2

1. Phased Conversion
2. Installation Database
3. Conversion Plan

Check Your Progress 3

1. Help Desk
2. Corrective Maintenance
3. Legacy System

11.6 FURTHER READINGS

Joey George, J. Hoffer and Joseph Valacich; *Modern Systems Analysis and Design*, Third Edition, 2001, Pearson Education.

Alan Dennis, Barbara Haley Wixom; *Systems Analysis and Design*, 2002, John Wiley & Sons.

Reference Websites

<http://www.rspa.com>

<http://www.dur.ac.uk/csm/jsm>