Objectives of the Chapter

- System Implementation.

- System Testing.

- Implementation Process.

- Implementation methods.

System Implementation

- The Implementation Phase has one key activity: developing the new system in its target environment.

- Supporting actions includes training end-users and preparing to turn the system over to maintenance personnel.

- After this phase, the system enters the Operations and Maintenance Phase.

System Implementation

- Successful completion of Implementation Phase should comprise:
1. System deployment.
2. Training on the System.

- The purpose of the Implementation Phase is to deploy and enable operations of the new information system in the production environment.

- The final phase in the SDLC is the Implementation Phase.

System Implementation

- This is the phase that usually gets the most attention, because for most systems it is the longest and most expensive single part of the development process.

- This Phase has three steps:

1. System Construction is the first step. The <u>system is built and tested</u> to ensure that it performs as designed. Since the cost of fixing bugs can be immense, testing is one of the most critical step in implementation.

System Implementation

2. The system is installed. Installation is the process by which the old system is turned off and new one is turned on.

- There are several approaches that may be used to convert from the old to new system.

- Most important aspects of conversion is the training plan, used to teach users how to use the new system.

System Implementation

3. The analyst team establishes a support plan for the system. This plan includes – formal & informal post-implementation reviews as well as systematic way for identifying major and minor changes needed for the system.
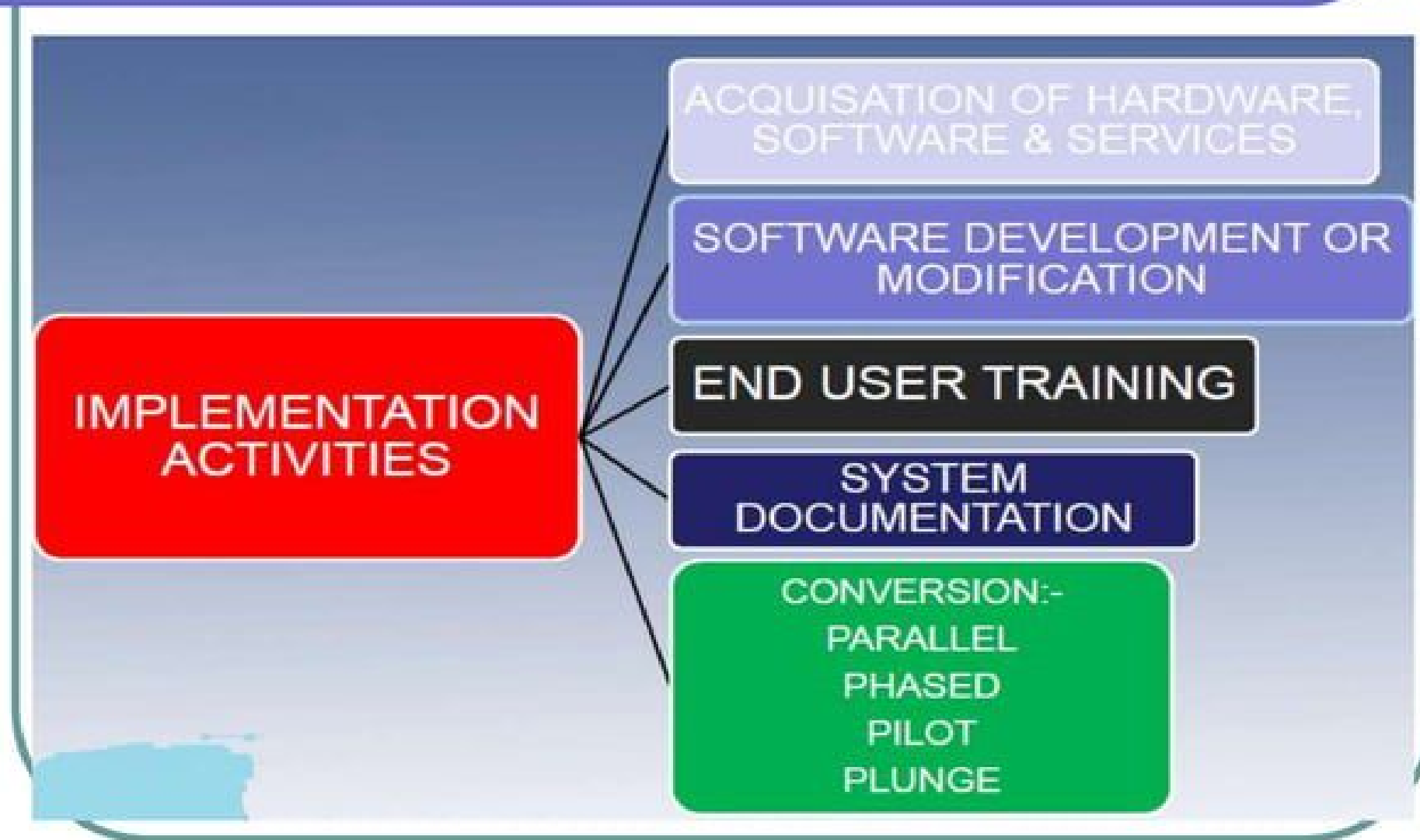
Cont..

- <u>Acquisition of hardware software and services</u>: This activity incorporates to implement the project, required hardware and software are acquired along with its services.

- <u>Software development or modifications</u>: This activity incorporates development or modification of software as per actual functional requirements during the process.

Cont..

- <u>End User Training</u>: During this activity user of the systems are trained for the actual functioning on the system. Necessary inputs by considering their role and key aspects of functioning is made aware to the end user.

Cont..

- <u>System Documentation</u>: During this phase system development authority reviews all system documentation to confirm that it is complete and correct. These documents includes Operations Manuals or system Administration Manuals, Maintenance Phase.

System Implementation Methods

- <u>Direct/plunge/crash approach</u> – entire new system completely replaces entire old system, in one step.

- <u>Parallel approach</u> – both systems are operated side by side until the new system proves itself.

- <u>Pilot/location approach</u> – launched new system for only one group within the business – once new system is operating smoothly, implementation goes company-wide.

System Implementation Types of Conversion

- <u>Phased/staged/incremental approach</u> – individual parts of new system are gradually phased-in over time, using either crash or parallel for each piece.

Steps in Implementation

1. Planning the implementation activities.
2. Acquiring and laying out facilities and offices.
3. Organizing the personnel for implementation.
4. Developing procedures for installation and testing.
5. Developing the training program for operating personnel.
6. Completing the system's software.

Steps in Implementation

7. Acquiring the required hardware.
8. Generating files.
9. Designing forms.
10. Testing the entire system.
11. Complete cutover to the new system.
12. Documenting the new system.
13. Evaluating   the   MIS.(debugging   &
    Improvement).

Risk Management

- The project Manager conduct risk management activities during the Implementation Phase:

1. Identification of Risk- Determination of risks and its effect.

Risk Management

2. Risk Analysis – Conduct quantitative and/or qualitative analysis of each identified risk.

3. Response Planning – as per identification response plans are prepared.

4. Monitoring and Control – response plan execution along with its monitoring and control.

System Testing

- Writing programs is a fun, creative activity.

- Testing and documentation aren't fun; consequently, they receive less attention than writing programs.

- Most professional organizations devote more time and money to testing than to writing the programs in the first place.

- The attention paid to testing is justified by the high costs associated with downtime and failures caused by software bugs.

System Testing

- A program is not considered finished until it has passed its testing, because of this Programming and testing are tightly coupled.

- Testing must be performed and documented systematically so that the project team always knows what has and has not been tested.

System Testing

- Test Plan: Testing starts with the tester's developing a test plan that defines a series of tests that will be constructed.

- A test plan often has 20 to 30 pages, with a separate page for each individual test in the plan.

- Each individual test has a specific objective, describes a set of very specific test cases to examine.

System Testing

- It is impossible to test every possible combination of input and situation.

- A really good test would include a test case with nonsensical, but potentially valid, data.

- There are four general stages of tests:

  1. Unit Tests
  2. Integration Tests
  3. System Tests
  4. Acceptance Tests

System Testing

1. <u>Unit Tests</u>: This test focus on one unit – a program or a program module that performs a specific function that can be tested.

- Ensures programs functionality as per specifications.

- It is performed after the programmer has developed and tested the code and believes it to be error free.

- There are three approaches :
  a. Black-box testing
  b. White-box testing
  c. Gray-box Testing

System Testing

a. Black-box testing is commonly used, the test plans is developed directly from the program specification.

- Each item in the program specification becomes a test, and several test cases are developed for it.

- The technique of testing without having any knowledge of the interior workings of the applications.

System Testing

b. White-box testing is reserved for special circumstances in which the tester wants to review the actual program code, usually when complexity is high.

- Is detailed investigation of internal logic and structure of the code.

- Is also called as glass testing or open box testing.

- Tester needs to possess knowledge of the internal working of the code.

System Testing

c. Gray-box Testing is used to test the application with limited knowledge of the internal working of an application.

- In gray box testing, the tester has access to design documents and the database.

System Testing

Comparison between the Three Testing Types

	Black Box Testing	Grey Box Testing	White Box Testing
1.	The Internal Workings of an application are not required to be known	Somewhat knowledge of the internal workings are known	Tester has full knowledge of the Internal workings of the application
2.	Also known as closed box testing, data driven testing and functional testing	Another term for grey box testing is translucent testing as the tester has limited knowledge of the insides of the application	Also known as clear box testing, structural testing or code based testing
3.	Performed by end users and also by testers and developers	Performed by end users and also by testers and developers	Normally done by testers and developers
4.	-Testing is based on external expectations -Internal behavior of the application is unknown	Testing is done on the basis of high level database diagrams and data flow diagrams	Internal workings are fully known and the tester can design test data accordingly
5.	This is the least time consuming and exhaustive	Partly time consuming and exhaustive	The most exhaustive and time consuming type of testing
6.	Not suited to algorithm testing	Not suited to algorithm testing	Suited for algorithm testing
7.	This can only be done by trial and error method	Data domains and Internal boundaries can be tested, if known	Data domains and Internal boundaries can be better tested

System Testing

2. **Integration Test:** assess whether a set of modules or programs that must work together do so without error.

- It ensures that the interfaces and linkages between different parts of the system work properly.

- Focus on the flow of control among modules and on the data exchanged among them.

System Testing

- Integration testing is often done by a set of programmers and/or system analysts.

- There are four approaches to integration testing: User interface testing, use scenario testing, data flow testing and system interface testing.

System Testing

3. System Tests are usually conducted by the system analysts to ensure that all modules and programs work together without error.

- Is similar to integration testing, but is much broader in scope.

System Testing

- It tests how well the system meets business requirements and its usability, security, and performance under heavy load.

- It also tests the system's documentation.

System Testing

4. Acceptance Tests are done primarily by the users with support from the project team.

- The goal is to confirm that the system is complete, meets the business needs that prompted the system to be developed, and is acceptable to the users.

- It is done in two stages:

System Testing

a. Alpha Testing: in which users test the system using made-up data.

b. Beta Testing: in which users begin to use the system with real data and carefully monitor the system for errors.

- The user's perceptions of the new system will be significantly influenced by their experiences during acceptance testing.

System Documentation

- Two types of documentations:

1. System Documentation is intended to help programmer and system analysts understand the application software and enable them to built it or maintain it after the installation.

System Documentation

2. User Documentation is designed to help the user operate the system. Ex. User Manuals, Training manuals and online help system.

Review

- System Implementation.

- System Testing.

- Implementation Process.

- Implementation methods.