# Chapter No. 7

**Implementation**

**Implementation**

Implementation is the carrying out, execution, or practice of a plan, a method, or any design for doing something. As such, implementation is the action that must follow any preliminary thinking in order for something to actually happen. In an information technology context, implementation encompasses all the processes involved in getting new software or hardware operating properly in its environment, including installation, [configuration](http://searchexchange.techtarget.com/sDefinition/0,,sid43_gci833457,00.html), running, testing, and making necessary changes. The word deployment is sometimes used to mean the same thing.

# 7.1 System testing

[Softwaretesting](http://en.wikipedia.org/wiki/Software_testing) is an integral and important part of the software development process. This part of the process ensures that [defects](http://en.wikipedia.org/wiki/Software_bug) are recognized as early as possible.

# Why System Testing is important?

System Testing is a crucial step in Quality Management Process.

In the Software Development Life cycle System Testing is the first level where   
the System is tested to verify if it meets the functional and technical   
requirementsThe application/System is tested in an environment that closely resembles the   
production environment where the application will be finally deployed   
The System Testing enables us to test, verify and validate both the Business   
requirements as well as the Application Architecture

### Prerequisites for System Testing:

The prerequisites for System Testing are:

* All the components should have been successfully Unit Tested
* All the components should have been successfully integrated and Integration
* Testing should be completed
* An Environment closely resembling the production environment should be   
  created.

When necessary, several iterations of System Testing are done in multiple environments.

# Steps needed to do System Testing:

The following steps are important to perform System Testing:

1: Create a System Test Plan

Step 1: Create Test Cases   
Step 2: Carefully Build Data used as Input for System Testing   
Step 3: If applicable create scripts to build environment and to automate Execution of test cases

Step 4: Execute the test cases   
Step 5: Fix the bugs if any and re test the code   
Step 6: Repeat the test cycle as necessary

# System Test Plan

As you may have read in the other articles in the testing series, this document typically describes the following:

* The Testing Goals
* The key areas to be focused on while testing
* The Testing Deliverables
* How the tests will be carried out
* The list of things to be Tested
* Roles and Responsibilities
* Prerequisites to begin Testing
* Test Environment
* Assumptions
* What to do after a test is successfully carried out
* What to do if test fails
* Glossary

# Way of writing a System Test Case

A **Test Case** describes exactly how the test should be carried out. The System test cases help us verify and validate the system.   
The System Test Cases are written such that:

* They cover all the use cases and scenarios
* The Test cases validate the technical Requirements and Specifications
* The Test cases verify if the application/System meet the Business & Functional   
  Requirements specified
* The Test cases may also verify if the System meets the performance standards

# 7.2 Method of implementation

[Implementation](http://en.wikipedia.org/wiki/Implementation) is the part of the process where [software engineers](http://en.wikipedia.org/wiki/Software_engineering) actually [program](http://en.wikipedia.org/wiki/Computer_programming) the code for the project.

* Factors affecting how much your project will cost and how much effort it will take
* Techniques for project management and control
* Planning project tasks and scope definitions
* Analysis techniques for your business and technical systems
* Designing ERP solutions for your business
* Techniques to enable the system

Managing change, transitioning from old to new systems, and supporting users after

The phase in those implementation processes can be divided.

# Solution Design phase

used to create designs for solutions that meets future business requirements and processes. The design of your future organization comes alive during this phase as customizations and module configurations are finalized.

#### Build phase

Coding and testing of customizations, Enhancements, interfaces, and data conversions happen. In addition, one or more conference room pilots test the integrated enterprise system. The results of the build phase should be a working, tested business system solution.

#### Transition phase

During this phase, the project team delivers the finished solution to the enterprise. End-user training and support, management of change, and data conversions are major activities of this phase.

#### Production phase

Starts when the system goes live. Technical people work to stabilize and maintain the system under full transaction loads. Users and the implementation team begin a series of refinements to minimize unfavorable impacts.

# 7.3 Deployment and maintenance

[Deployment](http://en.wikipedia.org/wiki/Software_deployment) starts after the code is appropriately tested, is approved for [release](http://en.wikipedia.org/wiki/Software_release) and sold or otherwise distributed into a production environment.

[Software Training and Support](http://en.wikipedia.org/wiki/Technical_support) is important and a lot of developers fail to realize that. It would not matter how much time and planning a development team puts into creating software if nobody in an organization ends up using it. People are often resistant to change and avoid venturing into an unfamiliar area, so as a part of the deployment phase, it is very important to have training classes for new clients of your software.

[Maintaining](http://en.wikipedia.org/wiki/Software_maintenance) and enhancing software to cope with newly discovered [problems](http://en.wikipedia.org/wiki/Software_bug) or new requirements can take far more time than the initial development of the software. It maybe necessary to add code that does not fit the original design to correct an unforeseen problem or it may be that a customer is requesting more functionality and code can be added to accommodate their requests. If the labor cost of the maintenance phase exceeds 25% of the prior-phases' labor cost, then it is likely that the overall quality of at least one prior phase is poor.[[citation needed](http://en.wikipedia.org/wiki/Wikipedia:Citation_needed)] In that case, management should consider the option of rebuilding the system (or portions) before maintenance cost is out of control.

[BugTrackingSystem](http://en.wikipedia.org/wiki/Bug_Tracking_System) tools are often deployed at this stage of the process to allow development teams to interface with customer/field teams testing the software to identify any real or perceived issues. These software tools, both open source and commercially licensed, provide a customizable process to acquire, review, acknowledge, and respond to reported issues. (Software maintenance)