**SYSTEM IMPLEMENTATION**

Implementation is used here to mean the process of converting a new or revised system design into operational one; conversion is one aspect of implementation. the other aspect is post implementation review and software and maintenance

There are three type of implementation:

* Implementation of a computer system
* Implementation of new computer system
* Implementation of a modified application

**Implementation of the computer system**

It’s should be replace a manual system the problems encountered are converting files, training users creating accurate files, and verifying printouts for integrity

**Implementation of new computer system**

It’s should be replace an existing one this is usually a difficult conversion. if not properly planned there can be many problems. Some large computer system have taken even years to convert

**Implementation of a modified application**

It’s should be replace an existing one using the same computer. This type of conversion is relativity easy to handle, provided there are no major changes to the file

**6.2 SYSTEM TESTING**

Software testing is an important element of software quality assurance and represents the ultimate review of specification, design and coding. It increasing visibility of software as a system element and the costs associates with a software failure are motivating forces for all well planned through testing .The system is tested with giving wrong information. Cascade deletion and, the software developer checks updating. Testing and debugging are different activities, but debugging must be accommodated in any testing strategy.

**6.3 Types of testing**

**6.3.1 Unit testing**

The first step in testing is Unit testing. Individual testing are tested to ensure that they operate correctly. Each component is tested independently, without other system components. The module interface is tested to ensure that information properly flow into and out of the program.These are tested that the module operates at boundary established to limit or restrict processing. Unit testing is normally considered as an adjunct to the coding step. After the coding has been developed, received and verified for correct syntax, unit testing begins. Here each module is tested to provide its correctness, validity and determine any missing operations and to verify whether the objectives have been met, errors are noted down and corrected immediately.

Unit testing is the important and major part of the project. So errors can be rectified easily in each module and program clarity can be increased. In this project, the entire system is divided into several modules and is developed individually. Hence, unit testing is conducted to individual modules.

**6.2.2 Integration testing**

The second step in the testing process is the Integration testing. Integration testing is the systematic technique for constructing the program structure while conducting tests to uncover errors associated with integrating. After the unit test, each module is gradually integrated to form one final system.

All the modules when unit tested will work properly but after integrating the data can cause error one module can have an inadvertent, adverse effect on another; sub functions when combined may not produce the desired major function; global data structures can cause problems, etc.

Hence, the objective of integration testing is to take unit tested modules and build a final program structure. In this project, modules are combined to find the overall performance of the system.

**6.2.3 Performance testing**

A type of [Physical test](http://en.wikipedia.org/wiki/Physical_test) covering a wide range of engineering or functional evaluations where a material, product, or system is not specified by detailed material or component [specifications](http://en.wikipedia.org/wiki/Specification): rather, emphasis is on the final measurable performance characteristics. Testing can be a [qualitative](http://en.wikipedia.org/wiki/Qualitative_data) or [quantitative](http://en.wikipedia.org/wiki/Quantitative_data) procedure.

# 6.2.4 Acceptance testing

The **types of acceptance testing** are:

* The **User Acceptance test:** focuses mainly on the functionality thereby validating the fitness-for-use of the system by the business user. The user acceptance test is performed by the users and application managers.
* The **Operational Acceptance test:** also known as Production acceptance test validates whether the system meets the requirements for operation. In most of the organization the operational acceptance test is performed by the system administration before the system is released. The operational acceptance test may include testing of backup/restore, disaster recovery, maintenance tasks and periodic check of security vulnerabilities
* **Contract Acceptance testing**: It is performed against the contract’s acceptance criteria for producing custom developed software. Acceptance should be formally defined when the contract is agreed.
* **Compliance acceptance testing:** It is also known as regulation acceptance testing is performed against the regulations which must be adhered to, such as governmental, legal or safety regulations