**TESTING**

**Testing Techniques**

***Black-box*** tests are used to demonstrate that software functions are operational, that input is properly accepted and output is correctly produced, and that the integrity of the external information is maintained.

***White-box*** testing of software is predicted on close examination of procedural detail. Providing test cases that exercise specific sets of conditions and/or loops tests logical paths through the software.

**Testing Levels**

There are several levels in testing phases. These are unit testing, integration testing, system testing and acceptance testing. Initially the tests are focused on each module individually to test whether it is functioning as a unit.

In conventional applications, unit-testing focuses on the smallest combinable program unit, the sub-program (e.g. module, sub-routine, procedure, and component). After testing them, individually, it is integrated into a program structure and does the remaining tests.

**Unit Testing**

The first level of testing is unit testing. When object-oriented software considered the concept of unit chances, rather than testing an individual module, the smallest testable unit is the encapsulated class or object. Class testing for object-oriented software is the equivalent of unit testing for conventional software. Unlike unit testing of conventional software, which tends to focus on the algorithmic detail of the module and the data that flows across the module interface, class testing for object oriented software is driven by the operations encapsulated by the class and state behaviour of the class.

**Integration Testing**

This testing is second level in testing process. After completion of unit testing, which confirms the module’s functionality, we integrated modules to form subsystems. These subsystems are tested under this integration testing. It checks whether data lost or preserved between interface calls. In this module whether data flowed properly across the procedures is tested. Modules are integrated by moving downward through the control hierarchy beginning from the main control module.

**System Testing**

System testing is responsible to ensure total software is worked as per requirements specified in requirement documents. The main reference for this level of testing is requirement document. This goal of testing is to see, whether the system meets its requirements or not.

**Acceptance Testing**

Acceptance testing was top level testing which tests with some realistic data of the client to demonstrate that the software is working satisfactory. Testing here focuses on the external behaviour of the system.

**TEST METHODOLOGY**

This test plan assumes the reader and tester has a comprehensive knowledge of Engineering Connect. The purpose is to test the Engineering Connect system against the system specifications.