Test Plan

for

Strength & Conditioning Online Report System, Release 1.0

Prepared By

Josh Norris and Kejiro Suzuki

Solution Argonauts

SCoRS Team:

Anderson John
Joshua Norris
Preston Flake
Kejiro Suzuki
Ryan Keown

1. Introduction

This document is to serve as the draft test approach for the SCoRS software development project. This document sets the scope of various tests to be conducted, the activities to be completed, the general resources required and the methods and processes to be used to test before release of a deliverable.

1.2 Product Overview

SCoRS provides a way for coaches in the WFISD district to easily and effectively report strength and conditioning test results to the athletic coordinator. SCoRS stores all of the athlete's information in a SQL database and produces various district-wide reports for the coaches and coordinator to view.

1.3 Test types

This is a list of test types we follow.

- **Unit testing**

  This is a White Box test to find any incorrect code in each module.
  It focuses on variables and operations in the object. The testing is mostly done by the programmers, but also by the tester.

- **Integration testing**

  This is a Black Box testing to make sure that individual forms implemented correctly with appropriate modules. It focuses on input and related output, not on the process in modules. Test cases are required.
  It includes database testing which tests the control query from modules such as creating / deleting tables and entities, and up-dating data entries.

- **System testing**

  This testing is performed when all the components (Forms) is put together.
  Use proper test cases to find any incorrectness in the system.

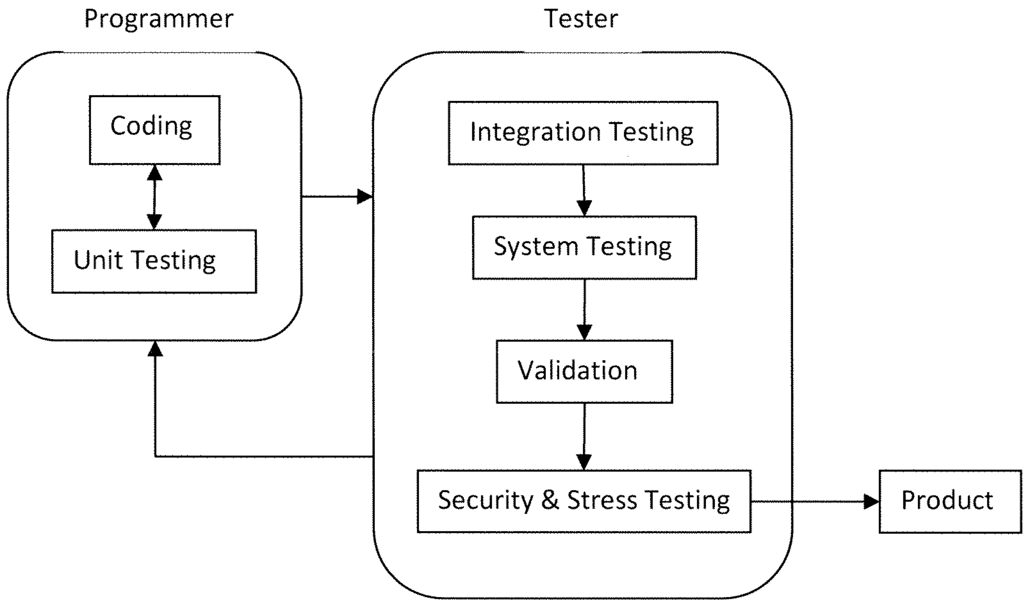
- **Validation**

  This testing and System testing is done simultaneously. It is to make sure that the product meets the requirement appropriately.

- **Security & Stress testing**

  Before the product release, we should try to hack the system. And we should use it in abnormal way as possible, in terms of order and speed of interface usage and volume of data entries.

1.4 Test process flow chart



2.1. Unit testing

At agreed upon regularly scheduled intervals, the system's developmental progress will be subject to peer review. If necessary the testing schedule will correspond to the completion to a system module. The objective of these reviews is to ensure correctness and to test the functional integrity within each individual module. Issues to consider are matching of parameters, arguments, relative attributes, I/O interface and memory management. Symbolic execution will be tested utilizing "basic path testing" also known as "white box" testing, (Sommerville, pg 448).

Entrance Criteria – At least one module should be coded and there should be at least 100 lines of code to be tested to schedule a formal review meeting. As the group meets every Tuesday and Thursday in a week, the development team will try to code at least 100 lines or at least one module by every Tuesday and Thursday so that it can be formally reviewed in the next group meeting.

Exit Criteria – All errors identified, during formal reviews and unit testing are fixed and tested.

2.2. *Integration testing*

This test proves that all areas of the system interface with each other correctly and that there are no gaps in the data flow. Final integration test proves that system works as an integrated unit when all the fixes are complete. "Test harnesses" will be constructed so that a "dummy" caller can invoke each method, and any database interaction will be with a fictitious database. The actual testing method used for this phase will be the "Black box" method, (Sommerville, pg 443). "Bottom-up testing strategy" will be followed throughout the integration testing phase, (Sommerville, pg 453).

Entrance Criteria – Enough code is developed and unit tested to complete atleast one class.

Exit Criteria – All high priority errors from Integration tests must be fixed and tested. All low priority errors left non-fixed should be documented.

2.3. *System testing*

This test intends to prove that the functionality delivered by the team is as specified by the SCoRS team in requirements document. It also assesses the quality of the software and ensures that the software will successfully replace/support the intended Athletic functions required by the client. The testing strategy for the validation of the system as a whole will also utilize the "black box" method. All possible user input must be examined and any deficiencies addressed. Like the integration testing, this phase will be performed in a "bottom-up" manner.

Entrance Criteria – All modules and classes are implemented, unit tested and Integration tested. Complete data base is created with actual user input and the system is put into actual working environment.

Exit Criteria – All high priority errors from system test must be fixed and tested. If any low-priority errors are left not fixed, they should be documented.

3. Testing process time schedule

Testing and coding is processed together in rest of the semester. Here is a brief time requirement for each testing stage.

Coding & Unit Testing – As necessary
Integration Testing – 7 days
System Testing & Validation – 4 days
Security & Stress Testing – 2 days

4. Resources

4.1 Human

The Solution Argonauts consist of:
Anderson John
Joshua Norris
Preston Flake
Ryan Keown
Kejiro Suzuki

Joshua Norris and Kejiro Suzuki will be coordinating the testing phase of the software project. All members will participate in the reviews of the testing.

4.2 Hardware /Software

- Windows 2000 or later version of operating system
- Connection to the Wichita Falls ISD network
- Microsoft .Net Framework 2.0 (included in the installation package)
- 20 MB hard disk space
- 128 MB RAM
- Compact Disk drive

5. *RECORDING PROCEDURES*

During Unit tests, Integration tests and System test, errors will be recorded as they are detected on Error Report forms. Errors which are agreed as valid by the team will be categorized as high-priority errors and low-priority errors based on their impact on system performance and on test implementation and handed over to the development team to make necessary changes to the system implementation. All the errors identified during formal reviews as well as test phases will be properly documented in a formally approved format to record various test results for reporting purposes.

5.1. *Error Report Forms*

A sample error report form that will be used to record a detected error during a formal review or during a unit test phase is given in appendix 6.1. A sample error report form to record an error detected during an integration test phase is given in appendix 6.2 and a sample error report form to record an error during system test is given in appendix 6.3.

5.2. *Unit test results*

All the errors recorded during unit tests of each prototype along with the action taken to correct the error will be documented as a Unit test results document.

5.3. *Integration test results*

All the errors recorded during Integration test phase of each prototype along with the category of the error, status of the error and action taken will be documented as an Integration test results document.

5.4. *System test results*

All the bugs identified while applying each of the following test cases during System test along with category of the error, status of the error and action taken to correct the error will be documented as a System test results document.

Testing Results

for

Strength & Conditioning Online Report System, Release 1.0

Prepared By

Josh Norris

Solution Argonauts

SCoRS Team:

Anderson John
Joshua Norris
Preston Flake
Kejiro Suzuki
Ryan Keown

Log In/Log Out:

- Issue - When a user is logged in and clicks the log-off button the log-in screen reappears but the main window stays open.
  Action Taken – Issue was resolved by changing the code to shut the main window down when log-off is clicked
- Issue – When a non-administrator logs off and an administrator logs on immediately after the program stays in non-administrator mode
  Action Taken – Issue was resolved by changing the code to completely restart all settings to default

Menus:

- Issue – When a menu item is hit multiple times multiple instances of said item open\
  Action Taken- Issue was resolved by checking to see if a menu item was open when it was clicked and if it is open just making it active
- Issue – Test Results menu item has no functionality
  Action Taken – functionality added to forgotten menu item
- Issue – About menu item has no functionality
  Action Taken – functionality added to forgotten menu item

Database:

- Issue – When the database name is wrong there is an unhandled exception thrown
  Action Taken – Issue was resolved by adding code to handle the exception
- Issue – When changing from Access database to SQL database exception was thrown in Remove User function
  Action Taken - Issue was resolved by handling the exception thrown

Reports:

- Issue – Coaches should not be able to access other schools information but in the Rankings report an All Schools option was still available for selection
  Action Taken – Option was removed