

# Automated Commentaries for Simulated Soccer

## Testing Report

Audience	All
Author	Ahsan Mussa
Scope	Testing
Date Created	03/01/2007
Version Number	0.1
Version History	0.1 – Original Document (AMu)
Reviewed	Yes
Last saved by , 20/03/2007	

## **Sign-off sheet**

**Date: 20/03/2007**

**Document Author Signature: AM\***

**Document Author Name: Ahsan Mussa**

**Quality Assurance Signature: AS\***

**Quality Assurance Name: Akbar Sherwani**

**Project Manager Signature: AM\***

**Project Manager Name: Ahsan Mussa**

\* By signing this document you approve that the entire contents of the deliverable has been reviewed and is in line with the objectives of the project.

## Testing Report

Software Testing is the process used to help identify the correctness, completeness, security, and quality of developed computer software. Testing is a process of technical investigation, performed by the member of the group, that is intended to reveal quality-related information about our system with respect to the context in which it is intended to operate. This includes, but is not limited to, the process of executing a program or application with the intent of finding errors. Quality is not an absolute; it is value to some person. With that in mind, testing can never completely establish the correctness of arbitrary computer software; testing furnishes a criticism or comparison that compares the state and behaviour of the product against a specification.

There are many reasons for testing, some of which are:

1. To evaluate the product quality
2. To reduce costs
3. To minimise risks
4. To increase our confidence to an acceptable level
5. To assess and improve the development process

We used a V-Model to guide us with our testing approach. The V-Model maps the software development lifecycle to the testing stages and is shown in figure1 below.

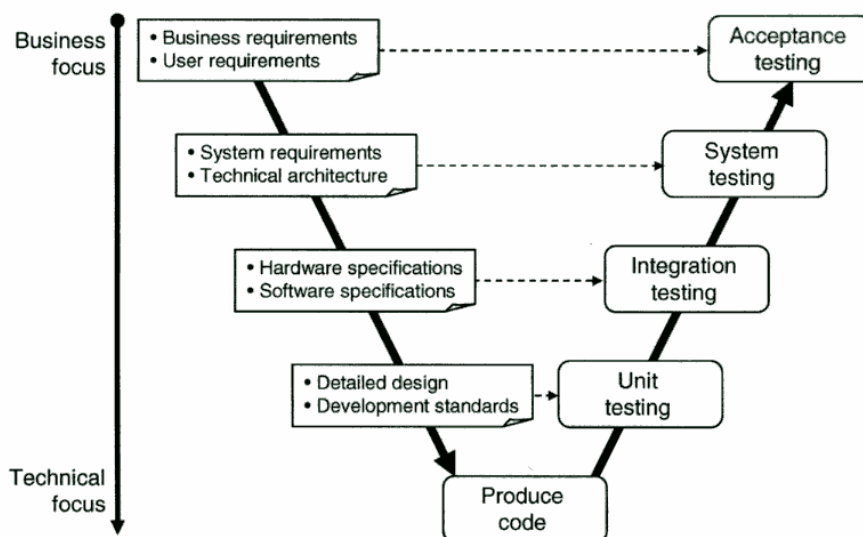


Figure1

Our approach to developing the system was to use extreme programming (XP), therefore using the XP methodology we conducted testing at each iteration of the development cycle.

As our project is based around producing automated commentary for simulated soccer it was very difficult to perform thorough testing as the output

generated is voice. We undertook a visual testing approach for each of the unit testing versions. This gave enabled us to detect errors or bugs in our system by visualisation the activity during the game with the commentary produced. This was not an effective way but it was the only option we had available to us during the first few iterations.

A more detailed testing analysis was created by the use of automated testing on the various classes. This enabled us to design pre-defined set of testing conditions and await results from the system. Our automated testing was very useful in terms of error checking and debugging the system.

Acceptance testing was also carried out on our system where we sampled 50 random people to give their evaluation of our system based on several questions they had to answer while watching the match. These have been documented in the Acceptance Testing section.

In summary we acquired the tools of many testing techniques and bug tracking systems (Bugzilla). All these help make our system more robust and efficient. All the errors and bugs in our system were corrected and fixed. However we cannot be entirely 100% sure that our system is fully correct and complete. There were some errors in the actual components that make up our system i.e. soccer monitor, server and the players, that we did not have control over during the development of our system and were not in scope of the project.