TEST PLAN

PROJECT: THE MOVIE DB

Table of Contents

1 Introduction

The client is a perfect site that has gone through the whole cycle of manual tests. . It includes the objectives, scope, schedule, risks and approach. This document will clearly identify what the final test results will be and what is considered inside and outside the scope.

1.1 Objectives

TestLink is an open source tool, that is, it can be adapted to your needs. This is to facilitate testing and ensure the quality of software.
Testlink is an open source test management tool. It allows you to register plans and test cases as well as control the execution of the tests.The platform supports test case, test plan, unit test among others, also has support for reports and statistics. TestLink has native support for MySQL and PostgreSQL databases
The test team is responsible for testing the product and ensuring that it meets your needs. Test team is the client and the tester in this project.

1.2 Team Members

Name	Role
Fabiano Oliveira	Project Manager
Alaine Furtado	Test Analyst
Rosangela Alecrim	Tester

2 Scope

The initial phase will include all 'must have' requirements. These and any other requirements that get included must all be tested. At the end of Phase 1, a tester must be able to:

1. Create a manual test with as many steps as necessary
2. Save it
3. Retrieve it and have the ability to view it when running the test
4. Enter results and appropriate comments
5. View results

As the team works with the product they will define the needs for the second phase.

Load testing will not be considered part of this project since the user base is known and not an issue.

Rewriting, moving or porting existing test cases from the existing Word documents is not considered part of this project.

3 Assumptions / Risks

3.1 Assumptions

This section lists assumptions that are made specific to this project.

1. Delivery of the product is in format that the test team can check it into CVS.

3.2 Risks

The following risks have been identified and the appropriate action identified to mitigate their impact on the project. The impact (or severity) of the risk is based on how the project would be affected if the risk was triggered. The trigger is what milestone or event would cause the risk to become an issue to be dealt with.

#	Risk	Impact	Trigger	Mitigation Plan
1	Scope Creep – as testers become more familiar with the tool, they will want more functionality	High	Delays in implementation date	Each iteration, functionality will be closely monitored. Priorities will be set and discussed by stakeholders. Since the driver is functionality and not time, it may be necessary to push the date out.
2	Changes to the functionality may negate the tests already written and we may loose test cases already written	High – to schedule and quality	Loss of all test cases	Export data prior to any upgrade, massage as necessary and re-import after upgrade.
3	Weekly delivery is not possible because the developer works off site	Medium	Product did not get delivered on schedule	
4				

4 Test Approach

The project is using an agile approach, with weekly iterations. At the end of each week the requirements identified for that iteration will be delivered to the team and will be tested.

Exploratory testing will play a large part of the testing as the team has never used this type of tool and will be learning as they go. Tests for planned functionality will be created and added to TCT as we get iterations of the product.

4.1 Test Automation

Automated unit tests are part of the development process, but no automated functional tests are planned at this time.

5 Severity List

The tester entering a bug into GForge is also responsible for entering the bug Severity.

Severity ID	Severity	Severity Description
1	Critical	The module/product crashes or the bug causes nonrecoverable conditions. System crashes, GP Faults, or database or file corruption, or potential data loss, program hangs requiring reboot are all examples of a Sev. 1 bug.
2	High	Major system component unusable due to failure or incorrect functionality. Sev. 2 bugs cause serious problems such as a lack of functionality, or insufficient or unclear error messages that can have a major impact to the user, prevents other areas of the app from being tested, etc. Sev. 2 bugs can have a work around, but the work around is inconvenient or difficult.
3	Medium	Incorrect functionality of component or process. There is a simple work around for the bug if it is Sev. 3.
4	Minor	Documentation errors or signed off severity 3 bugs.

6 Test Environment

A new server is required for the web server, the application and the database.

7 Milestones / Deliverables

7.1 Test Schedule

Task Name	Start	Finish	Effort	Comments
Test Planning				
Review Requirements documents			2 d	
Create initial test estimates			1 d	
Staff and train new test resources				
First deploy to QA test environment				
Functional testing – Iteration 1				
Iteration 2 deploy to QA test environment				
Functional testing – Iteration 2				
System testing				
Regression testing				
Resolution of final defects and final build testing				
Deploy to Staging environment				
Performance testing				
Release to Production				

7.2 Deliverables

Deliverable	For	Date / Milestone
Test Plan	Project Manager, Test analyst	
Script for automation test	Project Manager, Test analyst	
Bugs Report	Project Manager	
Test Report	QA Manager, QA Director	
Metrics	All team members	