

# Chapter 1

## TEST PLAN (TP) TEMPLATE

Version 1.1, October 2013

### FOREWORD

This template was created to provide system and software development projects with a model Test Plan (TP) document template. The template is based on IEEE 829 Format. It has been edited and updated by Dr. Clint Jeffery for use in UI CS 383.

The TP template begins on the next page. Just throw away this page and enter your project specifications into the following template. Don't forget to change the headers and footers as necessary. The following conventions are used to guide you in developing your TP:

[ Text ] **Replace** this text with your project design text.

*text in italics* Notes/instructions to the author. **Delete in your finished document.**

**TEST PLAN (TP)**

**FOR**

**TEST Plan for Spells and Characters**

Version 1.1.0

May 1st, 2014

Prepared for:

**CS 383 Course Project**

Prepared by:

**Tao Zhang, Cameron Simon**

**University of Idaho**

**Moscow, ID 83844-1010**

**CS383 TPD**

### RECORD OF CHANGES (Change History)

[illegible]

A - ADDED    M - MODIFIED    D - DELETED

[ put program /system name here ]

## TABLE OF CONTENTS

Section Page



<b>1</b>	<b>TEST PLAN (TP) TEMPLATE</b>	<b>1</b>
1	IDENTIFIER . . . . .	2
2	REFERENCES . . . . .	2
3	INTRODUCTION . . . . .	2
4	TEST ITEMS . . . . .	2
5	SOFTWARE RISK ISSUES . . . . .	2
6	FEATURES TO BE TESTED . . . . .	2
7	FEATURES NOT TO BE TESTED . . . . .	2
8	APPROACH . . . . .	3
	8.1 Testing Tools . . . . .	3
	8.2 Metrics . . . . .	3
	8.3 Configurations . . . . .	3
	8.4 Software . . . . .	3
	8.5 Hardware . . . . .	3
	8.6 Automated Testing . . . . .	3
	8.6.1 Unit Testing . . . . .	3
	8.7 Manual Testing . . . . .	3
9	ITEM PASS/FAIL CRITERIA . . . . .	3
	9.1 Reporting a failure . . . . .	3
	9.1.1 What Constitutes failure? . . . . .	3
	9.1.2 What Does Not Constitute a failure? . . . . .	3
	9.1.3 What to do when a failure is discovered? . . . . .	3
	9.1.4 What if a specification document is incorrect (e.g. outdated, misstated)? . . . . .	3
	9.1.5 What sections to include in an SCR? . . . . .	4
	9.2 Unit Testing Pass/Fail Criteria . . . . .	4
	9.3 Intergration Testing Pass/Fail Criteria . . . . .	4
	9.3.1 Character . . . . .	4
	9.3.2 Spells . . . . .	4
10	SUSPENSION CRITERIA . . . . .	4
11	TEST DELIVERABLES . . . . .	4
12	REMAINING TEST TASKS . . . . .	5
13	ENVIRONMENTAL NEEDS . . . . .	5
14	STAFFING AND TRAINING NEEDS . . . . .	5
15	RESPONSIBILITIES . . . . .	5
16	SCHEDULE . . . . .	5
17	PLANNING RISKS AND CONTINGENCIES . . . . .	5
18	APPROVALS . . . . .	6
19	GLOSSARY . . . . .	6
20	APPENDIX A. [insert name here] . . . . .	1
21	APPENDIX B. [insert name here] . . . . .	1

# 1 TEST PLAN IDENTIFIER

SWORDS & SORCERY TEST PLAN VERSION 1.0, CS 383 TEAM

## 2 REFERENCES

Use Cases and State diagrams are available at <https://github.com/cjeffery/sworsorc/tree/master/doc>

## 3 INTRODUCTION

The purpose of this test plan is to state the processes used by Game Rules and Play Team in testing the Spells and Characters for the Swords & Sorcery project.

## 4 TEST ITEMS

- Class Interfaces
- Class Interactions
- Spells Implementation
- Character

## 5 SOFTWARE RISK ISSUES

Testing will mainly focus on code written by Game Rules and Play Team members, more specifically spells and characters. This plan largely focuses on providing a testing plan that adequately covers both rules and implementation. As the code base is not yet complete, the complexity of the code cannot be yet determined. Also, since the documentation is often incomplete, risk also arises because of poor documentation. Complete and improved documentation of all areas of code with reduce this risk. Since our specific code for spells and characters is not dependent on the third party software, there is a little risk comes from new version of software or failure of any third parties.

## 6 FEATURES TO BE TESTED

- Select character on GUI
- Select Spell
- Effects of Partial Spells
- Manna Costing

## 7 FEATURES NOT TO BE TESTED

- Speed of software

## **8 APPROACH**

### **8.1 Testing Tools**

JUnit on NetBeans.

### **8.2 Metrics**

### **8.3 Configurations**

### **8.4 Software**

The software will be developed with Java 1.8.

### **8.5 Hardware**

### **8.6 Automated Testing**

The automated test process will be Unit testing ...

#### **8.6.1 Unit Testing**

Not yet.

### **8.7 Manual Testing**

Manual testing will be required for the portions of the program that can not undergo automated testing. This section applies to the testing of the user interface to simulate user orientated testing to verify conformance to the documented use cases.

## **9 ITEM PASS/FAIL CRITERIA**

### **9.1 Reporting a failure**

If a failure happens during the execution of any test, a failure report should be submitted to provide information of the failure. Failure should be reported in the issue page of the repository.

The name of the issue should be the name of the code file and the name of the test if applicable.

#### **9.1.1 What Constitutes failure?**

And deviation from a specification, e.g. SRS, UML diagrams.

#### **9.1.2 What Does Not Constitute a failure?**

Any unit or action that does not have any requirements documentation, cannot cause a failure.

#### **9.1.3 What to do when a failure is discovered?**

Produce a SCR to document each failure that needs to be corrected.

#### **9.1.4 What if a specification document is incorrect (e.g. outdated, misstated)?**

This also constitutes a failure and an SCR should be created.

### 9.1.5 What sections to include in an SCR?

Failure Identified  
Expected Outcome  
Actual Behavior  
Steps to reproduce

Each section should be brief and to the point, but yet convey enough information for the coder.

## 9.2 Unit Testing Pass/Fail Criteria

## 9.3 Intergration Testing Pass/Fail Criteria

### 9.3.1 Character

Rule Description	Test Description	Expected Result
TBD	createCharacter function is called with specific character name from Character-Maker.java.	A new character object is created and returned that contains all of the specified characters information.

### 9.3.2 Spells

Rule Description	Test Description	Expected Result
Character with Power Level should have a spell book.	Generate a spell book for the character.	A frame with a list of spells should be shown on the screen.
Show spell description.	Click on the Spell button.	A frame will be displayed with all information about the spell.
A target need to be selected for most of the spells.	Click on cast button on the frame of displaying information about the spell. Then select a target by right click the target on the map.	A target is selected to cast the spell.

## 10 SUSPENSION CRITERIA AND RESUMPTION REQUIREMENTS

*If the number or type of defects reaches a point where the follow on testing has no value, it makes no sense to continue the test; you are just wasting resources.*

*Specify what constitutes stoppage for a test or series of tests and what is the acceptable level of defects that will allow the testing to proceed past the defects. [Insert text here.]*

## 11 TEST DELIVERABLES

*What is to be delivered as part of this plan?*

- *Test plan document*



- *Test cases*
- *Relevant error logs or problem reports*

*One thing that is not a test deliverable is the software itself that is listed under test items and is delivered by development.* [Insert text here.]

## 12 REMAINING TEST TASKS

*If this is a multi-phase process or if the application is to be released in increments there may be parts of the application that this plan does not address. These areas need to be identified to avoid any confusion should defects be reported back on those future functions. This will also allow the users and testers to avoid incomplete functions and prevent waste of resources chasing non-defects.* [Insert text here.]

## 13 ENVIRONMENTAL NEEDS

*Are there any special requirements for this test plan, such as:*

- *Special hardware such as simulators, static generators etc.*
- *How will test data be provided. Are there special collection requirements or specific ranges of data that must be provided?*

[Insert text here.]

## 14 STAFFING AND TRAINING NEEDS

*Training on the application/system.*

*Training for any test tools to be used.* [Insert text here.]

## 15 RESPONSIBILITIES

*Who is in charge?*

*This issue includes all areas of the plan. Here are some examples:*

- *Selecting features to be tested and not tested.*
- *Ensuring all required elements are in place for testing.*

[Insert text here.]

## 16 SCHEDULE

*Should be based on realistic and validated estimates. If the estimates for the development of the application are inaccurate, the entire project plan will slip and the testing is part of the overall project plan.*

[Insert text here.]

## 17 PLANNING RISKS AND CONTINGENCIES

*What are the overall risks to the project with an emphasis on the testing process? Specify what will be done for various risk events.*

[Insert text here.]

## 18 APPROVALS

*Who can approve the process as complete and allow the project to proceed to the next level (depending on the level of the plan)?*

[Insert text here.]

## 19 GLOSSARY

*Used to define terms and acronyms used in the document, and testing in general, to eliminate confusion and promote consistent communications.*

[Insert text here.]

## 20 APPENDIX A. [insert name here]

*Include copies of test examples, etc. supplied or derived from the customer. Appendices are labeled A, B, ... n. Reference each appendix as appropriate in the text of the document.*

[ insert appendix A here ]

## 21 APPENDIX B. [insert name here]

[ insert appendix B here ]