**Issue Management Plan**

**Solutions Resource Inc.**

**Project Name :** Skosay

**Project Type:** Web app + Mobile app + API

**Date :** May 5, 2015

**Revisions**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Section | Description of Changes | Author | Date | Version |
| All | Initial draft | Shiela Marie C. Barbacena | 5/4/2015 | 1.0 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Table of Contents**

[1.0 Introduction](#h.gjdgxs)

[2.0 Definition of Defects](#h.30j0zll)

[2.1 Defect Tracking](#h.1fob9te)

[2.2 Defect Categorization](#h.3znysh7)

[2.3 Defect Lifecycle](#h.2et92p0)

# Introduction

The purpose of this document is to outline the process on how to manage issues that may arise during the actual usage of the application.

# Definition of Defects

* A defect is an error or a bug in the application.
* When actual deviates from the expected behaviour of the system.
* When the result of the software application or product does not meet with the end user expectations or the software requirements.

# Defect Tracking

To properly manage defects, these will be entered and tracked via issue tracking system. Each entry will include detailed information about each defect in order to provide useful information for the developers to fix the issues.

# Defect Categorization

Defects found in the application can be categorized by severity and priority.

1. Severity - It is the extent to which the defect can affect the software. In other words it defines the impact that a given defect has on the system. **For example:** If an application or web page crashes when a remote link is clicked, in this case clicking the remote link by an user is rare but the impact of application crashing is severe. So the severity is high but priority is low.

* Low - if problem involves with cosmetic feature of the system (E.g. information is correctly shown but the appearance is wrong, such as misspelled words, wrong font, wrong indentation, layout, etc.)
* Medium - there is no major system functionalities are impaired but there is workaround
* High - one or more major system functionalities are impaired but there is workaround
* Critical - one or more major functionalities are impaired and there is no workaround

1. Priority - Priority defines the order in which we should resolve a defect. Should we fix it now, or can it wait? This priority status is set by the tester to the developer mentioning the time frame to fix the defect. If high priority is mentioned then the developer has to fix it at the earliest. The priority status is set based on the customer requirements. **For example:** If the company name is misspelled in the home page of the website, then the priority is high and severity is low to fix it.

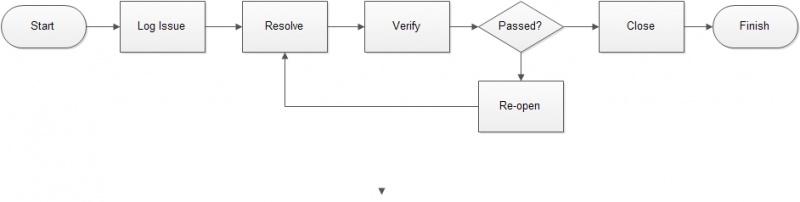
Priority can be of following types:

* Low: The defect is an irritant which should be repaired, but repair can be deferred until after more serious defect have been fixed.
* Medium: The defect should be resolved in the normal course of development activities. It can wait until a new build or version is created.
* High: The defect must be resolved as soon as possible because the defect is affecting the application or the product severely. The system cannot be used until the repair has been done.

*Note:* *Major and minor features will have to be identified by the stakeholder.*

# Defect Lifecycle

Defects must be clearly captured and escalated to ensure prompt resolution by development. Each defect can be re-categorized (if needed), re-tested, estimated, resolved, and verified by the development team prior to closure. The following is a snapshot of the standard defect lifecycle:



# Defect Reporting

Once a user encountered an issue, he can directly report it through the issue tracker on Github. The prescribed format is as follows:

**Issue Title:**

* Put a brief description of the issue encountered. State what happened after executing the test case like what was the prompt message, what was displayed, what was lacking or what was misplaced.

**Description:**

1. **Steps to replicate** - enumerate the steps on how to replicate the issue.
2. **Actual Result** - Describe the actual result after conducting the steps mentioned above
3. **Expected Result** - Describe the expected result after conducting the steps mentioned above.
4. **Other Details** – Describe the other details that might be useful for the replication of the issue like operating system, OS version, type of device and internet connection used.

**Labels:**

* The user will use labels to categorize the issue based on severity and priority.

**Attachments:**

* The user should upload screenshots of the issue encountered as much as possible.

# Response Plan to Defects

The following are the suggested standard action for each defect reported based on its severity and priority:

|  |  |  |
| --- | --- | --- |
| **Severity** | **Priority** | **Response Time** |
| Critical Severity | High Priority | Immediate |
| Critical Severity | Medium Priority | Immediate |
| Critical Severity | Low Priority | 1 day - 2 days |
| High Severity | High Priority | Immediate |
| High Severity | Medium Priority | Immediate |
| High Severity | Low Priority | 1 day - 2 days |
| Medium Severity | High Priority | Immediate |
| Medium Severity | Medium Priority | 1 day - 2 days |
| Medium Severity | Low Priority | 1 day - 2 days |
| Low Severity | High Priority | Immediate |
| Low Severity | Medium Priority | 1 day - 2 days |
| Low Severity | Low Priority | 1 day - 2 days |