Risk Plan

for the

Hotspotter Bug Prediction Software

CS 425 / CS 499 Senior Project

by

Nathan Reinhardt

Spencer Smith

Dylan Williams

of

Team HotSpotter

RISK-PLAN

Revision 1.1

As Of: 4 October 2015

Change Log:

Revision	Change Note(s)
1.0	• Initial release
1.1	• Added documentation and purpose prose

1 INTRODUCTION

This document is the risk plan for the Hotspotter Bug Prediction Software developed by Team HotSpotter.

1.1 PURPOSE

The purpose of this document is to outline team risk management and mitigation strategies. The document will also contain the Risk Analysis Chart.

2 MANAGEMENT

During weekly standup meetings, the team will discuss and review high impact risk. The group will come to consensus on each risk and update the risk analysis chart seen below. The team may set up monitoring on potential system breaking risk to enable fast mitigation response.

Table I: Risk Analysis Chart

No.	Risk Label	Description	Likelihood	Impact	Score	Mitigation Strategies
1	Software stack issues	The software stack does not provide proper functionality to implement system. Software interface may change over time.	2	8	16	Push scanning algorithm to external batch job.
2	Inefficient metrics	Some metrics may be a waste of time to research.	4	3	12	Eliminate poor metrics early.
3	Storing metadata in database	The repository must be pulled from the git url and translated into a portable format for database storage.	6	1	6	Establish standard translation early.
4	Graphics library problems	The graphic's library used to display the repository visualization is inadequate or proves difficult to use.	5	6	30	Switch to different graphics library for next sprint.
5	Browser compatibility issues	The web interface breaks down in some browsers.	2	2	4	Use standard functions supported by required browsers.

6	Repository size	The Repository is so large it won't completely fit in memory when running metrics analysis.	3	2	6	Build the scanner to be flexible and handle chunks of the repository.
7	New software	The software stack is new to most members on the team which may cause errors due lack of experience.	9	5	45	Design and teaching session with team members
8	No server	No server is available to test and deploy the system.	1	5	5	Test locally and build virtual machine for deployment
Note: Likelihood and Impact ratings are on a scale of 1 (Low) to 10 (High). The score is the product of the Likelihood and Impact.						