Project Plan

for the

Hotspotter Bug Prediction Software

CS 425 / CS 499 Senior Project

by

Nathan Reinhardt

Spencer Smith

Dylan Williams

of

Team HotSpotter

PROJECT-PLAN

Revision 1.1

As Of: 11 October 2015

Change Log:

Revision	Change Note(s)
1.0	• Initial release
1.1	• Modified project schedule, milestones, and architecture diagram

1   INTRODUCTION

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

1.1 PURPOSE

The purpose of this document is to define high-level estimation of the work needed to be done, the time expected for the work, the resources needed for the work and the schedule in which the work is expected to be accomplished. The document will also include a detailed exit strategy for CS 425 and CS 499.

2   ESTIMATION

The team estimates that a completed product fulfilling all customer requirements will contain the following components:

- User Web Interface
    - Allows user to submit a new repository to watch and analyze
    - Allows user to review data of currently watched repositories
- Web Backend
    - Handle all user requests from the User Web Interface
    - Adjust Metadata into user-friendly format for viewing or export
- Git Cloner Component
    - Clone and update watched git repositories to server disk
- Metadata Parser Component
    - Parse git repositories on disk, collecting metadata used in analysis
- Database
    - Store the metadata collected by Metadata Parser Component
    - Return requested data to Web Backend
- Git repository (Provided by User)
    - Provided by user in User Web Interface
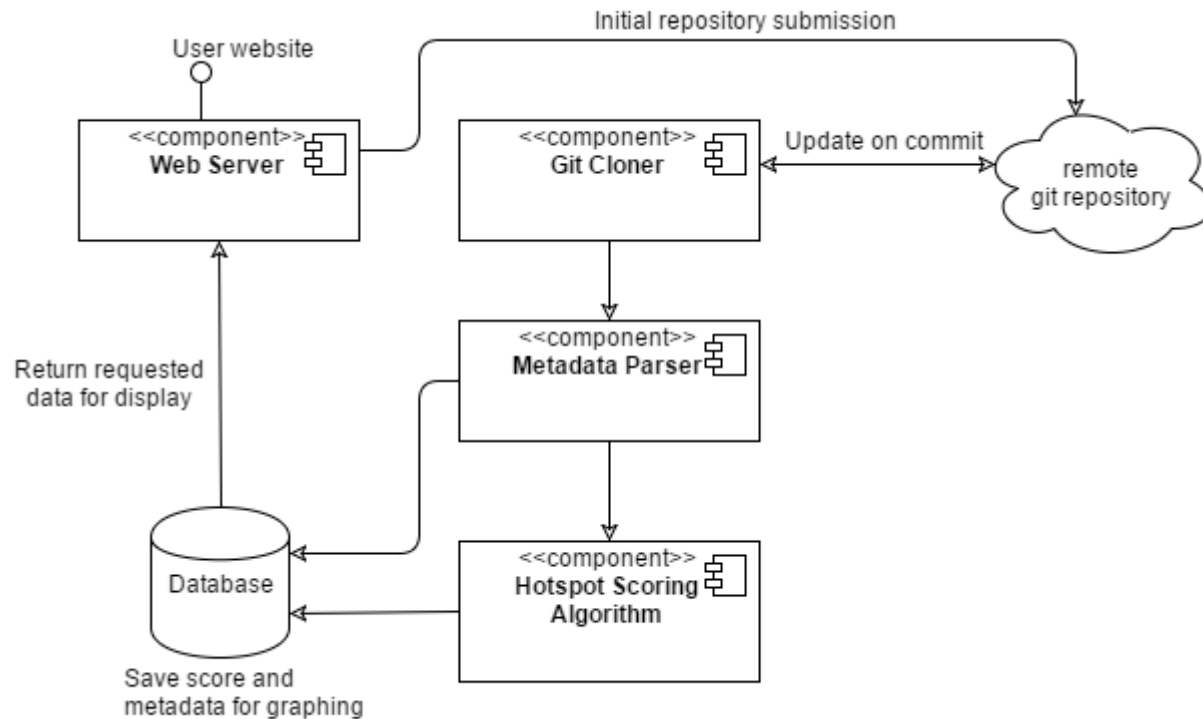   - Cloned by Git Cloner Component

Figure 1 - High-Level Architecture of Estimated Components

In the pursuance of the completed product, the following milestones will be reached after an estimated accumulated amount of person-hours:

- Prototype display of repository metadata   – 120 p-h
  - CS425 Exit
- Web backend finalization   – 160 p-h
- Hotspot scoring algorithm implemented   – 190 p-h
- Full implementation of system   – 210 p-h
- Post-Mortem Report   – 240 p-h
  - CS499 Exit

3 RESOURCES

- Server
  - A server will be needed when the final proof of concept is finished. Team Hotspotter will then use this server to transition the concept from being a locally deployed application to a server deployed application.

4 SCHEDULING

4.1 FALL 2015

Sprint	Week	Day	Date	Milestone	Deliverable	Resource
1	6	T	9/29	Project Plan, Risk Plan, Quality Plan, Sprint Planning Doc	Project Plan, Risk Plan, Quality Plan, Sprint Planning Doc	N/A
		R	10/1			N/A
	7	T	10/6	Presentation: Project Plans	Presentation: Project Plans	N/A
		R	10/8			N/A
2	8	T	10/13	Sprint Planning Doc	Sprint Planning Doc	N/A
		R	10/15			N/A
	9	T	10/20	Status Report #1	Status Report #1	N/A
		R	10/22			N/A
3	10	T	10/27	Sprint Planning Doc	Sprint Planning Doc	N/A
		R	10/29			N/A
	11	T	11/3	Status Report #2	Status Report #2	N/A
		R	11/5			N/A

4	12	T	11/10	Sprint Planning Doc Presentation	Sprint Planning Doc Presentation	N/A
		R	11/12			N/A
	13	T	11/17	Status Report #3	Status Report #3	N/A
		R	11/19	Prototype Display of repository data	Full working proof of concept	N/A
		F	**11/20**	**PRESENTATION**	**PRESENTATION**	N/A
Thanksgiving Break Week						
PM	14	T	12/1			N/A
		R	12/3	Customer Representative Report	Customer Representative Report	N/A
	15	T	12/8	Status Report #4	Status Report #4	N/A
		R	12/10	Post-Mortem Report	Post-Mortem Report	N/A
E	Exam	T	**12/15**	**Project Portfolio Due at Noon**	**Project Portfolio Due at Noon**	N/A

4.2 SPRING 2016

Sprint	Week	Day	Date	Milestone	Deliverable	Resource
5	1	T	1/12	Sprint Planning Doc	Sprint Planning Doc	N/A
		R	1/14			N/A
	2	T	1/19			N/A

		R	1/21			N/A
		M	1/25	Completed Back End	Server/Batch Job Code	Server
6	3	T	1/26	Sprint Planning Doc		N/A
		R	1/28			N/A
	4	T	2/2	Status Report #5	Status Report #5	N/A
		R	2/4			N/A
7	5	T	2/9	Sprint Planning Doc	Sprint Planning Doc	N/A
		R	2/11			N/A
	6	T	2/16	Status Report #6	Status Report #6	N/A
		R	2/18			N/A
		M	2/22	Completed Hotspot scoring algorithm	Code components integrated	Server
8	7	T	2/23	Sprint Planning Doc	Sprint Planning Doc	N/A
		R	2/25	Team Meetings		N/A
	8	T	3/1	Software Beta Release Status Report #7	Beta Release Demo	N/A
		R	3/3			N/A
Spring Break Week						

9	**9**	T	3/15	Sprint Planning Doc	Sprint Planning Doc	N/A
		R	3/17			N/A
	10	T	3/22	Status Report #8	Status Report #8	N/A
		R	3/24	Team Poster (Draft)	Team Poster (Draft)	N/A
		M	3/28	Full Implementation of Hotspotter	Hotspotter Application	Server
10	**11**	T	3/29	Sprint Planning Doc	Sprint Planning Doc	N/A
		R	3/31	Team Poster (Final)	Team Poster (Final)	N/A
	12	T	4/5	Presentation Status Report #9	Status Report #9	N/A
		R	4/7			N/A
		M	4/11	Acceptance Tested Code	Test Results Doc	Server
P	**13**	T	4/12			N/A
		R	4/14			N/A
		F	**4/15**	**PRESENTATION**		N/A
PM	**14**	T	4/19	Delivery; Acceptance Testing	Delivery; Acceptance Testing	N/A
		R	4/21			N/A

15	T	4/26	Customer Representative Report; Status Report #10	Customer Representative Report; Status Report #10	N/A
	R	4/28		Post-Mortem Report	N/A
E	**Exam**	**T**	**12/15**		**Project Portfolio Due at Noon**	Post-Mortem Report

5   COMMUNICATION

- Google Hangouts –Used for day to day communication for the team.
- BitBucket.org – Primary code review forum.
- Email – Used for primary communication with client and formal communication between team members.
- Jira – Used to collaborate tasks and problems within the project each sprint.

6   QUALITY ASSURANCE AND RISK MANAGEMENT

To ensure quality assurance the team will maintain biweekly meetings, weekly communication with the client, behavior driven development and version control. See Quality-Plan.pdf for more detail. The risk plan will mitigate risks by providing a viable backup plan for high impact risks. See Risk-Plan.pdf for specific details and risks.

7   CS425 EXIT STRATEGY

The end of the CS425 semester will produce a core component of the final product. The core component will be created from refining prototypes that model system functionality. The component will be the framework for the client server system and provide basic functionality of the file visualization. The system will have a functioning database to store the repository metadata. This component will demonstrate the basic functionally needed to select a repository, store the metadata and display a simple visualization of the repository. This will require each subsystem to undergo unit testing and integration testing the subsystems' interfaces.

8   CS499 EXIT STRATEGY

The CS499 semester will add more advance functionally to the system and usability documentation for longevity quality assurance. The main focus will be the scoring algorithm for the bug hotspots which will incorporate many dynamic metrics. Each metric will require proof-of-concept analysis on effectiveness which entails performing many tests. The test will compare results against known static metrics and if possible existing code analyzing programs. The metrics will be approve or rejected based off the results and since time is a factor only so much research and testing can be done. Next the visualization output will be improved more advance display options and with the addition of the scoring algorithm results integrated. All server side functionality will be implemented as APIs and documentation will be created for integration and use. Once all the metrics have been tested and approve the system will be demonstrated using different existing git repositories found online. The project will be delivered in two part the web page front end and server backend. The backend will contain a majority of the project and be built modular so more functionality can be added via new APIs. This also means the front end can be rebuilt and modified by using the backend APIs. The client will have admin access and complete control over the system when delivered.

9   SUMMARY

The Project Plan describes the overall work, expected time to complete the work and when the work is expected to be done. The project requires a server for final implementation and deployment. The team uses various management and communication such as email, google hangout, BitBucket and Jira. A detailed schedule shows the planned work effect for the entire project.

10 REFERENCES

1. RISK-PLAN
2. QUALITY-PLAN