1.4 Product Overview

1.4.1 Product Perspective

System Interfaces

The web server will interface with the web page and database. The web server will be handle calculating data, sending data to web page and sending data to the database for storage. The web page will display the data from the web server and send commands to the server. The database will handle storing and retrieving data for the web server.

User Interfaces

The user interface will be a simple web page to interact with the remote database through the web server.

Hardware Interfaces

N/A

Software Interfaces

The software stack used by the system will be MEAN. The application will be web based so it will be able to interface with any operating system will a supported browser. For simplicity the supported browser will be Chrome.

Communication Interfaces

The web page will communicate through standard Secure HTTP since some of the information could be sensitive.

Memory

The database will store every revision from the git repository so the amount of active repositories in the system will be limited.

Operations

Normal user operations include:

Connect to repository

Calculate hotspots

Filter hotspots

Update repository

Special user operations include:

Specify update timeframe

Remove active repository from database

Site adaptation requirements

N/A

1.4.2 Product Functions

The system will first connect a git repository specified by the user and download the said repository to a remote database. The system finds the hotspots in the repository with a user specified date range and dynamic metrics. The system displays the results in a visual and/or text format for the user to view. The results can be filtered by a file, function, developer or intensity of hotspot. The repository data is automatically pushed to the server by a given time frame or can be manually updated by the user. Multiple user can look at the results through different web interface instances.

1.4.3 User Characteristics

This software will be mainly used by software project team leaders and resource managers. The individual software coders may also find it useful for self-evaluation. The system won’t require the user to have programming experience but will act as tool for them to understand where the troublesome code is located.

1.4.4 Limitations

The system will only be able to use git repositories for calculating hotspots. The data will be store on a remote database for processing and presented through a web interface.

3.7 Software System Attributes

No system attributes issued by client.

3.8 Supporting Information

References:

<http://google-engtools.blogspot.com/2011/12/bug-prediction-at-google.html>

<https://github.com/igrigorik/bugspots>

<http://landley.net/writing/git-bisect-howto.html>

<http://macbeth.cs.ucdavis.edu/fse2011.pdf>

Problems Solved:

This software searches through git repositories and finds hotspots at the function level using dynamic metrics. The hotspots in the files are visualized with different shades of colors showing the intensity of the hotspot.

3.9 Course-specific

Semi-Agile Software Engineering (SAGE)