

**Assignment 9: Individual Requirements  
Analysis: Spring 2020 CS4320/7320  
Software Engineering**

Created by: Levi Olson

## **1. Introduction**

Many Technology and IT companies rely on various open source projects. Since these sorts of projects tend to be worked on by many different groups of people over different periods of time, it is imperative that companies are able to deduce whether a given project is both healthy and sustainable. CHAOSS (community health analytics open source software) aims to tackle this issue by providing a vast array of different metrics for a given project. However, metrics on their own are hardly very meaningful, therefore this software aims to create visualizations of the data derived from the metrics. The goal is that the visualizations will allow someone to view how healthy a project is, they can view things like how actively it is being updated, how much documentation is given for it, how diverse the work base is, and other similar traits which are all indicative of how healthy and sustainable an open source project is. The data is compiled by members of workgroups who specialize in a certain set of metrics. Overall implementation of this software should allow someone to view the metrics of a given project in a way that is both logical and that provides meaningful insight into the overall health of the project.

## **2. Software Product Overview**

The aim of CHAOSS is to take the metrics of a project and derive useful data for various different companies and analysts. The software would track the metrics in a given project and then present the visualization of the metric as a function of time so that users can better determine the overall health of the project. Furthermore the software will allow comparisons between metrics of different projects, this gives users a way to compare the metrics of a project with another project which may be deemed more or less healthy. The metrics would be selected based on the work of CHAOSS community members who have worked on a particular repo. The metrics themselves would be selected based on what someone finds to be the most meaningful for a given project. The range of metrics is also wide and varied and most likely different project managers will view certain metrics as more valuable than others. Lastly, in order to better organize data users shall be able to search their various repos by ID so as to keep metrics from different projects separate from each other as well as allow for easy comparisons between two different projects.

### 3. System Use

This system has a few key actors, first of all there are the technology company managers who are the users who actually look at the presented metrics. Last there are the members of the workgroups who are the ones that compile the data in a way that makes sense to the analysts/company managers.

#### 3.1 Actor Survey

##### **Manager/Analyst**

Manager/Analyst is the user that actually views the data generated from the metrics in a given project. They will interact with the system by selecting the repos which they are interested in viewing as well as selecting the metrics/data that they wish to view for a given metric.

##### **System Features:**

- Select repo
- Select metrics/data
- Select visualization

##### **Workgroup member**

A workgroup member is the user who collects the metrics based on input from the project contributors. It is the job of a workgroup member to compile the data and derive a meaningful visualization of the metrics. This actor interacts with the system by selecting metrics and creating visualizations of them as well as determining what metrics are most valuable for representing the health of a project.

##### **System Features:**

- Compile metrics
- Select repo
- Select time range
- Generate metric visualization

##### **Project Contributor**

A project contributor is someone from the CHAOSS community who volunteers to submit their metrics whilst working on a given open source project. This actor interacts with the system by submitting their metrics from a project.

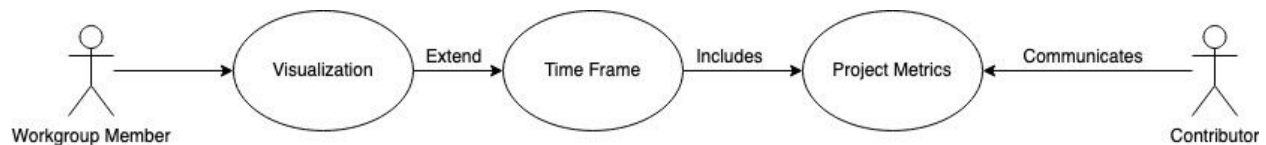
## System Features

- Submit metrics

## 4. System Requirements

### 4.1 Use Case Diagrams

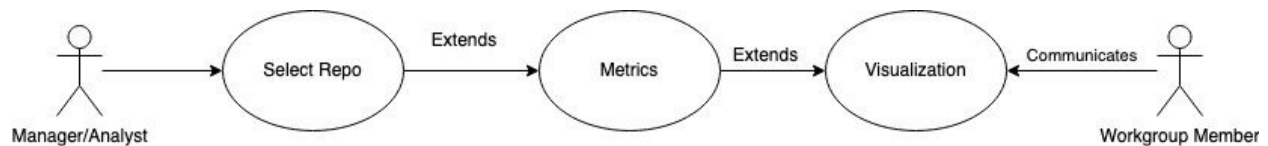
#### Use Case 1: Create Visualization



Use-case name	UC1 Create Visualization
System or subsystem	CHAOSS
Actors	Workgroup Member, Contributor
Brief Description	This use case explains how a workgroup member creates a visualization of metrics submitted by a project contributor.
Basic flow of events	<p>Basic flow begins when a workgroup member wants to create a visualization of a set of metrics for a given project. The workgroup member selects create visualization from a menu:</p> <ol style="list-style-type: none"><li>1. The system displays a time frame range to be selected. The User can choose a timeframe based on day, week, month, or year. There will be a dropdown window for the time period.</li><li>2. The user searches for the project metrics that they are interested in.</li><li>3. The workgroup member then organizes the metrics in a way that makes sense and creates a visualization from the metrics</li></ol>
Special Requirements	<p><u>Usability:</u> The system will present information in a clear and intuitive manner so as to make it easy for a user to select or submit information.</p> <p><u>Performance:</u> The system should be able to generate a</p>

	visualization relatively fast. <u>Reliability:</u> When an error occurs control should be returned to the user.
Pre-conditions	The workgroup member must log in to a main menu screen
Post-conditions	A visualization is generated for review by the workgroup member

### Use Case 2: Select Visualization



System or subsystem	CHAOSS
Actors	Manager/Analyst, Workgroup Member
Brief Description	This use case explains how a manager can view the visualization of metrics from a particular project repo.
Basic flow of events	<p>Basic flow begins when a manager wants to view the metrics for a particular repo:</p> <ol style="list-style-type: none"> <li>1. The manager enters in a repo ID into a dropdown menu.</li> <li>2. The system displays a list of possible metrics to display for that repo. The manager selects the metric that they are interested in.</li> <li>3. The system displays the visualization generated by the workgroup member.</li> </ol>
Special Requirements	<p><u>Usability:</u> The system will present the type of metrics available to a user in a concise manner so as to allow a manager to easily find the metric they are interested in.</p> <p><u>Performance:</u> The system should take less than 10 seconds to display metrics and pull up the appropriate metric for a visualization.</p> <p><u>Reliability:</u></p>

	When an error occurs, control should be transferred to the user.
Pre-conditions	none
Post-conditions	A concise visualization of the metric is displayed for the user.

## 4.2 System Functional Specification

CHAOSS is a web application, so the system will be a client server style of architecture. From the perspective of the client the following interface functions will be supported:

- Main menu screen with different options based on the user
- Select repository by entering the ID
- Select the specific metric you are interested in
- Save or download the visualization that is created for that visualization
- Create/upload a visualization for a certain metric.

## 4.3 Non-functional Requirements

### Usability

- Users will be taught about the different metrics and what they mean
- Work group members should learn what makes certain metrics useful to determining project health
- Contributors will learn how to submit their metrics to the system

### Reliability

- The application should be available for a large period of time, with moments of downtime for system repair
- Users should be able to access the system at any time of the day
- Time measurements of metrics should be accurate

### Performance

- There should be fast response times for a reasonable number of users
- Multiple users should be able to access and generate metric visualizations simultaneously
- The system should make efficient use of server storage

### Supportability

- The project should be well organized as to allow for multiple people to work on it
- Naming conventions should be followed. Names of classes, methods and variables should make sense
- Detailed comments explaining the purpose for blocks of code

## 5. Design Constraints

- The system should be supported on every major web browser which includes Internet Explorer, FireFox, Chrome and Safari
- New categories of metrics should be easy to add to the system
- Web standards will be adhered to
- Each category of metric should have some sort of unique icon to help distinguish it from other metrics
- Development should be split between in a way that front end developers can work independently from back end developers

## 6. Purchased Components

- Some sort of database server is needed to store all of the different metrics generated from the different projects

## 7. Interfaces

### **User Interface**

The user interface will be implemented over a web-based browser. The user interface should be able to support communication between the client and the server. This communication should allow for the user to send and receive data in the background without causing the page to reload constantly.