**Methodology:**

**Dataset:**

With memory, storage, and processing power increasing at a dizzying pace and with new data processing frameworks including MapReduce and Spark, researchers in the field of software engineering are able to mine larger repositories of code data. Despite this advantage, developing mining programs using frameworks such as MapReduce tends to be far too low level and time-consuming for the needs of researchers. Since much research in software evolution tends to require the use of large quantities of data, many different solutions have been developed to aid researchers in mining code repository data. One such solution, developed by researchers at Iowa State University, is a domain specific language (DSL) called Boa. By using a well-defined schema, users can write queries that can pull data from different version control systems in a uniform way [1]. Based on their schema, queries can be written that include data pertaining to software projects, repository contributors and code changes.

What makes this attractive is that, in addition to the Boa DSL, researchers at Iowa State developed a repository of GitHub data from September, 2013 that contains over 70,000 repositories [2]. While this is a large number of repositories, the data is stored on Iowa State's servers and Boa queries targeting the data are run on a Hadoop cluster. Despite the drawback of not being local, there are advantages such as being able to offload computational power and being able to submit Boa queries through a rich web interface, Eclipse plugin or Java API. As such, for this project, we propose to use Boa and its associated GitHub repository data. Based on this, this project will use the Boa DSL and the accompanying GitHub data from September 2013 to achieve the stated goals.

While Boa has many advantages that make it preferable to other GitHub data dumps, one limitation is Boa’s lack of GitHub Issue data in its GitHub dataset. Since our project relies on this issue data, we will need to overcome this problem. To do this, the project will build a simple program that uses the GitHub API to mine a repository’s Issue data. The GitHub API is service provided by GitHub that allows clients to receive desired GitHub data in the form of JSON [3]. The service is easy to access and there are many different open-source software libraries that automate parsing this data. Based on these factors, our project will obtain the Issue data using either raw JSON requests or by using an external open-source library.

**Tool:**

An extra component delivered by this project will be a software tool that will attempt to automate the methodology described above.

**Timeline:**

To achieve our objectives as stated here, this project has been split into several small milestones. Importantly, while the submission deadline for this project is due on the 29th of March, we aim to continue our work to further work on the software tool. This section highlights our proposed timeline and describes each of the milestones in detail.

*Milestone #1 (March 7th, 2016) –*

This milestone is concerned with understanding the Boa DSL and GitHub API in greater detail. In particular, we would like to work on developing queries that allow us to pull relevant data from GitHub repositories. By the end of this milestone, our project will have the Boa queries completed and will have started developing a tool that allows us to pull Issues from the GitHub API.

*Milestone #2 (March 14th, 2016)* –

In this milestone, there are two major goals that will be achieved. First, we will identify a collection of GitHub repositories that would like to analyze. This includes

**Challenges:**

As with any project, there are a variety of challenges that will need to be overcome for our project to achieve its stated goals. These challenges can be split into several categories: *data challenges* and *methodology challenges.*

Data challenges refer to problems encountered when dealing with the data itself. Since we will be dealing with GitHub repositories, there are certain considerations that need to take place when conducting the experiment. Previous research has found several challenges that can arise when mining GitHub for data. Since GitHub hosts professional open-source projects and hobbyist projects alike, most repositories tend to have very little commit activity [4]. Luckily, since our project will focus on larger, more active projects, this should not be a problem. One challenge associated with our project may come from the fact that GitHub is a pull-based development system. For instance, the Rails project has over 8,000 project forks and over 50,000 commits [4]. However, of those 50,000 commits, GitHub notes that there are only 34,000 commits to the core Rails repository, there are only 34,000 commits present. As such, this project must consider forked repositories as well to get a bigger picture of commit and contributor information.

 Methodology challenges refer to issues that are likely to be encountered when conducting the experiment. One major challenge that may occur will be linking the mined Issue data to the data collected from GitHub. For instance, it is likely that the two datasets will arrive in differing formats. This means that some sort of process that swizzles the data will have to occur so that the two collected datasets are compatible with each other. Figure 1 illustrates this problem. The data swizzling will need to take place before we can create a combined table that contains all this data.

# References

|  |  |
| --- | --- |
| [1] | R. Dyer, H. A. Nguyen, H. Rajan and T. N. Nguyen, "Boa: Ultra-Large-Scale Software Repository and Source Code Mining," *ACM Transactions on Software Engineering and Methodology,* vol. 1, no. 1, pp. 1-33, 2015. |
| [2] | Laboratory of Software Design, "The Boa Programming Guide," Iowa State University, 27 July 2015. [Online]. Available: http://boa.cs.iastate.edu/docs/index.php. [Accessed February 2016]. |
| [3] | G. Gousios and D. Spinellis, "GHTorrent: Github’s Data from a Firehose," in *Mining Software Repositories (MSR)*, Zurich, Switzerland, 2012. |
| [4] | E. Kalliamvakou, G. Gousios, K. Blincoe, L. Singer, D. M. German and D. Damian, "The Promises and Perils of Mining GitHub," in *Mining Software Repositories (MSR)*, Hyderabad, India, 2014. |