



---

# PROJECT PROPOSAL

---

Bug Report Classification



AUGUST 16, 2017

MD. HABIBUR RAHMAN  
IIT, DU

Project Proposal

SE 801

Submitted by

Md. Habibur Rahman

BSSE0623

Supervised by,

Shah Mostafa Khaled

Assistant Professor

IIT, DU

Submission Date

August 16, 2016

## Contents

Introduction.....	3
Motivation.....	3
Literature study .....	3
Proposed project.....	4
Methodology .....	4
Timeline .....	5
References .....	5

## Introduction

A bug is a defect in a system which indicates the abnormal behavior of a system. Software bugs are managed and tracked using different bug tracking tools like Bugzilla, JIRA etc. Software bug repositories contain the software bug information reported by the users. When a bug is reported the QA manager manually inspects the bug report and then he takes necessary steps like assigning to the developers, predict the severity of the bug, classify the bug report as bug or enhancement etc.

## Motivation

All bug reports reported by the users are not actually bug. A bug report can be a feature request, a change in documentation etc. which are called non-bug. The QA manager manually checks a bug report whether it is bug or non-bug. The QA manager classifies the bug report based on the bug report attributes. The attributes of bug report are title, description, assigned to, date, comment etc. Everyday a huge number of bugs are reported in bug repositories. For this, classifying huge number of bug reports by the QA manager is time consuming. An automated approach in classifying bug reports as bug and non-bug can reduce the time and effort of QA manager. But the existing bug tracking tools do not support this automation approach.

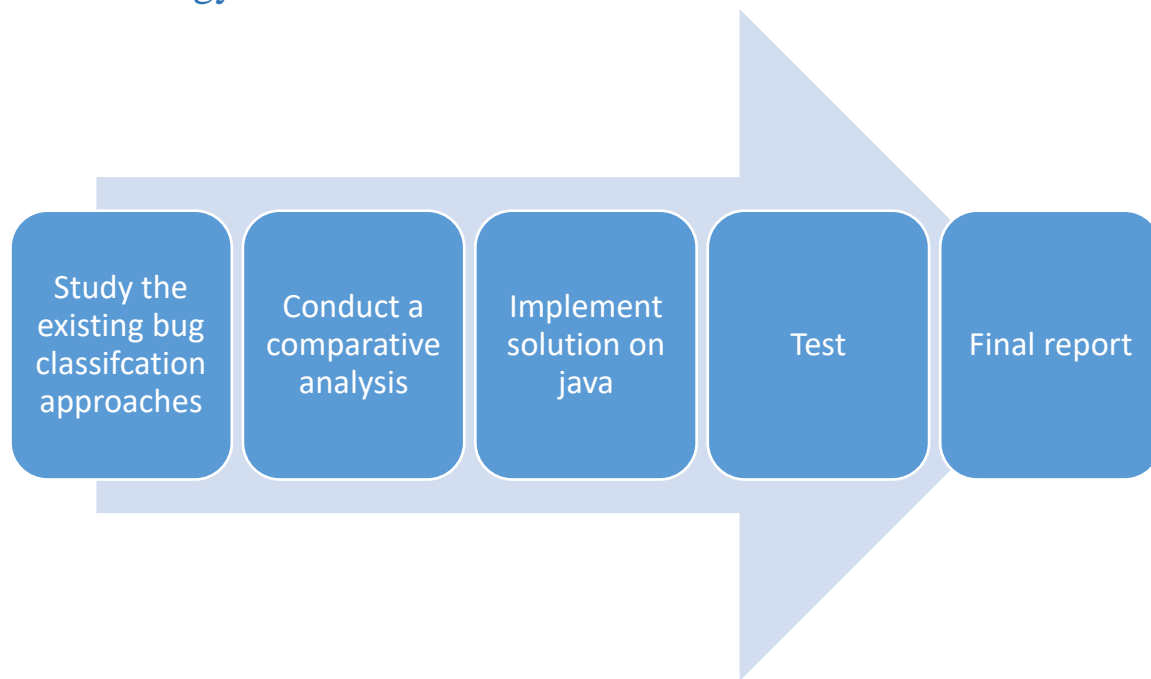
## Literature study

Recent approaches in automatically classifying bug reports as bug and non-bug are based on text mining. Antoniol et al. [1] have investigated the automatic classification of bug reports by utilizing conventional text mining techniques, which demonstrated the feasibility. They have used title and description of bug reports. They have collected about 1800 bug reports from Mozilla, Eclipse and Jboss. The approach consists of three pipeline phases. First, they have manually classified the bug reports as bug or non-bug. Second, the classifier is trained by the labeled data. Third, they have predicted the bug reports of test data using the trained classifier. The precision and recall was about 72-75% and 75% respectively. N K Nagwani and S Verma [2] proposed an approach to classify the bugs using clustering technique. Their proposed algorithm works in three major steps. In the first step text clusters are created using software bug textual attributes data and followed by the second step in which cluster labels are generated using label induction for each cluster, and in the third step, the cluster labels are mapped against the bug taxonomic terms to identify the appropriate categories of the bug clusters. J Xuan et al. [3] proposed a semi-supervised approach for automatic bug triage based on text classification. This approach trains a classifier with a fraction of labeled bug reports. Then the approach iteratively labels numerous unlabeled bug reports and trains a new classifier with labels of all the bug reports. They also employ a weighted recommendation list to boost the performance by imposing the weights of multiple developers in training the classifier.

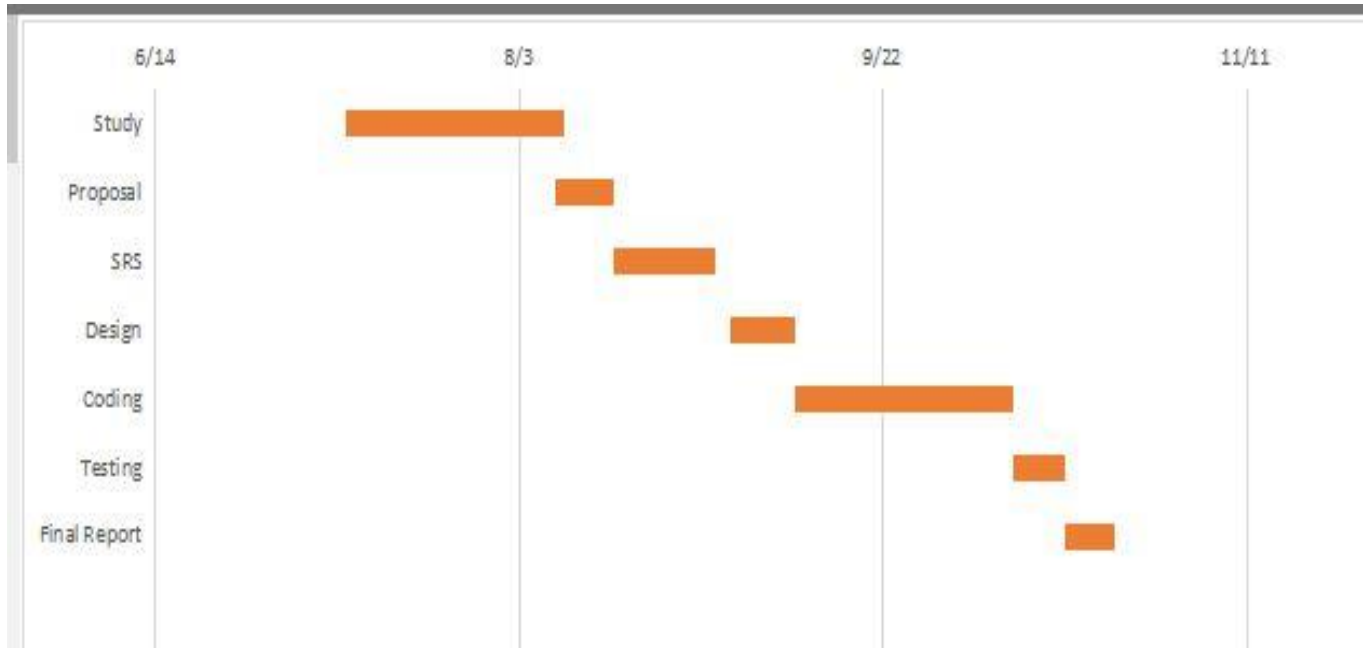
## Proposed project

The proposed project is to conduct a comparative analysis of different machine learning approaches like supervised, unsupervised and semi-supervised for bug classification. We use same dataset for this comparative analysis.

## Methodology



## Timeline



## References

- [1] Antoniol, Giuliano, et al. "Is it a bug or an enhancement? a text-based approach to classify change requests." *Proceedings of the 2008 conference of the center for advanced studies on collaborative research: meeting of minds*. ACM, 2008.
- [2] Nagwani, Naresh Kumar, and Shrish Verma. "CLUBAS: an algorithm and Java based tool for software bug classification using bug attributes similarities." *Journal of Software Engineering and Applications* 5.06 (2012): 436.
- [3] Xuan, Jifeng, et al. "Automatic bug triage using semi-supervised text classification." *arXiv preprint arXiv:1704.04769* (2017).