

Bug Triaging Based On Developers Expertise and Questioning and Answering Capability

Daruthi Rani Chauhan

RSR Rungta College of Engg. & Technology
Dept. of Computer Science and Engineering
Bhilai, Chhattisgarh, India
chauhan.nickey@gmail.com

Prof. Sachin Harne

RSR Rungta College of Engg. & Technology
Dept. of Computer Science and Engineering
Bhilai, Chhattisgarh, India
Sachin.harne2027@gmail.com

Abstract

Bugs are one of the essential issue for a product association. As indicated by explore, Software association spends more than 45 rates of their income in taking care of these bugs. To manage these bugs physically is exceptionally troublesome and furthermore blunder inclined. Along these lines, a programmed approach of occurrence determination technique and highlight choice strategy is consolidated together to deal with the bugs, at that point the bugs are dispersed to bug explaining specialists. An avoidable stride in settling the bugs is allotting a bug to fathoming master. The issue is dominant part of bugs are allocated to understanding specialists who has less involvement in that field which can leave the bugs as it may be. Thusly utilizing term determination strategy for a bug fathoming master is anticipated naturally relying on which sort of bugs it is. In this paper, we propose a novel method to find expertise developers to resolve the particular bug. Our methods combines questioning and answering ability and take into account the recent change in developer interest.

Keywords— *bug data reduction, feature selection technique, instance selection technique, prediction for reduction orders, bug triage.*

I. INTRODUCTION

Bugs are one of the essential issue for a product association. A considerable lot of the product organizations require to manage gigantic measure of programming bugs day by day. Programming bugs are inescapable and settling these product bugs is an exceptionally costly errand. Truth be told, a hefty portion of the Software association spends heaps of their assets in taking care of these bugs. Bug store are there for dealing with this bugs. Bugs vault assumes a critical part.

In programming improvement and Maintenance, a bug storehouse is an essential programming archive for putting away the bugs which has been presented by the clients.

The greater part of the product which works for an open source ventures has an open bug archive framework which permits each designers and in addition clients to submit issues or imperfections in the product that proposes conceivable arrangements and their comment on existing bug reports. The huge disadvantage is that substantial scale programming ventures are so much tremendous which makes the triaging procedure so much troublesome. The wasteful information and additionally indistinct information add repetition information to the information archive framework and make an awesome issue to the product specialists. In bug storehouse, every product bug has their own particular bug report. The bug report as a rule comprises of literary data identified with the bug and updates in regards to status of bug settling.

A bug archive gives an information arrange bolster about different sorts of errands on bugs, e.g., bug restriction, blame forecast, and revived bug examination. Tremendous programming ventures gives bug archives which is additionally called bug following frameworks that bolster data accumulation and to help the engineers to deal with bugs.

The gigantic measure of normal happening bugs for open source enormous programming ventures is so much huge that makes the triaging procedure most troublesome and testing. At the point when a bug report is framed, a human triage gives this bug to a designer, who will attempt to settle this bugs. This engineer normally recorded in a thing relegated to. The technique for giving a right designer to settling the bug is known as bug triage. Bug triage essentially is a standout amongst the most tedious stride in overseeing of bugs in

programming ventures. Physically bug triage by a human triage is extremely tedious process and mistake inclined in this way the quantity of every day bugs is enormous and absence of learning and involvement in designers about all bugs. On account of every one of these things, bug triage result in costly time misfortune, high cost and absence of precision.

Before changing and checking a bug, each bug report must be given to an important designer who could settle it. In conventional bug storehouses framework, all the tremendous measure of bugs are physically triaged by some specific designers. The point is to diminish the human work costs, some administered content arrangement approaches have been proposed for programmed bug triage. At that point the idea of the bugs is anticipated by utilizing a prescient calculation and afterward forecast of important designers for the approaching bug reports with these classifiers.

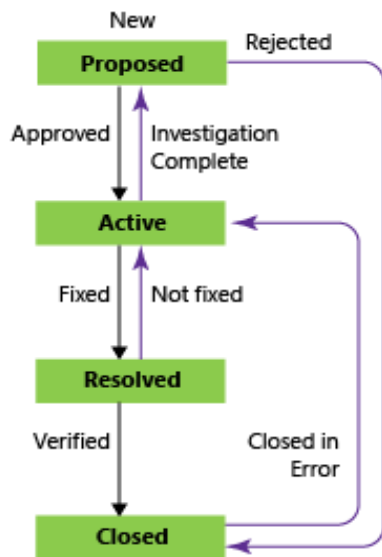


Fig.1. shows basic approach for bug triage

A bug repository provides a data stage support system about types of tasks on bugs, e.g., fault prediction, bug localization, and reopened bug analysis. Huge software projects provide bug repositories which is also called as bug tracking systems which helps to support information collection and to assist developers to handle bugs.

II. LITERATURE SURVEY

B Jifeng Xuan[1], Author focused Towards Effective Troubleshooting with Data Truncation deals with reducing the data present in the bug repository and improve the quality of data then reduce time and cost of

bug triaging, it represent an automatic approach to predict a developer with relevant experience to solve the new coming report.. The bug data sets are obtained and techniques such as instance selection feature selection are applied simultaneously.

Suvarnaa Kale [2], focused on A Technique to Combine Feature Selection with Instance Selection for Effective Bug Triage .It addresses the issue of data reduction for bug triage by text classification techniques. Conventional software analysis is not totally suitable for the large-scale and complex data in software repositories. Data mining has developed as a promising means to handle software data. There are two difficulties related to bug data that may influence the effective use of bug repositories in software development tasks, namely the huge scale and the low quality. Therefore unfixed bugs are deleted from the bug repositories.

D. Cubranic [3], focused on Automatic Bug Triage using Semi-Supervised Text Classification propose a semi-supervised text classification approach for bug triage to avoid the deficiency of labeled bug reports in existing supervised approaches. This approach combines naive bayes classifier and expectation maximization to take advantage of both labeled and unlabeled bug reports. 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. Then it employs a weighted recommendation list to boost the performance by imposing the weights of multiple developers in training the classifier. Before training a supervised classifier for bug triage, a necessary step is to collect numerous labeled bug reports, which are bug reports marked with their relevant developers. The semi supervised text classification approach to improve the classification accuracy of bug triage. This semi supervised approach enhances a NB classifier by applying expectation-maximization (EM) based on the combination of unlabeled and labeled bug reports. First, this semi-supervised approach trains a classifier with labeled bug reports.

Phuc Nhan Minh [4], focused on An Approach to Detecting Duplicate Bug Reports using N-gram Features and Cluster Chrinkage Technique to duplication detection which is an important issue for software maintenance in recent years. In this study, we propose a

detection scheme using n-gram features and the cluster shrinkage technique. From the empirical experiments on three open source software projects, the proposed scheme shows its effectiveness in duplication detection.

Anjali [5], focused on Bug Triaging: Profile Oriented Developer Recommendation. Author proposed a Domain Mapping Matrix (DMM) based developer recommendation approach for predicting the best suited developers list who could resolve the newly reported bugs. Unlike other approaches, our approach does not find a matching with historical bug reports and recommend the developers who fixed historical bug report; rather, it utilizes the expertise profile of developers maintained in DMM. our proposed approach uses a wider area for token matching.

III. METHODOLOGY

This section presents proposed workflow of system architecture in detail. Fig. 2. Shows the proposed architecture.

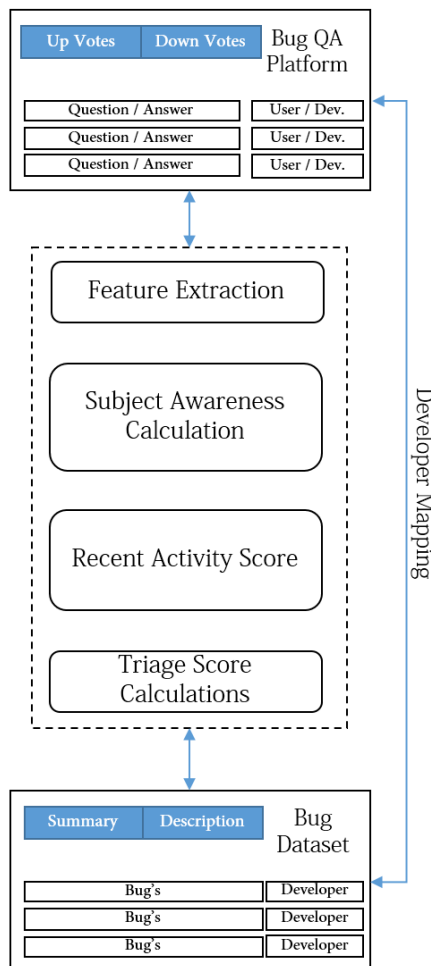


Fig. 2. Proposed System Architecture

There are several modules:

1. Q and A Platform
2. Feature Extraction
3. Subject Awareness Calculation
4. Recent Activity Score
5. Triage Score

A. Q&A Platform

The Q&A platform, where users from around the world ask technical questions and developers and users try to answers the questions. It just a collection of Q and A.

B. Feature Extraction:

There are various features which are need to be extracted from the question and answers of the users. These features may be the extraction of question tags, up votes, down votes etc.

C. Subject Awareness Score

This metrics is the calculation of individual subject score. It is used to find the developers area of interest in particular language. The score is generated based on the answer and question he/she asked in Q&A platform. It is given by formula

$$SA - Score_{user,bug} = \frac{(A_{user,bug} - Q_{user,bug})}{\sqrt{(A_{user,bug} - Q_{user,bug})}}$$

D. Recent Activity Score

The recent activity is used to calculate the recent expertise of the user. The expertise may be change during the time so it need to be updated every time when expertise changes. It is given by equation

$$RecentActivityScore_{user,bug} = \sum_{i \in bugs\ of\ u} \frac{1}{1 + |\{d \mid d \in bugs\ of\ user \wedge t_d > t_i \wedge t_d < t_b\}|}$$

E. Triage Score

Based on the triage score the developer is assigned to particular bug. The higher is the triage score for particular language the more will be the developer expertise in that particular field. The triage score is calculated by:

$$TriageScore_{user,bug} = \alpha \cdot (SA - Score_{user,bug}) + \beta \cdot (RecentActivityScore_{user,bug})$$

IV. RESULTS

We have performed experiment using Stack overflow dataset. The dataset is available freely. We have also considered Bit Bucket website for finding developers and assigning bugs to them. The input file of Q&A is shown in fig. 3. Various equations presented at above section are applied to the dataset and finally the developers Q score, A score, SA Score, RA Score are calculated. The output are presented in fig. 4 to 6.

```

Id PostTypeId OwnerUserId ParentId Score Tags CreationDate
AnswerCount Title
9 1 1 747 [c#;.net;.datetime] 2008-07-31T23:40:59.743 44 How
do I calculate someone's age in C#?
11 1 1 749 [datediff] 2008-07-31T23:55:37.967 31 How do I
calculate relative time?
19 1 13 143 [performance;.algorithm;.language-agnostic;.unix;.pi]
2008-08-01T05:21:22.257 22 What is the fastest way to get the value of ?
88 1 61 46 [linux;.winapi;.visual-c++;.unix;.timer]
2008-08-01T14:36:18.640 10 Is gettimeofday() guaranteed to be of
microsecond resolution?
234 1 91 29 [mysql;.sql-server] 2008-08-01T22:13:34.100 4 SQL
Server 2005 implementation of MySQL REPLACE INTO?
263 1 61 13 [c++;.c;.linux;.gtk;.x11] 2008-08-01T23:27:24.993 3
GTK implementation of MessageBox
264 1 104 11 [c++;.berkeley-db] 2008-08-01T23:28:51.577 5
BerkeleyDB Concurrency
330 1 63 16 [c++;.oop;.class;.nested-class] 2008-08-02T02:51:36.470
10 Should I use nested classes in this case?
337 1 111 35 [python;.xml] 2008-08-02T03:35:55.697 16 XML
Processing in Python
387 1 134 20 [search;.lucene] 2008-08-02T09:28:17.683 3 Lucene
Score results
502 1 147 17 [python;.windows;.image;.pdf] 2008-08-02T17:01:58.500
5 Get a preview JPEG of a PDF on Windows?
514 1 151 10 [ruby-on-rails;.ruby;.crash] 2008-08-02T17:26:39.793
5 Frequent SystemExit in Ruby when making HTTP calls
626 1 136 207 [ruby;.lambda;.proc] 2008-08-03T06:40:54.120 12 When
to use lambda, when to use Proc.new?
698 1 173 124 [python;.ide;.code-completion;.wing-ide]
2008-08-03T14:22:28.963 51 Is there an IDE that provides code completion
for Python?
833 1 186 6 [sql-server;.database] 2008-08-03T21:23:41.077 7
Editing database records by multiple users
    
```

Fig. 3. Shows the snapshot of SO dataset

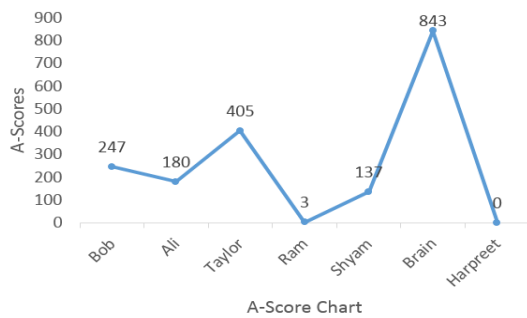


Fig.4. Shows are A-Score Calculation of Developers

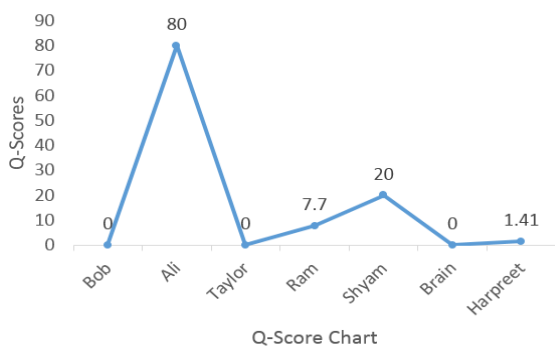


Fig.5. Shows are A-Score Calculation of Developers

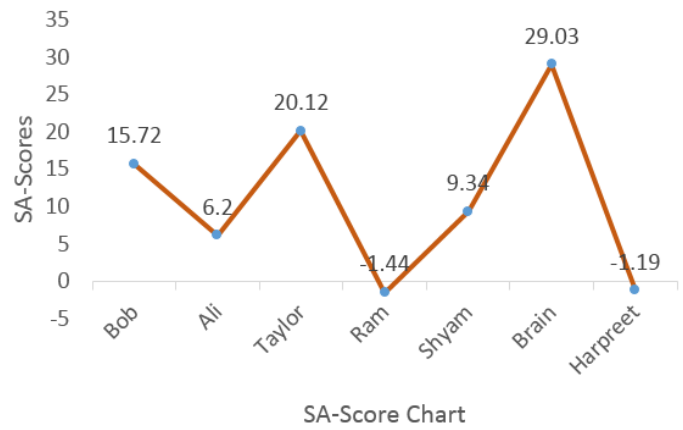


Fig.6. Shows are A-Score Calculation of Developers

V. CONCLUSION

In this paper we have proposed a novel mechanism for finding expertise developer to solve particular bug. The bug are given only to those developer who are expertise and has interest to solve it.

The SA Score, Q Score, A Score are used to calculate the expertise subject of the developers. The recent activity is also tracked so that if there is any change in the developer interest, it should also be accounted.

REFERENCES

- [1] B Jifeng Xuan, He Jiang, Yan Hu, Zhilei Ren, Weiqin Zou, Zhongxuan Luo, and Xindong Wu, "Towards Effective Bug Triage with Software Data Reduction Techniques" *iee transactions on knowledge and data engineering*, vol. 27, no. 1, january 2015.
- [2] Suvarnaa Kale, Ajay Kumar Gupta, "A Technique to Combine Feature Selection with Instance Selection for Effective Bug Triage", "International Journal of Science and Research (IJSR) ISSN (Online): 2319-7064".
- [3] D. Cubranic and G. C. Murphy , "Automatic bugs triaging usings text categorization," in Proc Sixteenth International Conference on Software Engineering, Citeseer, 2004, pp. 92–97.
- [4] Phuc Nhan Minh, "An Approach to Detecting Duplicate Bug Reports using Ngram Features and Cluster Chrinkage Technique", International Journal

of Scientific and Research Publications, Volume 4, Issue 5, May 2014, ISSN 2250-3153.

- [5] Anjali, Sandeep Kumar Singh, “Bug Triaging: Profile Oriented Developer Recommendation”, International Journal of Innovative Research in Advanced Engineering (IJIRAE), ISSN: 2349-2163, Volume 2 Issue 1 (January 2015).
- [6] D. Matter, A. Kuhn, and O. Nierstrasz, “Assigning bug reports using a vocabulary-based expertise model of developers,” in Mining Software Repositories, 2009. MSR '09. 6th IEEE International Working Conference on, May 2009, pp. 131–140.
- [7] T. T. Nguyen, A. T. Nguyen, and T. N. Nguyen, “Topic-based, timeaware bug assignment,” SIGSOFT Softw. Eng. Notes, vol. 39, no. 1, pp. 1–4, Feb. 2014.
- [8] M. K. Hossen, H. Kagdi, and D. Poshyvanyk, “Amalgamating source code authors, maintainers, and change proneness to triage change requests,” in Proceedings of the 22nd ICPC. ACM, 2014, pp. 130–141.
- [9] H. Kagdi, M. Gethers, D. Poshyvanyk, and M. Hammad, “Assigning change requests to software developers,” Journal of Software: Evolution and Process, vol. 24, no. 1, pp. 3–33, 2012.
- [10] M. Bahrami Zanjani, K. Huzefa, and C. Bird, “Using DeveloperInteraction Trails to Triage Change Requests,” in MSR. IEEE, 2015