International Journal of Engineering Research in Computer Science and Engineering (IJERCSE) Vol 2, Issue 4, April 2015

Search-Based Duplicate Defect Detection

[1]Ajit Padwal,[2]Ashish Kane,[3]Prof . Harish Barapatre
Dept. of Computer Engineering, University of Mumbai
Yadavrao Tasgaonkar Institution of Engineering Technology
[1]login2ajit@gmail.com,[2]kane.ashish@gmail.com,[3]harish.barapatre@tasgaonkartech.com

Abstract: **Redundancy is major concern in any computational or development system, Specifically redundant defect is a key challenge in any database oriented system. If any system contains redundancy then its processing overhead is automatically increases which will affect overall efficiency & performance of a system. Removal of such redundancy using manual approach is time consuming & not up to the mark. Therefore there is necessity of some automatic & efficient tool for searching the this kind of redundant defect ,which in turn saves time of developer to reprocess the same defects or Defect repeatedly. This idea made us to develop search based duplicate defect or Defect detection system.**

I. INTRODUCTION

Any software project is said to be successful only if it totally Defect free & user friendly i.e. Easy to use for end user. This Defect can be at large level or even tiny level but its impact to system is always hazardous to its overall performance.

Defect detection & reporting of Defect to developer is key job of tester. In an organization a testing team do this job, due to presence of multiple tester there is more chances of reporting the same Defect repeatedly. Thus for efficient system its prior task to identify the duplicate defects or Defect, so we can save developer time to process the same Defect. So preventing occurrence of such duplicate defect or Defect is key challenge for any software system.

So our systems main aim is to track the redundant defect or Defect & filter out them from overall Defect, thus the final Defect list will be free from duplicate Defect which will be reported to developer. Thus it will save overall cost & time to process those redundant defects.

Reporting the same Defect repeatedly by different manner is some time gives more key information to solve that Defect easily but this benefit comes with the expense of high cost & large processing time. Thus overall system throughput will automatically degrades drastically.

Thus our system main focus is to make such powerful tool which will restrict the redundant defects & improve the system performances. If any tester submits the same Defect then our system should automatically detect it & filter out the defect before it reaches to developer for processing.

II. DUPLICATE DEFECT DETECTION

A Duplicate Defect Detection system or defect tracking system is a system which is used to trace the path of reported Project Defect in software development process. A important element of a Duplicate Defect Detection system is a database which maintain the details of defects. Details like defect category, location of defect and the timestamp of defect reporting , its severity. As well as the detailing about the tester who detect and report about the defect to the developer.

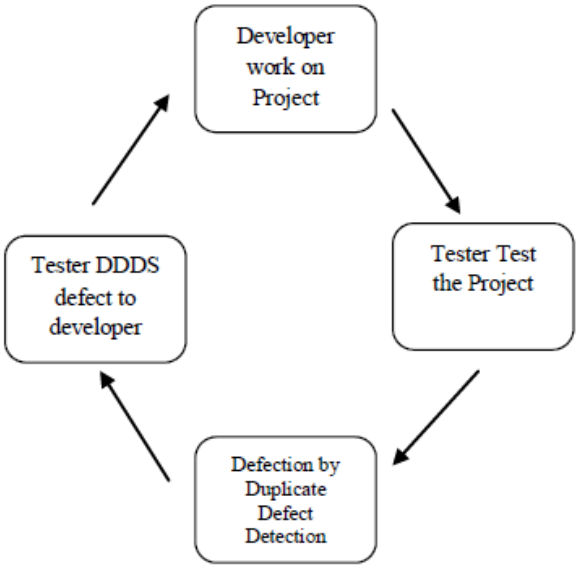

Figure 2.1 Test Processing Cycle

Work flow of system :
A tester find a Defect and report it

- Developer Develop the application and pass it to the Team Leader

- The Team Leader assign the work to the respective tester .

- Tester search for defect in given application.
- The tester will pass this defect list to Duplicate Defect Detection system
- If system report this defect as a genuine defect then it will pass to developer for resolving

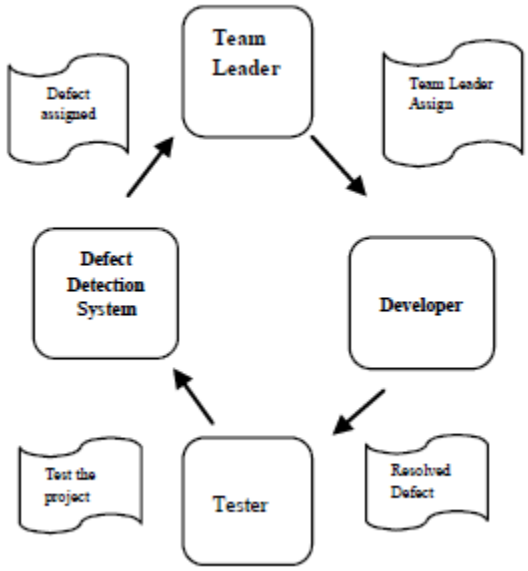

Figure 2.2 Duplicate Defect Detection Workflow

III. PROPOSED DUPLICATE DEFECT DETECTION SYSTEM

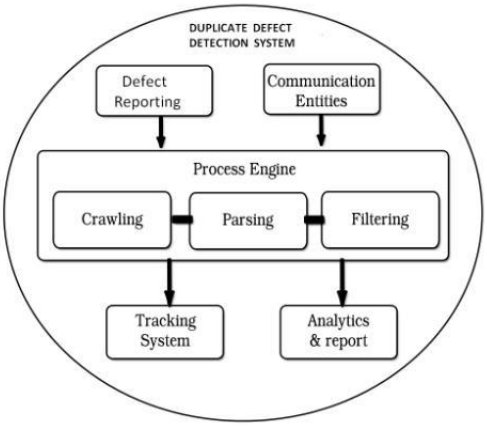

Fig.3.2. Proposed system

We describe the overall process for extracting data from Duplicate Defect Detection systems as follows –

1. Defect reporting
2. Communication elements
3. Processing element

Crawling

Parsing

Filtering

4. Tracking system
5. Analytics and report

3.2.1 Defect reporting

This is the main module of Duplicate Defect Detection system. Tester reports the Defect in the system. Duplicate Defect Detection system has the facility to add the Defect i.e. reporting of Defect. Reporting of Defect consist of following things –

Defect type

Priority

Defect category

Description

Related attachments

Location/ module of the system

3.2.2. Communication entities

This is also most important module of Duplicate Defect Detection system. This very helpful to track the Defect status. Communication entities consist of who is involved in the current product testing i.e. developer list, testers list and Team Leader.

Following list represent entities –

1. Testers
2. Developers
3. Team Leader
4. End users

3.3.3 Process Engine

Process engine is main module of Duplicate Defect Detection system. Process engine consist of Crawling, Parsing and filtering modules in system. Crawling gets the Defect list and classifies it by using Naïve Bayes classifier and removes the duplicate. Crawling returns or maintains result in JSON format that need to parse it by JSON Parser. Here Parser module of process engine parse it accordingly and provides data to filtering module.

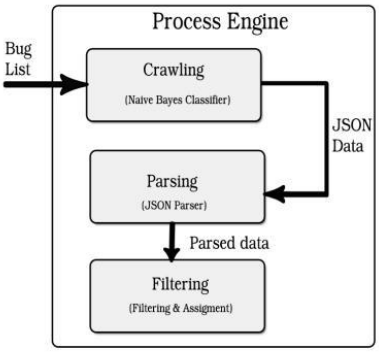

FIG.3.3 **PROCESS ENGINE**

Filtering module gets parsed data from parser module then it filters and sorts that data according to priority and make ready to assign to respective entities.

3.3.1 Crawling:-

Process engine consists of three different main modules such as crawling, parsing and filtering. Crawling is first and very important module of Duplicate Defect Detection system. Crawling takes Defect list as input and returns JSON data as output. It works on Defect list, which consist of all reporting data. There may be possibility that tester can make the mistake while reporting the Defect. So system should be intelligent that it need to be classifying Defect by using classification technique. Here we propose Naïve Bayes classifier to classify the Defect perfectly. Naïve Bayes best classifier than other classifiers such as decision tree, neural network and many more. Naive Bayes is conditional probability based classifier.

Naïve Bayes:-A naive Bayes classifier is a simple probabilistic classifier based on applying Bayes' theorem with strong (naive) independenceassumptions. A more expressive term for the underlying probability model would be "independent feature model. Plus point of Naive Bayes is that, it only requires a small amount of training data to estimate the parameters necessary for classification. Because independent variables are assumed, only the variances of the variables for each class need to be determined and not the entire covariance matrix. According to Bayesian theorem the formula in plain English is as follows –Mathematically it is represented as

Let $\mathbf{Y_1, Y_2, Y_3 \ldots . Y_n}$ be a partition of Ω (space) such that P $(\mathbf{Y_n}) \neq 0$ for any
n = 1, 2, 3… and let P (X) ≠ 0.Then,

$$P(X|Yn) = \frac{P(Yn \mid X)P\ (Yn)}{\sum P(Yn|X)\ P(Yn)}$$

Where, n =1, 2, 3, 4…

By using Bayesian network and feature variable Naïve Bayes classifies the data.

3.3.2. Parsing:-

Crawling returns result in JSON format after classification by Naïve Bayes theorem. So that JSON need to parse for reading of data. Following snippet represent JSON code –

```
[
  {
   name: "object1",
   message: "Greetings from object1",
  },
  {
   name: "object2",
   message: "Greetings from object2"
  },
  {
   name: "object3",
   message: "Greetings from object3"
   code: alert("This will be executed when evaluated!")
  }
]
```

Json (javascript object notation) is a trivial data-interchange format. Json is a text format that is completely language independent but uses conventions that are familiar to programmers of the c-family of languages, including c, c++, c#, java, javascript, perl, python, and many others.

3.3.2. Analytics and report:-

Analytics and report gives overview of the current testing project. It may represent graphs, statistical reports and simple reports. Analytics refers to the skills, technologies, applications and practices for continuous iterative exploration and investigation of past business performance to gain insight and drive business planning. Business analytics focuses on developing new insights and understanding of business performance based on data and statistical methods. In contrast, business intelligence traditionally focuses on using a consistent set of metrics to both measure past performance and guide business planning, which is also based on data and statistical methods. It also helps to work out the project development track. Tracks time, resource money utilizations.

CONCLUSION

We proposed a Defect rule based Defect report classification technique with feedback for duplicate Identification and Defect reports retrieval. By performing textual analysis, clustering and classification, the Defect reports which are crawled in Mantis BT have been classified into a taxonomy structure according to 4C model. In order to improve the accuracy of duplicate identification and Defect report retrieval, we applied the feedback process to rate these Defect reports. Developers decide whether a Defect report is

a duplicate, the report is valid and it belongs to an appropriate Defect category in the feedback process.

REFERENCES:-

[1]Yongsoo Yuk, Woosung Jung, Comparison of Extraction Methods for Duplicate Defect Detection System Analysis, IEEE, 2013

[2] Search Based Duplicate Defect Detection An Industrial Experience MehdiAmoui, NilamKaushik, Abraham IEEE, 2013

[3]Zatul Amilah Shaffiei, Mudiana Mokhsin and Saidatul Rahah Hamidi, Change and Duplicate Defect Detection System: Anjung Penchala Sdn. Bhd., International Journal of Computer Applications (0975 – 8887), Volume 10– No.3, November 2010http://en.wikipedia.org/wiki/Mantis_ Defect_Tracke

[4]Thomas Zimmermann, Rahul Premraj, Jonathan Sillito and Silvia Breu, Improving Duplicate Defect Detection Systems, IEEE 2009.