IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 9, Issue 1 (Jan. - Feb. 2013), PP 26-29
www.iosrjournals.org

A cost reduction method in SQA by using inspection and static analysis tool

Dr. Md Abul Kashem, Rowsan Jahan Bhuiyan
Associate Professor, Dept. of Computer Science and Engineering, Dhaka University of Engineering and Technology, Gazipur-1700, Bangladesh.
M.Sc in Engineering Programme, Dept. of Computer Science and Engineering, Dhaka University of Engineering and Technology, Gazipur-1700, Bangladesh.

Abstract: *Quality and cost have close relationship in software development process. That's why quality has to be better and cost must to be lower for a software developer organization. In our paper, we are presenting an approach that will give details analysis how cost can be reduced by detect defection in code using static analysis tool and formal inspection together with. Our proposed method will help software industry to think acutely about cost reduction by defect detection in code.*

Keywords*- Cost reduction, Code inspection Quality assurance, Static analysis tool, inspection.*

I. Introduction

How well a software is developed can be measured by quality of a software. In this paper, we will explain how proper code inspection techniques decrease considerable amount of quality assurance cost. KI. Emam, Briand and Laitenberger [1] already demonstrate that a software design inspection can reduce 44% (forty four percent) of the usual testing cost. They also proved that accurate code inspection can save on average 39% (thirty nine percent) of the total QA cost [1]. It has stated by Boehm and Basili [2] that it is100 times more economical to detect and remove a fault in design and coding phase than to detect and correct it after shippment of the product to customer.

Any Organization's business plan is to reduce costs and to increase profit. By using inspection this aspiration can be fulfilled. Generally, a product is developed and used for different business motivation and business depends on product quality. Our paper has identified the cost affecting factor when inspection is not used and it also has described a new method to reduce the cost by using inspection combinely with static analysis tool to detect fault in code.

II. Related Work

A defect detectiontechnique in code [3 and 4] has developed by Barry Boehm and S. Chulani. They also revealed the removal techniques.According to these author , Various kind of defect can be detect at different section of development life cycle, They could be in requisites, design, code, documentation and other phases. Another model named PAF [5] by Mandeville has illustrated software quality costs by defect detection . Software quality economics [6] by Humphrey has also Introduced to reduced QA cost by using inspection. The more conventional inspection technique [7] presented by Fagan , Gilb and Graham to utilize inspection as a new cost effective tool for defect detection. They proposed the initial approach for a formal inspection process. Another important related work is Static analysis tools [8]. They are class of programs which discover defects in code. It works like a compiler. Static analysis is a decisive component of an entire quality development process. It helps developers to check and remove defects by using a lot of rules to find out code error. It helps to achieve reliability and security of a development process.

III. Quality Assurance By Inspection

Inspections techniques have initiated about 30 years ago and from that time they are applied to defect detection and enhance software quality. Proper inspection activity decline considerable amount of time and also cut huge amount of development costs. Inspection is very essential part to discover faults in a software development process. A Quality cost study [6] has showed that the manufacturers spend more than 50% (fifty percent) of their development effort for testing and analytical SQA also confirmed about this amount. When a test process encounters a failure, extensive efforts need to waste for finding the fault. But in the time of inspection these Faults can be detected directly without wasting of time and testing cost. Additionally, rework expenses can be reduced remarkably by inspection. The inspection activities generally consist of some sections. They are planning, defect discovery, bind the defect, and correction of the defect. The inspection tasks generally done by inspectors or software developers or in a group of meeting. Inspection part exploit 10–20% of overall

development effort of software, it is stated by Industrial knowledge [9]. But the cost is lower than rework and re-tests. Inspection should be applied in the beginning of the development stages.

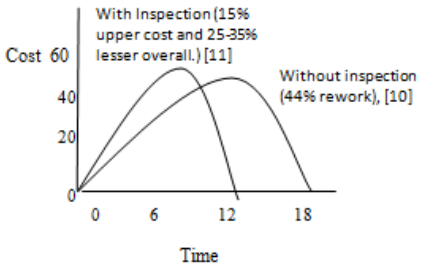


Figure -1 software development project (With and Without Inspection)

Fig- 1 shows cost and time agenda with and without inspections of a software development project .The higer curve illustrates the increasing cost and time in the earlier of the development process when inspection is used but after a certain period the higher curve falls down rapidly. It means the cost and time also reduced after inspection . The wider curve point to the project area where inspection is not used. At the begning of development process, the cost and time is lower but it is incrased and caused more time and cost because rework and retest needed when inspection are not used.The amount of Rework is 44% while inspection is not used. Overall development cost of inspected area is about 30% less than without inspection area of the curve.

IV. Proposed Method

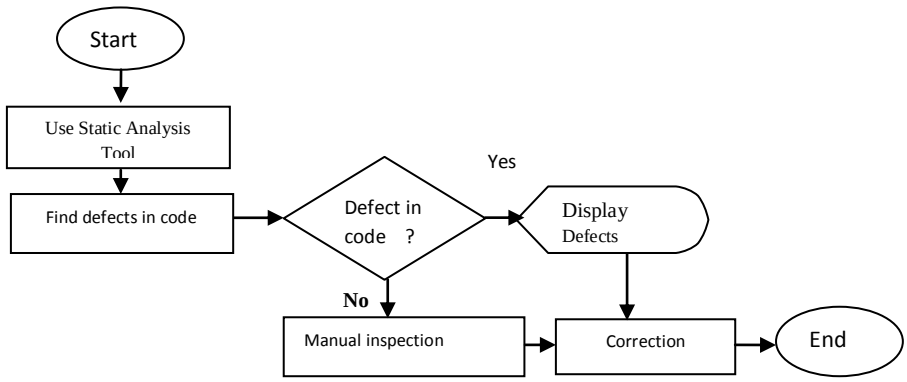
In our paper, we have proposed to use static analysis tool [8] and manual inspection [7] combinely to detect the defection of code.

After completion of coding Static analysis tool will find defects in code . It will concern if any part of code become complex . It uses different techniques to discover critical code position. One is to find out usual bug pattern that are obtain from knowledge and experiences and recognized drawback in a particular programming language. Additionally, coding rule and principles can be verified by static analysis tool. The dataflow and control flow analysis method are also used by SAT.

If static analysis tool find out defects in code it will display and will suggest for correction.

In the second step, formal code inspection method [7] will check for expected quality of detailed design, coding standards, and accuracy. This method will give attention on data definitions, relationship of parameters, control logic, various interfaces like internal or external. It also will find out if there are any execution and storage issues. If the process detect any violation of a coding standard and design or any piece of code that is unable to compile, the list of defect will be prepared . After that the correction phases will correct the defect and the next process will go on.

4.1 Flowchart of Proposed Method



4.2 Performance Evaluation

Example of a data sorting Code [12]

```
1    #include <iostream.h>
2
3    const int TABLE1 = 50;
4
```

```
5     void swap(int& x, int& y)
6     {
7        int t = x;
8        x = y;
9        y = t;
10    }
11
12    int max(int x, int y)
13    {if (x > y) return x; else return y;}
14
15    int main()
16    {
17       int size;
18       int table[TABLE1];
19
20       cin >> size;
21
22       if(size >= TABLE1)
23     cout << "A lot of elements, maximum is " << TABLE1 << endl;
24       else {
25          for(int i = 0; i < size; i++){
26             cout << "Data " << i << ";
27             cin >> table[i];
28          }
29          for(i = size - 1; i > 0; i--)
30             for(int j = 0; j <=i - 1; j++)
31                if(table[j] > table[j+1])
32                   swap(table[j], table[j+1]);
33
34          cout << endl << "Sorted lits" << endl;
35          for(i = 0; i < size; i++)
36          cout << "Data  " << i << " " << table[i] << endl;
37       }
38       return(0);    }
```

The code is made for simple sorting program. After using our code inspection method the defect report will be prepared.

Table 1: Report of code inspection

Defect no.	By	Report	Position	Type	Brutality
1	Static Analysis tool	Parameters are not passed by reference. "swap" function doesn't properly swap the numbers .	Function swap() , Line 5	Function call	Failure
2	Formal Inspection + SAT	"list" is written wrongly into message.	Function main(), line 34	Format of output	Displays of wrong output or output quality will be bad
3	Formal Inspection	Function max() is never used. But defined,	Function max(), line 12	Function calls	Violation of checklist. Minor brutality.
4	Static Analysis tool	Should be use of only >. The program allow one less than the true maximum number of elements.	Function main(), line 22	Comparisons	Sometimes identify as error.

From the above Table, we can see that Combination of static analysis tool and formal code inspection detect error in code and suggest for correction. It saves a considerable amount of test time and cost. Rework

overhead also can be reduced by this method because defect cannot propagate to the next phases. The final outcome is increasing the quality of software.

V. Conclusion

In this paper, we have presented a model for defect detection in code, which will increase quality and decrease QA cost of a software. Even every phase should be inspected properly. Otherwise the defect will be propagated to the next phases in software development process and they will be reason for more cost. Thus, overall quality will be affected. We intend to implement and integrate our model in future with other existing inspection techniques.

REFERENCES

[1] Laitenberger, L. Briand, and K. Emam, C.1998. *Simulation to Build Inspection Efficiency for development Projects*,
[2] Boehm and Basili .2001. *Top 10 List of Defect Reduction in Software*. *IEEE Computer*.
[3] Barry Boehm and S. Chulani, " *Modeling Software Defect Introduction*", University of Southern California .
[4] S. Chulani, 1999. " *Software Cost and Quality Models in Bayesian Analysis*". PhD thesis. University of Southern California.
[5] W. Mandeville , J. "*Costs of quality in software* ", International Journal of IEEE.
[6] W. S. Humphrey , " *Discipline of Software Engineering*, Addison-Wesley".
[7] Fagan, M. E. "*Inspections of code and design for error reduction in program development*. IBM Systems"
[8] Reasoning Inc. "*Automated Software Inspection: A New Approach to Increase Software Quality and Productivity*".
[9] O.Laitenberger, T.Beil, and Schwinn, "*A case study to inspect a non conventional Inspection for Requirements Specifications".*
[10] Boehm Barry W.1987. "*Improving Software Productivity*", IEEE Computer.
[11] Fagan, Michael E. "*Advance of Software Inspections*," Software Engineering, IEEE, 1986.

Chapters in Books:

[12] Ge Fan, Fang Fang and Stacy Lacy *" Using Formal Inspections in Software Quality Assurance" chapter 3,Web Book , 1999.*