# **RESEARCH ARTICLE**

## **OPEN ACCESS**

# **Software Engineering: Re-Engineering Metrics**

Satya Sobhan Panigrahi<sup>1</sup>, Pappu Sharada<sup>2</sup>, Dr,Sujit Panda<sup>3</sup>

<sup>1,3</sup>Associate Professor, Department of Computer Science Engineering, Gandhi Institute For Technology (GIFT), Bhubaneswar

<sup>2</sup>Assistant Professor, Department of Computer Science Engineering, Gandhi Engineering College, Bhubaneswar

## ABSTRACT

Software Reengineering is a term which is used to analysis of legacy system or an existing software. The Reengineering en compasses of other processses such as Forward and Reverse Engineering. This approach to understand the existing over all structure and then we have to compute the optimal metrics of cost and quality to related software system .Reverse engineering is an efficient technique to find the best documentation approach of any software .Inthis paper, we explore how different metrics related to differentarea ofsoftwareproject, can be use ful to manage thes of tware with in our requirement .The focus of our requirement is ability tomeet the target value which is our propose to cost ofproject

Keywords:Reengineering, Legacy system, Cost and Quality Management

## I. INTRODUCTION

Reengineeringisindemandsonbusinessfunctionsandin formation technology that supports them is changing at a pace that puts commercial organization. Reengineering strives away to break from oldrules and organize the changes in theiroldrulesaccordingtonewtechnologies, reenginee ringisperformedindifferentlevelsofsoftwaresystem.a ndqualityaccordingtotimewehaveobtainanoptimalp ointatwhichwehaveproposedthatsoftwareforwebsite withminimum cost and quality. This requirementachie veaccordingtocustomersatisfaction.Byusingallinter nalandexternaleffectsconsideredinsoftware, weestab lish we bsites of tware with customer satisfaction is calledReverseengineering.Now,wehaveconsideredfromi nitialpoint and obtain product by formulating all key factors that effect the cost and quality of product according to time.

- At organizationallevel
- At softwarelevel
- Hardwarelevel
- Custom built level
- Commercial level

At these levels, maintenance cost of reengineering depends on de composition of system. Thenumber of differenttasksset-tasks, milestones,workproductsandqualityassurancepointsar emeasure due to completion of project. After the completion of project, if any error, bugs etc are finding at completion time we have to use reverse engineering and follow at which stage system is unreliable.

#### II. PROPOSEDAPPROACH

Due to Forward and Reverse Engineering used in software system, reverseeng ineering approach is thatfromwhichwehavetostudy at which stage or point we have justify our cost and quality of software according totime.

#### A. Example

Mostenterpriseshavelimitedresearchandde velopmentrequiring software development. Development is the issue to initially find plan and design for new products. Different enterprises have different plan for different products that arises according to time .Software like Microsoft Word Excel, Power point etc built according to their quality and cost. If any customer wants software or product for Web sites differently rather than other internal used software. Some items such as initial graphics are covered but others are not. The discussion related to web site cost and quality is contained in separate report. The cost of the web site is minimum and quality tends to be an efficient. If we increase the functionality with up gradation of time, cost will be high. For such software, we have considered only initial use, not covers the modification. By formulating the metrics of cost

International Journal of Engineering Research and Application www.ijera.com ISSN : 2248-9622, Vol. 6, Issue 9, (Part -5) Sepamber 2016, pp.84-87



Fig. 1: Optimum Solution Curve for Two Variable Function

#### 1. How to find the OptimalSolution

The plane x=a is vertical and parallel to the yzplane. Thus, the crosssectionwithx=aisthelinez=1+a-

ywhichslopesdownward in the y-direction. Similarly, the plane y=b is the line z=1+x-b which slopes upward in the x-direction. Since all the cross-sections are lines, you might expect the graph to be a flat plane, sloping down in the y-direction and up in the x-direction. This is indeed the case.





Fig. 2: The Optimum Solution Will be a Line and not a Point

# **III. IMPLEMENTATION**

By using Reverse engineering, we have to detect the bad effects andsmellsin Lehman's LawusingwithSOMOXandRECLIPSE tools. We have to use the matrix and GUI interpretation for detecting the bad smells in softwareproduct.

### A. Matrix withSOMOX

We use tool SOMOX (software model extractor) that calculates the number of metrics such as name resemblance, coupling and cohesion. These values are applied into matrix form. Thematrix from these elements proceeds to calculate new matrix valuesfor detected basic component and continues together intoc omposite element. The results of these matrixes are implemented into the mapping format of X-axis and Y-axis.

#### B. GUI with Reclipse

Reclipsetoolisusedforpatterndetectionwhichwa sdevelopedatuniversityofpadeborn.Itoffersthestatist icalstructureofdetected elements into the graphical notation. With the help of Reclipse toolwedraw the detectedmatrixinto

graphicalstructureandfindthebadsmells,duetodetecti ngthesebadsmellswehaveto recover it by using restricting. Restructing is a technique which helps to maintain only detected values, not the overall structure.

WehavetoapplythesetoolswiththehelpofReverseengi neering in Lehman'sLaw.

Lehman's Law: Lehman and Belady identified several laws of

system change.

## 1. ContinuingChange

Aprogramthatisusedinrealworldenvironmentsmustc hange,orbecomeprogressively less useful in that environment.

#### 2. IncreasingComplexity

As a program evolves, it become more complex, extra resourcesare needed to preserve and simplify its structure.

ByusingthetoolsofSOMOXandRECLIPSEinLehma n'sLaw,wehavetodetectbadsmellsanddetectedoccurr encesvaluesgive authenticate results.





Fig. 3: Proposed Approach

## **IV. OBJECTIVES**

- To achieve the optimal solution related to metrics of cost and quality.
- Atthatoptimalpoint, we have to develop as of tware product with customersatisfaction.
- The software model extractor (SOMOX) tool help to built matrix which is easily understandable and detect the bad smell.
- Frompatterndetectiontool, we have to easily detect at which documentation point error is occurred.
- We have easily designed a pattern for a softwareproduct.
- Cost of data conversion activities except for costs incurred to develop or maintain software that allows conversion of data.

#### A. My Method of Software Re-Engineering

On the right hand we take the software to re engineer. Uncode the links and the functions to be re-engineered. Find the data matrix and the interactive blocks for the inputs and link them with functional Blocks. Join then with the output blocks to see theresults. Wecancalculatetheaccuracyandchecktheerrorby using NN function.This has not been donebyanyresearcherand the result can be calculated by forward as well reverse analysis. Thetimetakentoexecutetheprocesscanbecalculatedw ithvery precise accuracy. This can also help the programmer toreorient the software however big it may be. The result can becompared with the previous software and the improvement can be sought. This method is fool proof and scientific andwecaneasilyfindthe bugs and repair them in no time.



Fig. 4: Implementation Detail

#### **V. CONCLUSION**

The proposed approach has promising results required less detected values. Lehman's Lawdetect the badsmells by combining the matrix with SOMOX and GUI with RECLIPSE and finally compare withany software project, the values which are obtained from Lehman's Law has minimum effect than other's. Lehman's Law has continuing

changestorealworldobjectswhicharerapidly

changing the detected values of bad smells and finally givesthe less value of bugs, which are detected by others. No one can change or remove any detected value in Lehman's Law. These lawrulesarejustifyingthesebadsmellswithintimeand maintain their software easily within minimum amount oftime.

#### REFERENCES

- [1] Shahid Iqbal, M.Naeem Ahmed Khan,"Yet another setoff requirement metrics for software projects", 2012.
- [2] Phuc V.Nguyen, Ph.D candidate, "The study and approach of software reengineering", 2011.
- [3] DavidEichman,"Factorsinreuseandreenginee ringoflegacy software".
- [4] Er.Anand Rajavat, Dr. Vrinda Tokekar,"A Decisional Framework Measure Technical

Dimensions of Legacy Application For Rejuvenation Through Reengineering", 2011.

- [5] Dr.Ying Zou, "Software Reengineering (Software Metrics and Quality)".
- [6] Dr. Ashok Kumar, Bakhshish Singh Gill,"Cost Model For Reengineering an Object Oriented Software System", 2011.
- [7] MarkusVonDetten,SteffenBecker,"Combinin gClusteringAndPatternDetectionfortheReeng ineeringof Component based Software Systems.

Satya Sobhan Panigrahi "Software Engineering: Re-Engineering Metrics "International Journal of Engineering Research and Applications (IJERA), vol.6(9), 2016, pp 84-87.

\_\_\_\_\_