Software Superiority Achievement through Functional Point and Test Point Analysis

Amol K. Kadam\(^1\), S.D. Joshi\(^1\), Debnath Bhattacharyya\(^2\) and Hye-Jin Kim\(^3\)

\(^1\)Bharati Vidyapeeth University College of Engineering, Pune
akkadam@bvscoep.edu.in, sdj@live.in
\(^2\)Department of Computer Science and Engineering, Vignan’s Institute of Information Technology,
Visakhapatnam-530049, India
debnathb@gmail.com
\(^3\)Sungshin Women’s University,
2, Bomun-ro 34da-gil,
Seongbuk-gu, Seoul, Korea
hyejinaa@daum.net
(Corresponding Author)

Abstract
Software cost estimation is the important activity while the development of the software. Expenditure assessment is bit complex task as it can be affected by many factors. This factors aids in the calculating of maintenance cost of software. In this paper we have implemented the function point analysis and test point analysis in order to discover the maintenance cost. This is accomplished by using the various techniques to calculate the function point analysis and test point analysis. Along with the calculation of maintenance cost we have also presented the module to assess the reliability of the software from the context of white box testing. Software reliability growth models are aids to evaluate the reliability of the software. This paper presented the analysis of code based SRGM to estimate the reliability.

Keywords: Function Point Analysis (FPA), Test Point Analysis (TPA), Software Reliability, Software Reliability Growth Model

1. Introduction
While facing the new challenges of the today’s growing age in every step we have to deal with various types of softwares. Some softwares are complex some are simple. The software which follows hard deadlines should design carefully with the greatest precision in order to acquire the accuracy. So to prevent the software from defects and bugs, software reliability required to estimate.

Because of the modification of the three lines in the code of single program the entire telephone system in California and eastern has totally distorted [1]. Due to software failures aircraft industries have also confronted bunches of air transport accidents, strange flight conditions because of contrary reaction to the pilots [2]. These major problems occur due to lack of reliability and accuracy.

Reliability is the vital aspect of the software quality. In short the reliability and quality are associated with each other [3]. So to maintain the quality of the software we also have to sustain the superiority of the software. For the assessment of the reliability, Software reliability Growth model (SRGM) is utilized [4] [8]. Various reliability growth models have been proposed. Some models are applicable in some circumstances. There is no such model exist which can be applicable in all
circumstances due to several environment dependent assumptions. Figure 1 and Table show the various software reliability growth models.

![Reliability Model Classification](image)

**Figure 1. Reliability Model Classification [2]**

**Table 1. Reliability Models [2]**

<table>
<thead>
<tr>
<th>Models</th>
<th>Proposed by</th>
<th>Year</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>JMM</td>
<td>Z. Jelinski Paul and B. Moranda</td>
<td>1972</td>
<td>Binomial</td>
</tr>
<tr>
<td>GM</td>
<td>Amrit Goel and Kazu Okumoto</td>
<td>1979</td>
<td>Poisson</td>
</tr>
<tr>
<td>ETM</td>
<td>Jolan Musa</td>
<td>1975</td>
<td></td>
</tr>
<tr>
<td>HEM</td>
<td>Ohba</td>
<td>1984</td>
<td></td>
</tr>
<tr>
<td>WM</td>
<td>Weibul</td>
<td>1983</td>
<td>Binomial</td>
</tr>
<tr>
<td>SSM</td>
<td>S. Yamada, M. Ohba, and S. Osaki</td>
<td>1983</td>
<td>Gamma</td>
</tr>
<tr>
<td>DM</td>
<td>J.T Duane</td>
<td>1984</td>
<td>Poisson</td>
</tr>
<tr>
<td>GM</td>
<td>Paul B. Moranda</td>
<td>1979</td>
<td>Binomial</td>
</tr>
<tr>
<td>LPM</td>
<td>Musa –Okumoto</td>
<td>1984</td>
<td>Poisson</td>
</tr>
<tr>
<td>LVRG</td>
<td>A. Ghaly, P. Chan, &amp; B. Littlewood</td>
<td>1986</td>
<td>Gamma</td>
</tr>
</tbody>
</table>

Software is used in many industrial, engineering applications which are bit complex. For such framework, exact appraisals of the software product expenses are the significant components of efficient program administration [5].

Software cost estimation is somewhat difficult to measure for both software designer and customer. There are several methodologies to estimate the Expenditure for the software.

Function point is the entity of the dimension to represent quantity of software functionality performance [6]. And function point analysis is the procedure to calculate the software size.

Test point analysis is used to estimate the test efforts [7]. Test point analysis does not focus on white box testing efforts and includes only black box testing strategy. So, TPA is combined FPA in order to estimate both glass box testing and black box testing efforts. Martin L. Shooman [11] put focus on the errors that occurs within the program. They have classified errors as follows:

1. Bohrbugs
2. Mandelbugs

Bohrbugs means the software will generate the similar errors after taking the similar inputs. And in contrast to that Mandelbugs means software will not generate the identical errors also after taking the similar inputs. So based on above circumstances Martin L. Shooman estimates the reliability model which has considered these two types of errors.

Qiuying Li[12] have presented a novel approach towards the SRGM which comprises S-shaped testing coverage with the phenomenon of field environmental
randomness. Ankur Choudhary et. al., [13] focused on prediction of software reliability considering the parametric and non-parametric aspects. Jie Zhang et. al., [14] have focused on NHPP dependant reliability model considering two factors:

1. testing efforts
2. rate of fault detection

Zainab Al-Rahmaneh et. al., [15] have considered the genetic programming approach for reliability growth model. Through this approach they have analyze the hoarded faults.

2. Background and Motivation

Evaluating the expenditure assessment is the greatest challenge in front the software management as the involvement of the many factors in the cost estimation. The development of efficient approach towards the expenditure assessment is the main motivation behind this work. And also assessment of the reliability by means of the reliability model is the second motivation behind this work.

3. Methodology

We have to use Threshold value for the compare the results

- **Threshold value:**
  The threshold value is interpreted based on previous projects experience and historical information. While considering the threshold value, benchmarks designed by industries also taken into grant. From the team experience and various processes involved the threshold is monitored and updated [16].

  A) Function Point Analysis
  “Function Point Analysis is the method for evaluating the software dimension”
  Function Point Analysis Includes:
  ✓ Quality & Testability
  ✓ Skill of the Staff Members
  Quality & Testability Include:
  - **Calculate Size of Project:**
    It calculates the lines of code, blank spaces as well as comment lines in the project. If the number of lines per class is greater than threshold value (>750) then it advice for split up the class and delegate.
  - **Calculate Number of Object in Class:**
    Total numbers of object in class are identified. For the project given to system then after clicking on calculate number of object of class it calculates number of attributes of the class. If the total number of object in class are more than threshold range (>10) then suggest advice for splitting the class.
  - **Calculate Number of Methods:**
    In this module it calculates how many methods in each class of the project. If method values don’t match between threshold range 3-7 then provide suggestion for decompose the class having more methods.

  B) Test Point Analysis
  “Test Point Analysis is the mechanism for the test evaluation.”

  Test Point Analysis Includes:
Team Structure
- Environmental Factor
- Calculate Project Cost

Team Structure:
Mainly focused on how many employees are required for completing project, suppose employees are project manager, team leader, seiner developer, junior developer and trainee. All these employees perform various task such as how many lines written per day, how many days he worked on that project then based on these parameters it estimates cost of the team structure.

Environmental Factors:
Environmental Factors plays keen role in estimating the software cost, Infrastructure, electricity, Maintenance & other miscellaneous are such kind of factors which I have to consider for budgeting the cost of the project.

Calculate Project Cost:
By considering the parameters which come under structure of team as well as environmental factors, which are used in development process, total amount of expenditure taken into consideration it calculates total cost of the software.

4. Algorithm

A) Functional Point Analysis

Quality and Testability

1) Calculate Size of Project:
   THRESHOLD_VALUE = 750
   function StatementCoverage(file):
   
   Input: A file from the directory containing code files
   Output: None
   Side-effect: Conditional splitting of file
   if (file.countLines() > THRESHOLD_VALUE)
   SplitFile(file)

2) Calculate number of objects of a class
   THRESHOLD_VALUE = 10
   function TotalObjects(CLASS):
   
   Input: Class
   Output: None
   Side-effect: Conditional splitting of file
   objectsCount = class.memberObjects.length
   for (method in class.methods):
   objectsCount += method.memberObjects
   if (objectsCount > THRESHOLD):
   RefactorClass(classname)

3) Calculate Number of Methods.
   THRESHOLD_MIN = 3
   THRESHOLD_MAX = 7
   function ProcedureCoverage(class):
   
   Input: Class
   Output: None
   Side-effect: Conditional splitting of class
   methodsCount = class.methodsCount
if (methodsCount < THRESHOLD_MIN):
    DecomposeClass(classname)
if (methodsCount > THRESHOLD_MAX):
    Splitclass(class)

B) Test Point Analysis:
Team Structure
function TeamStructure():
    Input: Employee details from user
    Output: Employees with their daily Line of code and scale
    employees = Input()
    return employees

function EnvironmentalFactor():
    Input: Environmental expenses from user
    Output: Total environmental expense
    expenses = Input()
    return expenses.total

function ProjectDevelopmentCost(projectSize):
    Input: Size of the project in line of code
    Output: Project Development Cost
    employees = TeamStructure()
    totalScaleperday = 0
    totalLOCperday = 0
    for (employee in employees):
        totalLOCperday += employee.loc
        totalScaleperday += employee.scale
    totalDays = projectSize / totalLOCperday
    developmentCost = totalDays * totalScaleperday
    return developmentCost

function TotalCostCalculation(projectSize):
    Input: Size of the project in line of code
    Output: Total Project Cost
    Totalexpenses = EnvironmentalFactor()
    developmentCost = ProjectDevelopmentCost(projectSize)
    totalCost = Totalexpenses + developmentCost
    return totalCost

5. Proposed System Architecture

Figure 2 shows the system architecture. Software system comprises the various modules. Each module or component has its own functionality. This system comprises the three modules.
• Software processes the input and generates the output. So the input for this product is the entire software product along with its coding part. Therefore the input to this product is nothing but the software product. As it analyses the coding part it is like the white box testing.

• Subsequent to the input files, the analyser module evaluates the code and it assess the various quality aspects among the code and by means of this it appraisals the reliability of the software. This module also estimates the function point analysis and test point analysis.

• After the analysis the results in the graphical format have been generated. From this result we are able to predict the overall reliability and maintenance cost of the software.

6. **Result Analysis**

Analysis of result comprises the various analytical measures to estimate extent of superiority. SRGM is analyzed with the help of various comparison criterions. Following are some parameters on the basis of which we analyze the SRGM.

- Size of the Project
- Number of Objects of the Class
- Number of Methods of the class

**Screen No 1: Home Screen**

**Description:** This is home screen of the research work. In that screen we can see the different tabs like Home, Functional Point, Test Point & Graph.

**Screen No 2: Calculate Size of Project**

**Description:** In this screen analysis all coding files of the project and calculate size of the project. Display the details about the code files. In details display File Name, File Path, Code Lines, Blank Lines, Comment Lines, Threshold Range and Advice. If line of code increases more than 750 lines then suggest the split the class.
Screen No 2
**Screen No 3**: Calculate Number of Object Class.
**Description**: In this screen calculate how many objects of each class in the project. Objects more than 10 in each class then provide suggestion to the user as the class requires decomposition in order to better manage the complexity of the model.

Screen No 3
**Screen No 4**: Calculate Number of Methods and Events.
**Description**: In this screen calculate how many methods and events in each class of the project. If method values don’t match between range 3 to 7 then provide suggestion for splitting “Decompose the Class”.

Screen No 4
Screen No 5: Team Structure.
Description: In Test Point Analysis displays Team Structure, Environmental Factor & Calculate Project Cost and New Project Cost Estimation. In Team Structure provide the how many employees are required for completing project like how many Project Manager, Team Leader, Senior-Junior Developer and Trainee are required for complete project.

Screen No 6: Environmental Factor.
Description: In environmental factor provide values for infrastructure, electricity, development and other cost. Environmental factor is very important in calculating project cost.

Screen No 7: Calculate Project Calculation.
Description: This screen is the final screen of the research work. In that calculate the final cost of the project using team structure cost & environmental factor cost.
Screen No 7

Screen No 8: New Project Cost Estimation

Description: This screen calculates new project cost estimation.

Screen No 8

Resultant Graph:

Screen No 9: Graph

Description: In graph section display graph of line of code, Object of Class and Number of Methods in inputted project code. This graph present the all lines before the threshold values that means don’t changes in generated errors.
Screen No 9

**Description:** In graph section display graph of line of code, Object of Class and Number of Methods in inputted project code. This graph present the all lines after make changes in programs that achieve minimum threshold values.

Screen No 10: Graph

Above Screen no 09 and 10 shows the screen outputs for the proposed models. Screen no 09 shows the graph for test coverage function versus the time interim. This graph shows the various test coverage function values for various time interims which indicate the coverage degree.

In this graph displays of line of code, Object of Class and number of methods in inputted project code. This graph present the all lines before the threshold values that means don’t changes in generated errors.

Screen number 10 shows the graph for make changes in project to increase the reliability. This graph shows the how to increase reliability of the software. Both
graphs show the superiority of proposed SRGM in terms of both before and after function. In this graph displays line of code, Object of Class and number of methods in inputted project code. This graph present the all lines after make changes in programs that achieve minimum threshold values.

7. Conclusion

This paper presents a novel approach towards the expenditure assessment and reliability appraisals. By combining the both approaches we definitely achieve the technique for accurate cost estimation and reliability analysis. This software will definitely become beneficial for the software industry.

References
