

Measuring the Quality of Software on the basis of Time Gap and Quality Gap with Standard Model-II

Ekbal Rashid¹ and Srikanta Patnaik²

¹*Department of Comp. Sc. & Engg.
C.I.T, Tatisilwai, Ranchi*

²*Department of Comp. Sc. & Engg.
Siksha 'O' Anusandhan*

University, Bhubaneshwar, Orissa

ekbalrashid2004@yahoo.com¹, patnaik_srikanta@yahoo.co.in²

Abstract

While dealing with students in making their projects, we found that students are ready to cope up the standard software defined by standard body like IEEE/ACM. In this paper we have set the quality value of standard software on the basis of quality function for each month and comparing with standard software and software being developed. The novel idea of this paper is to calculate the difference in quality of the software being developed and the standard software which has been decided upon as the criteria for comparison. The system also calculating how much effort needed and what should be the speed of software being developed. This work is the second part of a Understanding the State of Quality of Software on the basis of Time Gap, Quality Gap and Difference with Standard Model, which moves towards practical implementation from theoretical foundation [1]. In order to obtain results we have used an indigenous tool for calculating the quality gap and time gap at a particular point of time and for the graphical representation of data, we have used Microsoft office 2007 graphical chart. For finding the mean, standard deviation, T value and P value, we have used SPSS version 16.

Keywords: *Software quality, time gap, quality gap, quality difference, effort, rate*

1. Introduction

Software quality is a set of characteristics that can be measured in all phases of software development. It is not an absolute term. Somewhat, like other scientific parameters, it should be viewed as something precise, relative and concrete. Even measurement of software quality has to be contemplated with the correct approach, which is on the basis of comparison with some specific standard. For instance, what do we mean when we say that something is ten meter long. clearly, we have some standard length, which we call one meter, and when compared to that, our object in discussion in ten times that standard length. Similar is the case of measuring other scientific parameters. The comparisons are made with some specific standards which are globally recognized. However in the case of software engineering, quality measures do not have globally recognized concrete values. There are set of standards defined by the IEEE/ACM as to how a software quality should be estimated. Only when some concrete software has been identified as the measuring standard, can we precisely calculate the quality of any software by comparing it to that given standard. This research work outlines the possible comparison parameters to measure the quality of software on the basis of

time gap and quality gap. The parameters that have been used are novel ideas and have not been previously used in the realm of software engineering. Moreover, an attempt has been made to chalk out a mathematical model, and a concrete strategy so as to precisely determine the quality of the software being developed. As elucidated earlier, the basis of quality determination is chiefly by drawing out an analogy between what is being developed and what is already developed. Although, the author considers that an international standard has to be established, regarding this quality, unless that is achieved, we can define our own standards in our own institutes and organizations and use the methods given herewith to determine concretely the state of quality of the software being developed. The mathematical methods employed are basic and can be used by any people. The parameters and terms may be enriched and advanced to form newer concepts and ideas. On the whole after the required comparisons have been made and the particular measuring of the quality has been achieved, decisions can be taken suitably to augment the proper and rapid quality software development.

2. Background and Related Works

The client requires a system that can predict the quality of the software which is being developed on the basis of time gap and quality gap. Quality software is reasonable bug-free, delivered on time and within budget, meets requirements and is maintainable. So, the method of determining the quality of software is unavoidably linked to the ideal of software testing. There are many methods employed in software testing such as – on the basis of hundred lines of code, on the basis of per hours of development time, on the basis of per hundred tests conducted [3]. There are works regarding the technique and framework to measure the quality of software. This technique leverages technology that automatically analyzes 100% of the paths through a given code base, thus allowing a consistent examination of every possible outcome when running the resulting software. Using this new approach to measuring quality, there has been a target to give visibility into how various open source projects compare to each other and suggest a new way to make software better [4]. There have been study where the research objective is to build a parametric model which utilizes a persistent record of the validation and verification (V&V) practices used with a program to estimate the defect density of that program. The persistent record of the V&V practices are recorded as certificates which are automatically recorded and maintained with the code [5]. In-process quality metrics are less formally defined than end-product metrics, and their practices vary greatly among software developers. On the one hand, in-process quality metrics simply means tracking defect arrival during formal machine testing for some organizations. On the other hand, some software organizations with well-established software metrics programs cover various parameters in each phase of the development cycle [6]. It is especially in context of the last idea that this paper has been designed. The target is to achieve the quality during the development process and that has been explained in the section of objectives as below. Rashid et. al emphasized on the importance of Estimation and evaluation of change in software quality at a particular stage of software development [2].

The rest of the paper is organized as follows: section 2 gives brief overview of the various related work, section 3 describes the aim of research, section 4 describes the methodology. In section 5 we describe the problem statement, in section 6 we describe measurement the development speed & efforts, in section 7 statistical analysis has been presented, in section 8 analysis and results has been given , Section 9 presents the conclusion. The Appendix provides the details of algorithm and flow chart which is used in this research paper.

3. Aim of Research

The aim of this research work is to calculate accurately the status of the software during the course of its development. This aim is further linked to the focus of trying to analyze whether the development process is going on the right direction. If the development process is not going on as per the expected lines, corrective or remedial measures may be undertaken to bring it back on the desired track. The concrete quality measurement is not the quality of the end product. Rather, it is the quality of the software at various stages. Usually, the final quality is talked about more in the domain of Software Engineering. However, we have to understand that the final quality can only be achieved if the intermediate quality levels are assured. Particularly, if we follow a model of Software development such as the Boehm's Spiral Model, where there are consistent checks at regular stages to see if the software is developing on the required lines or not, the method given in the present work will prove to be very useful. This method will also prove to be effective in all forms of iterative models of software development. The research work also fulfils the aim of defining the necessary metrics to make the above measurements. This paper also has the objective to present an alternative mathematical model and a very simple one so as to enable the organizations take decisions in the right direction.

4. Methodology

The measurement of software quality has been done through indigenous tool. The environment of our study is the college campus and our target group is the B. Tech students of computer science and engineering. Students are guided with their project supervisor, supported by the technical staffs, resources like computers, software etc. The quality values for software being developed were given by the supervisor on the basis of performance of the students during development of the projects in month wise. The project duration is a period of one year.

5. Problem Statement

The following points give the steps in the process:

1. First, there should be a quality function related to the type of software being developed. This quality function should be a function of time. This quality function can be decided either by any international body (only then can the measurement be universalized) or by any organization interested in determining the quality of the software at various phases of software development. To put it more elaborately, we can have a separate quality function for a particular type of software. To decide what this quality function should be one has to go back and trace the history of development of similar types of software developed in accordance to the principles and methods laid out in the IEEE/ACM standards. The quality function can be linear or polynomial depending upon the complexity of the software and upon the known history of its development. How do we arrive at this function? We have shown the method to arrive at such a standard function by using some synthetic data. Let us assume for the sake of simplicity that it takes twelve months to develop some typical software and the quality of the software develops as shown below. See Table 1.

Table 1. (Synthetic Data Set)

Time (in months)	Quality
1	5
2	8
3	11
4	14
5	17
6	20
7	23
8	26
9	29
10	32
11	35
12	38

We can plot the graph of quality versus time with the help of the above data as shown below in Figure 1.

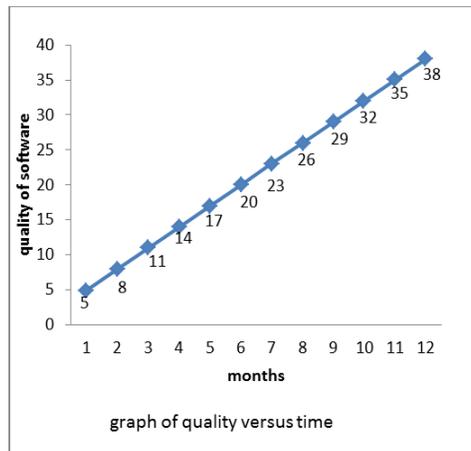


Figure 1. Quality versus Time

The quality function then can be easily determined which will be as under for the given data. In this case, we have set the quality value of standard software on the basis of quality function. (See Table 1)

$$Q(t) = 3t + 2$$

(1)

Where Q = Quality function
t = Time in months

```
C:\WINDOWS\system32\cmd.exe - tc
Months      Quality value of standard s/w
-----
1           5
2           8
3          11
4          14
5          17
6          20
7          23
8          26
9          29
10         32
11         35
12         38

*****MAIN MENU*****
1.Display Quality value of standard software
2.Enter Quality value of software being developed & display
3.Calculate Time Gap
4.Calculate Quality Gap
5.Calculate Rate
6.Calculate Effort
7.Exit
Enter your option
```

Snapshot 1(months wise quality value of standard software)

2. Second, after close monitoring, we have framed the check point months wise when software is being developed by the B. Tech students of computer science in group. In this case, we are using student's data after checking the quality value achieved by the students in every month with respect to standard software. On the basis of the data for the software being developed, we can plot the points on the same graph. There is less possibility that this graph will be linear or of any regular form. The quality of software being developed is improving as per the following data. See Table 2.

Table 2. (Real Data Set)

Time (in months)	Quality
1	3
2	6
3	7
4	9
5	10
6	13
7	15
8	18
9	20
10	22
11	25
12	27

On the basis of the above data we can plot a graph shown in Figure 2.

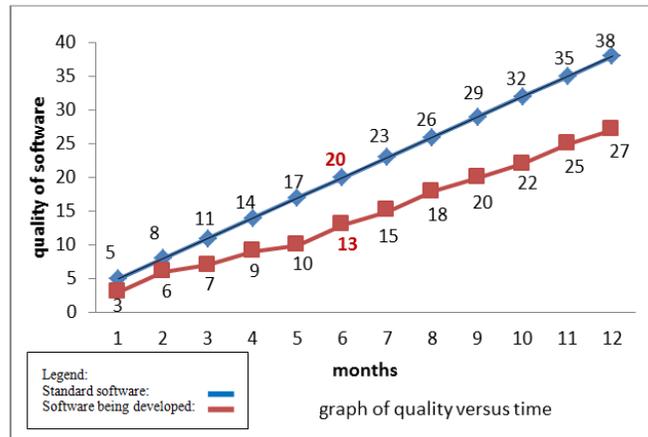
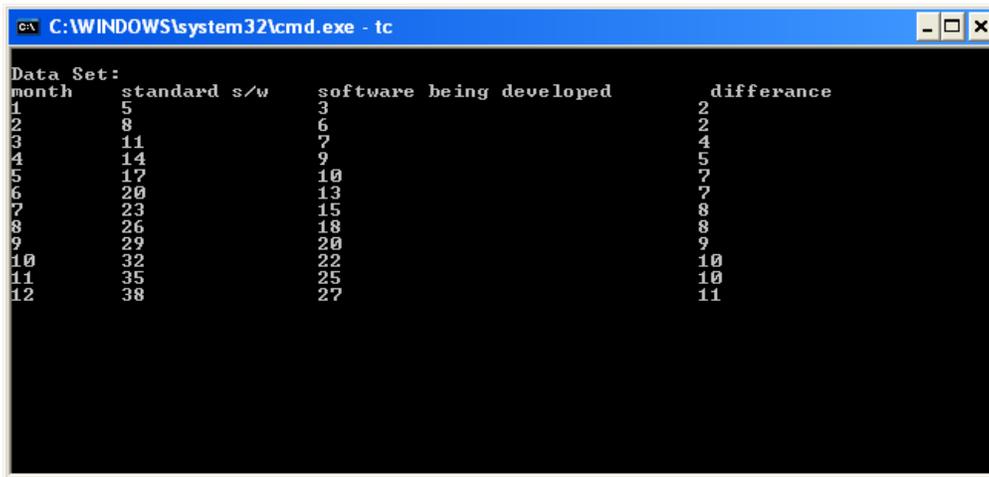
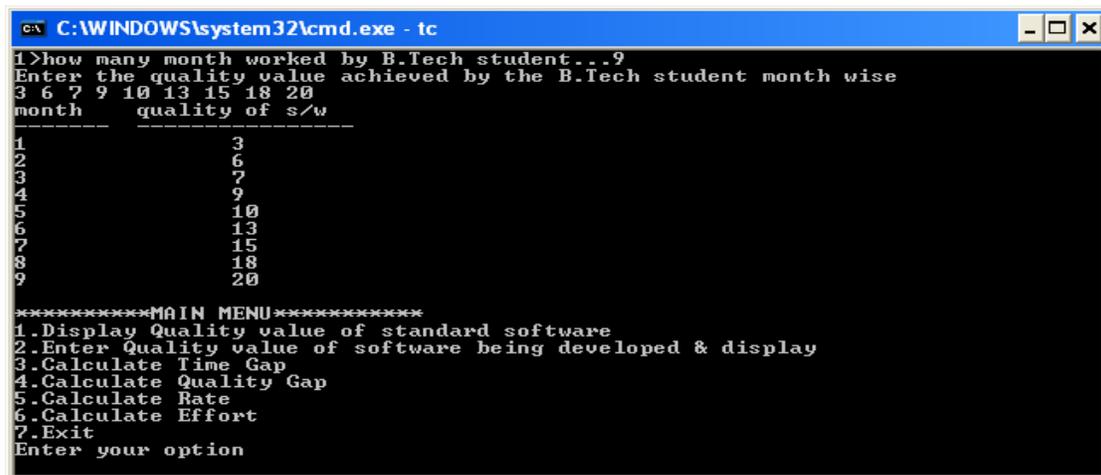


Figure 2. Quality versus Time



Snapshot 1 shows the difference between standard software and software being developed



Snapshot 2 (months wise quality value of software being developed)

3. Third, we calculate the quality gap. For this, at a particular point of time we identify the quality of the standard software (q_1) and then we identify the quality of the software being developed (q_2). Then the relation $q_1 - q_2$ gives the quality gap. We have taken the above data to calculate the quality gap between the software being developed and the standard software available with us for comparison. We are calculating the quality gap just after six months of software development work has been completed. We see that the quality of the standard software at six month time point is 20 while the software being developed has reached quality of 13 after six months. Then we can say that the quality gap at six months is 7. This fact is being illustrated through figure 3 as well as in snapshot. (See snapshot 3).

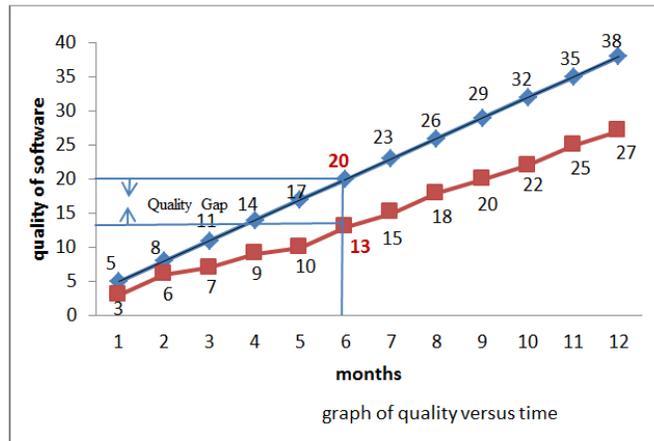


Figure 3. Quality Gap

```
C:\WINDOWS\system32\cmd.exe - tc
Enter Quality of software being developed
3 6 7 9 10 13 15 18 20
Time<in month>Us Quality<Standard Model>
1 5
2 8
3 11
4 14
5 17
6 20
7 23
8 26
9 29
Time<in month> Us Quality<Software being developed>
1 3
2 6
3 7
4 9
5 10
6 13
7 15
8 18
9 20
Enter month on which you want to check progress....
6
Quality gap = 7
```

Snapshot 3 (Measuring the Quality Gap after six months)

4. Fourth, we calculate the time gap. In order to calculate the time gap, we first pick a particular quality value of the standard software for which we want to calculate the time gap. Next, we identify the time (t_1) at which the standard software had acquired that quality and

then we identify the corresponding time (t_2). Then $t_1 - t_2$ gives the time gap between the two for the particular stage of quality. We then draw a horizontal line to meet the graph of the software being developed. Where the line meets the graph of the software being developed. As for example, we have identified the time when the standard software acquires the quality 20 as 6 months. For the same quality we see that the software being developed is taking 9 months. So for the quality level 20, we say that the time gap is 3 months. This fact has been illustrated in the form of snapshot 3 and figure 4. In the same manner we can tabulate a series of data showing the quality gap at a particular time of software development and the time gap at a particular level of quality as parameters of comparison.

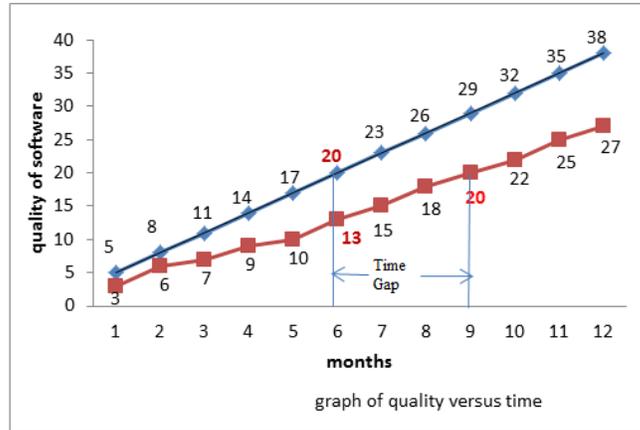
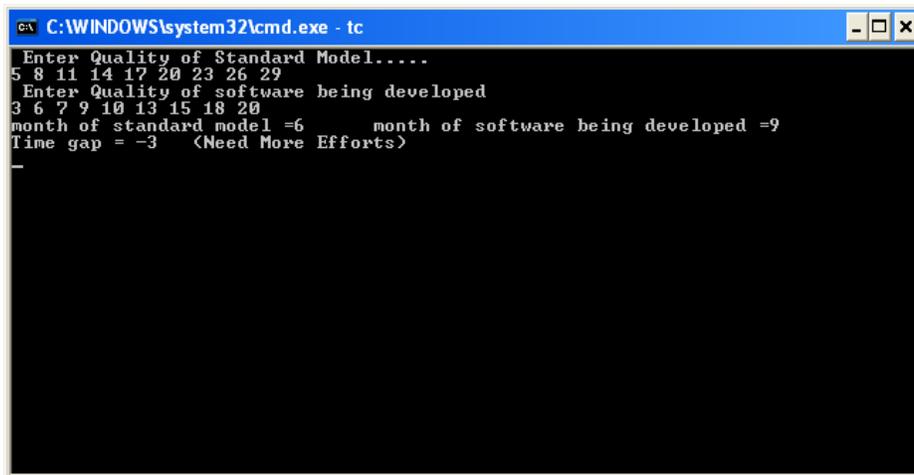


Figure 4. Time Gap



Snapshot 4 (Measuring the Time Gap after six months)

6. Measuring the Development Speed (Rate) and Efforts

This section gives the detail information regarding efforts needed on the basis of time gap and quality gap. Both parameters are responsible for measuring the quality between standard software and software being developed. It means how much efforts required reaching the standard software. For example, if large difference between standard software and software being developed then, we can calculate through this system that how much effort needed to

achieve the quality of standard software and what should be the speed of developing the software being developed. After seeing the snapshot 5 and snapshot 6, we can say that we need more efforts as well as rate of development speed must be increased. The following formulae have been used for rate calculation as well as effort calculation, which are given below:

$$\text{Rate} = (\text{Rate} + \text{Q_O_S_B_D} * 5) / \text{Number}$$

Where

Rate=0

Q_O_S_B_D = Quality of software being developed

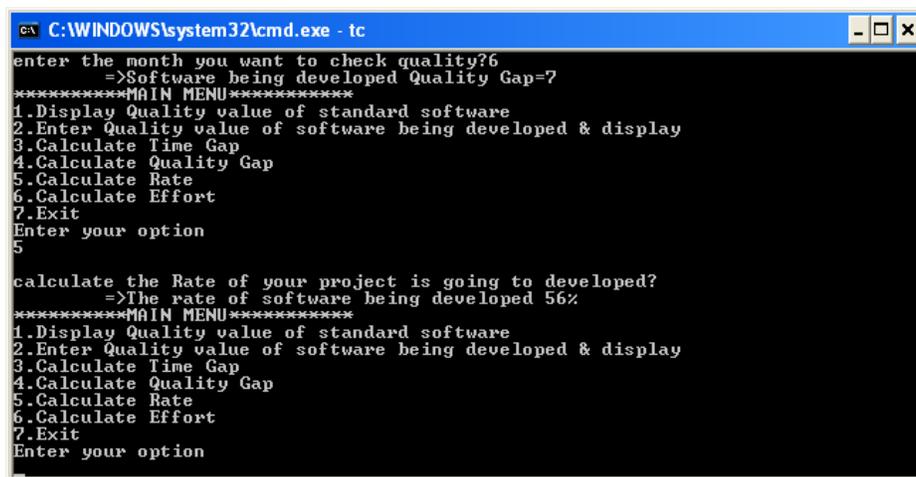
Number = How many months worked by B. Tech student

$$\text{Effort} = \text{sqrt}(\text{pow}(\text{quality}^2) + (\text{time}^2))$$

Where

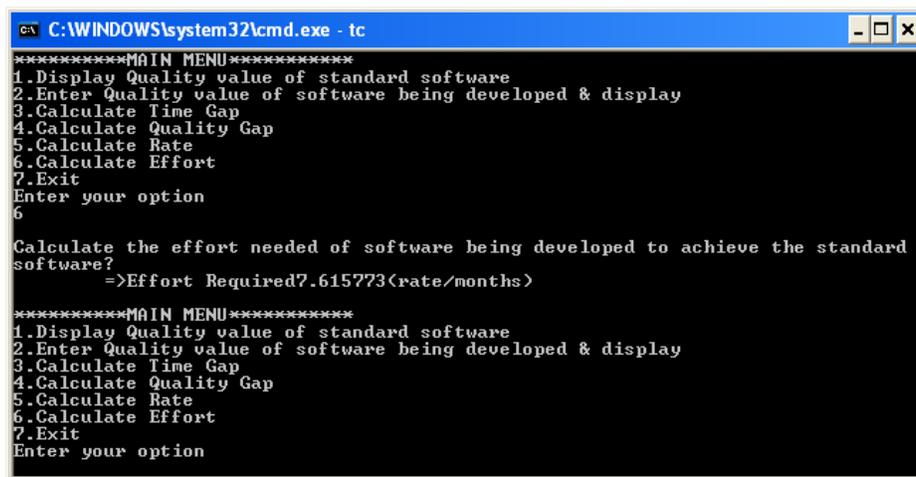
Quality = quality of standard software – quality of software being developed

Time = quality gap of standard software.



```
C:\WINDOWS\system32\cmd.exe - tc
enter the month you want to check quality?6
=>Software being developed Quality Gap=7
*****MAIN MENU*****
1.Display Quality value of standard software
2.Enter Quality value of software being developed & display
3.Calculate Time Gap
4.Calculate Quality Gap
5.Calculate Rate
6.Calculate Effort
7.Exit
Enter your option
5
calculate the Rate of your project is going to developed?
=>The rate of software being developed 56%
*****MAIN MENU*****
1.Display Quality value of standard software
2.Enter Quality value of software being developed & display
3.Calculate Time Gap
4.Calculate Quality Gap
5.Calculate Rate
6.Calculate Effort
7.Exit
Enter your option
```

Snapshot 5 (rate of software being developed)



```
C:\WINDOWS\system32\cmd.exe - tc
*****MAIN MENU*****
1.Display Quality value of standard software
2.Enter Quality value of software being developed & display
3.Calculate Time Gap
4.Calculate Quality Gap
5.Calculate Rate
6.Calculate Effort
7.Exit
Enter your option
6
Calculate the effort needed of software being developed to achieve the standard
software?
=>Effort Required7.615773(rate/months)
*****MAIN MENU*****
1.Display Quality value of standard software
2.Enter Quality value of software being developed & display
3.Calculate Time Gap
4.Calculate Quality Gap
5.Calculate Rate
6.Calculate Effort
7.Exit
Enter your option
```

Snapshot 6 (Efforts required)

7. Statistical Analysis

In this section, we have done statistical analysis of standard software and software being developed. Below table shows that mean with standard deviation of the quality of standard software and software being developed is 21.5 ± 10.82 and 14.58 ± 7.86 respectively. After applying the student's t test, we found that the quality of the standard software and software being developed is not statistically significantly different ($p < 0.087$). It concludes that software being developed is matching with the standard software as far as quality is concerned.

Table 3: Mean and Standard Deviation of the Quality of Standard Software and Software being Developed

Group A1+A2	No of months	Mean	Std. Deviation	T value	P value
Standard software (A1)	12	21.50	10.817	1.792	.087
Software being developed (A2)	12	14.58	7.856		

8. Analysis and Results

The quality gap between the standard software and the software being developed helps us to understand how much more effort needs to be put in terms of removing faults/defects and improving code quality. More the quality gap more is the amount of effort needed to be put in this regard.

The time gap between the standard software and the software being developed shows us how much the software being developed is lagging behind the standard software in terms of time. More the time gap more is the effort needed in terms of speeding up the process of software development. If necessary the number of software development people has to be increased in order to match the standard quality.

In this way we can say that the quality gap speaks qualitatively about the software. The quality of the software developers have to be improved to bridge that gap. On the other hand, the time gap speaks quantitatively about the software. The number of software developers need to be increased in order to bridge the gap with respect to the standard software.

The difference between the standard software and the software being developed gives the overall view about the status of the quality of the software being developed. It helps us in understanding the overall effort needed in terms of achieving a particular quality of standard software. (See Table 4).

Table 4. Summary of Standard Software and Software being Developed

Months	Quality of standar d softwar e	Quality of software being develope d	Time gap between standard software and	Quality Gap between standard software and	Rate (Speed) of software being develope	Efforts Require d for software being develope
--------	---	---	---	--	--	--

				software being develope d	software being develope d	d	d
6 (Standar d software)	9 (Softwar e being develope d)	20	13	3	7	54%	7.615

9. Conclusions

This research work deals about the status of the quality of the software that is being developed. The importance of the work lies in the fact that the parameters that have been introduced here give a clear picture about the qualitative and the quantitative states of the software stage. The parameters can be used for other kinds of software analysis as determining how much effort needs to be put in as regards to which aspect to improve the quality of the software. Only when we compare the software with any particular standard can this be made possible. For this reason we stress upon the necessity of having an international standard of various types of software that are typically being developed nowadays. It would be better to express the standards as functions of time so that the comparisons can be easily made. The conceptions used here are significant as they can be further enhanced to develop newer understandings about the quality. Here, we have taken development time in terms of number of month. For smaller software, we can take the development time in terms of number of days.

We can also review the concept by using number of software developers instead of taking the number of months for software development. Then, we shall be calculating some other parameters instead of time gap but then the parameter quality gap would remain. Only this time it would be in terms of number of working hands instead of being calculated with respect to time. The above mentioned parameters as well as the significance of the ratio between the quality gap and time gap are areas for future work.

10. References

10.1. Journal Article

- [1] E. Rashid, S. Patnaik and V. Bhattacharjee, "Understanding the State of Quality of Software on the basis of Time Gap, Quality Gap and Difference with Standard Model" has been published in International Journal of Engineering and Technology (IJET), ISSN: 0975-4024, vol. 5, no. 3, (2013), pp. 2821-282.
- [2] E. Rashid, S. Patnaik and V. Bhattacharjee, "Estimation and evaluation of change in software quality at a particular stage of software development" has been published in Indian Journal of Science and Technology (INDJST), pp 5370-5379, ISSN: 0974 -5645, vol. 6, no. 10, (2013).

10.2. Book

- [3] S. Desikan and G. Ramesh, "Software Testing", Principles and Practices Pearson, (2012), pp. 444-450.
- [4] B. Chelf, "Measuring Software quality", A study of Open Source Software, Coverity, Chief Technology Officer Coverity, Inc.
- [5] M. Sherriff and L. Williams, "Defect Density Estimation Through Verification and Validation", North Carolina State University Raleigh, NC, USA 27695 {mssherr, lawilli3}@ncsu.edu).
- [6] S. H. Kan, "Metrics and Models in Software Quality Engineering", second edition by Addison Wesley p. 100.

APPENDIX

Algorithm

Steps:

1. Declare Variables.
2. Set sum=0.
3. Repeat step 4 to 14 step.
4. Create menu containing option 1 to 7.
5. Choose your option.
6. Execute switch case as switch (option).
7. Case 1:
 - a. Initialize i=1 as month;
 - b. Repeat 1 up to 12
 - c. Compute quality of s/w corresponding to each month using quality function formula.
 - d. Set quality[i]=(3*i)+2;
 - e. Increment i;
8. Case 2:
 - a. Initialize j = 1 as month.
 - b. Repeat j up to user defined month (value of num).
 - c. Enter quality of s/w being developed.
9. Case 3:
 - a. Compute Time gap
 - b. Repeat i from 1 to 12.
 - c. Repeat j from 1 to 12.
 - d. Check whether quality[j] is not equal to quality[i].
 - e. Compute Time gap using Time=[i-j];
 - f. Display result.
- 10 case 4:
 - a. Enter month (n), you want to compare quality of s/w being developed to standard s/w.
 - b. If(n<=num) then
 - c. Compute quality gap using quality=quality1[n]-quality2[n].
 - d. Else print error message.
 - e. Display result.
- 11 . case 5:
 - a. Repeat j from 1 to num.
 - b. Set sum = sum+qualty2[j];
 - c. Compute rate as rate= (sum*5)/num.
 - d. Display Rate.
12. case 6:
 - a. calculate required effort using formula .
 - b. and set effort= sqrt(pow(quality,2)+pow(3,2)).
 - c. display effort.
13. case 7:

Exit.
- 14 default :

Print error message.

Flow chart

