Volume 6, Issue 1, January 2016 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Methodologies for Measuring Efficiency of the Programmer

[1]Deepalakshmi.J, [2]A. V. Ramani
[1]Research Scholar [2]Associate Professor
[1, 2] Department of Computer Science, Sri Ramakrishna Mission Vidyalaya College of Arts and Science
Coimbatore-20, Tamilnadu, India

Abstract: Computer programs are result from a creative process under strong formal restriction. Effective testing can find more and more deficiency in the programs. Software measurement is aessential component of a healthy and highly capable software engineering culture. It is an integral part of the state-of the practices in software engineering. This paper proposed method to measure efficiency of the programmer. Here Methodologies are provided to measure the efficiency of the designer, efficiency of coder, efficiency of testers and efficiency of the maintainer.

Key words: Average time, Expected time, Software efficiency, repairer time.

I. INTRODUCTION

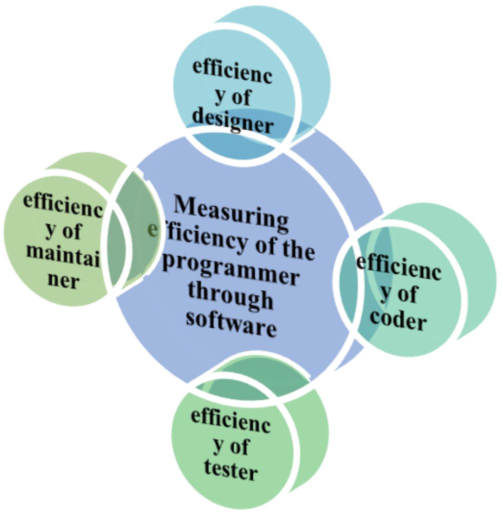
There are many ways in which the organization measures the performance of project in terms of efficiency and productivity. Some aspects of efficiency/productivity are important for measuring software and project quality, and estimating new projects for efforts and schedules. It has direct relationship with size of software under development. [14].
The efficiency is measured as size expressed in number of code lines/functions points/use cases ect. Developed by a project team per unit time,the size measurement may vary form organisation to organisation, customer to customer and project to Project. The size measurements can be debated over any length of period with advantages/disadvantages associated with each option.
The process of software development has been characterized as self emerges, iterative, community-situated, synergistic, and after preparatory. A focus on programming and programs as "knowledge craftship" is of particular interest from a learning analytics perspective. There is much less analytic research in the field of technology enhanced learning addressing the computational, analysis of craftship than there is on interaction analysis. However, from a creativity point of view, it is of high interest to identify creative contributions or "creative style elements" in the craftship themselves. [4,11]

II. LITERATURE SURVEY:

Computer programs are a specific type of knowledge that result from a creativity, it has been argued that programming supports intellectual development and knowledge building. At present, a system can to automatically detect the creativity of solutions to programming exercises that address general mathematical and algorithmic skills. The evaluation of software products made it potential to ascertain that themodel is an effective tool for scrutinize product quality, that can be used to compare different software products and also to detect weaknesses in the product that need to be improved. It revealed that Process Efficiency and Effectiveness influence product quality; an improvement in Auditing and Quality authority in particular has a repercussion on productimprovements.[3] Applying questionnaires to users, analysts/planners and project leaders, enabled information to be obtained that could be used by them for product improvement[3].Artificial, in that the creativity of the sentences are judged with respect to their respective keywords, which are assumed to be known beforehand. This allows us to design features around the keywords. Creativity is a key resource in producing competitive advantages, Identification ofSpecific criteria for creativity can lead to an improved level of specific criteria for creativity can lead to an improved level of sophistication in both software and management techniques. In an of flattened organizations where career paths are diminished, achievement goals must substituted for advancement goals. Creativity provides a new category for achievement [2]. More and more customers are specifying software and/or quality metrics reporting as part of their contractual requirements. Software Engineering has always been a matter of concern for every individual involved in software development starting from analysis phase to delivery phase or even at the maintenance time. There have been novel approaches for developing program complexity metrics. In this regard we have proposed the Efficiency Metrics, which can calculate the efficiency of a programmer and can also calculatethe exact time taken by the development team to complete the software development under various complexities. Over and above we have also developed a relation between time and efficiency have to be find out[5].Though the changes in the analysis, design and code are certain, we can still calculate the efficiency of the manpower (Programmers) have to be calculated involve in a project of development. The programmer's efficiency table shall be able to calculate the efficiency of the programmer to an appropriate level based on his aptitude, typing andprogramming skills. This efficiency shall allow to forecast the manpower required for testing his integral, which upsurges planner productivity. The planner has more time to focus on design and code .vs. testing.[21]

III. METHODOLOGIES FOR CALCULATING EFFICIENCY

Hear methodologies are provided to measure the efficiency of designer, efficiency of coder, efficiency of tester, efficiency of maintainer.



3.1 Efficiency Of A Designer [ED]:

Design could refer to many things but often refer to functional design or internal design, Architectural design and detail design are important levels of designs.

Once the requirements statement satisfies all characteristics, the system architect starts with system high-level architectural design. The designs made by system architects must be traceable to requirements. The system designer starts building the low-level or detail design with the help of architectural design. Low-level design must be traceable to architectural design which in turn is traceable to requirements. [12]

A design must define all functions, components, tables, stored procedures, and reusable components very clearly.

A design must be complete in all respect. It must define the parameters to be passed/received, formats of data handled, etc. Once the design is finalized, programmers must do their work of implementing designs mechanically.

A design must be made in such a way that it can be implemented easily with selected technology and system. It must guide the developers in coding and compiling the code. The design must include interface requirements where different components are communicating with each other and with other systems. [12]

In principle, the program design which is estimated to achieve the highest net assistance should be selected. However, optimizing [5] distributional assistances (to either primary or secondary recipient) typically conflicts with optimizing aggregate social welfare. Some times this may be relevant (i.e., targeted programs). At other times some might criticize the selection of sub-optimally efficient programs as pork-barrel spending. Making the trade-offs inherent in each design can help inform the debate. [2, 13]

The Total average time and efficiency of design is calculated using the following formula.

If Ri is the number of repetitions, of the design I, Ti is the time required for the design I, N is the number of designs.

Total avg time=$\frac{\sum Ri\ Ti}{N}$

Efficiency=$\left(1-\frac{\sum Ri\ Ti}{N}\right)$**X100**

The calculation of efficiency of designer is illustrated below. The number of repetition of the design and time taken for each repetition of the design are given in the following table.

PROGRAM NO	EXPEXTED TIME	ACTUAL TIME	EXPETEDTIME/ACTUAL TIME(ET/AT) min
1	10	5	2
2	5	2	2.5
3	2	5	0.4

Total avg time=$\frac{33.5}{3}$

Total avg time =10.625

Efficiency of designer=$\left(\frac{1}{100}\frac{\sum Ri\ Ti}{N}\right)$**X100**

= (0.665) X 100

= 66.5

Efficiency of designer =66.5%.

3.2 Efficiency of Coder.[EC]

Code efficiency is precisely linked with algorithmic efficiency and the speed of execution of software.

The goal of code efficiency is to reduce resource consumption and completion time as much as potential with minimum risk to the field or operating surroundings. The software product quality can be accessed and inspected with the help of the efficiency of the code used. Code efficiency plays a symbolic role in applications in a high-execution-speed surroundings where performance and scalability are paramount. One of the recommended best proceeding in coding is to ensure good code efficiency. Well-developed programming codes should be able to handle complex algorithms. [12]

The efficiency of the coder is calculated using following formula.

The calculation of efficiency of coder is illustrated below. The expected time to complete the coding and actual time taken to complete the coding is given in the following table. The efficiency is calculated using the following formula.

Efficiency of coder (%) = $\frac{1}{N}\sum\frac{ET}{AT}$**X 100**

If, ET- Expected Time, AT-Actual Time and then N-Number of programs.

DESIGN NAME	NO OF REPETIT IONS Ri	T 1	T 2	T 3	T 4	T5	AVG TIME (Ti)	RiTi
DESIGN 1	5	2	1	6	4	1/2	2.7	13.5
DESIGN 2	3	2	1	3	-	-	2	6
DESIGN 3	4	5	4	3	2	-	3.5	14

Formula for coding efficiency is,

Efficiency of coder (%) =$\frac{4.9}{3}$X 100

Efficiency of coder (%)=163.33%

The efficiency of the coder is 163.33% .Here the coder finished his work with in the given time. Hence the efficiency is more than 100%. Here the actual time is the time taken by the programmer to produce error free program with user satisfaction. Here all the test data are sample data.

3.3 Efficiency of a Tester [ET]:

A technician who conducts prescribed tests on softwareprograms and applicationsprior to their implementation ensuresquality, designintegrity and proper functionality. They apply rigorous testingmethods including extensive end-usersimulations to uncover program "bugs" which are then eliminated by the software programmers.

The responsibilities of a Software Tester is go through the software requirements and get clarifications on one's doubts , become familiar with the software under test and any other software related to it, understand the master test plan and/ or the project plan, Create or assist in creating own test plan, generate test cases based on the requirements and other documents, Procure or create test data required for testing, Set up the required test beds (hardware, software and network),Create or assist in creating assigned test automation, Test software releases by executing assigned tests (manual and/ or automated),Re-test resolved defects, Update test cases based on the discovered defects, Update test automation based on the updated test cases, Provide inputs to the team in order to improve the test process, Log own time in the project management software or time tracking system, Report work progress and any problems faced to the Test Lead or Project Manager as required,(If applicable) Support the team with testing tasks as required.[22,21]

Time required by a test team to execute one test case may be measured as testing efficiency. Testing efficiency is calculated on the number of test cases per unit time. It can also be measured in terms of coverage of functions, use cases, and lines of code by testing team.[8]

The following formula is proposed to calculate the efficiency of tester.

EFFICENCY OF TESTER = $\frac{1}{N}\sum\frac{NDR}{NDS}$ **X 100**

Where,

NDR- No. of deficiency Resolved.

NDS- No. of deficiency Submitted.

The calculation of efficiency of tester is illustrated below. The number of effects resolved and number of deficiency submitted to the tester is given in the Following table.

PROGR AM NAME	NUMBER OF EFFECTS RESOLVE(NDR)	NUMBER OF DEFICIENCY SUBMITTED(NDS)	$\frac{NDR}{NDS}$ X100
1	9	10	90
2	5	5	100
3	2	3	66.6

$$=\frac{1}{3}\ (256.6)$$

=256.6/3

Efficiency of tester=85.53%

3.4 Efficiency Of A Maintainer [EM]

Program maintainability and program understand ability are parallel concepts; the more difficult a program is to understand, the more difficult it is to maintain. [15]

Maintainable software exhibits effective modularity. It makes use of design patterns that allow ease of understanding. It has been constructed using well-defined coding standards and conversion, leading to source code that is self-documenting and understandable. It has undergone a variety of quality assurance technique that have uncovered potential maintenance problems before the software is released. [10, 14]

The calculation of efficiency of implementer is illustrated below. The work starting time work ending time, time to repair and error report time are given in the following table.

The efficiency is calculated using the following formula

Average repair time = $\frac{\sum X}{N}$

Process for calculate efficiency of program or Maintenance:

PROG RAM NO	PROGRAM REPECTED TIME	PROGRAM STARTING TIME	WORK ENDING TIME	TIME TO REPETE D (X)
1	10.00	11.00	12.00	+1HR
2	9	8	10	+1hr
3	8	8	7	-1

Average repair time =$\frac{4}{3}$

Average repair time =1.33

If $\frac{\sum X}{N} < T$ the efficiency of maintance is good.

T is set by the company.

IV. CONCLUSIONS

A literature survey was carried out on software testing. The study of various books and journals resulted the problem of "measuring efficiency of programmer skills through software".

The methodologies are proposed to measure the efficiency of designer, efficiency of coder, efficiency of tester and efficiency of implementer. The literature study does not show any mathematical model for measuring efficiency of coder, efficiency of designer, efficiency of tester and efficiency of implementer. Since the mathematical model is highly reliable than other models, methodology are deviated using mathematical models. This methodologies are tested with sample data. This shows that new methodologies are very useful. A software is also developed for measuring efficiency of designer, efficiency of tester, efficiency of coder and efficiency of implementer.

Now a days the team of people are involved in development of a software project. So programs are are developed concurrently. The issues of concurrency may be considered in measuring .programmers efficiency. The easiness of use of the project is depends on how the internal components are coupled each other. The effects of module coupling may be considered when measuring .programmers efficiency.

REFERENCE

[1] Adrienne Andrew, Gaetano Borriello Understanding Self Efficacy and the Design of Personal Informatics Tools. CHI 2011, May 7–12, 2011, Vancouver, BC, Canada. ACM 978-1-4503-0268-5/11/05.

[2] Albert bandura, A. Self-efficacy: Toward a unifying theory of behavioral change. Psychological Review 1977 84, No 2, 191-215.

[3] Barry W. Boehm Software pioneers, Software engineering economics, Springer- Verlag New York, Inc.New York, NY, USA©2002

[4] Beizar, boris Software testing techniques.Edition2, Dreamtech New Delhi -110002. 2003.

[5] Boehm.B, Software Engineering Economics, Prentice- Hall, Englewood Cliffs, NJ, 1981.

[6] Bulletin OEPP/EPPO Design and analysis of efficacy evaluation trials Bulletin (2012) 42(3), 367–381 ISSN 0250-8052. DOI: 10.1111/epp.2610.

[7] Daniel.J Cougar,Measurement of creativity of I.S.products. University of Colorado,[0073-1129-/92$3.001992IEEE]

[8] Houcheng Zhang, Shanhe Su, Guoxing Lin, Jincan Chen Efficiency Calculation and Configuration Design of a PEM Electrolyzer System for Hydrogen Production Published Int. J. Electrochem. Sci., 7 (2012) 4143 - 4157. 1 May 2012

[9] Kenyer Domínguez, María Pérez, Anna C. Grimán, Maryoly Ortega, Luis E. Mendoza, Software quality model based on software development approaches…

[10] Ortega, Maryoly, Pérez, María , Rojas, Teresita. Construction of a Systemic Quality Model for evaluating a Software Product. Software Quality Journal, 11:3, pp. 219-242. Kluwer Academic Publishers, 2003...

[11] Pressman.R.S, Software Engineering: A Practitioners Approach,edition 7 Publisher (McGraw Hill, NY, 2005).

[12] Prasad K.V.K.K,Software Testing Tools: Covering WinRunner, Silk Test, LoadRunner, JMeter and TestDirector with case studies. Dreamteach publication.

[13] Sdstin, L.firide, Effective of software testing tools.

[14] Supraja1.K,Srinivasa Reddy.A, Lakshmi Tulasi.R Evaluating Efficacy of Forward Error Correction Coding, India International Journal of Computer Trends and Technology- July to Aug Issue 2011 ISSN: 2231-2803 .

[15] Stu Messier, Measuring the Effectiveness of FPGA Programming Languages csm1@seas.wustl.edu

[16] Sven Manske,H,Ulrich Hoppe Duiburg Automated indicators to assess the creativity of solution to programming exercises.[978-1-4799-4038-7/14$31.00"2014IEEE].

[17] Tamanna Siddiqui, Munior Ahmad Wani and Najeeb Ahmad Khan, Efficiency Metrics Vol. 3 No. 2; ISSN 0973 – 5658 377, July – December, 2011;

[18] Warren F. Kuhfeld Experimental Design: Efficiency, Coding, and Choice Designs 53 MR-2010C.

[19] Xiaojin Zhu, Zhiting Xu and Tushar Khot How Creative is Your Writing? A Linguistic Creativity Measure from Computer Science and Cognitive Psychology Perspectives, USA 53706 CALC '09 Proceedings of the Workshop on Computational Approaches to Linguistic CreativityISBN: 978-1-932432-36-7

[20] William E.PerryEffective Methods for Software Testing, 1995. ISBN: 978-0-7645-9837-1 3rd Edition

[21] Wasif Afzal ·Ahmad Nauman Ghazi·Juha Itkonen· Richard Torkar·Anneliese Andrews·Khurram Bhatti An experiment on the effectiveness and efficiency of preparatory testing © Springer Science+ field Media New York 2014.

[22] Warren F. Kuhfeld, Experimental Design: Efficiency, Coding, and Choice Designs MR-2010C-project.com/Articles/31029/Efficient-Software-Development.

[23] https://www.google.co.in/search?
Sclient=psyab&site=&source=hp&btnG=Search&q=efficient+software+development+methodology

[24] http://www.codeproject.com/Articles/31029/Efficient-Software-Development