ISSN: 0976-3104
SPECIAL ISSUE: COMPUTER SCIENCE

ARTICLE

HYBRID APPROACH OF CODE ANALYSIS AND EFFORTS CALCULATION FOR SOFTWARE RELIABILITY GROWTH MEASUREMENT AND COST ESTIMATION

Abhyuday Nivrutti Patil*, Amol Kadam , Sachin B. Wakurdekar, S. D. Joshi

Department of Computer Engineering, Bharati Vidyapeeth (Deemed to be university) College of Engineering, Pune, INDIA

ABSTRACT

Background: *Software reliability modeling of information gathered amid the testing of an expansive scale mechanical framework (System T) was utilized to gauge software quality from the client point of view. Software Reliability is the likelihood of disappointment free software task for a predefined timeframe in a predetermined situation. Software Reliability is additionally a vital factor influencing framework reliability.* ***Methods:*** *Framework proposed software reliability model which comprises of code examination and cost estimation model. There are a few software estimation attributes, for example, Function point, Line of Code, and models Constructive Cost Model (COCOMO).* ***Results:*** *We consolidate investigation of code and cost estimation to compute proficiency of give software. For the each model we will likewise reveal insight into the distinctive segments, and how their parameter impact on the precision of software cost estimations improvement ventures.* ***Conclusions:*** *This paper displays a successful software reliability model, efforts calculation and cost estimation model which got from different characteristic.*

KEY WORDS
Software reliability model, code analysis, COCOMO II, efforts calculation

INTRODUCTION

The business of software should to be productive. Because of fast change in innovation, usage of complex software frameworks at less expensive cost and the desire to keep up better quality software are a portion of the real difficulties for the software organizations. One of the hardest works is cost estimation, in the field of software engineering. It is the estimation of aggregate cost required in creating software.

Cost estimation incorporates the procedure or techniques that assistance us in anticipating the real and aggregate cost that will be required for our software and is considered as one of the mind boggling and testing movement for the software organizations. They will likely create software which is shabby and in the meantime convey great quality. Software cost estimation is utilized fundamentally by framework experts to get a guess of the basic assets required by a specific software venture and their calendars. Critical parameters in assessing cost are estimate, time, and exertion and so on. Procedure of software estimation essentially centers around four stages. At first we appraise software estimate, at that point the required exertion after this we infer the timetable and finally figure out cost of the software. Constructive Cost Model (COCOMO) is a mainstream and broadly utilized Algorithmic arrangement of models [1]

We evaluate system in two parts. Code analysis and cost estimation. In code analysis, first we complied software for bug report. We have used five parameters for software reliability i.e. function point, statement coverage, operator analysis, reserve word analysis and comment line analysis. COCOMO II model consists early design model, reuse model and post architectural model.

Received: 28 June 2018
Reviewed: 16 July 2018
Accepted: 25 July 2018

RELATED WORK

This paper shows the method to dissect unwavering quality with the blend of testing time analyser, test scope, and dependability analyser. Through this method we have endeavoured to break down the software from its interior structure i.e. coding structure. And if the code structure is enhanced the unwavering quality consequently increments. This paper unquestionably turns into a helpful factor in testing conditions so developer can distinguish the multifaceted nature inside the code and attempt to make it straightforward and dependable so software is exceptionally solid before genetic algorithm it for the testing [2].

Reliability of Software has been significant Issue in Software industry. Disappointment Free Code Development is to be accomplished inside time of Software adaptation updates. Despite the fact that this offer ascent to various issues. To beat these deformities Software area has concocted software Reliability and Growth Models (SRGM's). Displayed on this Concept a mathematical Analyzing and Modeling framework for numerous arrival of Software item is planned and created. Testing is impossible consistently and research tries to limit testing, esteem minimization reliability expansion are results of research [3].

***Corresponding Author**
Email:
patilabhay@outlook.com
Tel.: 7276299900

Software Reliability is characterized as the likelihood of free-disappointment activity for a predetermined timeframe in a predefined situation in a given timeframe under indicated conditions. Software Reliability Growth models (SRGM) is constructed for evaluate software reliability parameters, for example, number of residual issues, software disappointment rate and software reliability. Software testing can be characterized as a procedure to recognize blames in the totality and worth of created PC software. Testing

IIOAB JOURNAL
ISSN: 0976-3104

is vital in guaranteeing the nature of the software by recognizing shortcomings in software, and potentially expelling them [4].

The primary test in the improvement of huge and complex ventures is the cost estimation with more precision. Numerous estimation models are presented over the span of time, which infers that software cost estimation isn't exact and new techniques or models ought to be proposed all the time. A point by point diagram of existing software cost estimation procedures or models are given by this model. The models are significantly characterized in two kind's algorithmic and non-algorithmic models. Enter factor in the improvement of new software is the choice of the reasonable cost estimation model and it additionally delineates the qualities and shortcoming of different cost estimation models. The fundamental goal is to give a relative writing examination of different cost estimation strategies or procedures in this paper [5].

The industry of software ought to be productive. Because of fast change in innovation, execution of complex software frameworks at less expensive cost and the inclination to keep up better quality software are a portion of the significant difficulties for the software organizations. Scientists have proposed different strategies for cost estimation. This paper gives knowledge into the different models and methods utilized as a part of evaluating cost of the software. The advantages and disadvantages of the current cost evaluating procedures have been featured in this paper. There is all things considered no single strategy which can be viewed as the best technique so in this paper it is proposed that a blend of the strategies ought to be utilized to get a precise cost gauge [6].

Reliability is a most important point in software system. To achieve better performance reliable software need to work properly in specified environment. It is impossible to test the software for making it hundred percent defect and bug free. Many software reliability growth models support the accessibility of the software reliability. Software reliability growth model evaluates the software reliability parameters like number of remaining faults, software failure rate and software functionality and performance [7].

Software reliability is a vital part of software quality. And accomplishing reliability is the need of the present worldwide rivalry. Estimation and prediction are the approaches to break down software reliability. Software reliability growth demonstrate is utilized to assess the reliability through mathematical articulation and it additionally used to translate software disappointments as a random procedure. This paper portrays a novel software reliability growth display in light of non-homogeneous Poisson process with taking into consideration defective investigating. Keeping up and enhancing nature of the software is an extremely troublesome errand because of numerous components like equivocal necessity determination, absence of required assets and so forth. Numerous reliability growth models have been proposed as of not long ago as indicated by various setting and subsequently there is no all-around acknowledged model. Software quality metric features the quality parts of item, process, and task. As there is corresponding connection amongst quality and reliability, analyzing quality measurements is additionally an approach to evaluate reliability. Thus, we investigate quality measurements alongside keeping up the deformity database [8] [1].

MATERIALS AND METHODS

Software reliability is the possibility of the software causing a system letdown over some specified operating time. Software does not fail due to wear out but does fail due to defective functionality, timing, sequencing, data, and exception handling. We categorize model code analysis and cost estimation. In first part, i.e. compiler will compile given uploaded file. Complier will debug file. Admin will select any one software category from organic, semi detected and embedded.
We have selected five parameters for code analysis. Following are parameter for analysis.

- Function path: -We define functions, i.e., and method in given code
- Statement coverage: - We define statement i.e. loops, if-else function, for loop, switch cases etc.
- Operator analysis: Operator like plus, minus, equal to, less than and greater than etc. are define in this cost driver.
- Reserve word analysis: - The words which are allocated in java library and cannot be used for other purpose. These word are defining in this cost driver.
- Comment line analysis: - We calculate number of comment used in given code of line.

Further, Cost estimation is calculated using COCOMO II model. In early design model, file will be uploaded, software category is selected. We used reliability calculator attributes like product reliability and complexity, reuse required, platform difficulty, personal capability, personal experience, schedule, support facilities etc.
In reuse model, reuse LOC will be consider for this model. We consider three attributes i.e. ASLOC, AT, ATPROD.
ASLOC is the number of adaptive LOC of reusable components AT is the percentage of adapted generated code
ATPROD is productivity of engineers integrating the code, usually approx... to 2400 LOC/ month/person.
In post architectural model, 17 attributes are consider i.e. program capability, required system reliability, complexity of system modules, extent of documentation required, size of database used, required percentage of reusable components, execution time constraints, volatility of development platform, memory constraints, capability of project analysis, personal continuity, programmer experience in project

IIOAB JOURNAL
ISSN: 0976-3104

domain, analyst experience in project domain, language and tool experience, use of software tool, development schedule compression, extent of multisite working of inter site community.

COCOMO II model use above model. It contains additional attributes for calculating exact cost estimation. Our proposed system utilizes all Post Architectural attributes combine with proposed scale drivers likewise precendentedness, development feasibility, architecture, team cohesion, process maturity. Also in addition to these attribute, we proposed our attributes they are frequency of program specification change, process performance, database complexity, code skill level. Using these parameter, report will generate which shows estimated cost of software. Refer below [Fig. 1]: Architecture diagram of proposed system.

COCOMO strategies make utilization of conditions and arithmetic to play out the process of estimation... Constructive cost model, one of the prominent and broadly utilized algorithmic model for the estimation of cost and in the meantime get the calendar of a creating software was given by Barry Boehm and is known as the Constructive Cost Model (COCOMO). The parameters and conditions that are utilized as a part of this model are acquired through past software ventures. The span of code is typically given in KLOC (thousand lines of code) and the acquired exertion is in Person Months (PM).

There is three models of COCOMO which is proposed by Boehm as follows:

1) Basic COCOMO – It is the first of the COCOMO set of models, formula used for this model is

Effort = a*(KLOC) b

KLOC is the code size and the constant are a and b. The constant value is depends on the type of project, whether the project is semi-detached, organic, or embedded.

2) Intermediate COCOMO – In this model get the nominal value of constant a and b and effort estimation which is differs from earlier basic COCOMO. Formula used as:

Effort = a*(KLOC)b * EAF.

Here the effort adjustment factor is represented by EAF

3) Detailed COCOMO – This model works separately on each sub-system and serves as a boon for large systems made up of non-homogenous subsystems. To predefine and stable the software requirement Constructive Cost Models believes the system.

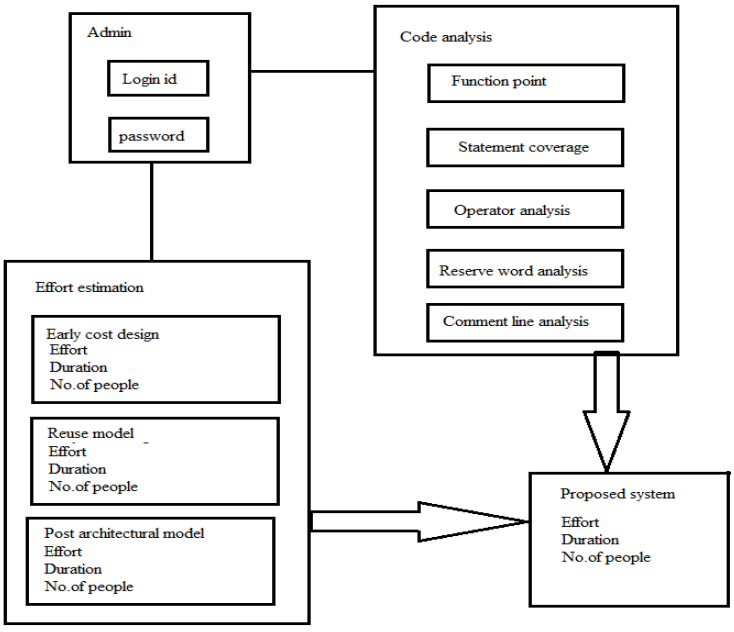

Fig. 1: Architecture diagram of proposed system.

RESULTS

Admin will be provided with login credentials. Admin will enter login id and password to login to application. For Code analysis, file is uploaded and attributes are selected for analysis. Following is report of code analysis. It is shown in [Fig. 1]. We have calculated threshold value for current project and actual value of current project is given. For example, Threshold value of current project is 7824 ad obtain value is 276 for function path. Likewise, report will be generated for all attributes. Here, analysis table shows the reliability of the software in Fig 2.

IIOAB JOURNAL
ISSN: 0976-3104

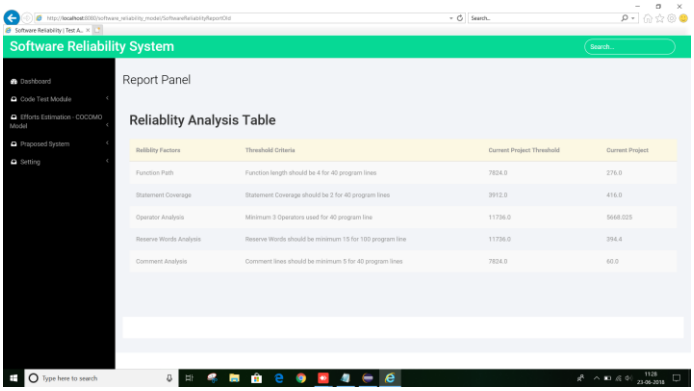

Fig. 2: Reliability analysis table

Here, graphical representation is shown for each attributes. Threshold value and obtain value are given on x– axis. Graph of statement coverage, functional point, operator point, reserve word and comment analysis is shown in [Fig.3]

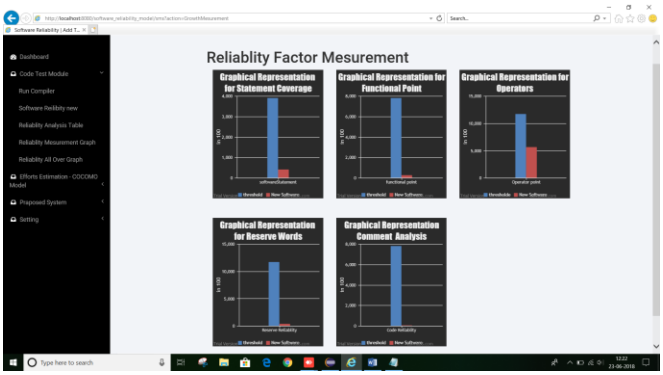

Fig. 3: Graphical representation of reliability factors

0
Here, pie chart is generated for all above factors as shown in [Fig.4]

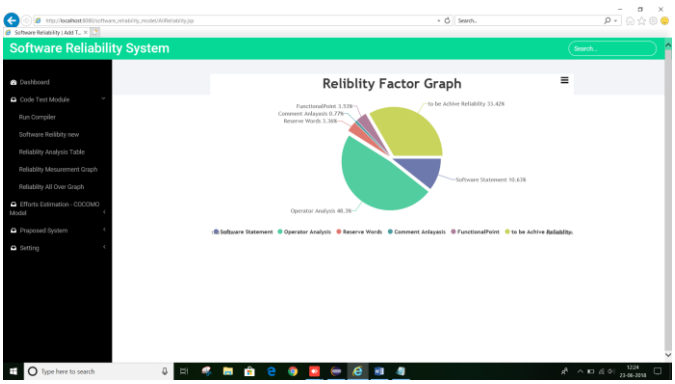

Fig. 4: Reliability factor graph

To estimate cost, we have compare exiting model with our proposed system. In existing system, report of early design model, reuse model, post architectural model is shown. Using these model, LOC, KLOC, efforts, no. of people and duration is shown. Our system gives estimate all parameter. As shown, Proposed system is efficient as efforts, people and duration is less than existing system. Repost is given in [Fig.5]

IIOAB JOURNAL
ISSN: 0976-3104

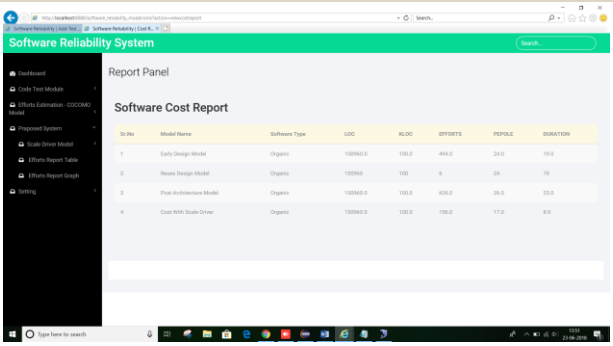

Fig. 5: Effort calculation and software cost report

CONCLUSION

Arranging and planning of software venture is generally influenced by cost estimation, hence it is a fundamental process in software estimation. Our point ought to be to create such software which are both shabby and offer a decent quality and solid. There are numerous strategies for checking reliability and assessing cost yet as unmistakably we can't view any single system as the best one as every one of the procedures have their own preferences and impediments. Endeavors ought to be made to utilize a blend of the estimation strategies to touch base at a superior cost and quality estimate. To create solid estimates, it is needed legitimate learning and comprehension of every procedure and the connection between the software qualities of each.

CONFLICT OF INTEREST

No Conflict of Interest

ACKNOWLEDGEMENTS

To design and build proposed result paper on “HYBRID APPROACH OF CODE ANALYSIS AND EFFORTS CALCULATION FOR SOFTWARE RELIABILITY GROWTH MEASUREMENT AND COST ESTIMATION" has been prepared by Mr.Abhyuday N. Patil and Mr.Amol Kadam. Author would like to thank his faculty as well as his whole department, parents, friends for their support. Author has obtained a lot of knowledge during the preparation of this document.

FINANCIAL DISCLOSURE

None

REFERENCES

1. Patil AN, Kadam A. [2017] Software Reliability and Cost Estimation Model, JETIR, 4(6):25-29.
2. Mengmeng Z, Pham H. [2017] A multi-release software reliability modeling for open source software incorporating dependent fault detection process. Annals of Operations Research: 1-18.
3. Chi j et al. [2017] Defect Analysis and Prediction by Applying the Multistage Software Reliability Growth Model. Empirical Software Engineering in Practice (IWESEP), 8th International Workshop on Empirical Software Engineering in Practice (IWESEP). DOI: 10.1109/IWESEP.2017.16
4. Choudhary A, Singh A, Sangwan OP. [2016]. Software reliability prediction modeling: a comparison of parametric and non-parametric modeling. 6th International Conference - Cloud System and Big Data Engineering (Confluence) DOI: 10.1109/CONFLUENCE.2016.7508198
5. Chengyong M, Li Q.[2016] A Testing-Coverage Software Reliability Growth Model Considering the Randomness of the Field Environment. IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C). DOI: 10.1109/QRS-C.2016.62
6. Iqbal J. [2017] Software reliability growth models: A comparison of linear and exponential fault content functions for study of imperfect debugging situations, Iqbal, Cogent Engineering, 4: 1286739
7. Kumar A. [2016] Software Reliability Growth Models, Tools and Data Sets- A Review, ISEC '16, February 18-20, 2016, Goa, India ISBN 978-1-4503-4018-2/16/02