

Justifying Quality in Software Cost Estimation

Gande.Shankar

Research student, Rai university ,Ahemadabad,India:

Dr. Lokesh kumar Sharma

Scientist-B, NIOH, Ahemadabad India

ABSTRACT: *software research mainly cost estimation became complex problem. many researchers applied different methods, but artificial intelligence based techniques have come in limelight which are promising to solve given problem. Case based reasoning has emerged recently is promising approach, with comparable accuracy and it is potentially helpful for people to understand and apply. This paper propose model for cost estimation relies on determining the software features defined by software quality requirements specification to the analysis at an early stage of the software development with support of case-based reasoning tool*

1.INTRODUCTION: software quality became increased attention and need to improve quality software products within cost and time became highly prioritized. [1] Delivery of unsatisfactory and unsuitable software leads to waste of money, time and consequently loss of productivity .firm may bear cost of cost of replacing the software but software is too difficult to learn by the users requirements that need to acquire policies like cryptographic controls for the protection of the information transmission in order to provide enhanced and more realistic results when understanding the price. Estimating process may be included to identify method based on quality attributes that can be specified and quantified by the software project managers in the early phase of the software development. Introduce case – based reasoning tool with just enough of each attribute to satisfy the desires of the system. Relative number of quality attributes may be used to determine cost of the software.[2] In this paper, the next section described about the term artificial intelligence and defying of . Then in next section described existing quality models and justification for choosing specified quality attributes in next section provided introduction of proposed model and its mechanism and equations finally conclusion is explained.

2.CASE BASED REASONING: AI techniques may be adapted for assisting many industries.[3] AI is simplest form of teaching of computers to solve problems. Human reasoning can be reviewed on the collection of experiences may be trend address such problems. human being solving problems similar or related problems in the dynamic memory. When new type of problem is faced , the knowledge is require to solve the problem . the gained knowledge information in the dynamic memory to be used in future to solve similar problems. [4]This type solving method can be defined as case based reasoning. case based reasoning is problem solving paradigm which uses the specific knowledge of earlier experiences and finding similar past case.[5] And reusing its solution in the new problem situation. CBR is not model based approaches to use rules; the developers does not have to explicitly define causalities and relationships with specific domain.

3. SOFTWARE QUALITY: software quality comprises all characteristics and significant features of a product or an activity which relate to the satisfying of given requirements.[6] Software ability that bear on to satisfy given needs and the degree to which software

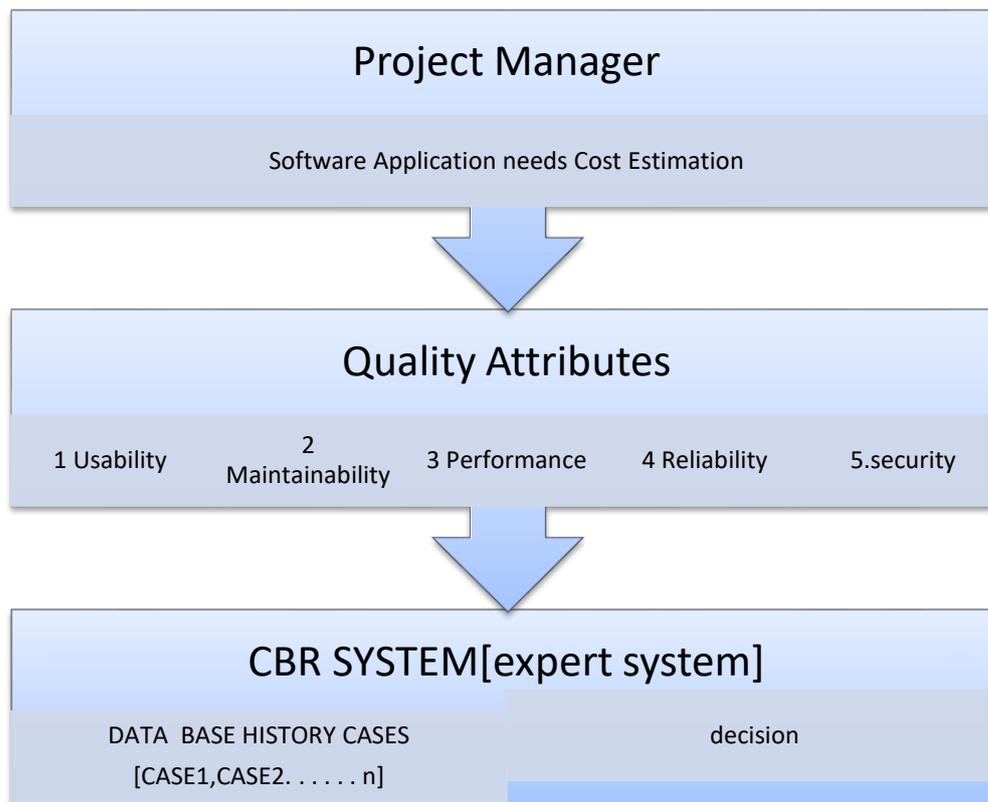
possess a desired combination of attributes. The degree to which a customer or user perceives that software meets expectations. Software market is rapidly growing, users are often dissatisfied with software quality. User satisfaction is often considered as critical outcome of quality management. To address the issues of software product quality, software quality attributes is defined but rather the most important attributes need to be identified and specified. Software is usable, correct and reliable then efficiency, interoperability and integrity can be considered. meeting the non-functional requirements in the determination of the system's perceived success or failure today's task is to fulfill functional requirements[7] and also characteristics which are non-functional requirements such as portability and extensibility which may affect the global architecture. quality depends on quality measurement robust, reliable and efficient may be defined as quality attributes. Unfortunately in spite of many approaches some times quality implementation more problem. We are unable to provide solution..

3.1 SOFTWARE QUALITY MODELS: software quality models is something perceived by the user, it establishes a framework to perform some kind of measurement of the specific desirable characteristics that are needed in the final system. product operation refers to the ability[8] of system efficiently operated and capable to provide results required by the user. Product transition refers to distributed processing, together with rapidly changing hardware-portability, reusability and interoperability. The McCall model has got advantage is the relationship created between its quality characteristics and disadvantage is its lack of consideration for the functionality of the system. Another model is the ISO/IE9126, which classifies software quality into four types namely process quality, internal quality, static and dynamic model and external quality. ISO 9126 model splits into six quality characteristics they are functionality, usability, reliability, effectiveness portability and maintainability. the important features of this model is to quality evaluation decomposes the concept of quality in to set of lower level quality characteristics that recognizable properties of a product which refine quality into something more concrete and measurable. it defines the internal and external quality characteristics of a system.[9] It defines a three-level hierarchical structure of quality concepts. Even evaluation procedures are defined in a separate standard to explain procedures for conducting product evaluations. finally ISO/IEC 9126 is preferable board consensus and widely accepted by researchers and used in practice. Dromey's model consisted of three types of constructs with causal linkages between them which are, set of components, set of high level quality attributes, set of tangible, quality-carrying properties of components this model may be better than ISO/IEC9126 model. Most of the software quality definitions, the software quality attributes, the various quality models used for measuring these attributes were reviewed and the software quality metrics were discussed in detail from two different views. Who defined software quality metrics as a subset of software metrics that focus on the quality aspects of the process and product metrics more than project metrics. product metrics that quantify useful attributes of the products generated like usability, reliability maintainability and portability. Process metrics that quantify useful attributes of the software development process and its environment. people metrics that quantify useful attributes of those generating the products using the available processes research is to produce a metric that will guide developers in choosing techniques to achieve the desired product quality. reliability is expressed in Mean Time Between Failure[MTBF] parameter, which is usually specified in hours. if quality can change quickly relative to new threats or changes in the operating environment. Then quality is based on attribute set on the execution environment, then in turn means that the environment can potentially modify any attribute's value for given piece of software. Attaining quality requires for each attribute according to its importance to the system into which the software will be embedded. The key attributes would probably be reliability, performance, safety, fault tolerance and availability. quality attributes in percentages and then be able to get a cost estimate for such future requirements as it is assumed that an attribute is a set of required feature model

3.2 Justifying the Selection of the Following Quality Attributes

Software quality requirements or program specifications as readability, fault-tolerance, the data which I when software exhibit undesirable[10] behavior, the data which is processed, the machinery on which the software runs. The quality requirements needed for a software product relies on the environment that it would be implied the accessibility, attractiveness; learn ability understandability, user friendliness maintainability changeability scalability, testability traceability. Authors suggest that the most important attributes that should be considered at the analysis of the project requirements are usability, maintainability, performance testability. adequate comments so that their purpose is clear or deviations for logical flow adequately commented. Feasible to perform difficult operations. And user interface responsive may be slow. If user interface self explanatory, usability questions include learn ability, readability and understandability requires attention to the needs of the novice and unlimited users. The uninitiated user is one who has no previous experience with software. The novice user has either had some experience with similar software readability is characterized, concise code that is immediate understandable, understandability is the capacity of the software product to enable the user to understand whether the software is suitable, and how it can be used for particular tasks and conditions of use. It is the extent that the software is convenient and practicable to use, which is affected by such human-computer interface. the component of the software which has most impact on this is the graphical user interface. The requires that the purpose of the design and user documentation must be clearly written so that it is easily understandable which is obviously subjective in that the user context must be taken into account. Next dealing with such a difficult software interface leads to unusable software where you end to replacing it by new software.. it is defined as the capability of the software product to be modification a she corrections improvements or adaptation of the software to change in environment. Performance is the key attribute for a software system, it expresses its functionality. It has a practical significance as it describes the number of operations that can be done or completed in a given time and the delay in user operations request and to ensure minimum number of customers is capable to serve. And dealing with effectively is measured by evaluating process speed and response time, the degree to which something effectively uses its resources. These resources may include all type of resources. It is important to check the extent that the software fulfills its purpose without waste of resources in the sense of memory utilization and process speed. [11]It is defined as the capability of the software product to enable modified software to be validated. Requirements to what extent complies with specifications and, it supports the evaluation of the software performance. reliability is very important to determine the extent to which the software system can be expected to perform its intended functions satisfactorily under stated conditions. Reliability is parameter value greatly affects the development cost. software to economic and production processes. [12] its important is the need to assess the software reliability is quality factor, it can be measured such as mean time between failures (MTBF) that can be specified as the number of failures during given time.[13] Reliability increases as the(MTBF) increases and also availability of the system which is the probability that the system is operating at a specified time. modern software systems such as the aircraft control financial loss can result from their failure. Secure system is a system which does exactly performs.

5. PROPOSED MODEL: proposed model may be used to determine for effort estimation to achieve to answer questions relating to quality attributes 1. Usability 2. Maintainability 3. Performance 4. Reliability.

Proposed model

5.1 MODEL MECHANISM: a filling in there required for the new project by project managers has to the model all quality requirements with relevance to each quality attribute that to that are required for the new project by filling in the given to him effort so the model able to reason with case base to suggest the appropriate effort for the project, using the nearest neighbor matching search technique to the entered threshold value. The project manager can accept the effort suggested or he can modify the result by himself and store the result

Steps as follows

The project manager has to input to the model all quality requirements with relevance to each required for the new project.

And then calculate the percentage of needed quality attributes

Then cost estimate

]

6. CONCLUSION: case-based reasoning provides solution to predict cost estimate. This paper proposed software project effort estimation with quality attributes . quality attributes measures are discussed. This model works for limited period only. Requirements may Be achieved with development effort as it is linear relation for more desirable high quality non functional requirements need to be attain research is needed to estimate the weight age of software quality attributes. project managers is acquainted to the level quality attributes in applying desired system.

1. Mukhopadhyay,T,Kekre,S;software effort models for early estimation of process control applications .IEEE Transactions on software engineering, 18(10),915-924 (1992)
2. Akingbehin,kiumi: A quantitative supplement to the definition of software quality in proceedings of the 2005,3rd ACIS Int. conference on software engineering research, management applications(SERA`5), 2005
3. Partridge, D : Artificial Intelligence and software engineering, Alex publishing corporation, New Jersey, USA (1991).
4. Pradeep, R : case based reasoning :an implemented methodology, Auburn University, Technical Report CSE 95-04,(March 1995).
5. Kolodner,J : Case based reasoning. MorganKaufman Publishers,California. (1993)
6. Fitzpatrick, Roman: Software Quality: Definitions and Strategic issues. School of Computing Report Advanced ResearchModule,StaffordshireUniversity.(April 1996).
7. Jung, Ho-Won, Seung- Gweon, Chung, Chang-Shin: Measuring software product Quality: A Survey of ISO/IEC9126. IEEE Computer Society. (2004)
8. Losavio, Francisca ,Chirinos, Ledis, Perez, Maria A: Quality Models to Design Software Architecture. In proceedings of the Technology of Object- Oriented Language System. (2001).
9. Suresh, N, Babu, A.J.G: Software quality Optimization: a linear programming Model, Association of management proceedings Vol. 10, No 2, pp.219-242. (2003).
10. Dromey,R.G.: A Model for Software Product Quality. IEEE Transactions on Software Engineering, Vol. 21,No. 2, pp. 146-162. (1995).
11. International Standard :ISO/IEC 9126-1. Institute of Electrical and Electronics Engineers part1,2,3,:Quality model.(2001)
12. Pressman, Roger: Software Engineering a Practitioner`s Approach McGraw-Hill,Inc (1992)
13. Burris,Eddie: Software Quality Mangement, The Practice of Software Engineering (2005)