International Journal of Engineering Research and Development
***e*-ISSN : 2278-067X, *p*-ISSN : 2278-800X, www.ijerd.com**
Volume 12, Issue 11 (November, 2016), PP.73-81

Software Quality and Productivity Enhancement Model

Sanjay Kumar[1], Rahul Rishi[2], Rajkumar[3]
[1]*university Campus School, Mdu Rohtak, India,*
[2]*university Institute Of Engineering and Technology Mdu. Rohtak, India.*
[3]*university Institute Of Engineering and Technology Mdu. Rohtak, India*

Abstract: In this paper, architecture is designed in such a way that will enhance quality of specification of modules that result to enhance quality of software module design. It does not directly depends only upon specification factor (feasibility study), but also depends upon some other attributes that may be directly/indirectly(Reusability, Under-stability, Correctness, Completeness, Well documented, Flexibility). Hence quality of specification is directly proportional to design of software module to enhance reliability of modules. The proposed model designed in such a way that it remove the maximum faults or errors in early phase of development using Risk analysis(by cause and effect diagram), fault prediction, validate SRS by using testing and evaluation.

Keywords:- Quality, productivity, cause and effect diagram, fault prediction, reliability

I. INTRODUCTION

Software quality is one of the pivotal aspects of a software development company. Software quality assurance starts from the beginning of a project, right from the analysis phase. According to the *Webster's Dictionary,* "quality" is "a degree of excellence; a distinguishing attribute." That is, quality is the degree to which a product lives up to its performance, endurance, maintainability, and other attributes that a customer expects to receive from purchasing this product. In present time, quality is defined and judged by the customer. Therefore, it acknowledges a customer-driven economy. It focuses on continuous process improvement to achieve high quality of product (or service). It suggests that any improvement that is made in the business, be it a better design of a component or a better process of a system, will help to improve the quality of the organization and the quality of the final product. Under this philosophy, the view of quality is very different from the traditional one. The proposed model includes the systematic process of assuring that standards and procedures are established and are followed throughout the software development life cycle and test cycle as well. The proposed model ensure that hard work done in the beginning(Requirement and analysis phase) of the product not only reduces the risk, also results in faster and accurate development in the further processes.

II. LITERATURE REVIEW

Quality can be described in five different perspectives: user view, transcendental view, manufacturers view, product view and value based view. Quality must be monitored from the early phases of development such as requirement analysis, design, implementation and maintenance phases. A number of models are presented by software developers to enhance the quality. Some of the standard models are listed here:

McCall's et al model (1979) is one of the earliest model regarding the productivity gave emphasis on Triangle of Quality. McCall's quality model [1] addresses three perspectives: (i) Product operation (ii) Product revision (iii) Product transition. It covers some characteristics of quality these are efficiency, integrity, re-usability, reliability, usability, accuracy, testability, flexibility, maintainability, interface, transferability, interoperability.

Boehm Quality Models (1976, 1978) attempts to define equality of software using a predefined set of attributes and metrics. Boehm classified [2] General utility into Asis utility, portability, maintainability, which are further classified into characteristics like device independence, self-contentedness, Integrity, Consistency, Device efficiency, Accessibility, Accuracy, Completeness, communication, Self-descriptiveness, Legibility, Structuredness, Conciseness, Augment ability.

FURPS - The following characteristics are considered in FURPS model are Functionality (include feature sets, capabilities and security), Usability (human factors, aesthetics, consistency in the user interface, online and context-sensitive help, wizards and agents, user documentation, and training materials), Reliability (frequency, severity of failure, recoverability, predictability, accuracy, mean time between failure), Performance (speed, efficiency, accuracy, throughput, response time, availability, recovery time, and resource usage), Supportability (testability, extensibility, maintainability, compatibility, adaptability, configurability, serviceability, installability, Localizability)

Dromey's Quality Model [3] presented a framework to evaluate requirement determination, design and implementation phases. The framework consists of three models, i.e. Requirement quality model, Design quality

model and Implementation quality model. The high-level product properties for the implementing quality model include (i) Internal measures (ii) Contextual measures (iii) Correctness measures (iv)Descriptive measures. It includes high level quality attributes: functionality, efficiency, usability, reliability, maintainability, portability, reusability and process maturity.

Kitchenham (1996) defined quality as "Quality is a complex concept, because it means different things to different people, it is highly context-dependent. Just as there is no one automobile to satisfy everyone's needs, so too there is no universal definition of quality. Thus, there can be no single, simple measure of software quality acceptable to everyone. To assess or improve software quality in your organization, you must define the aspects of quality in which you are interested ISO 9000 provides guidelines for the quality assurance. ISO 9001: 2000 [4] structured gives the requirement forthe development of quality management system. The standard has eight clauses related to references, scope, terms & definition, Requirement (documents, quality manual, control of documents, control of records), management responsibility (commitment, customer focus, quality policy, planning, management review, responsibility of authority and communication), resource management, product realisation, measurement, analysis & improvement)

ISO 9126 [5] – ISO 9126 is an international standard for software development and evaluation. The standard is divided into four parts which addresses, the following subjects: quality model; internal metrics; external metrics and quality in use metrics. The ISO 9126-1 software quality model identifies 6 main quality characteristics, these are: Functionality (Suitability, Interoperability, Compliance, Accurateness, Security), Usability (Learnability, Operability), Reliability (Maturity, Recoverability, Understand ability, Fault tolerance,), Efficiency (Resource behaviour, Time behaviour, Analysability), Portability (Installability, Replaceability, Conformance,). Maintainability (Stability, Testability, Adaptability, Changeability)

According to R. Fitzpatrick(1996) Software quality is the extent to which an industry-defined set of desirable features incorporated into a product to enhance its lifetime performance. [6]

and then decide how you are going to measure them". [7] Eldon Y. Li.(2000) contrasts the Quality Seven (Q7) and the Management Seven (M7) tools commonly used in the TQM process. It describes the Deming's quality management concept and its fourteen point management method. It briefly explains the similarities between software development process and product development process. [8]

According to Card (2006), Simple Model of Productivity considers the entities such as Product, Process or Sub-process, Requirements, Value, Cost, and Effort.

According to Dr. Rao(2011) Seventy-two percent of the survey population was of the opinion that quality assurance should be introduced in all phases of the SDLC. The remaining 28% fell into a different category. Other trends that affected the inclusion of QA in all phases of the SDLC included (a) too much time spent in the process, (b) inconsistent methodologies, (c) over commitment. [9]

According to Amit C. Joshi(2012) Total quality management is not only a philosophy of work but also an ethic of workers. It is coming from the wisdom and the teachings of many quality improvement gurus. It has helped many companies to improve quality of products and processes, and in turn, increase the productivity and the profitability.[10]

According to David Chappell, Both functional quality and structural quality are important and they usually get the lion's share of attention in discussions of software quality. Yet the third aspect, process quality, is also critically important. The quality of the development process significantly affects the value received by users, development teams and sponsors and so all three groups have a stake in improving this aspect of software quality.[11]

According to SadiaRehman(2012) Global Software Development (GSD) is a modern software engineering paradigm adopted by many client organizations in developed countries to get high quality product at low cost in low wage countries. Production of high quality software is considered as one of the key factor in the rapid growth of GSD.[12]

KarineMordal(2012) states that most software quality metrics are defined at the level of individual software components, there is a need for aggregation methods to summarize the results at the system level. Second, because a software evaluation requires the use of different metrics, with possibly widely varying output ranges, there is a need to combine these results into a unified quality assessment. [13]

According to Wallin(2012), the effort of designing and evolving the architecture is often neglected during system development.

Mrinal Singh Rawatet al.(2012) examine the realm of software engineering to see why software metrics are needed and also reviews their contribution to software quality and reliability. Results can be improved further as we acquire additional experience with variety of software metrics.[14]

Dr. DeepshikhaJamwal(2012) discussed different quality models (McCall's Quality Model, Boehm's Quality Model, Dromey's Quality Model, FURPS Quality Model, ISO 9126 Quality Model.**)** and concluded that only "reliability" is the common attribute in all models. A criterion has been defined based on some questions in order to choose quality model for any organisation that will save organisation's time. [15]

According to IderpalSingh(2013) Different quality models include different factors in relation to functional and non-functional requirements but reliability is an important factor for quality measurement. It is important to have good software specification requirements to have best results. It should be noticed that phases like planning, requirement, coding implementation should be done in proper manner to have quality.[16]

According to Hongyi Sun et al.(2014) Except for the time to market, quality, productivity and cost performance are all significantly better with more re-used modules

SabaAwan et al.(2015)in their studiescompute multiple features of different software quality models and evaluated their relative significance.[17]

1. Proposed model:-

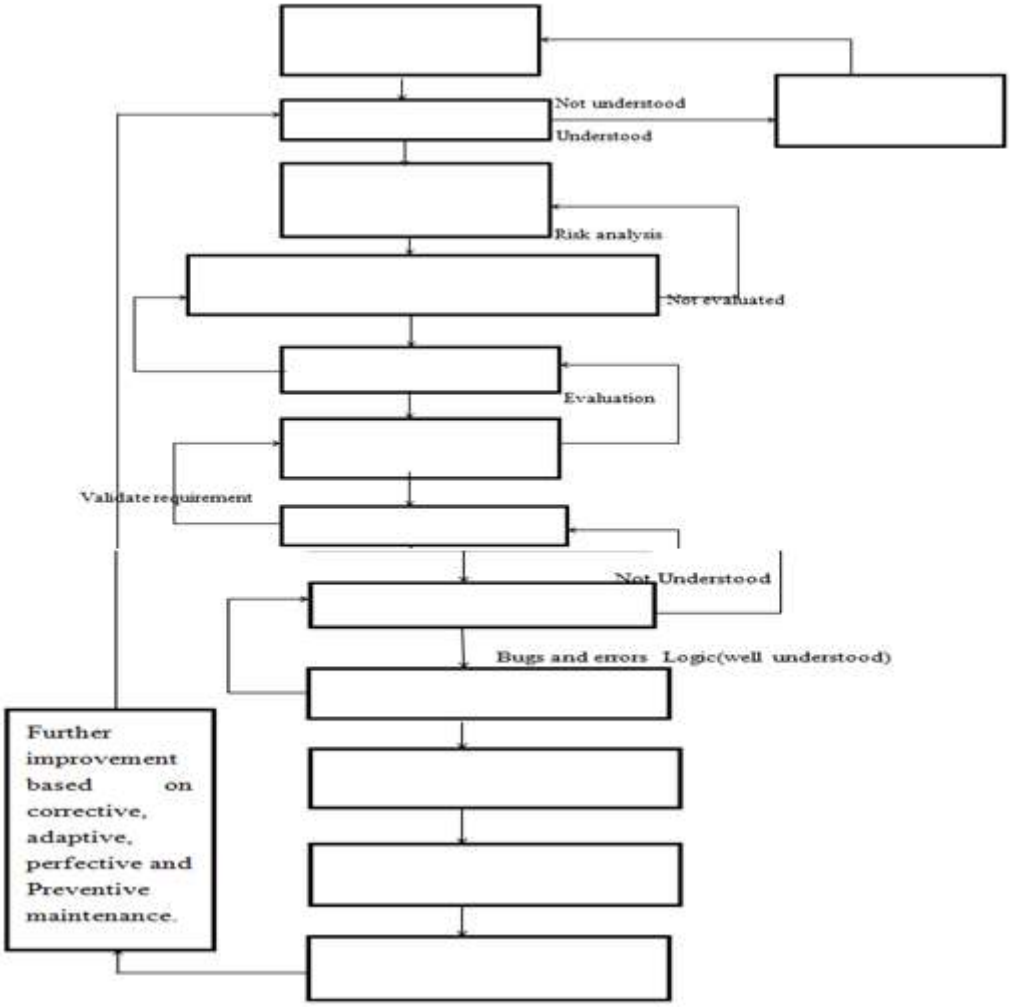

Fig. 1.Software Quality and productivity enhancement model

The proposed model includes the systematic process of assuring that standards and procedures are established and are followed throughout the software development life cycle and test cycle as well. The proposed model designed in such a way that it remove the maximum faults or errors in early phase of development using Risk analysis(by cause and effect diagram), fault prediction, validate SRS by using testing and evaluation. It includes

1.1 Requirement analysis:- it help the annalist to understand, interpret and organise the software requirement to assess its physical, logical, economical, legal, social feasibility, as well as completeness and consistency of the requirement. Requirement analysis also includes

- Communication
- Initiation
- Objectives and scopes
- Limitations

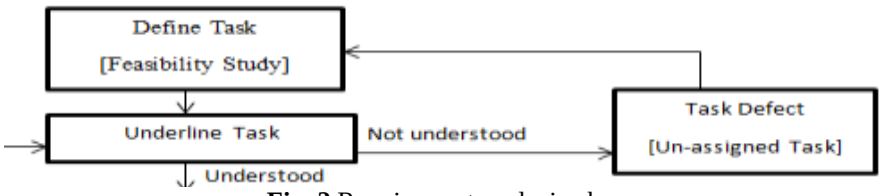

Fig. 2 Requirement analysis phase

In proposed model this phase includes Define task, underline task, task defects etc.

A "School Management Software" is designed using the proposed model. The main task is to develop school management software. The requirement analysis process is started, the entire requirement are collected from principal, staff and students as well as from their parents. The task is further defined and divided into subtask. Subtask is further defined and broken into its lowest level according to the requirement analysis. These task and subtask are further checked to identify any unassigned or hidden task in the underline task. The task that is not properly understood is moved to task defect. During this process we find 8 unassigned task(requirement) which are either not assigned properly or not as per the requirement. These tasks are further send to define task panel. These requirements are further defined again and verified that they are as per the requirement. This phase also helps to discuss the objectives and scope of the proposed software, what can be achieved with the proposed software and its limitations? As the entire requirement of the customer are properly defined and as per the requirement, the quality attributes(Reusability, Understability, Correctness, Completeness, Well documented, Flexibility) are assigned.

1.2 Property Specification:- In property specification quality attributes are applied. Quality comprises all characteristics and significant features of a product or an activity which relate to the satisfying of given requirements, Quality is achieved when all activities are carried out to satisfy the total requirements of every customer, internal as well external. Quality attributes are assigned to these task or sub task. The quality attributes are as follows.

i. Reusability
ii. Efficiency
iii. Understability
iv. Well documented
v. Correctness
vi. Completeness
vii. Usability
viii. Flexibility
ix. Integrity

These are the quality attributes that help to create SRS of required requirement of customer. These quality attributesspecifies and evaluates the quality of a software product in terms of internal and external software behaviour and their connection to attributes. The model categorizes software quality attributes into three categories such as: Operational specification (Correctness, Reliability, efficiency, Integrity etc.)Product Revision(Maintainability, Flexibility, Traceability etc.)Product Transition(Portability, Reusability, interoperability etc.)

1.3 Risk Analysis:- During risk analysis, not only analysis the risk associated with technical, potential, financial but also related to requirement of customer. Such kind of risk are validate by applying cause and effect diagram, to justify the quality and productivity of customer requirement i.e. used to find the alternate solution of the problem that is associated with highest risk(Requirement, time or cost). It helps us to think through causes of problem thoroughly. Its major benefit is that it pushes us to consider all possible causes of the problem, rather than just the one that is most obvious. This phase includes risk analysis by using cause and effect diagram, fault prediction, testing and evaluation.

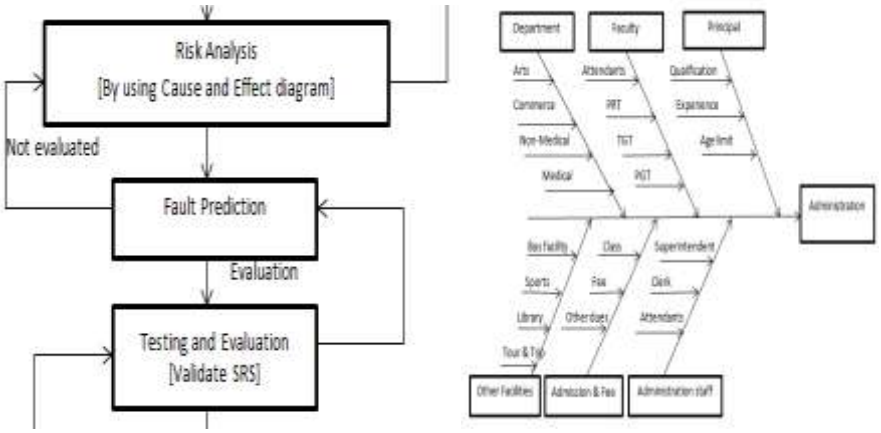

Fig. 3(a)Risk analysis phase and Fig. 3(b) use of cause and effect diagram to find defects.

1.4 Fault prediction model:- The major goal of this model is to detect faults as early as possible in the development life cycle. This model[11] helps us in better designing of modelling algorithm in incremental development, towards improving quality and productivity. Before the delivery of final SRS, to find the pre-release defects the fault prediction model is applied. It helps to validate the model.Before the final SRS, if any how any failure or faults remains after applying quality attributes, the cause and effect diagram is applied. This help to remove these faults. Before the delivery of SRS to design phase, to find the pre-release defects the fault prediction model is applied.

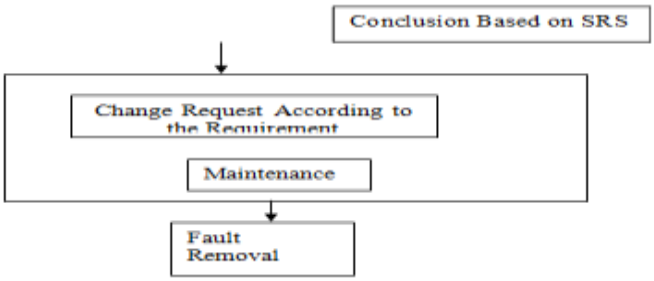

Fig. 4 Fault prediction phase

It helps to identify the process of providing input, how output comes and how it fulfils the requirement of user. To find projects limitation, the boundary analysis is done. Whatever number of times analyst is executing the process, it is running exactly that number of times or not. Independent path is verified and process validates whether every requirement gathered is once achieved or not. All validate requirement:-every requirement is checked n number of times or not. During this process three modules or components detected that are not validates during the system progress are moved back to the risk analysis for further consideration and defect removal.

1.5 Architectural Design:- on the basis of user requirements and the detailed analysis by fault prediction model high level design and detailed design is done. High level design includes overview of modules, number of modules and what these modules do, irrespective how they do? In detailed design view the logic of the modules is discussed.

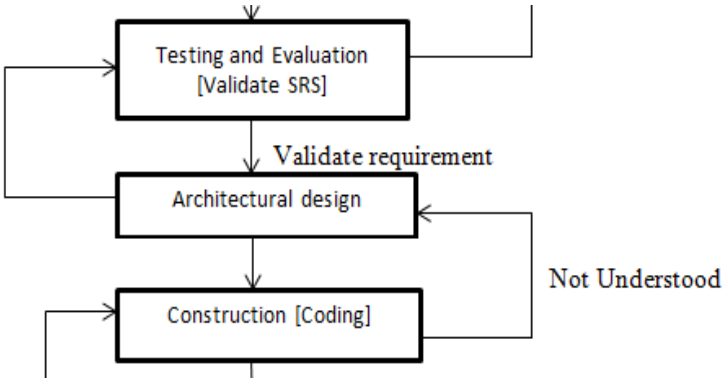

Fig. 5 Architecture design phase

In the design phase the SDLC process continues to move from what questions of the analysis phase to the how. The logical design produced during the analysis is turned into a physical design - a detailed description of what is needed to solve original problem. There are several tools and techniques that are used for describing the system design of the school management. These tools and techniques are: Flowchart, Data flow diagram (DFD), Data dictionary, Decision table and Decision tree etc.

1.6 Construction (Coding):- After completing the design documents, the work is divided in modules/units and actual coding is started. Since, in this phase the code is produced so it is the main focus for the developer. This is programming phase in which the programmer converts the program specifications into computer instructions. It's an important stage where the defined procedures are transformed into control specifications with the help of a computer language. The programs coordinate the data movements and control the entire process in project.

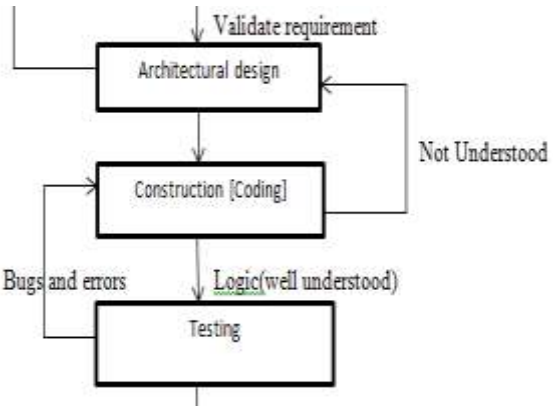

Fig. 6 Construction (coding) phase

A well written code reduces the testing and maintenance effort. Programming language effect the quality of project, right programming language should be chosen.

1.7 Testing :- After the coding is developed, it is tested against the requirements to make sure that the product is actually solving the needs addressed and gathered during the requirements phase. It brings all the pieces together into a special testing environment, then checks for errors, bugs and interoperability.During this phase all types of functional testing like unit testing, integration testing, system testing, acceptance testing are done with non-functional testing such as defect testing, path testing etc.

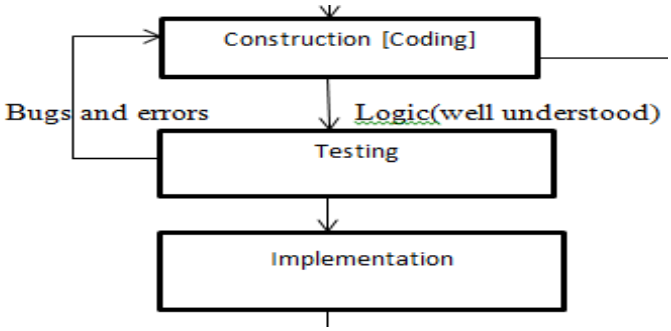

Fig. 7 Testing phase

During the testing phase four errors detected in school management software which are further rectified and removed. When it is ensured that the 'School Management System" is running error-free, the users are called with their own actual data so that the system could be shown running in their environment, conditions as per their specified requirements. So, that prerelease defects are identified and removed before the final product.

1.8 Implementation:-When the project satisfies the entire user's requirement and is free from bugs and errors, the implementation phase begins. Implementation is that stage of a project during which theory is turned into real practice. The major steps involved in this phase are:

- Acquisition and Installation of Hardware and Software
- Conversion
- User Training

- Documentation

1.9 Review and feedback:- Once when the customers starts using the developed system then the actual problems comes up and needs to be solved from time to time. The feedback is taken from the user for any modification if however required. The review of the system is done for:

- knowing the full capabilities of the system
- studying the performance
- knowing the required changes or the additional requirements.

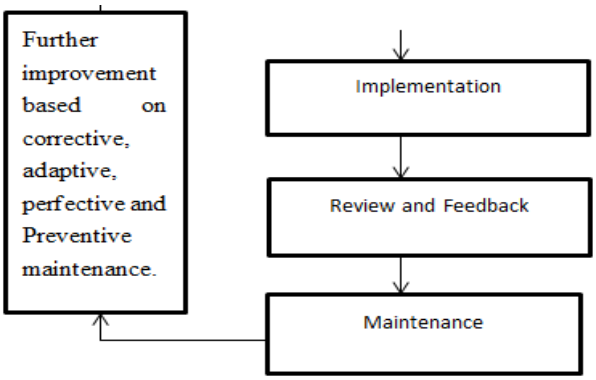

Fig. 8 Review, feedback and maintenance phase

1.10 Maintenance:- Maintenance is necessary to eliminate errors in the system during its working life and to tune the system to any variations in its working environments. It must meet the scope of any future enhancement, future functionality and any other added functional features to cope up with the latest future needs. It has been seen that there are always some errors found in the systems that must be noted and corrected. Further improvement of school management is based on corrective, adaptive, perfective and preventive maintenance.

All activities takes place recursively, each and every phase of model is reliable to Every requirement of software module, until software project not complete, not only today, but also during software maintenance and post maintenance of software to enhance quality and productivity of software. It validates the final product, weather it is as per the user requirement or not. Now direct and indirect attributes are applied and measured with the help of metrics. A number of metrics are applied to measure the quality and productivity of school management software. The final product so obtain has higher quality and productivity as compared to other product.

III. JUSTIFICATION OF PROPOSED MODEL

There are lots of quality software development models available in market/ software engineering. Quality characteristic are found in number of models. Efficiency, reliability, Maintainability, Usability and Functionality are considered by most of recent models. The table I shows the quality factors/ attributes/ characteristics supported by various models.[15,18,20,21,22]

Factors/ Attributes/ Characteristics	McCall	Boehm	Dromey's	FURPS	ISO 9126	Quality Enhancement Model
Maintainability	□		□		□	□
Flexibility	□	□				□
Testability	□	□				□
Correctness	□		□			□
Efficiency	□	□	□		□	□
Reliability	□	□	□	□	□	□
Integrity	□					□
Usability	□	□	□	□	□	□
Portability	□	□	□		□	□
Reusability	□		□			□
Interoperability	□					□
Human Engineering		□				□
Understandability		□				□
Modifiability		□				□
Functionality			□	□	□	□
Performance				□		□
Supportability				□		□

Table I Represents comparison of various models on the basis of factors/ attributes/ characteristics

The above table shows that the quality attributes Reliability, efficiency, usability, portability, functionality and maintainability are supported by most of quality models. Other attributes such as reusability, integrity, testability, understability, correctness and well documented are not supported by every model. These attributes also have their direct or indirect impact on quality and productivity.The table given below[23] shows the comparison of the different quality models with the proposed model on the basis of their structure, number of levels, relationship, their advantages and disadvantages.

Characteristics Model	McCall	Boehm	FURPS	Dromey's	ISO 9126	Quality Prediction Model
Structure	Hierarchical	Hierarchical	Hierarchical	Hierarchical	Hierarchical	Network& Relational
No. of Levels	2	3	2	2	3	n- level
Relationship	Many-Many	Many-Many	One-Many	One-Many	One-Many	Many-Many
Major advantage	Evaluation Criteria	Hardware factors included	Separation of FR & NFR	Different Systems	Evaluation Criteria	Evaluation criteria based on reliability
Major disadvantage	Components Overlapping	Lack of Criteria	Portability is not considered	Comprehe-nsiveness	Generality	----------

Table II shows comparison of quality models and their architecture

		McCall				Boehm				Dromey's				ISO 9126				Quality Enhancement Model			
Quality attributes		Poor	Average	Good	Best	Poor	Average	Good	Best	Poor	Average	Good	Best	Poor	Average	Good	Best	Poor	Average	Good	Best
Requirement Analysis	Are the basic requirements of software fulfilled?		Y					Y			Y				Y						Y
Reusability	Either part of application / entire application		Y				Y				Y				Y					Y	
Testability	Is the software tested and validated?			Y				Y				Y				Y					Y
Correctness	Is software removes all faults/ errors?		Y				Y				Y					Y					Y
Completeness	Is the software complete as per user needs in all respects?		Y					Y			Y					Y					Y
Well documented	Is software properly documented?														Y						Y
Understability	Is software easy to understand?				Y							Y				Y					Y
Functionality	Are the required functions available in the software?						Y				Y					Y					Y
Maintenance			Y				Y					Y				Y					Y
Flexibility	Is the software flexible?			Y			Y				Y					Y				Y	
Usability	Is software easy to use?		Y				Y				Y				Y					Y	
Efficiency	How efficient is software?		Y				Y					Y				Y					Y
Reliability	How much reliable is the software?		Y				Y					Y				Y					Y

Table III comparison between quality models w.r.t. characteristics.[18,20,21,22]

Most of the software development models provide the basic method to develop a software product. They do not consider the quality attributes and their impacts while developing the software product. Table given

below shows the comparison of quality prediction model with the most common software development models methodology.

IV. CONCLUSION

A School Management Software is developed by using the proposed model. The proposed model ensure that hard work done in the beginning(Requirement and analysis phase) of the product not only reduces the risk, also results in faster and accurate development in the further processes. The use of cause and effect diagram help in risk analysis and fault prediction help to reduce the maximum fault in early phase of development. Different metrics of quality, productivity and their attributes are applied and results are observed. It ensure that to increase the quality of the product the analyst has to understand the requirement thoroughly and reduces the maximum faults at every phase of development and measure it quality by using various metrics. Metrics not only help to measure the quality of product but also help to increase the quality of product, which automatically increase the productivity of the product. In further work we measure the quality and productivity of products developed by using quality and productivity enhance model.

REFRENCES

[1]. J. A McCall, P. K, Richards and G. F. Wallers, "Factors in software quality ", Griffiths Air . Force Base, N. Y: Rome Air Development Center Air Force Systems Command, 1977.

[2]. B.W. Bohem et al. J. R. Brown, H. Kaspar, M. Lipow, G. McLeod, and M. Merritt, "Characteristics of Software Quality", North Holland, (1978)

[3]. R.G. Dromey, "Software product quality: Theory, model and practices. Software quality Institute", Griffith University, Brisbane, Technical Report, 1999.

[4]. ISO 9001: 2000 Quality management systems Requirements, 2001.

[5]. ISO / IEC, IS 9126-1, "Software Engineering – Product Quality – Part 1: Quality Model", Geneva Switzerland: International Organization for Standization, 2001.

[6]. R. Fitzpatrick, "Software Quality: Definitions and Strategic Issues," in School of Computing Report. County of Staffordshire, UK: Staffordshire University, 1996.

[7]. B. Kitchenham and S. L. Pfleeger, "Software Quality: The Elusive Target," IEEE SOFTWARE 0740-7459/96/$05.00, vol. Vol. 13, No. 1, 1996.

[8]. Eldon Y. Li,Total Quality Management in Software Development Process The Journal of Quality Assurance Institute, Vol. 14, No. 1, January 2000, pp. 4-6 & 35-41.

[9]. Dr.RaoNemani et al, Int. J. Comp. Tech. Appl., Vol 2 (4), 716-725 IJCTA, JULY-AUGUST 2011

[10]. Amit C. Joshi Online International Interdisciplinary Research Journal, {Bi-Monthly}, Volume-II, Issue-II, Mar-Apr2012www.oiirj.org, ISSN2249-9598 Page 143-149

[11]. David Chappell Principal of Chappell & Associates (www.davidchappell.com) in San Francisco, California. Sponsored by Microsoft Corporation

[12]. SadiaRehmanSwot Analysis Of Software Quality Metrics For Global Software Development: A Systematic Literature Review Protocol IOSR Journal of Computer Engineering (IOSRJCE) ISSN: 2278-0661 Volume 2, Issue 1 (July-Aug. 2012), PP 01-07

[13]. KarineMordal ''Software quality metrics aggregation in industry'' JOURNAL OF SOFTWARE: EVOLUTION AND PROCESS J. Softw. Evol. and Proc. (2012)

[14]. Mrinal Singh Rawatet al. "Survey on Impact of Software Metrics on Software Quality", (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 3, No. 1, 2012

[15]. Dr.DeepshikhaJamwal "Analysis of Software Quality Models for Organizations" International Journal of Latest Trends in Computing (E-ISSN: 2045-5364) 19 Volume 1, Issue 2, December 2010

[16]. Inderpal Singh "Different Software Quality Model" International Journal on Recent and Innovation Trends in Computing and CommunicationVolume: 1 Issue: 5 p 438 –442 May 2013

[17]. SabaAwan, Faizah Malik, "An Efficient and Objective Generalized Comparison technique for Software Quality Models"I.J.Modern Education and Computer Science, 2015, 12, 57-64 Published Online December 2015 in MECS

[18]. S. Malathi and S. Dr S, "Analysis of Size Metrics and Effort Performance Criterion in Software Cost Estimation," Indian Journal of Computer Science and Engineering (IJCSE), vol. 3, 2012.

[19]. C.S.Lamba and Sanjay Kumar "Design of model for incremental development of software modules to evaluate the quality of software modules" International Journal of Modern Engineering Research (IJMER) Vol.2, Issue.4, July-Aug 2012 pp-1540-1542

[20]. Sheikh Fahad Ahmad, Mohd. Rizwan Beg, Mohd. HaleemA Comparative Study of Software QualityModels International Journal of Science, Engineering and Technology Research (IJSETR) ISSN: 2278 – 7798Volume 2, Issue 1, January 2013

[21]. Brijendra Singh, Suresh Prasad KannojiaA "Model for Software Product Quality Prediction" Journal of Software Engineering and Applications, 2012, 5, 395-401http://dx.doi.org/10.4236/jsea.2012.56046 Published Online June 2012 (http://www.SciRP.org/journal/jsea)

[22]. Mrs.Manisha L. Waghmode, Dr.Pallavi P. Jamsandekar " Software Quality Models: A Comparative Study'' Asm's International E-Journal Of Ongoing Research In Management And It E-Issn-2320-0065 Incon13-It-070 Incon Viii-2013

[23]. SyahrulFahmyet al. "Evaluating the Quality of Software in e-Book Using the ISO 9126 Model " International Journal of Control and Automation Vol. 5, No. 2, June, 2012