SQAREM-A Customized Model for SQA, Reuse

Ashwin Tomar, V.M.Thakare

Abstract—In the era of Information technology, Quality is major competing weapon in all sectors to gain the market share, productivity, performance and profitability. The software development with reuse promises many benefits like reducing efforts, cost, time, and schedule but involves many constraints. Quality assurance means assuring the quality. In this a model SQAREM is proposed which involves Input, Output factors and metrices, tools. The Input factors are monitored, controlled to imbibe quality into the system, application. The model is based on ISO 9126 model and can be customised as per the user needs. It can be evaluated on basis of model's criteria using metrices of quality and reusability.

Index Terms—Analytic hierarchy process (AHP), SQAREM - Software Quality Assurance Reuse Evaluation Model, SQCP - Software Quality Control Process, and SQAP – Software quality assurance process.

1. Introduction

SOFTWARE plays a pivotal role in sectors like banking, railway, medical services, telecommunication, transportation & defense services. In fact, it is woven into the threads of our daily life and almost impossible to think without software as an integral element. It is becoming omnipresent in mobile, website, and internet and embedded technologies. Seeing the ever increasing dependence over software there is need for further research in the field of software. Thus it may be said that the ravaging effect of the software is not only on individual and organization level but also at national and international levels. The pressing need arises for speedily software development with quality. This is achieved through reuse. This leads to quality software by improving requirement, removing hundred percent deficiencies from various phases of software development. In general, software is a troubled technology plagued by defects, project failures, cost overruns, schedule overruns, and poor quality level.

. Software quality evaluation is defined as "the systematic examination of software capability to fulfill quality requirement". The software quality model is defined as "a set of characteristics and sub-characteristics, as well as the relationships between them that provide the basis for specifying quality requirements and evaluating quality"[4]. The software development is dynamic process. The process adopted for developing software needs to take into account various factors. Identifying factors of software quality assurance are human centric process, involving time and will optimize the software development activities to bring profit to industry. Based on these factors a model is proposed. The factors are classified and grouped into Input, Output and metrics, tool. The quality, reusability characteristic can be evaluated using metrics, tools which are considered as the Output of models. If the system is developed from the beginning then the Input parameters can be controlled, monitored to build quality into the system.

This paper is organized as follows: In Section 2, we attempt to summarize the survey of literature on quality models and factor associated with SQA, reuse. In Section 3, deals with identification and classification of factor on software quality assurance. In Section 4, the parameters and metrices are described for evaluation of process, quality characteristics and quality models. Section 5, proposes a model which comprises Input, Output Factor and metrics. Finally we present conclusions and future work.

2. Literature Survey

The quality of software is affected by elements like defect level, origin, severity, complexity, reliability, schedules, budgets, portability and many other factors. The research papers on different type of model were searched and found associated with different words like Prediction, Classification, Analysis, Measurement, Estimation, Improvement and Quantification. Pendharkar et.al. [1], proposed a model for estimation of software development efforts. He suggested use of Quality models which bridge the gap between metrics and characteristics of software. Balaswaminathan et.al. [2], considered process to be built with a set of various activities which are controlled to have better control over the entire process. Metrics are used to control the process.Ting-Peng et.al [3] tried to show that Software quality is influenced by information diversity, task conflict, learning in project team. Taghi khoshfgoftar et.al [4, 5, 6, and 7] used quality model to identify program modules that are defective and improve the fault detection process. Finding faulty components in a software system during early stages of software development process can lead to a more reliable final product and can reduce development, maintenance costs & resources effectively. His research paper shows use of technique like genetic algorithm, rule based decision, evolutionary

- *Ashwin Tomar is presently working in Siddhant Institute of Computer Application for MCA course under University of Pune, Maharashtra, India, Email - mcatomarashwin@gmail.com*
- *V.M.Thakare is currently working as Professor and Head in Computer Science, Faculty of Engineering & Technology, Post Graduate Department of Computer Science, SGB Amravati University, Amravati Country- India, and E-mail: Vilthakre@yahoo.co.in.*

sampling, fluzzy & semisupervised clustering, case based reasoning, regression tree algorithm[8], Principal component analysis[9], neural network[10], tree based approach[11] related to software quality classification, its enhancement & quality modelling. O.Alshathry et.al [12], suggested use of defect containment matrix for optimization of software quality assurance. Defect containment matrix should be used for defect detection and removal. D.C.Kirk et.al [13], proposed a framework for modelling software processes that supports representation and comparison of different kinds of software process. J.Eckroth et.al [14], suggested a method for determining how functional requirements affect software quality. A functional modeling framework is proposed that includes a controlled language for requirements specification and assess software qualities. Information entropy metric is suggested to measure the significance of each software requirement. Using this method the designer can identify which requirements, when implemented, will affect software quality. Metrics are used by the software industry for analyzing, designing, developing, implementing and maintaining of software. Metrics were surveyed based on quality assurance and reuse which can be seen in paper [15]. Metrics provide basis for analyzing, measuring, comparing, evaluating all characteristics of product, process (quality assurance and reuse). The different types of metrices were classified into traditional metrics, object oriented metrics, quality metrics, defect metrices, reuse metrics, repository metrices. Ejaz, Zafar et.al [16], proposed a quality assurance model for systematic verification and assessment of the analysis phase. If defect is not detected during requirement i.e analysis phase it increases the cost of the product. As per them fifty percent development efforts get reduced if defects are removed at analysis phase itself. Hence the author proposed a model. The model provides a roadmap to the quality assurance people to conduct their activities in a systematic manner during the requirement analysis phase. The main elements of evaluation in this phase are the people (capable team to generate checklist), the process (standard and should be in manner) and the product documents (SRS, IRS i.e. software requirement specifications and interface requirement specification). A. Corporation et.al [17], suggested that predicting software defect introduction and removal rates, COQUALMO variants are useful for identifying appropriate defect reduction strategies. COQUALMO model can be used in different ways to reason about and optimize quality processes. Caper Jones et.al [18], has worked on defects. He suggested three bad quality metrices a) cost per defect; b) lines of code; and c) technical debt. The two good metrics are: a) function points, for normalization of data; and b) Defect removal efficiency, for measuring the percentage of bugs found prior to release and afterwards via maintenance. He suggested defect removal efficiency i.e DRE as one of the commonly used metrices for measuring the efficiency of defect removal at various stages of software development life cycle. An organization can measure the effectiveness of their quality assurance process based on the number of defects found in the product before and after its release. Klas, Elberzhanger et.al [19], suggested framework - Quality Improvement Paradigm (QIP) for systematically improving an organizations development processes in a continuous manner to support small and medium sized enterprises in developing, optimizing, and adapting a Quality Assurance strategy best suitable in their context. The framework and process model for the balanced optimization of quality assurance integrates approaches of defect flow model. Zander Nowicka et.al[20], presented an Quality Assurance framework is proposed for modeling the system services, transforming them into failure detectors, deriving failure mitigators, and test cases based on those services. Ominguez Mayo et.al[21] proposed quality evaluation framework (QuEF-Ts) to manage quality in model driven web engineering (MDWE) and extended to cover all quality management process. The framework has a model which manages quality. Becker, Lew et.al [22], suggested specific strategy called SIQinU (Strategy for understanding and Improving Quality in Use) which allows recognizing problems of quality in use through evaluation and proposes product improvements by understanding and making changes on product attributes. As per L. Zhao et.al[23] open source software development is a methodology to improve software quality assurance, it is faster and safer rather than traditional methodology, especially in case of large scale systems. Behkamal, Akbhari et.al [24] proposed a model after customizing ISO 9126 model and applied for web and B2B application. Phongaibul & Boehm et.al [25] showed in "Improving Quality through Software Process Models in Thailand" that Thailand people have different culture values and hence found it difficult to implement the Software Process Models. Models given by SEI and USA (e.g. CMM, CMMI) are more tailored to western cultures.

3. IDENTIFICATION, CLASSIFICATION & GROUPING FACTORS

Identifying factors are human centric process, involving time and will optimize the software development activities to bring profit to industry. The software development is dynamic. The process adopted for developing software needs to take into account factors. Mapping is required between various factors to optimize the process of software quality. The journey of software development involves major activities like i.e Quality Assurance Process SQAP (Input), Quality Control Process SQCP (Measure) & Software as end product SAEP (Output).

SQAP - Software Quality Assurance Process - is the process involves building quality into the product through various phases which are requirements gathering, converting requirement into the design phases, implementing the product developed, maintaining the product. There are developmental approaches like structured, object oriented, component, web, embedded based development [8]. For any process the goal has to be defined. SQAP determines whether everything is going according to polices standards and procedures and practices.

SQA is about providing assurance about the working of the product. It is an expensive time consuming process. SQA is the responsibility of separate independent group. It is involved in improvement of process and product. It works like the road police, here it is process police. Thus it is a preventive approach and prevents faults from occurring by enforcing rules, methods & policy. It prevents defects from occurring. There are many factors influencing the quality assurance process. The papers were downloaded from Internet and read number of times to find factors of SQA listed in research paper [29]. They are Planning, Standards(for Code, Design), Rule, Legal Procedure, Documentation (of Process & Product), Guidelines, Responsibility, Technology, Right Conduct, Authority, Approvals, Environment, Culture, Risk, Size, Report, Reuse (code and design), Virtue & Ethics, Cost involved, Efforts, Practice, Schedule, Schedule Pressure, Approaches, Techniques, Feedback/Customer satisfaction, Output, Tool, Time limit(deadlines), Revision, Checklist, Manuals, Inspection (code), Budget Pressure[28], Resources, Process Metrics. There are some more factors affecting SQA like Information Diversity, Task conflict, Learning Management, Methodologies, Cycle Time, Complexity(in process, code), process maturity, Security. Some factors are classified under influencing factor which are type of Development(model used, language used, reuse level), Goals (for Quality attributes e.g. reliability, Quality levels i.e. high, medium, low), use of new Technology e.g. Ajax, Resources (efforts i.e.7/8 hours, experience, time, team size), Artifact (based on Size, Complexity, Defect density, Extent of quality), Tools e.g. QTP. Some factors often vary and hence classified under as variable factors like Focus (Quality attribute, defect class), Resources (Effort, Team Size, Experience),Scope, Entry/exit criteria (reading techniques), Inspection, Testing (tool, test execution, test methods).These factors in some or other way influences the quality assurance process thereby imbibing, accumulating adding quality drop by drop into software. Many times it is seen that these factors if not taken care starts eroding, eating the quality from software. So all the factors have to be taken care of.

SQCP - Software Quality Control Process - It is corrective approach. It finds faults, corrects faults when occur. Testing and Quality are like two sides of same coin. The following factors or activities are associated with SQCP are Examining, Monitoring, Inspection, Measuring/Measurement (process, product) [8] [29], Evaluation, Testing (test plan), Continuous Improvement (ongoing effort to improve the quality of products, services or processes), Audit (fail/pass) Improvement (process, product), Product Metrics, Product Evaluation, Feedback (output), Evaluation Report [29].

SAEP - Software As End Product - SAEP i.e. output - Product which is obtained is measured, checked and used further. Various techniques like regression tree, case based reasoning, neural network, genetic algorithm, Bayesian Network, Principal component analysis, Fuzzy logic, Function points, metrics based are employed by authors for estimating the quality of the product[29].

4. PARAMETERS & METRICES USED FOR EVALUATION

Software is the set of instructions to acquire the inputs and to process them to produce the desired output in terms of function and performance as determined by the users of the software. While developing an application different methodology, approaches are involved like ISO, CMM which impacts quality factors. The various quality models like Mc Calls, Dromey's, ISO 9126, and FURPS are proposed which are based on Characteristics/ Factors, Sub-Characteristics, Attributes. Quality models bridge the gap between metrics and characteristics of software. Quality models have many uses and are proposed to improve, built, predict & assure quality of software product. They are proposed to optimize the cost of quality. They serve as tools for focusing software development efforts. Metrices are used for monitoring, controlling, measuring the process and its activities like requirement gathering, removing defects, testing, quality factors[15]. The reuse of quality components are used to speed up the development of application, system. The quality factors can be measured using metrics. The similar type of relationship can be shown in Fig 1.1. The figure shows use of different approach methodology (ISO, CMM) impacts quality factor like reliability, functionality, usability, portability, maintainability.

IEEE Standard 12207 defines Quality Assurance as “a process for providing adequate assurance that the software products and processes in the product life cycle conform to their specific requirements and adhere to their established plans” [30]. Software reuse is the process of creating systems from existing software rather than building them from scratch. Software reuse means artifacts are used in more than one project or system. Reuse of requirements, code, modules, components, design, architecure, test plans, test cases can be done in form of artifacts. The benefits from reuse are reduction in cost, development time and increase in productivity. Reuse and reusability are two major aspects in object oriented software which can be measured from inheritance hierarchy. Reusability is a property of a software asset that indicates its probability of reuse. Metrics can be used to evaluate reuse and reusability of object oriented software like Depth of Inheritance

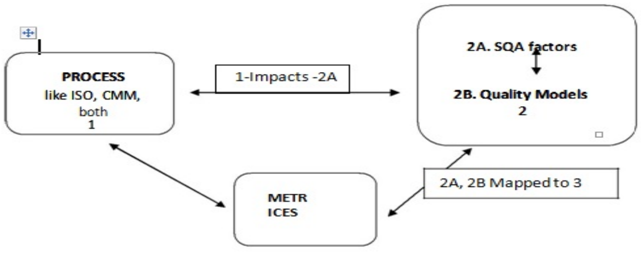


Fig: 1.1 Use of Metrices for Evaluation

Tree (DIT), Number of Children (NOC), Method Inheritance Factor (MIF) and Attribute Inheritance Factor (AIF) [32]. The paper estimates quality in terms of characteristics, sub-characteristics.

5. PROPOSED MODEL & ITS CRITERIA FOR EVALUATION

The Proposed model is shown in form of chart, Fig 1.2. The model shows criteria for Evaluation of model. The model represents different level 1, 2, 3, 4. **The Level 1** represent the classification Criteria. The **Level 2** represents the ***Input Factors/ Characteristics*** which are Requirement, Process-Progress-Control-Improvement, Testing & Defect removal, Reuse, Organization & Business, Tool Technique Technology, Human resources. ***Output Factors*** evaluates the Quality i.e functionality, usability, reliability, portability, maintainability, efficiency. The Evaluation is also controlled, checked by using metric, tools and auditing. **Level 3** represents Sub-characteristics under Characteristics. In this Characteristic i.e. FACTOR i.e of requirement the Sub-characteristics are complete, unambiguous, certified correct, modifiable. **The Level 4** represents the Attributes. The fig 1.2 represents Model SQAREM.

SQAREM is multidimensional model which involves use of metrics for evaluation. It is based on various Factor/Characteristics like quality, process, components, testing, defect, technique, tools, metrices, and human resources. Each FACTOR has many attributes/sub characteristics which influences it. It involves quality assurance of various parameters of system, application, and project. The criteria for evaluation (output result) are based on parameters ISO 9126 model. The factors evaluated for quality are functionality, usability, reliability, portability, maintainability and efficiency. The Factors are classified into Input (internal), Output (external) and Metrices measuring them. If the project is already developed and its Quality parameters is to be estimated in that case Characteristics or Factors are only considered. There are the Output Factors like i.e functionality, usability, reliability, portability, efficiency. The Input factor is listed under 5.1, 5.2, 5.4, 5.7, and 5.8. These Factors/Characteristics and Sub-characteristics are controlled using metric or tools. The model is explained in more details here [33]. The criteria associated with the model are explained herewith. The complete model with its criteria is shown.

5.1 Requirements

Requirement is associated with following parameters.

i. Consistent
ii. Clear
iii. Usable
iv. Unambiguous
v. Complete
vi. Verifiable
vii. Correct
viii. Modifiable

5.2 Process, Progress, Control & Improvement

The identified factors are logically grouped under process.

i. Planning
ii. Standard
iii. Rules
iv. Documentation
v. Guidelines
vi. Technology
vii. Team involved
viii. Approvals
ix. Schedule
x. Techniques
xi. Approaches
xii. Input and output
xiii. Resources used
xiv. Improvement
xv. Maturity

5.3 Reuse

Reuse is associated with following requirement specification, architectural design, code, components, template, tables, patterns, test cases, documents etc.

5.4 Testing & Defects Removal

Process Testing (Input)	*Product Testing (Output)*
i) Checklist	vii) Usability testing
ii) Review	viii) Functionality testing
iii) Inspection	ix) Interface testing
iv) Audit	x) Performance testing
v) Walkthrough	xi) Security testing
vi) Planning	xii) Compatibility testing

5.5 Metrics, Measurements & Audit *(Metrics, Audit)*

Process and product metrics are used for measuring.

5.6 Project & Product

Project involves factors like size, complexity, risk, Reuse & Performance. **Product** involves quality factor.

5.7 Human Resources

5.8 Right Conduct
5.9 Authority
5.10 Responsibility
5.11 Attachment among members
5.12 Skills
5.13 Experience
5.14 Team size
5.15 Interpersonal conflict
5.16 Commitment towards work

5.8 Organization(O), Management(M) & Business (B)

i) O_Environment
ii) O_Culture
iii) O_Virtue and Ethics
iv) O_Maturity
v) M_Commitment
vi) O_Structure & Size
vii) M_Expertisaton
viii) B_dependencies
ix) B_drivers
x) B_cust_satisfaction
xi) B_payments
xii) Risk Management

5.9 Evaluation parameter for Quality (based on ISO 9126)

i) Functionality
ii) Usability
i) Reliability
iv) Portability
ii) Maintainability
vi) Efficiency

The model in fig 1.2 shows input parameters which are controlled to imbibe quality into the system. When the system is being developed various parameters of Requirement gathering are controlled so that maximum error is detected and quality increases. The requirement gathering should be unambiguous, clear, correct, verifiable, consistent, usable, modifiable and reusable. The software complexity is the major problem in software quality assurance as suggested by Alsultanny Yas An et.al [26].

The process progress should be planned, controlled, monitored and improved. The standards should be fixed to control quality while developing the system. Best, Cheapest Technology bringing benefits to masses should be accepted. The schedules should be such that it minimizes resources and should be used.

Testing is the very important activities to remove defect and improve quality of the product. Checklist, Review, Inspection, Walkthrough, Inspection are various activities to remove defects.

Reuse of requirements, design, architecure, code, test case, components, and documents are done which are stored in repositories for building a system. Reuse speeds the process, builds quality into system and lower downs the cost.

Organization vision, mission and policy speak about the quality. Management is responsible for building the type of organization culture. Ethics plays an important part in molding behavior of human beings. The maturity of process, organization is decided with time. The business of an organization is dependent of many drivers, factors. The customer satisfaction depends on the services given by the human being in an organization. The risk and its type is very crucial part in an organization. It should be reduced to optimize quality.

The evaluation of system can be performed using metric and tools i.e implementation of model. The result of system can be evaluated using FURPM E parameters which are listed fewer than 5.9.

The reuse or reusability parameters evaluation is described in research paper [34].

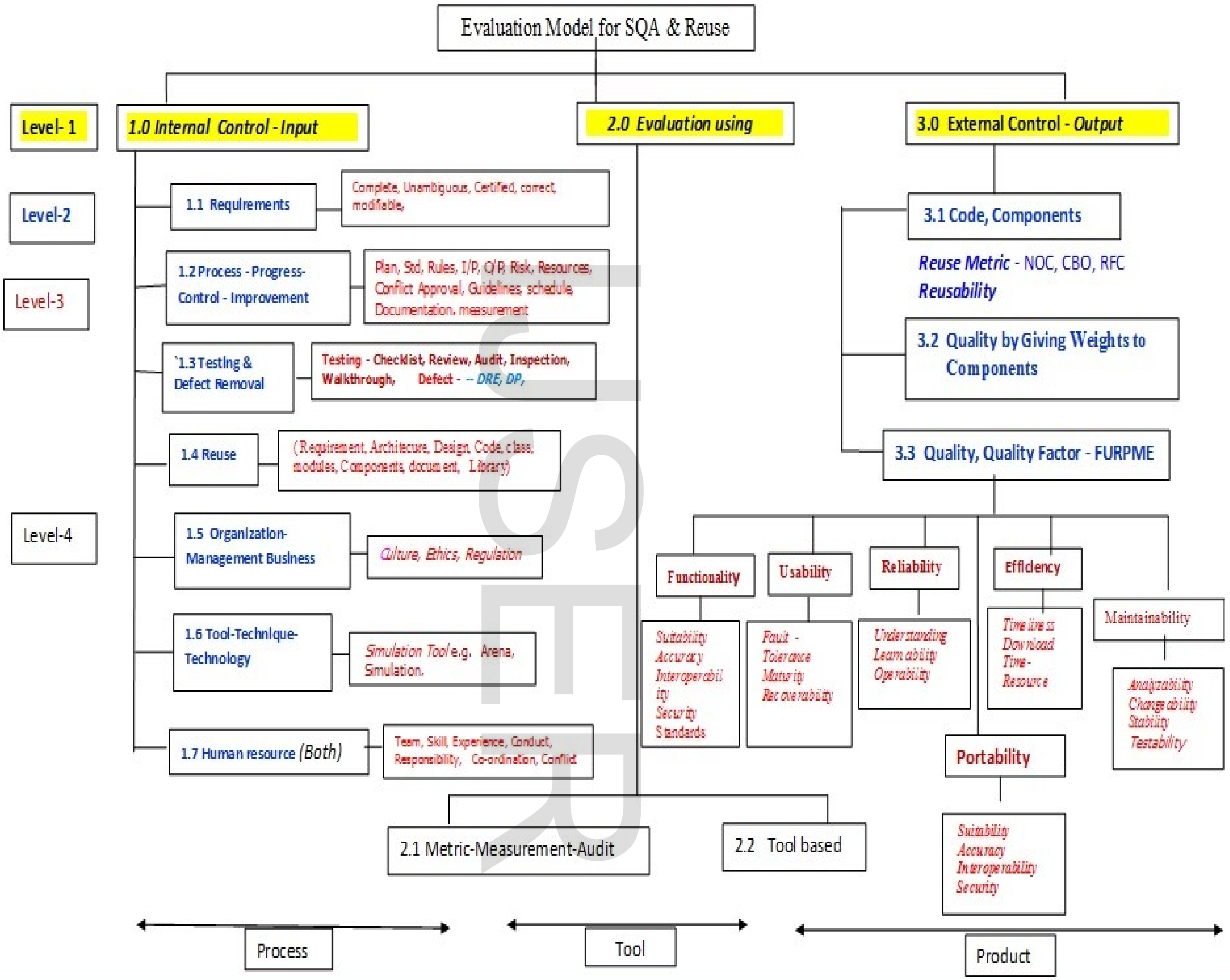
The model proposed can be CUSTOMISED and modified to in terms of criteria to satisfy a customer as per their needs. It provides the user the flexibility to change its factors, criteria as per needs and evaluate it.

6. RESULT

SQAREM is a multidimensional conceptual model which can use metrices and tools for evaluation. It involves various input factors which are controlled to optimize quality and reusability. The output of a system is based on ISO 9126 model. The factors which can be evaluated are functionality, usability, portability, maintainability, reliability, efficiency.

7. CONCLUSION

This model can be implemented in case of website development, component development etc. After implementing the model cost will be saved, the productivity will be better, the performance of project, system will be much better, maintenance cost will be low and quality will increase. The evaluation of model with its comparison is our future work.


Evaluation Model for SQA & Reuse
Level- 1
Level-2
Level-3
Level-4
1.0 Internal Control - Input
2.0 Evaluation using
3.0 External Control - Output
1.1 Requirements
Complete, Unambiguous, Certified, correct, modifiable,
1.2 Process - Progress- Control - Improvement
Plan, Std, Rules, I/P, O/P, Risk, Resources, Conflict Approval, Guidelines, schedule, Documentation, measurement
`1.3 Testing & Defect Removal
Testing - Checklist, Review, Audit, Inspection, Walkthrough, Defect - -- DRE, DP,
1.4 Reuse
(Requirement, Architecture, Design, Code, class, modules, Components, document, Library)
1.5 Organization- Management Business
Culture, Ethics, Regulation
1.6 Tool-Technique- Technology
Simulation Tool e.g. Arena, Simulation.
1.7 Human resource (Both)
Team, Skill, Experience, Conduct, Responsibility, Co-ordination, Conflict
2.1 Metric-Measurement-Audit
2.2 Tool based
3.1 Code, Components
Reuse Metric - NOC, CBO, RFC
Reusability
3.2 Quality by Giving Weights to Components
3.3 Quality, Quality Factor - FURPME
Functionality
Suitability Accuracy Interoperability Security Standards
Usability
Fault - Tolerance Maturity Recoverability
Reliability
Understanding Learn ability Operability
Efficiency
Time iness Download Time - Resource
Maintainability
Analyzability Changeability Stability Testability
Portability
Suitability Accuracy Interoperability Security
Process
Tool
Product

REFERENCES

[1] P.C. Pendharkar, James A. Rodger, “A Probabilistic Model for Predicting Software Development Efforts”, IEEE *Transactions on Software Engineering,* Vol. 31, No .7, July, 2005.

[2] S. Balaswaminathan, “A Composite model used for Software Quality Assurance”, *www.stickyminds.com.*

[3] Ting-Peng Liang, James Jiang, Gary S. Klein, & Julie Yu-Chih Liu, “Software Quality as Influenced by Informational Diversity, Task, Conflict, and Learning in Project Teams", IEEE *Transactions On Engineering Management*, Vol. 57, No. 3, August 2010.

[4] Kehan Gao, Taghi M. khoshfgoftar, “An Empirical Investigation of Filter Attributes Selection Techniques for Software Quality Classification”, IEEE conference on IRI, July 10-12, 2009.

[5] Taghi M. Khoshgoftaar, Yi Liu, Member, IEEE, “A Multi-Objective Software Quality Classification Model Using Genetic Programming”, *IEEE Transact ions On Reliability,* Vol. 56, No. 2, June 2007.

[6] T.Khoshgoftaar, Jenq-Foung Yao, Yi Liu, “Building a Novel GP Based Software Quality Classifier Using Multiple Validation Datasets“, *IEEE ICIRI*, pp. 644-650, 2007.

[7] Taghi M. Khoshgoftaar, Erik Grleyn, Laurent Nguyen, Lofton Bulalrd, “Cost Sensitive Boosting Quality Modeling”, *Proceedings of the 7th IEEE International Symposium on High Assurance System Engineering,* 2002

[8] Katsuyuki Yasuda Shigeru Yamada, “The Concept & Practice of Software Quality Assurance in Japan in the Global Era, Proceedings of' the 7th IEEE *International* Symposium *on High Assurance Systems Engineering (HASE)*, pp 124-133, 2002.

[9] Beatriz Marin, Giovanni Giachetti, Oscar Pastor, Tanja E. J. Vos, “A Tool for Automatic Defect De tection in Models used in Model-Driven Engineering", IEEE *Seventh International Conference on the Quality of Information and Communications Technology,* Sept 29-Oct. 2, 2010.

[10] Taghi M. Khoshgoftaar, Edward B. Allen, Member, John P. Hudepohl, Stephen J. Aud, “Application of Neural Networks to Software Quality Modelling of a Very Large Telecommunications System”, *IEEE Transactions On Neural Networks,* Vol. 8, No. 4, July 1997.

[11] Taghi M. Khoshgoftaar, Naeem Saliya, “Tree based software quality models for fault Prediction “, Proceedings of the *Eighth IEEE Symposium on Software Metrics*, 2002.

[12] Omar Alshathry, Helge Janicke, Hussein Zedan, Abdullah Alhussein, "Quantitative Quality Assurance Approach”, *International Conference on New Trends in Information and Service Science (NISS)*, 2009.

[13] D. C. Kirk, S. G. MacDonnell, “Modeling Software Processes a Focus on Objectives,” *ICSSP,* ACM, pp 941- 948, May 21-22, 2011.

[14] J.Eckroth, Guy-Alain Amouddou, “Improving Software Quality from the Requirements Specification,” Science *of Design Symposium*, pp. 38–39, 2007, ACM 978-1-60558-436-2/07/03.

[15] A. Tomar, V.M.Thakare “A Survey of Metrics on Software Quality Assurance and Reuse,” *NCIPET-2013*.

[16] R. Ejaz, M. Nazmeen, M. Zafar, “A quality assurance model for analysis phase,” Proceedings of the National *Software Engineering Conference on NSEC* ‟ pp.1-4, 2010.

[17] A. Corporation, “Modeling Software Defect Dynamics,” *Software Quality, Reliability, and Error Prediction*, STN, April 2010.

[16] C. Jones, “Measuring Defect Potentials and Defect Removal Efficiency", pp, March 1, 2008, www.rbcsus.com.

[17] M. Kläs, F. Elberzhager, R. van Lengen, T. Schulz, and J. Goebbels, “A Framework for the Balanced Optimization of Quality Assurance Strategies Focusing on Small and Medium Sized Enterprises,” 35th Euromicro Conference on *Software Engineering and Advanced Applications,* pp. 335–342, 2009.

[18] Zander Nowicka, "Quality assurance of automotive hybrid embedded software,”
Proceedings of the Warm Up *Workshop for ACM/IEEE ICSE* 2010, WUP, 2009.

[19] Dominguez Mayo. F.J, Escalona M.J, Ross M, "A Quality management based on Quality model life cycle", *Computer Standards & Interfaces,* Vol -34, pp. 396–412, 2012

[20] P.Becker, P.Lew, and L.Olsina, “Strategy to Improve Quality for Software Applications : A Process View,” ICSSP‟11, pp. 129–138, May 21–22, 2011.

[21] L. Zhao, S. Elbaum, “Quality assurance under the open source development model,” *Journal of Systems and Software*, vol. 66, no. 1, pp. 65–75, Apr. 2003.

[22] B.Behkamal, M. Kahani, M. K. Akbari, “Customizing ISO 9126 quality model for evaluation of B2B applications,” *Information and Software Technology*, vol. 51, no. 3, pp. 599–609, Mar. 2009.

[25] Phongpaibul Monvorath, Boehm Barry, “Improving the Quality Through Software Process Improvement in Thailand: Initial Analysis”, *3-WoSQ* May 17, 2005.

[26] Kemerer Chris F, Paulk Mark C, “The Impact of Design and Code Reviews on Software Quality: An Empirical Study Based on PSP Data”, *IEEE Transactions on Software Engineering*, vol 5, No-4, July-Aug 2009.

[27] Alsultanny Yas A, Wohaishi Ahmed M, “Requirements of Soft ware Quality Assurance Model”, IEEE *Second International Conference on Environment and Computer Science*, 2009.

[28] Ning Nau, Donal E. Harter, “Impact of Budget and Schedule Pressure on Software Development Cycle Time and Effort”, *IEEE Transaction on Software Engineering*, Vol -35, No-5, Sept-Oct- 2009.

[29] Tomar Ashwin, Thakare V M, *“Identification and listing of factors affecting SQA”* International Journal of Internet Computing, ISSN No: 2231-6965, Vol-1, No-3, 2012.

[30] W.A.Conklin, “Software Assurance: The Need for Definitions", IEEE, *Proceedings of 44th Hawaii International conference on system Science,* pp 1–7, 2011.

[31] A. Sharma, Rajesh Kumar, P.S. Grover, “Managing Component Based Systems with Reusable Component", *International Journal of Computer Science and Security, Vol 1,* No. 2, 2007.

[32] N. S. Gill, “Inheritance Hierarchy Based Reuse & Reusability Metrics in OOSD,”*IJCSE,* Vol. 3, no. 6, pp. 2300–2309, 2011.

[33] A.Tomar, V.M.Thakare, "The Study of Model for SQA, Reuse and Predicting a customized model", Thesis, 2015.

[34] Ashwin Tomar, V.M.Thakare, "Evaluation model for estimating reusability parameter using metrices", IJCSEITR, ISSN (P): 2249-7943, Vol 6, Issue 1, pp. 53-68, Feb 2016.