Available at: www.researchpublications.org

# A Customized Model on Software Quality Assurance & Reuse

Ashwin Tomar[1], V.M.Thakare[2]

[1]*Siddhant Institute of Computer Application, MCA Dept.*
*Maval, Pune University, India*
[1]mcatomarashwin@gmail.com

*P.G. Dept. of Computer Science*
*Amravati University, Amravati, India.*
[2]Vilthakare@yahoo.co.in

*Abstract*---**One of the major problems with software development process is of cost, quality and reuse which depend on many variable factors. There are many technical barriers and impediment to software quality and reuse. In this paper we have proposed a model which emphases on concept of common architecture design, formal reuse culture with systematic quality assurance process involving continuous monitoring, evaluation, improvement followed by certification activities. We proposed a guaranteed process of building quality into the software.**

***Keywords***---**Software Quality Assurance (SQA), Component Certification, Quality Evaluation and Certification Process.**

## I. INTRODUCTION

Software quality assurance and reuse are very important aspects in development of software. SQA monitors the software Engineering processes and methods to ensure quality. It is the process of verifying or confirming that whether products and services meet the customer expectation or not. SQA is process driven approach with specific steps to attain development goals. This process of QA considers design, development, production and service. It is preventive approach from occurring by providing rules and methods. It prevents defects from occurring. It is conducted in software development process from the early stages of SDLC. It gives confidence to customer regarding the software development process and the product. According to IEEE QA is a planned and systematic pattern of all actions necessary to provide adequate confidence that an item or product conforms to established technical requirements [1]. Software Quality Assurance involves both process and product assurance. Process oriented approaches deal with the establishment of rules, principles, standards, guidelines, manuals, process definitions, the evaluation and improvement of software quality processes [2]. The high quality development processes result in a high quality product. The process oriented models are ISO 9001, CMMI, SPICE, and ISO 12207. Some of them are studied and surveyed. Product oriented approaches such as models by Boehm, McCall [15] aim to assure the quality of the product by evaluating the characteristics. Product oriented standards such as ISO 9126 offers a multiplicity of measures to assure product quality targets.

Software reuse is the process of creating software system from existing software rather than building from scratch. Software reuse comes in two major forms: direct reuse and indirect reuse. A product may be designed to link to an existing software component, a form of direct reuse, or a software developer may cut and paste a portion of existing source code and use it in the new product with some modification, a form of indirect reuse. Typically in software development, they are different forms of software reuse practiced every day. It is very common for a software developer to search some existing code and find one or more pieces of code that is usable for a new product, so that they do not have to write every line of code from scratch. This practice usually exists even informally if there is no formally defined reuse process within an organization. There are lot of benefits from Software reuse like save time and cost. It Improve the quality of software. It retains high quality in new product from old existing product. It refers to developing and defining a standard formal reuse process for organization. When a project becomes large and complex, it is more difficult to find what they can reuse to satisfy the new requirements.

The proposed model is based on quality assurance and reuse process through building a repository for developing product lines, domain engineering product and components. The model describes how to do it. The scheme of the paper is as follows; section 2 deals with Literature review, section 3 deals with proposed model, section 4 follows conclusion and references.

## II. LITERATURE SURVEY

On exploring the domain of SQA and Software Reuse we came across number of models, authors who have worked in this area:

*Korean Software Process Quality Certification Model* is proposed by Sun Ming, Soobin [17] - The models is a

Korea standard software process certification model for small medium sized companies. The model consists of 3 levels, 5 categories 17 assessment items and 76 detailed items.

*ISO 9000 Series Standards [18]*-The Quality Management system to be developed as per ISO 9000 guidelines is used for quality assurance when supplier has t demonstrathe capability to the customer. ISO 9000:2000 QMS guides for fundamental and vocabulary, ISO 9001:2000 QMS guides for Requirements and ISO 9004:2000 QMS is for Performance Improvement. ISO 9001:2000 standard gives the requirements to develop a quality management system. It has eight clauses.

*Quality Assurance Model for Analysis Phase* [4] - Ejaz, Nazmeen & Zafar proposed a Quality Assurance model for analysis phase. If defect is not detected it increases the cost of the product. If quality produced at analysis phase then the
50% of development efforts is reduced. Hence the author proposed a model for systematic verification and assessment of the analysis phase. 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).

*Capability Maturity Model* [18]-addresses the process improvement in software development organizations. CMM identifies a set of guidelines that need to be implemented for producing quality software. The CMM framework has been accepted as very comprehensive framework for quality processes implementation. It consists of 5 maturity levels, 4 categories and 25 processes. Software capability is defined as range of expected results that can be achieved by following a software process. The actual performance achieved by process is considered as software process performance. The extent to which a process is explicitly defined, managed, measured, controlled and effective is defined as Software Process Maturity.

*Requirements of Software Quality Assurance Models i.e. Complexity Model* [11] - Author has suggested a software quality model to test the different factors that affect the quality of software increases the productivity of the software by taking in consideration the software complexity. The proposed model showed how to provide safe, reliable and quality engineering product to develop, by taking all factors affecting software quality to reach the ISO 9126. In short this model showed relationship between problem and constrain that might affect the quality and productivity of the software. Software complexity is considered that main factor affecting the quality and productivity of the software.

*Six Sigma [18]* - It is statistical measure of variation in a process. It is said that Process has achieved six sigma if the quality is 3.4 Defect per million opportunities. Main focus of six sigma is on reducing defects and variations in the processes. There are two models suggested for it which are DMADV i.e (D-define, M-measure, A - analysis, D-design,
V-verify & DMAIC (I-improve, C-control). Six Sigma is about techniques, techniques, tools, statistics and people who implements.

*SPICE* [18] - It is Software Process Improvement and Capability determination (SPICE) is for software process improvement. It provides framework based on process definition, process assessment, capability determination, and process improvement. In these standards various processes required for software development are categorized into nine processes with nine attributes.

*Quality Planning Model & Modified Framework* [13]-The framework proposed by Shahela Saif, Aliya Ashraf Khan & Dr Fahim Arif is an analysis of framework proposed by Frank Elberzhanger and Christian Denger. They have suggested improvements and overcome limitations in their proposed framework. They gave emphasis on various elements of models which are categorized under influencing factors (development context, goals, new technology, resources, artifacts, tool), variation factor (resource, focus, scope, entry/exit criteria, inspection process, testing), technique characteristics, QA gate, QA technique, QA Strategy.pestige, wealth and power i.e between boss and subordinate), uncertainty avoidance (is degree to which people in societies tolerate uncertain situations)

*Improving Quality through Software Process Models in Thailand* [14] - *Phongaibul & Boehm* showed 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. The models of culture differences from Edward T.Hall and Greet Hofstede are used as culture dimensions to explain the difference in cultures towards software process models and improvements. The result revealed that main culture dimension are power distance (*is degree of inequality in prestige, wealth and power*

*i.*

*e between boss and subordinate*), uncertainty avoidance ( *is degree to which people in societies tolerate uncertain situations*), masculinity/feminity , Monochronic/Polychronic time and high/low context languages *( refers to amount of information conveyed during communication including voice, gestures and facial expression*).The paper focused by

showing that Thailand culture is polychromic, high context languages, high power distance, highly uncertainty avoidance, more collective ( based decisions), feminine countries, long term oriented country ( long term expectation).

*A Composite Model for Software Quality Assurance, www.stickymind.com[20]*-This model integrates Quality Assurance review and measurement processes. Both these processes provide better visibility. The model identifies ten processes which are categorized as Quality Assurance Process and Measurement Process. All these process has to be monitored by using metrics. The variation (deviation) is assessed. The key parameters are identified and measured. The process is controlled by taking corrective action.

*Open Source Software Development Model* [9]:- It is faster, safer methodology/model to improve software quality assurance. In open source user has to have a license agreement and pay some money to the owner or maintainer of code. Author name has to be preserved and copyright statement under it.

*Quality Assurance: Much More than Testing [3]:* Quality assurance deals with process. Testing is finding errors, defect, and bugs. QA is preventive process while testing is detective process. More strictly the process of development is followed better is the level of Quality. If it is strictly followed the efforts required in removing the bugs while testing gets reduced.

*A Study of Software Reuse and Models [23]:* The Models proposed are on - Reuse cost-benefits, Maturity assessment, Amount of reuse, Failure modes, Reusability metrics, and Reuse library metrics.

*Reuse Component Model [12]* – by Arun Sharma proposed a model to bring out relationship among metrices and with a view to establish quantitative estimation of quality and validate the suggested model.

## III. PROPOSED MODEL

The proposed model is a generalized form of Software Quality Assurance Reuse model which involves many factors and variables. Model depends on factors like human (people), management skills, environment, formal reuse process, organization type, product monitoring division, evaluation division, improvement, certification processed. The model emphasis on common methodology, architecture design, reuse culture with systematic discipline quality assurance process involving continuous monitoring, evaluation, improvement and certification activities.

### *A. Problem solver, Estimation & Forecasting Division*

Different type of projects are undertaken to develop. The estimation division has idea and experience of type of project to be developed. It involves predicting *cost*, *time, quality, efforts and number of people* required to develop that software. They should predict and analyses the *risk and budget.*. The *complexity* part in software is very crucial. The requirements are mapped to the design part in software development. The *failures, risks* are taken into account and goal is set. There are various *factors* like variable factor, influencing factor which are influences the development of models.

### *B. Quality Assurance Planning & Reuse Centre*

This Center is involved in planning of quality attributes and reuse aspects involved in development of software. The Centre is associated with all other activities to implement and report the quality process within an organization. There are many activities which needs planning and managing like resource, risk, Technology, training, defect, efforts, time, budget, documentation, testing, test cases, strategies, market research, analysis, design. The technology should be selected for building reusable libraries. Training is required for developing reusable libraries. Sufficient budget and resource planning has to be taken care of.

Planning should be systematically done so that development goes smoothly. This need high investment and support from management. The management should support the developing reusable repository so that the cost of development can be brought down and hence high quality can be achieved. This needs technical engineers and hence they had to be trained. The tool support should be well planned. The risk involved should be reduced so that the process is conducted smoothly. The reusable product should be certified and for this a culture has be developed within an organization. The technology should be carefully adopted and accepted from future point of view. Complex project needs architecture development.

Software organization should have *risk rating* and classify each software development phase's deliverable according to the levels. Highest share of the budget should go to the most critical levels rather than distribute it evenly to the whole phase. Every phase of development should be classified into three types of rating i.e high risk, medium risk, and low risk. Every risk rating can be classified according to the type of work product with a known weight percentage (Wp1, Wp2, and Wp3). The practices (P) involved in work can be rated as P1, P2, and P3 [22] etc. Thus each deliverable can be classified into two or more

work products according to risk rating levels introduced by software development organization and high effective practices can be applied to high risk rating level work products.

### C. Standard and Rules Enforcing Division

This division is basically based on enforcing rules, regulation, principles, following guidelines & ethics, implementing policy, strategy, building documentation for activities thereby enforcing quality into the software. Ethics in this division creates atmosphere and environment and hence culture within an organization[6][16]. It is also responsible for implementing quality assurance planning policy and reuse strategies required within the organization. Strict implementation of quality assurance plan will prevent defects from propagating it to next stages [18] and builds quality in product. Lot of effort is saved as it prevents insertion and propagation of defects.

### D. Manufacturing Processing Division

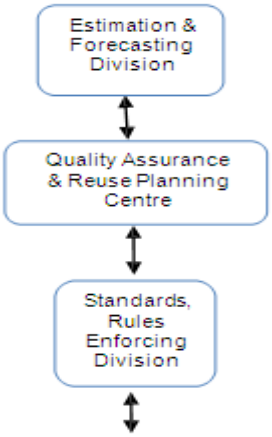
Development of software is a manufacturing process. It is broadly classified into four phases which are analysis, design, implementation and maintenance. Each phases is broken into sub process. Every sub process has input, output, control mechanism and transformation. The input of one phase is the output of other phases. Each complex process is associated with many activities. There are different approaches to software development like structured development, object oriented development, and component based development and web-based development [21].

*Analysis* - The requirement are taken and analyzed from reusable and quality assurance aspects.
Requirement specification and feasibility report is produced after seeing the feasible solution of the problem. The solution is obtained from technical, cost, social and legal aspects. Feasibility report if accepted is signed and approved to proceed for designing phase. In case of any deviation from normal process the action is taken to bring back the process to its normal track.
*Design* - After analysis phase once more review of requirements are done. They are mapped to designing aspects. They are converted into designing phase with help of available tools. Testing is followed by coding. The designing is done looking into the issues of reusing code, modules, and form, and report, tools like DFD, ERD, and Gantt Charts. UML diagrams like class diagram, object diagram, activity diagram, sequence diagram are quite helpful in case of object oriented development etc. Legal issues are taken into consideration. Reuse practices are adopted on basis of organization and management support, development process and techniques adopted. Reuse is made a formal process in this model.
*Implementation*-Software is implemented. There can be parallel, direct or indirect conversion of software. There are many critical success factors in implementation of software. *Maintenance* - This is followed by maintenance of software. Best practices are adopted for developing a repository for long run.

### E. Monitoring & Reviewing the Manufacturing Division

The main purpose of this division is to monitor the manufacturing process and all the activities involved in software development process. These activities associated with the manufacturing division are carefully and closely observed. If something goes wrong from normal process it is brought back by taking proper action.

The various tools are used to detect and remove the software defect originated in different phase of development/ manufacturing. Some of the tools used are root cause analysis, orthogonal defect classification, Defect Containment Matrix, work product Classification Matrix [8]. The matrix is applied to tract the defect injection and removal activity i.e. in which phase a defect was inserted and in which phase was removed. Their role is to verify that each phase artifact to software specifications and requirement and deviation from them.



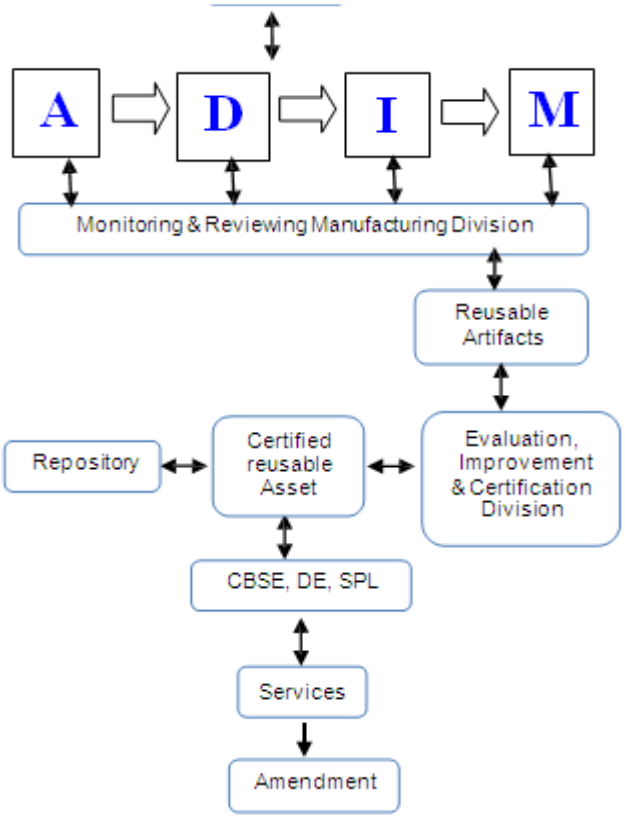



**Fig. 1 A Model on SQA and Reuse**

A – Analysis, I – Implementation, D-Design, M-Maintenance

*F. Reusable epository*

Software reuse is the process of creating software system from existing software rather than building from scratch. Software reuse improves quality, productivity and saves time. The reusable artifacts are collected in the repository. It can be used for reusing product for different reusable areas like Domain Engineering, Product Line, Component Based development, application framework etc**.** This will bring the cost down and improve quality of products in long run

*G. Evaluation, Improvement & Certification Division*

Quality control division involves evaluation, improvement and certification. This evaluation means measuring the quality This division is associated with metrics framework. The various metrics are used which indicate the development of the process. The reusable metrices should be incorporated. E.g. DRE i.e Defect *Removal Efficiency* is a common metrics which measures the effectiveness of defect removal of development phase by comparing defects found and removed before release to defect found after release[7]. It is given in percentage.

The evaluation process is lengthy process. i) Establish a team to evaluate the goal and scope of the process. ii) Specify what has to be evaluated. iii) Specify the quality, technique, goal and specification of evaluation. iv) Design the technique, tool, cost and schedule for evaluation. v) After that execute the evaluation process in particular environment. vi) Analyses the result and develop the report. During evaluation different aspects are monitored. In case of poor result the improvement are suggested [6] [13]. The process is improved for betterment of results [10].
Quality management system is a cyclic, continuous improvement process which involves management, resources, product development, measurement analysis and improvement [5]. Self-assessment and subsequent investigation provides introspection and give organization to strength and weakness [10].

Different techniques are proposed by different author for estimating quality of software. These techniques are studied and categories [24] which are Regression Tree,Poisson Regression, Zero-inflated Poisson, Case Based Reasoning, Decision Tree, Data Sampling, Boosting, Simulation, Neural network, Genetic Algorithm, Bayesian Network, Principal Component Analysis, Adaptive Boosting, Data Mining, Statistical Method, Fuzzy Logic, Software Life cycle management Method, Combination of estimates, Rule Based Induction (RI), Function Points, Rules based, Classification and Regression Tree algorithm, Product based, Process based, Metrics Based, Clustering, Classification Tree Disc Algorithm, Dependency Graph and Analogy Based Reasoning, Rule Induction Models.

The certified product means that the quality is assured after being tested and software do not presents unexpected and undesired behavior. The process can be certified. More over the reusable artifacts are certified by different ways so that reusable libraries can be developed. All this certified assets are listed with their function, classes and code. Their behavior is also explored and confirmed.

*H. Reusable Artifacts, Finished Product & User Satisfaction: The quality of product is determined by the internal factors*

generated during all the phase (analysis, design, implementation and maintenance) of development. This internal quality has impact on the external quality of the product.

*I. Services & Repository*

The components which are certified Guarantees the

services to the user. These certified components are stored in repositories and used for in Component based (CBSE) Engineering, Software Product Lines (SPL), Domain Engineering (DE).

## IV CONCLUSION AND FUTURE ENHANCEMENT

Model emphasis on quality assurance, reuse and making repositories of components to reduce, cost and speed the process of development. Model takes into account rules, policy, strategy, virtues which enforces quality into software. To make this model more usable and to increase its applicability we like to integrate it with different models. The model emphasis on making the reuse as a formal process. The model guarantees the development of quality in the product by integrating standards, rules, monitoring, evaluation, and improvement and certification techniques. This model is useful in developing repository and reusable assets which are used in Components based Engineering, Domain Engineering, Software Product Lines.

Our further work is to solve problem of reuse, repository and find ways to standardize it. Implementation of model using case study will be our next work. Efforts are required to modify model for estimating quality by using techniques like genetic algorithms, fuzzy logic, simulation.

## REFERENCES

[1] Wm. Arthur Conklin, "Software Quality Assurance: The Need for Definitions", Proceedings of the 44th Hawaii International Conference on System Sciences-2011.

[2] Ashwin Tomar, V M Thakare, "Identification and listing of factors affecting Software Quality Assurance", International Journal of Internet Computing ISSN No: 2231–6965, VOL-1, ISS-3 2012

[3] Feldman Stuart J, "*Quality Assurance: Much More than Testing",* www.acmqueue.com, 2005.

[4] Ejaz, Reham, Nazmeen. Mubina, Zafar Maryam, "*A Quality Assurance Model for Analysis Phase"* ,NSEC 10, 4 OCT-2010, ACM, DOI= http://doi.acm.org / 978-1-4503-0026

[5] Yasuda Katsuyuki (Hitachi Ltd) Yamada Shigeru, "*The Concept and Practice of Software Quality Assurance in Japan in the Global Era"*. In Proceedings of the 7th IEEE International Symposium of High Assurance Systems Engineering (HASE '2002), IEEE.

[6] Alan R. Peslak, Improving Software Quality: An Ethicsbased Approach, 2004.

[7] Omar Alshathary, Hlege Janicke, Hussein Zedam, Abdullah Alhussein, *" Quantitative Quality Assurance Approach",* 2009 IEEE, DOI

[8] Tobias Otte, Robert Moreton, Heinz D.Knoell, "*Development of a Quality Assurance Framework for Open source Development Model",* The Third International conference on Software Engineering Advances, 2008, DOI.10.1109/ICSEA.2008.17.

[9] Neelam Bawan, C.V. Srikrishna, *" A novel method for Quantitative Assessment of Software Quality "*, International Journal of Computer Science and Security, Vol 3 : Issue (6).

[10] Hardgrave Bill. C, Armstrong Deborah. J*," Software Process: It's A Journey Not a Destination"* in Communication of the ACM Nov 2005, Vol. 48. No.11.

[11] Alsultanny Yas A, Wohaishi Ahmed M, 2009, "*Requirements of Software Quality Assurance Model"*, IEEE 2009, Second International Conference Environment and Computer Science, DOI= http://doi.acm.org/10.1109/ICECS.2009.43.

[12] Arun Sharma, *"Design and Analysis of Metrics for component based Software System"*, Ph.D thesis.

[13] Saif Shahela, Khan Aliya Ashraf, Arif Fahim, "*An analysis of a Comprehensive Planning Framework for Customizing SQA"*, NSEC'10, 04-oct-2010, ACM 978-1-4503-0026-1/10/10.

[14] Phongpaibul Monvorath, Boehm Barry," *mproving the Quality Through Software Process Improvement in Thailand: Initial Analysis*,WoSQ' 05, May 17, 2005, USA.

[15] Nayan B. Ruparelia, *Software Development lifecycle Models,* ACM SIGSOFT Software engineering Notes, Vol 35, No-3, Page 8, May 2010, http://doi.acm.org/10.1145/1764810.1764814.

[16] Peslak Alan R, *Improving Software Quality: An Ethics Based Approach, SIGMIS'04, April 22-24, 2004,* 10.1109/NISS.2009.114.

[17] Sun Myung, Soobin, *Korean Software Process Quality Certification Quality Certification Model,* First ACIS/JNU International Conference on Computers, Networks, Systems, and Industrial 2011, DOI=doi.acm.org/10.1109/CNSI.2011.

[18] Dr. K.V.K.K. Prasad, Software Testing Certification Study Guide, Dramtech Press.

[19] Ana Sanz, Javier Garcia, Javier Saldana, Antonio Amescua, "*A Proposal of a process model to create a Test Factory"*, WoSQ'09, May 16, 2009.

[20] Swaminathan.Balan,"A, "A Composite Model for SQA", *www.stickymind.com.*

[21] Kenyer Dominguez, Maria Perez, Anna C. Griman, Maryoly Ortega, Luis E. Mendoza, *"Software Quality Model based on software Development approaches ".*

[22] Per Runeson, Martin Host, *"Guidelines for conducting and reporting case study research in Software Engineering*", Empirical Software Eng. 2009, DOI -10.1007/s10664-008- 9102-8.

[23] Ashwin Tomar, M.Thakare, "A Study of Software Reuse and Models", International Journal of Computer Applications, IJCA Proceedings on National Conference on Innovative Paradigms in Engineering Technology, Vol No-15, 2012.

[24] Ashwin Tomar, V. M. Thakare, "A Systematic Study of Software Quality Models", International Journal of Software Engineering & Applications (IJSEA), Vol.2, No.4, October 2011