Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 5.258

IJCSMC, Vol. 5, Issue. 7, July 2016, pg.30 – 35

A Requirement Model for Quality Product using Reuse

[1]ASHWIN TOMAR, [2]V.M.THAKARE
[1]Computer Science, MCA Department, under Pune University, Pune, India
Siddhant Institute of Computer Application
[2]Computer Engineering & Science, Amravati University, Amravati, India
P.G. Dept. of Computer Science
[1] mcatomarashwin@gmail.com, [2] Vilthakare@yahoo.co.in

Abstract: ***Software requirements play a very important role in producing quality products. Most failures in software products are due to errors in the requirements and design phases - 64% of total defect costs and 36% for coding, implementation phases. Requirement should be stable and there should not be requirement change more that 2.5%. So a Conceptual Requirement Model using Reuse (CRMUR) is proposed which improves quality of product. This model is implemented in case of system development.***

Keywords: ***SQA - Software quality assurance, SPL – special product line, CBSD – Component based software development***

I. INTRODUCTION

Quality assurance is defined as conformance with requirements, fit to be utilized, efficiently produced to meet what the market expects. Quality Assurance is assuring quality by controlling the factors which inhibits quality development during software development process. Software quality assurance and reuse are very important aspects in development of software. SQA monitors the software Engineering processes 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 "Software Quality Assurance 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. Product oriented approaches such as models by Boehm, McCall aim to assure the quality of the product by evaluating the characteristics.

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. There are lot of benefit from Software reuse like it

saves time, improves the quality of software and reduce cost. It refers to developing and defining a standard formal reuse process for organization.

When a project is large, complex, the quality is very important while gathering requirement. Many project's software requirements specification(SRS) are filled with badly written requirements. Because the quality of any product depends on the quality of the raw materials. Poor requirements cannot lead to excellent quality software[6]. There are many factors which should be considered while gathering of requirements. In this paper a requirement model is proposed based on quality assurance and reuse process. The scheme of the paper is as 2.Literature survey 3.Quality requirements 4.Requirement reusability 5.0 Conceptual Requirement Model using Reuse technology. 6. Result and Conclusion.

II. Literature Survey

Ejaz, Zafar et.al[3], 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 in analysis phase itself. Hence the author proposed a model.

G.Garcia, M.Arcilla, C.Pacheco et.al[4] proposed a model to reuse the software requirements in form of Requirements Catalog. Software reuse and requirements reuse are recognized as an effective way to improve software quality and productivity. The requirement reuse process is through four principal activities like searching, selecting, adapting, and implementing.

L. Zhao et.al[5] 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. In open source development user of the code has to have a license agreement and pay some money to the owner or maintainer of the code.

R.Rajnish, R.Vyas et.al[6] aim is to focus on relationship between quality requirements and project outcome. Constant changes(called as requirement volatility) i.e addition, deletion, modification in requirement during the software development have great impact on cost, schedule, performance and quality of final product. Every phase of software development is effected by requirement volatility. He proposed practices for writing quality requirements.

D.Pandey, U.Suman, A.K.Ramani et.al[11] presented requirement modelling framework with the analysis of modern requirement modelling techniques.

III. QUALITY IN REQUIRMENTS

Project failures are due to poor requirements. Project manager prevents failure of project through requirement management. There are reasons of project failures due to requirement gathering. They are inadequate expertise on part of participating requirement engineer, insufficient allocation of time and resources within larger development project, problem in defining scope, ambiguous understanding of processes, problems of volatility i.e changing requirements, high maintenance cost, unsatisfactory development of system, wrong selection of elicitation technique. Other reasons are that do not have technical knowledge, lack of trust and cooperation among stakeholder, non involvement of stakeholder, lack of management support, conflicting among stakeholder interest, insufficient input from stakeholder, bad requirements, costly rework, schedule overruns, dissatisfaction in stakeholder[7][8].

Requirements errors, defects detected in development, maintenance phase costs hundred times if not removed during requirement phase. It is an iterative process for bringing quality into requirements phase, following things should be carried out[6].

a) Peer review of requirement – costly method, detect only formal and cosmetic defects.
b) Theoretical reviews - Practically elicitation requirements to remove the deficiency of theoretical reviews.
c) Different reviews with validation – is effective method
d) Inspection
e) Trained stakeholder: Selecting education trained stakeholder for requirement gathering – high quality requirements depend on the right people who engage a number of RE technique.
f) Removing deficiencies: in requirement review
g) Approved Requirements: next level of requirements is started only after the **previous** level has been reviewed and approved and design is started only after the entire set of the requirements is approved.
h) Change Management Process: When design, coding or testing methods detect an error in the requirements, the change request is submitted and affected requirements are updated and reapproved as per the change management process.
i) Competent Reviewers: reviewers are not competent with a requirement which affects quality. Reviews is psychological. Requirement authors rely on the review, falling into illusion that a version submitted to review is not final and will be checked on before getting to the design phase.
j) Change management also affects quality.

k) Requirement review checklist should be developed. This checklist is translated into practical tasks like correctness, feasibility, completeness, verifiability and non ambiguity, clarity and non ambiguity, non-redundancy. The potential internal changes to the requirements to be resolved before the requirements are approved, the change management discipline should be strictly controlled. The quality should be evaluated against your requirements. Incomplete requirement causes lot of extra rework. The time & cost required for software development increases.

IV. REQUIRMENT REUSABILITY

Industry can be saved by reuse concept by speeding the work and decreasing the time and cost of its production**.** For this identify requirements that are candidates for long-term usage by the enterprise. Reuse is associated with project, enterprises, portfolio, programs.

Requirements to be reused can be stored in repository for using it any time**.** For example login Form requirements are reusable. Requirements is reusable at different levels i.e at project, organization or corporate level, program or system level, There are many benefits of requirement reuse.

a) Requirements *need not be re-validated* with stakeholders repeatedly.
b) It *ensure consistency* of Requirements & Business Rules within organization or program.
c) Test Cases/Test Coverage is *already available* and can be reused.
d) *Reduce Requirement*s work for subsequent uses.

V. CONCEPTUAL REQUIRMENT MODEL USING REUSE TECHNOLOGY

The proposed requirement model integrates quality assurance and reusability concepts. The model is the summation of various concepts involved while taking in the requirements for the projects. The SQA is affected by various factors like planning, standards(of code, design), procedure, documentation, guidelines, responsibility assigned, technology, right conduct, authority, approvals, environment, culture, risk, project size, report, reuse (of code and design), cost involved, efforts, practice, schedule, customer satisfaction, tools, deadlines, revision, checklist, manuals, Inspection, defects, certification, management, budget pressure, resources, process metrics, goal[2][10] etc.

The model suggested is the integration of various concept related to requirement gathering, reusability and quality assurance**.** The abbreviation used for the model is "CRMUR". The models suggest various step which are explained below.

5.1 Feasibility Study

The problem is studied followed by feasibility study. The feasibility study is performed from technical, legal, economical, social aspects etc. The feasibility means if the system feasible or not(possible). It involves fixing of scope, vision and objective of system. The strength, weakness of the system is studied.

5.2 Elicitation & Analysis

Elicitation is practices of gathering requirements from users, customers, and others stakeholders. Requirements elicitation can therefore be defined as the process of uncovering and documenting the capabilities required by users to solve a set of problems or achieve a set of goals. Errors in elicitation stage are serious and very difficult to remove. They cost more if not removed on time. Errors are due to inadequate system specification and design issues. Problem in Elicitation is related to scope, understanding and volatility[10]. Elicitation i.e requirement gathering is associated with various activities like (i) identification of key holders (ii) Training the stakeholder (iii) Collecting raw requirements from various viewpoints (business, customer, user, constraints, security, information, standards) using various techniques like interview, questionnaires (iv) classifying & prioritize the requirement like functional, operational, technical, transitional (v)Interpreting the requirements i.e prioritizing, analyzing, resolving the conflicts (vi) Establishing the relationship between requirement[7].

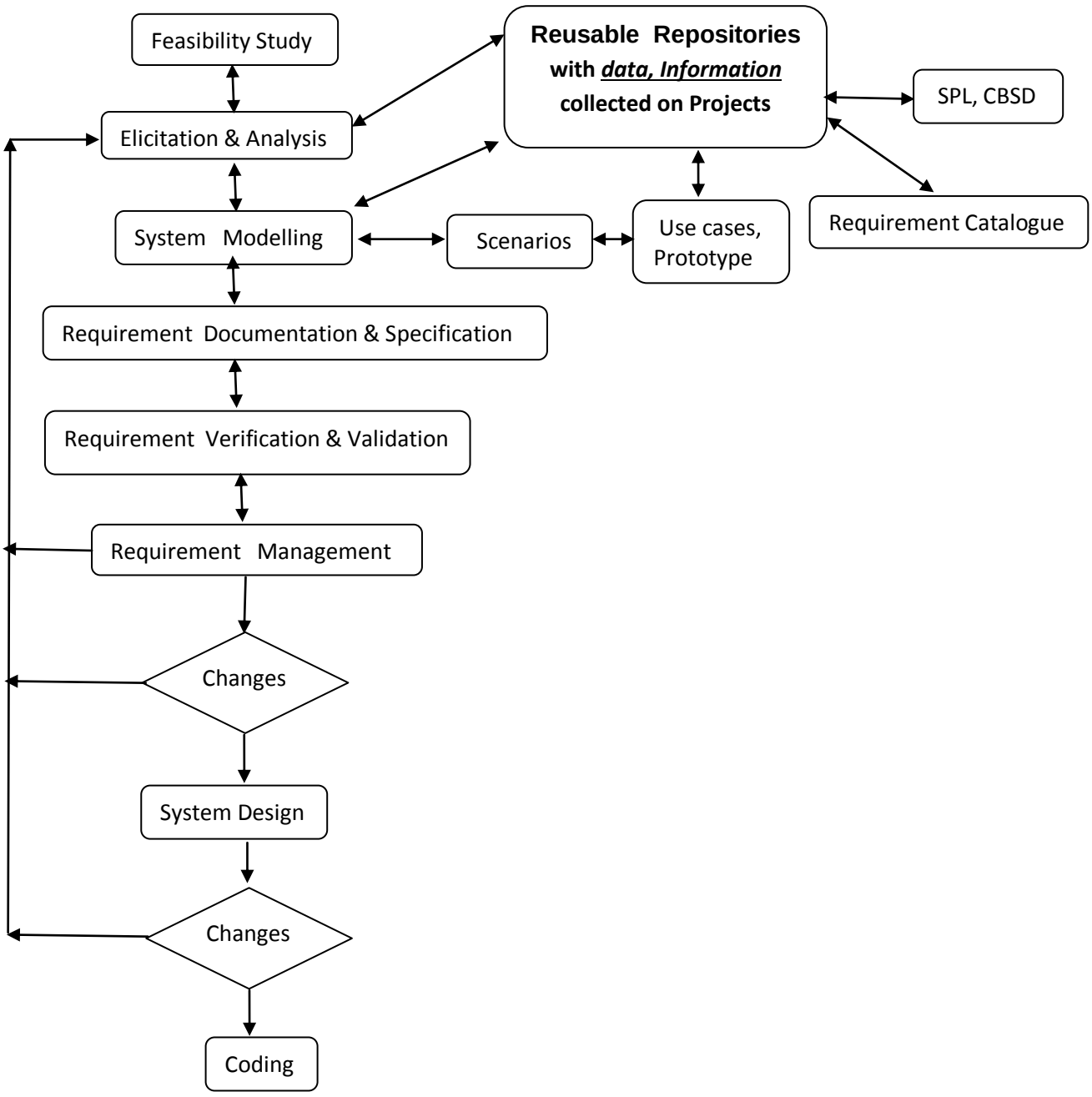

Fig.1. Conceptual Requirement Model using Reuse (CRMUR)

There are different ways of gathering requirement like Interview, Questionnaires, observation, prototyping, scenarios, viewpoints, Introspection, brainstorming, Introspection etc. Every methods has some benefits and disadvantages.

The detailed information of different projects is collected and stored in repository which is reusable. The 15 different types of information related to Project are its problem, mission, vision, context, scope, Deliverables(result of process, objective, overview), System(background, system), Objectives(related to project, system), Assumptions(related to project, system), Constraints(related to project, system), Environment (social & physical characteristics of the project), Opportunities (for improvement), Challenges(of projects), Risks, Stakeholders, Processes(details), Functional, Non-functional, Implementation[7]. Requirement information is associated with ID, Name, Description, Type(classification of objective), Source(can be document, person or an organization), reason for inclusion of objective, priority. There are seven support

Information Types which are required like Glossary(abbreviations, acronyms), Dictionary(data definitions), Issues(system related), Actions(system related actions),idea(system related), references(cited), Appendixes[9].

5.3 Requirement Reusable Repositories & Catalogues

Requirements are stored in repositories and in form of catalogues after classifying the requirements (based on functionalities), validate and verify and record before storing it in repositories. The reusable requirements can be partially and fully adjustable before using it. Catalogues helps in maintain the quality of requirements. It solve problems of incompleteness. The requirements can be traced to build relations. The traceability helps to trace details of stakeholder like source of a requirement, changes in requirement, impact and inform them. The tracing can be forward and backward. The reusable requirements are identified and their priority can be set as high, medium or low. Risk causes and consequences can be categorized (high, medium, low).The type of requirement can be classified. The status of requirement can be validated, adapted, accepted, accepted pending, incomplete, complete. The adapted requirement can be reused. Reusing Requirements can be done by different ways:

a) by sharing the business rules, files or records.
b) by sharing entity-attribute i.e person related attributes like surname, age, dob, address, city, state.
c) by using terms which are defined in glossary.
d) by using the requirements via << include >> relationship of use case scenarios i.e there are different scenarios which can be included via use cases.
e) by converting the scenarios test into diagram and then reusing it through use case scenarios. They can be reuse via << include >>.
f) by identifying the reusable use cases.
g) by reusing requirements via trace links
h) by reusing requirements via traceability i.e traceability matrix
i) link to same type of requirement in multiple use case scenarios
j) by tracing the requirements via traceability diagram
k) by tracing same requirements to multiple artifacts including across projects.
l) by tracing requirements to use case. Use cases are traced to test case.
m) via OLE Embedding of existing Spreadsheets and Visio Diagrams
n) Multi-level Impact Analysis/Trace Report shows reuse of Requirement via Trace
o) Outside artifacts such as Visio Diagrams/Excel Spreadsheets can be reused in your Projects via OLE embedding.
p) by setting permission to all types of users.

5.4 System Modelling

Modelling is visualizing, understanding and converting the conceptual into the design form. Tools can be used to model the system requirements. Tools like use case are to be used to model the system requirements.

5.5 Requirement documentation and Specification

Documentation is required while gathering the requirement. Software SRS is documented in detail. The document should provide details of purpose, scope, definition, interfaces(of system, hardware, software, communication), operations, user characteristics, constraints, assumptions, dependencies, specific requirements, performance requirements, logical database requirements etc.

5.6 Requirement verification and validation

It is the process of ensuring that the requirement document is unambiguous, consistent and complete and the stakeholder are satisfied with final requirement specification. The output of requirement documentation process is the input of the verification and validation process. The output of V & V process is the finalized requirements specification document agreed and authorized by the stakeholder. Inspection, Testing and Checklist generation are used For V & V[9].

5.7 Requirement Management

It is preventing changes to requirement, controlling requirement versions, tracking status of requirements, tracing requirements. Changes can be due to many reasons addition of new requirements, fixing errors, adding new Links, numbering, identifying[11]. The degree to which requirement changes over a time period is called volatility of requirements. Changes can be checked by using metrices. The volatility decreases as the requirements matures. Some tools are used for collecting, viewing and changing requirement like ARM (automated requirement measurement), Dynamic object oriented requirements systems, requirements use case[9]**.**

5.8 System Design & Coding

Most failures in software products are due to errors in the requirements and design phases as high as 64% of total defect costs and 36% for coding, implementation phases. Requirement should be stable and there should not be requirement change more that 2.5%.

VI. RESULT

The quality is an important part of any software. Assuring quality process start from very beginning. Model is proposed based on quality assurance and reuse technique. The requirement extracted can be stored in repositories which can be used in various applications when needed in different ways. This can save time, cost and speed of our development by reusing these requirements in future.

VII. CONCLUSION

The model is economical and save time and energy if taken care while elicitation of requirement. The model is useful in case of website development, component development and other type of projects. The Quality of project is assured by assuring the quality of requirement.

REFERENCES

[1] W.M Arthur Conklin,“ Software Assurance: The Need for Definitions”, Proceedings of 44th, Hawaii International Conference on System Sciences 2011, IEEE.

[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, no-3,pp 43-47,2012.

[3] R. Ejaz, M. Nazmeen, M. Zafar, “A quality assurance model for analysis phase,” *NSEC* ’ pp.1-4, 2010.

[4] . I.Garcia, M. Arcilla, C.Pacheco, J.A Calvo-manzano, “A proposed model for reuse of software requirements in requirements catalog,” pp. 1–45, 2016, www.researchgate.net/publication/ 267810166.

[5] 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.

[6] R.Rajnish, R.Vyas, “Writing Quality Requirements (SRS): An Approach To Manage Requirements Volatility,” Indian Journal of Computer Science and Engineering,,vol.1, no.1, pp. 28–37, 2007, www.ijcse.com/docs/ IJCSE10-01-01-06.pdf.

[7] M.Q.Hanif, “Issues and Challenges of Requirement Elicitation in Large Web Projects,” January, 2010,

[8] J.D.Procaccino, J.M.Verner, S.P.Overmyer, “Case Study : Factors for Early Prediction of Software Success & Failure,” pp. 1–10,

[7] A.E.K.Sahraoui,“Managing Complexity in Requirement Elicitation”,www.csdm2013.csdm.fr / MG/pdf/ Managing_Complexity_in_Requirement_Elicitation.pdf

[9] M.U.Bokhari, S.T.Siddiqui, “Metrics for Requirements Engineering and Automated Requirements Tools,” Proceedings of the 5th National Conference, INDIACom-2011.

[10] 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, September, October 2009.

[11] S.D.Pandey, U.Suman, A.K.Ramani,“A Framework for Modelling Software Requirements,” IJCSI, vol. 8, no. 3, pp. 164–171, 2011, ISSN:1694-0814.

