iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 2 **Issue:** IV **Month of publication:** April 2014

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

ISSN: 2321-9653

INTERNATIONAL JOURNAL FOR RESEARCH IN APPLIED SCIENCE AND ENGINEERING TECHNOLOGY (IJRASET)

Software Quality Assurance

Vishal Manocha

Student of Information Technology

Ch. Braham Prakash Government Engineering College

New Delhi-110073

Abstract: When **we see Software Quality Assurance then we understand that it is very essential to each software development process. It helps in reducing the number of incomplete deadlines and also the time spent on design and code as per our requirements. With the help of SQA we can examine those guidelines that are helpful in advancing the development process. Also, a study can be made to show stake holders that continuous progress is being made on project. Basically Software Quality Assurance (SQA) features the entire software development process which monitors and helps in improving the process that ensures correct standards and procedures are followed and also guarantee that problems are found and dealt with. Its main intention is towards protection and if properly followed then it could result in the production of good quality software. This paper basically stresses the need of a software quality process and also describes the ways by which it could easily be achieved. [1][2]**

Keywords**: Software Quality Assurance Group Activities-1, Group Activities-2, Quality Factors, Reason for Software Bugs, Capability Maturity Model (CMM)**

INTRODUCTION

As we know that a lot of money is already spent in attempting to make good quality software as software bugs are very ordinary. Software Quality Assurance determines the prevention of software which results in the production of good quality software .Since past numerous standards are followed in this area, including one of the most famous and acknowledged IEEE IS0 9000 software quality and management guidelines.

Mostly for all computer systems, the cost of software comprises a major part of system. We all know that software is very crucial and precious, such that if development stage of software lacks quality, then the software that is created will definitely neglect the quality. Thus we can say **"**Software Quality Assurance" includes the complete software development process that helps in monitoring and improving the whole process that ensures that only correct standards and procedures are followed and also guarantees that problems are found and dealt with. Software Quality Assurance mainly focuses on developing a sound software development methodology that will create good quality software [1] [2].

This paper has been separated into seven sections. First being Introduction, Section II refers to Software Quality Assurance, Section III mentions Quality Factors with software products. Section IV describes Why Software Is Generally Prone to Bugs. In the end, Section V discusses the most important topic of this paper Capability Maturity Model (CMM) and Section VI verbalize the conclusion of this paper. Section VII is References.

I. WHAT IS SQA?



Figure1.Software Quality Assurance [9]

It can be suitably defined as a set of organized activities which provide proof of the ability of a software process to generate a software product which is suitable to use [3].

- It supervise the processes and products during the software development lifecycle to assure the quality of a given products.
- It furnishes the management with objective result regarding process completion to approved plans, procedures, standard and analyses.
- For supervising the products, there is a need to concentrate on the superiority of product inside each phase of the Software Development Life Cycle e.g., requirements, test plan, architecture, etc.
- Its main objective is to discover and vanishes the defects throughout the lifecycle, as soon as possible [3].

After knowing the meaning of software quality assurance now we are going to study about two important group activities of software quality assurance in the coming sub sections.

a. SQA Group Activities – 1

- Study the SQA strategy for the project.
- Take part in the creation of the project's software process description.
- Analyze software engineering activities to examine compliance with the characterized software process [4].

b. SQA Group Activities – 2

- Inspect the designated software work products to examine the compliance with those defined as part of the software process.
- Make sure that any variation in software work products are documented and treated according to a prescribed documented procedure.
- Note down any information of noncompliance and complains them to management [4].

II. QUALITY FACTORS WITH SOFTWARE PRODUCTS

a. Correctness

A software product is correct, if various requirements as mentioned in the SRS document have been correctly implemented.

b. Reliability

Software Reliability is the probability of software to operate in a failure-free environment for a specified period of time. It also plays an important role in affecting the system reliability.

c. Portability

A software product is called portable, if it can be easily made to work in various operating systems, on different machines, with many other software products, etc.

d. Usability

A software product has best usability, if various types of users (i.e. both skilled and non skilled users) can easily understand the functions of the product.

e. Reusability

A software product has best reusability, if various modules of the product can easily be utilised to create new products.

f. Maintainability

A software product is maintainable, if errors can be well rectified as and when they come up, fresh functions can be easily added to the product, functions of the product can be easily altered, etc. [5].

After analyzing the developing strategy of a software product one is left wonder that why software is generally prone to bugs.

III. WHY SOFTWARE IS GENERALLY PRONE TO BUGS?

As all know developing good quality software is very important for software industries. There is a term called software crisis which is generally used in software industry to stress the difficulties in developing good quality software. Generally there are five major problems that developers face during the software development process. They are miscommunication, software complexity, programming errors, changing requirements and unrealistic schedule [2].

- *Miscommunication:* As we know it is a human tendency that they tend to assume and misinterpret a lot of things while communicating to each other, so there is a widespread miscommunication of information during all the phases of software development.
- *Software Complexity*: Sometimes the huge complexity of software that is developed to serve some useful

purpose makes difficult for a single person to understand it completely.

- *Programming Errors:* Since these are people itself that create the software and people are inherently prone to making errors. Therefore, software bugs are naturally produced because of the programming errors.
- *Changing requirements*: With the time when the requirements change then the software functionality also changes and when we are working on a system with rapidly changing requirements, additional functionality that is added to the system can alter the already existing modules in unexpected ways and these interdependencies among the modules makes the system error prone.
- *Time pressure and deadlines*: Due to the high competition between software development industries any schedule slippages are not acceptable. Sometimes due to the unrealistic schedules of some projects development methodology becomes far from perfect and the software which is developed lacks quality [2].
-

Now we come to know after seeing these problems that software bugs are very common and the coming section discusses the ways to reduce software bugs and make software more reliable.

IV. CAPABILITY MATURITY MODEL (CMM)

Capability Maturity Model (CMM) generally mentions the process improvement approach which is based on a process model. The Capability Maturity Model is helpful to develop and improve an organization's processes. A maturity model is an organized collection of elements that depict the characteristics of operational processes. Generally a maturity model provides

- A place to start
- The profit of community's earlier experiences
- A familiar language and a joint vision
- A strategy for taking actions
- A right method to specify what improvement means for a specific organization

CMM is the first model which is developed by the Software Engineering Institute (SEI) in the mid of 1980s [6].

There are five levels of the CMM which defines the methodology of the software development which is generally ensued by the organization. The coming subsection will describe each level of CMM in detail.

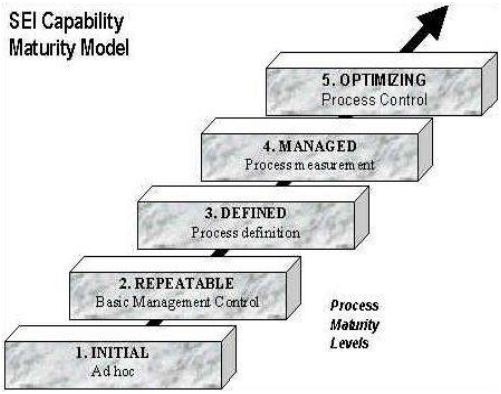


Figure2. CMM Maturity Levels [10]

Level 1: Initial level – This level targets the initial processes where there is a indication of one or more processes getting disorganized. Some of the issues are discovered and corresponding steps are taken to solve those issues in the initial processes.

Level 2: Repeat level – When we see the successes of some processes in the past, then we analyze that processes are reviewed and revised and certain processes which are tracked are the processes on costing, scheduling and many other associated functions.

Level 3: Documentation – Organizations examine a straight line of software development activities and these activities are recorded procedures which developed over the years and both the management as well as development team uses them. The Modules which are developed are combined into current organization.

Level 4: Managing the processes – It is essential to evaluate different product features and processes related to the software

process. Steps are taken to gather data from different processes on different parameters of the software and all are evaluated. The evaluation of all pronominal measures is done which are used in the performance evaluation of the process and the product.

Level 5: Optimization – Persisting betterment is one of the major goals of organization today and it is accomplished by means of some already defined characteristics which are naturally expected from the processes. The estimated result can be different for different processes. A valuable feedback is acquired for different processes and evaluated. The estimated results are made under those difficulties under which a process is optimized to perform. Therefore the results which are evaluated should be optimized results. It facilitates us in taking decisions on the betterment of the process on a persistent basis [7].

Level 5 "Optimizing"	Defect Prevention
	Technology Change Management
	Process Change Management
Level 4 "Managed"	Quantitative Process Management
	Software Quality Management
Level 3 "Defined"	Organization Process Focus
	Organization Process Definition
	Training Program
	Integrated Software Management
	Software Product Engineering
	Intergroup Coordination
	Peer Reviews
Level 2 "Repeatable"	Requirements Management
	Software Project Planning
	Software Project Tracking and Oversight
	Software Subcontract Management
	Software Quality Assurance
	Software Configuration Management
Level 1 "Initial"	(no Key Process Areas)

Figure3. Key Process Areas of Maturity Levels of CMM [10]

a. Benefits of CMM

- Consistency
- Cost Saving
- Self Improvement
- Market Demand
- Performance Demand
- Process Improvement [8]

V. CONCLUSION

Amongst the best profits of software quality assurance is the capability to confront those problems that are encountered during software development, by choosing a best software development process. My research paper mainly produces the need for quality assurance required for a software product and the methods by which software development teams can make to their goals easily. I have examined that software quality assurance is the indivisible part of countless domains.

VI. REFERENCES

[1] Matt Heinzelman et al.” Software Quality Assurance: Techniques and Tools”. Website: http://www.uwplatt.edu...Heinzelman,Matt%20%20Software%20Quality%20[2] Ramakrishnan Venkitaraman et al. “Software QualityAssurance”.Website:http;//www.utdallas.edu~rsv031000ProjectsEnglishSQA.

[3] Dr. K Cooper et al.” Software Quality Assurance for software engineering and architecture and design”. Website: http://www.utdallas.edu~kcooper6362materialSQAsqa.ppt.

[4] Dr. Bruce R. Maxim UM-Dearborn et al.” Software Quality Assurance”. Website: http:// groups.engin.umd.umich.eduCIScourse.descis375pptlec8.ppt.

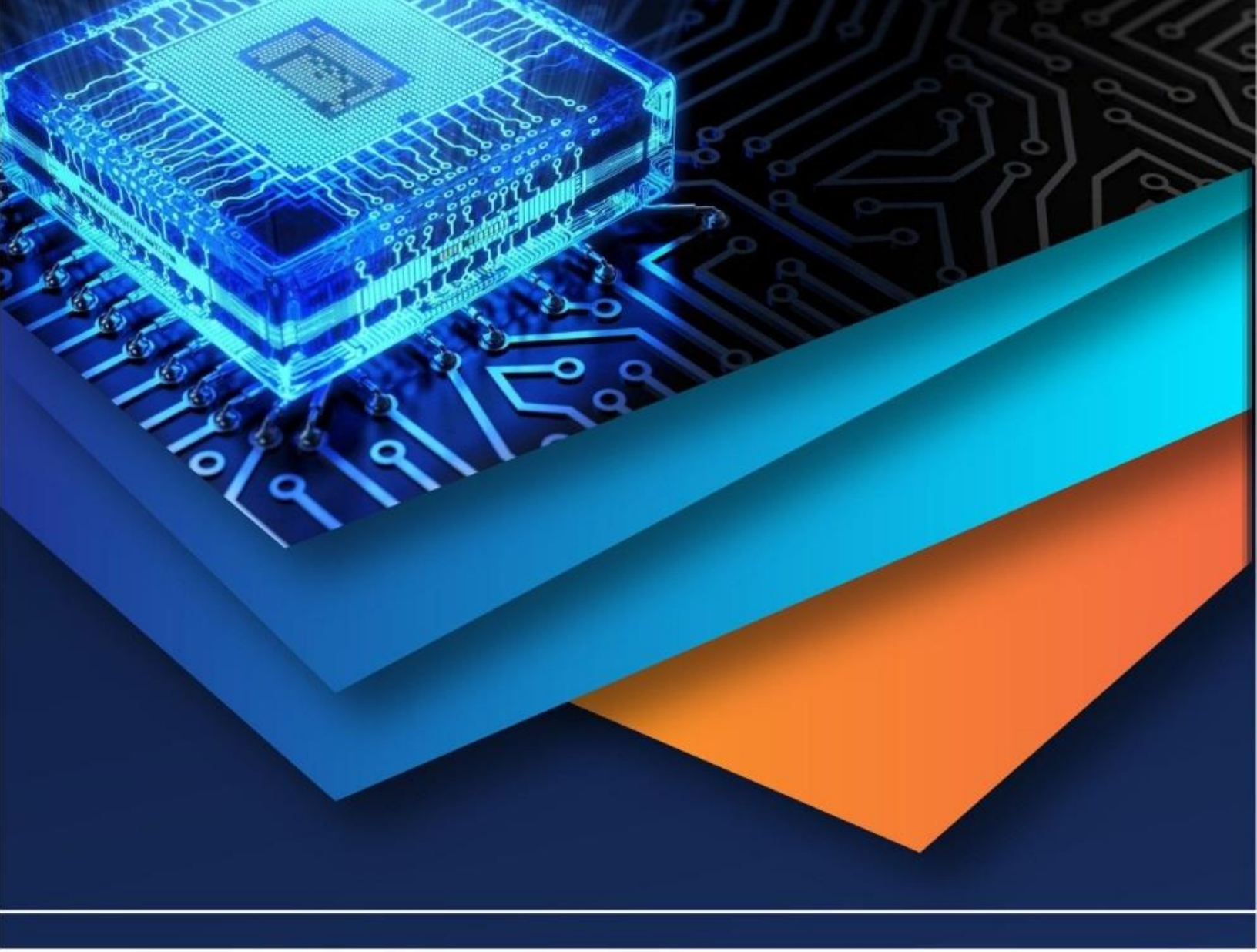
[5]Dr. R. Mall et al.”Software Quality Assurance”.Website:http://www.facweb.iitkgp.ernet.in~spplect14.ppt. [6] Website:http://www.selectbs.com/process-maturity/what-is-the-capability-maturity-model.

[7]Website: http://www.smusolutions.com/2010/ 12/mi0024-q1-discuss-cmm-5-levels.

[8]Website:http://www.vizteams.com/blog/top-6-benefits-of-adopting-capability-maturity-model-cmmi-focus-software-companies..

[9]Website:http://www.itransition.com/work/software-development-quality.php.

[10]David Lipton et al.” Function Points and the SEICapabilityMaturityModel”.Website:http://www.qpmg.com/seicmm2.htm



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)