Indian Journal of Communications Technology and Electronics (IJCTE)
Vol.2.No.2 2014 pp 75-80.
available at: www.goniv.com
Paper Received : 05-03-2014
Paper Accepted: 24-05-2014
Paper Reviewed by: 1. John Arhter 2. Hendry Goyal
Editor: Prof. P.Hendry

IMPLEMENTING QUALITY PROCESS IN DEVELOPING PHASES

[1] **C. SenthilMurugan**, [2] **Dr. S. Prakasam.**
PhD Scholar, Assistant Professor

[1,2]Dept of Computer Science & Application, SCSVMV University, Kanchipuram
[1]Dept of MCA, Thiruvalluvar College of Engineering and Technology, Vandavasi.

ABSTRACT

Software development and maintenance is used to make the error-free Software and also concentrate on time-consuming and complex activity. To evaluate the quality of a software product and to keep its level high is much more difficult than to do them for the other industrial products. In this paper We have concentrated to keep the quality level of software products high, firstly the Software Quality Assurance activity process are implemented in the early stages of software development, it may reduce the problem that was occurring at the time of development and the start and end time of the SQA processes are analyzed and necessary quality factors on organizational level and department/project level were made.

Key words: Software Quality Assurance Software development, SQA process.

1. INTRODUCTION

The Computer has been used for commercial purpose. With every aspect of computer development, Software engineers have been tasked to solve the large and complex programs and in a cost effective and efficient manner. Also the development and maintenance of the software product has become an important criterion.

In the early years, engineers faced many problems, without having a better knowledge in the software fields, such as "Late delivery of software, Development team exceeding the budget, poor quality, user requirement are not completely supported by the software, difficult maintenance and unreliable software and lack of systematic approach.

To develop a software product, the following criteria has to be satisfied:

- User needs and constraints must be determined and explicitly stated.
- The source code must be tested thoroughly.
- The product must satisfy the user needs
- Supporting documents such as user guide, installation procedures and maintenance documents must be prepared. [2]

2. SOFTWARE ENGINEERING

Software engineering is the establishment and use of sound engineering principles in order to obtain economically software that is reliable and work efficiently on real machines. It is the application of a systematic, disciplined and quantifiable approach to the development, operation, and maintenance of software.

The software engineering is useful

- To acquire skills to develop large programs
- Ability to solve complex programming problems
- Learn the techniques
- To acquire skills to be a better programmer

The primary goal of software engineering is to improve the quality of software products and to increase the productivity and job satisfaction of software engineers. [3]

3. SOFTWARE QUALITY ASSURANCE (SQA)

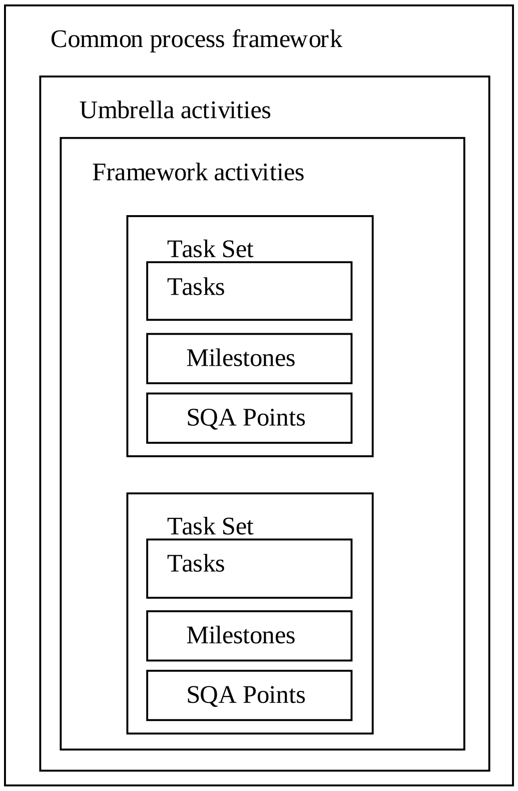
A systematic, planned set of actions necessary to provide adequate confidence that the software development process or the maintenance process of a software system product conforms to established functional, technical requirements as well as with the managerial requirements of keeping the schedule and operating within the budgetary confines.

The difference between the quality control and quality assurance should be recognized. Quality control activities are done to sort the products that do not qualify for the qualified products to not deliver the customer or to not sell in the market.

A Software process is a set of activities and associated results which produces software products. These activities are mostly carried out by software engineers. There are four fundamental process activities which are common to all software process. These activities are

1. Software Specification:
   The functionality of the software and constraints on its operation must be defined.
2. Software Development:
   The software to meet the specification must be produced.
3. Software Validation
   The Software must be validated to ensure that it does what the customer wants.
4. Software Evaluation
   The software must evolve to meet changing customer needs.

4. COMMON PROCESS FRAMEWORK



Task Sets: A number of task sets-each a collection of software engineering work tasks, project milestone,software work products and deliverables and software quality assurance points.

Framework Activities: The task sets enable the framework activities to be adapted to the characteristics of the software project and the requirements of the project team. The generic process framework activities are

1.Communication: The customer requirement information gathering is done by communication.

2.Planning: The planning activity defines the engineering work plan, describes technical risks, list resource requirements,work products produced and defines work schedule.

3. Modeling: The modeling activity defines the requirement analysis and design.

4. Construction: The construction activity implements the corresponding coding and testing.

5. Deployment: The software delivered for customer evaluation and feedback is obtained. [2]

Umbrella Activities

Umbrella activities –such as quality assurance, software configuration management and measurement. Umbrella activities are independent of any one framework activity and occur throughout the process. The umbrella activities are

1. Software project tracking and control: The activity helps to access the software team progress and take corrective action to maintain schedule.

2. Risk management: This activity analysis the risk that may affect the quality.

3. Software quality assurance: This activity maintains the required software quality.

4. Formal technical reviews: This activity helps to analyze the engineering work products to uncover and clear errors before they proceed to the next activity.

5.Software configuration management: This activity manages the configuration process when any change in the software process.

6.Work product preparation and production: This activity defines the model,document forms and lists to be carried out.

7.Reusability management: This activity defines the work product reuse.

8.Measurement: This activity defines the project and product measure to assist the software team in delivering the required software.

5. PROBLEM FORMULATION

Software Errors: The error resulting from bad code in some program involved in producing the erroneous result.

Software faults:Due to the Software errors the Software fault occurs.

Software failures: Due to Bug/defect/fault consequence of a human error.

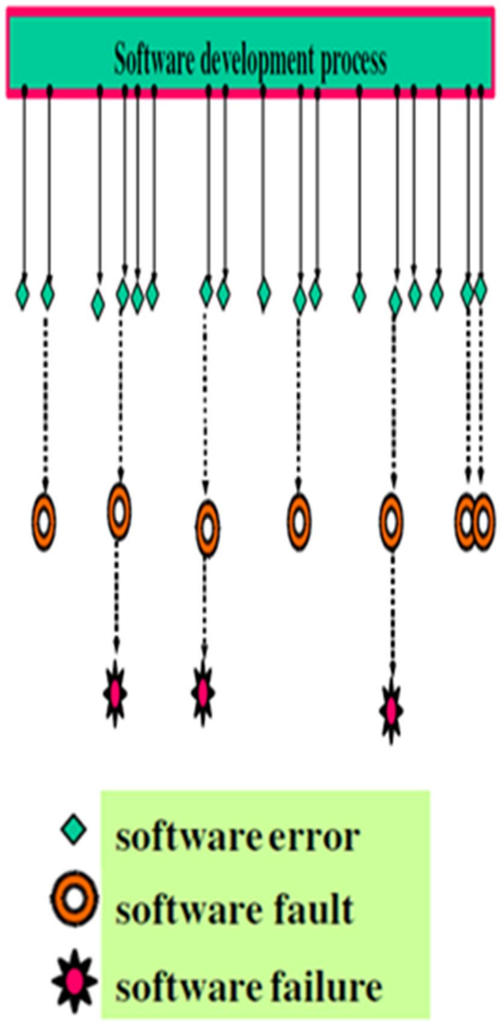


Fig-1: Software Errors, software faults and software failures

Nine Causes of Software Errors

1. Faulty requirements definition
2. Client developer communication failures
3. Deliberate deviations from software requirements
4. Logical design errors
5. Coding errors
6. Noncompliance with documentation and coding instructions
7. Shortcomings of the testing process
8. User interface and procedure errors
9. Documentation errors [1]

Development process relating to defects

Majority of defects is introduced in earlier phases

Table-1: Majority of defects

Phase	**Percentage of defects**	**Effort to fix defects**
Requirements	56	82
Design	27	13
Code	7	1
Others	10	4

Relative cost of fixing defects

Table-2: Cost of fixing defects

Phase in which found	Cost Ratio
Requirements	1
Design	3-6
Coding	10
Unit/Integration testing	15-40
System/Acceptance testing	30-70
Production	40-1000

6. PROCESS OF SQA SYSTEM

The processes are classified into two main groups:

1. Organizational processes, and
2. Departmental/Project processes.

For example,
Galin explains an SQA system as dividing its components five major classes
1). Pre-project quality components,
2). Project life cycle quality components,
3). Infrastructure error preventive and improvement components,
4). Software quality management components,
5). Standardization, certification and SQA assessment components,
6). Organizing for SQA-the human components.

Life-Cycle Phases of SQA System Development:

The implementation order, start point, duration and end point of the SQA processes are analyzed. First of all, the phases of Software Development Lifecycle and SQA SystemDevelopment Lifecycle should be explained. Software Development Lifecycle is a subset of the SQA System Development Lifecycle. Here, the classical waterfall model is chosen as Software Development Lifecycle model

1. *Requirements Definition/Analysis*: In this phase, software engineers gather customer requirements by defining them with the help of customer and domain experts. Most of the time, a developed software must have interfaces with existing hardware and software. Therefore the information about them helps to define the interface requirements. In this phase, software engineers must understand the nature of the program to be built. Therefore, understanding the information domain, system's required functions, behaviors, performance and interface are musts.

2. *Design:* In this phase, software designer identifies the system inputs, outputs and processes. Processing algorithms, data structures, databases and software structures are also defined.

3. *Code Generation:* In this phase, software engineers and programmers transform the design into a code by using a selected programming language. Code review, unit tests, unit integration tests are part of this phase.

4. *System Testing:* In this phase, the system is tested as a whole and system integration is realized. Activities are performed by the testing team.

5. *Installation and Conversion:* After customer approval, the software is installed to serve the customer. If the new software will be used to replace the existing software, a suitable conversion process must be performed to prevent the interruption in the organization's services.

6. *Operation and Maintenance:* Operation phase begins after the installation and conversion is completed. Maintenance activities are performed during the normal operation period, which generally continues a few years.

SQA System Development Lifecycle includes the above software development Lifecycle phases, but it has a few phases prior to software development Lifecycle phases:

1. *Pre-Project Phase:* In this phase, organizational level activities will be performed. None of the activities of this phase are related to a specific project. During this phase, the SQA system of the organization is initiated, organization quality policy and QA methodology are defined, QA staff is assigned an initial SQA component are developed.

2. *Proposal/Contract Phase:* Activities of this phase are performed by the proposal team and legal department for a contract between the customer and the organization. During this phase, firstly proposal team develops a proposal draft from the customer requirements document. After reviewing a proposal draft with a customer, a contract draft is developed from a final approved proposal document. After reviewing the contract draft with the customer, a mutually agreed contract which defines source, timetable and cost estimation for the project is achieved.

3. *Project Preparation Phase:* In this phase, project products, project interfaces, development methodology, development tools, standards, procedures, project schedule, resource and cost estimation, project milestones, staff organization, quality goals, QA activities are identified. [4]

Timing Activities of SQA Processes

The start and end time of the SQA processes are analyzed.

1. Conduct Management's Role in SQA System process is the first step to build an SQA system in an organization. Therefore, this process starts atthe very beginning of Pre-Project Phase and it never ends.

2. Construct Infrastructure of the SQA System is the base process to establish an SQA system in an organization. It starts at the beginning of Pre-Project Phase and it ends at design phase.

3. Do Standardization and Certification is important to show that the quality of the software products produced by the organization has the level of chosen standards. It starts in the middle of the Pre-Project phase after constructing the infrastructure of the SQA system and never ends.

4. Perform Contract Review starts at the Proposal/Contract Phase after having an offer from the customer and ends at the end of this phase.

5. Develop Project Management Plan, SDP and SQA Plan start at the beginning of Project Preparation Phase just after signing a contract with
the customer and ends at the second half of design phase.

6. Perform Activities to Force and Coordinate the SQA System is necessary to support the SQA system at the organizational level. Therefore it starts atthe Project Preparation Phase before entering Software Development Lifecycle phases and it never ends.

7. Develop CM System is necessary to control any changes at the project products (e.g., Source code, documents, data, etc.) and to get the correctversion or releases of the products. Therefore, it starts at the end of Project Preparation Phase just before the software development begins. It ends when the project is

delivered to the customer completely after completing maintenance phase.

8. Conduct Quality Assurance Audits is an organizational level process to control the departmental SQA activities. Therefore, it starts in the middle of Requirement Analysis Phase just before the project level SQA activity starts. It ends after the project is delivered to the customer.

9. Apply Software Quality Metrics Program is an auxiliary process to collect some data about the project and to use them to control the softwaredevelopment process. Therefore, it starts in the middle of Requirement Analysis Phase just before the project level SQA activity starts. It endsafter the project is delivered to the customer.

10. Perform Activities to Support the SQA System is necessary to support the SQA system at department level. Therefore, it starts in the middle of Requirement Analysis Phase just before the project level SQA activity starts. It ends after the project is delivered to the customer.

11. The Apply Risk Management Program is important to take precautions against any possible problems which can be occurred during the developmentLifecycle, and can cause undesirable deviations from project schedule and budget. Therefore, it starts in the middle of Requirement Analysis Phasejust before the project level SQA activity starts. It ends at the end of installation and conversion phase. Because for maintenance phase, a newrisk program must be applied.

12. Perform Documentation Control Activities is necessary to retrieve the project documents later. Therefore, it starts just before the Software Requirements Specification (SRS) document is released and never ends because some documents must be retrieved later.

13. Coordinate Review Meetings is a process which has different review processes. They are starting when a software product is produced and ends when all work listed in the action - item list are completed:

a. The Formal Design Review is performed by the customer after system design is finished by the firm.

b. SRS review is performed after the SRS document is produced at the end of Requirement Analysis Phase.

c. Software Design Description (SDD) review is performed after SDD document is produced at the end of Design Phase.

d. Software Test Plan (STP) review is performed after test plans are completed at the end of Design Phase.

e. Software Test Description (STD) review is performed after test case scenarios are generated at the end of Coding Phase.

f. Code Review is performed after the code is finished and ready to be tested at the end of Coding Phase.

g. The Test Readiness Review is performed to control whether the test environment and source code are ready to test, just before unit integration tests at Coding Phase.

h. Software Test Report (STR) review is performed after tests are completed and a test report is produced at the end of Coding Phase. Similar review process is performed after system testing is completed at the end of System Testing Phase.

i. Software Version Description (SVD) review is performed after all necessary testing and review processes are completed, and the product is ready to release at the end of System Testing Phase.

14. The Apply Test Program is a process to perform all necessary unit levels, system level and customer acceptance level tests. This process starts at the beginning of Design Phase and ends at the end of System Testing Phase.

15. The Apply Process Improvement Program is an auxiliary process to support feedback to the Software Development Process for improvements.Therefore, this periodical process starts when the SQA system frame is established and never ends.

16. Apply SQA Controls to External Participants is an optional process and only performed when at least one part of the project is developed by an external participant. Its purpose is to ensure the quality of the product developed by external participant. Therefore, it starts at the end of Requirement Analysis Phase and ends after the project is delivered to the customer. [1]

7. CONCLUSION

The SQA components were explained by considering the SQA system processes, and their inputs, outputs and sub-processes. They were classified as organizational level and department/project level. An implementation order was presented with implementation time (start time) and end time of the application of each component. Hence, a life cycle for the SQA system is presented by considering the software project development life-cycle. As a result, the aim was to make a better understanding of what will be modeled to create an SQA system.

REFERENCES

[1] Galin, Daniel (2004), Software Quality Assurance – From theory to implementation, Pearson –Addison Wesley, England.

[2] Chandramouli, pradiba, Software Quality Assurance

[3] Omer Korkmazsystem, Simulation for software Quality Assurance (SQA).

[4] David Hooker, Seven Principle of Software Development.

[5] Darrel Ince, ISO 9001 and Software Quality assurance

[6] Abdel-Hamid, T.; Madnick, S. E. (1991), Software Project dynamics: an Integrated Approach,Prentice-Hall Software Series, Englewood Cliffs, New Jersey, USA.

[7]Ambler, Scott W. (2002), Examining the Agile Cost of Change Curve, Ambysoft Inc.

[8] ARENA Rockwell Automation (2007), Forward VisibilityFor Your Business,

[9] Balan, S. (2003), A Composite Model for Software Quality Assurance

[10] Bass, Len; Paul Clements; Rick Kazman (2003), Software Architecture in Practice, Second Edition.Boston: Addison-Wesley, p. 21-24. ISBN 0-321-15495-9.

[11] Bertsekas, Tsitsiklis (1996), Neuro-Dynamic Programming, Athena Scientific.

[12] Boehm, B. W. (1981), Software Engineering Economics, Prentice-Hall, NJ, USA.

[13] Brennecke, Andreas; Keil-Slawik, Reinhard (1996), Position Papers for Dagstuhl seminar 9635 on History of Software Engineering August 26-30, 1996, Germany.

[14] Chikofsky, E.J.; J.H. Cross II (January 1990), Reverse Engineering and Design Recovery: Taxonomy in IEEE Software, IEEE Computer Society: 13–17.

[15] Pressman, Roger S. (2000), Software Engineering - A Practitioner’s Approach, European Adaptation by D. Ince, 5th Edition, McGraw Hill International, London

[16] The Linux Information Project (2004 – 2007), Bellevue Linux, Bellevue, WA, USA.

[17] Warden, R. (1992), Software Reuse and Reverse Engineering in Practice. London, England:Chapman & Hall, 283–305.

[18] Zave, P. (1997), Classification of Research Efforts in Requirements Engineering, ACM ComputingSurveys, 29(4): 315-321

[19] Padberg, Frank (1999), A Probabilistic Model for Software Projects, Proceedings ESEC/FSE 7 109-126, Lecture Notes in Computer Science 1687, Springer 1999.

[20] Padberg, Frank (2000), Estimating the Impact of the Programming Language on the Development Time of a Software Project, Proceedings International Software Development and ManagementConference ISDM/AP-SEPG 287-2

[21] N.Gross; M. Stepanek; O. Port; J. Carey (December, 1999), Software Hell:Glitches cost billions ofdollars and jeopardize human lives. How can we kill the bugs? Business Week Online – International,

[22] NITS (2006), NIST/SEMATECH e-Handbook of Statistical Methods,

[23] Ntourntoufis, Panos (2002), From Software Quality Control to Quality Assurance, UPSPRING Software, UK.

[24] Nuseibeh, Bashar; Easterbrook, Steve (2000), Requirements Engineering: A Roadmap, Future of Software Engineering, Limerick Ireland ACM 2000 1-58113-253-0/00/6

[25] Warden, R. (1992), Software Reuse and Reverse Engineering in Practice. London, England:Chapman & Hall, 283–305.