ISSN: 2321-7782 (Online)
Volume 2, Issue 12, December 2014
International Journal of Advance Research in Computer Science and Management Studies
Research Article / Survey Paper / Case Study
Available online at: www.ijarcsms.com

Quality Assurance and Its Standards: Importance in Various SDLC Models

Anu Gupta
Computer Science And Engineering
Dronacharya College of Engineering
Gurgaon, India

Abstract: ***In recent years customers are more interested in user friendly softwares and best quality products, which make them more satisfied. That is the only reason why Software Quality standards should be considered. Software Quality management ensures that software has low number of defects and it reaches the required standards of maintainability, reliability and portability. Software standards are important to quality assurance as they represent an identification of best practice. The best way to check whether software is deliverable or not is through accessing software quality and getting reviews from its users. This may be based on different types of existing SDLC models. In this paper we will study how quality of software varies in different SDLC models and how it could be corrected through quality assurance standards.***

Keywords: SDLC process models, quality management, quality assurance.

I. INTRODUCTION

Software process models: Software processes are the activities involved in producing a software system. Software process models are abstract representation of these processes. It describes the sequence of phases for the entire lifetime of a product. Therefore it is sometimes also called Product Life Cycle. They include software specification, software design, implementation, verification, validation and most importantly quality control. There are various software process models:

A. Waterfall model

B. Evolutionary development model

C. Reuse oriented model

D. Spiral model

II. TYPE OF SDLC MODEL

A. Waterfall model:

This model takes the fundamental process activities as input and separates each phase independently in a linear manner. Premature freezing of requirement leads to significant rework. During the final life cycle phases (operation and maintenance) software is put into use where quality is tested from user point of view. Maintenance phase was big in size. As the process only allows for a single run through the waterfall, this could be only a first sample phase which means that the further development is squeezed into the last never ending maintenance phase and virtually run without a proper process because of lack of quality.

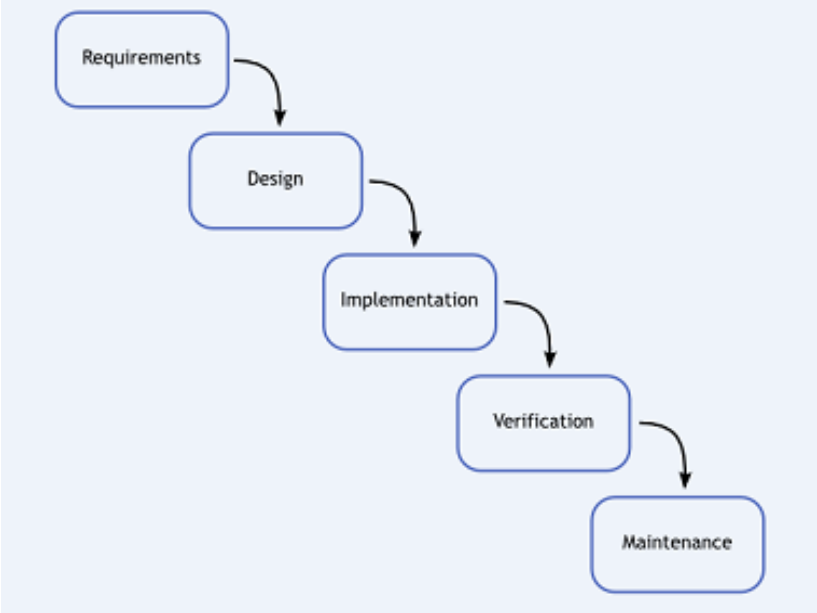

Fig 1: WATERFALL Model

B. Evolutionary development model:

Basic theme of Evolutionary model is to refine the software until an adequate quality system has been designed. Each iteration delivers a functionally operational product and thus customers can get to see the working version of the product at each stage. All process activities are carried in a concurrent manner with parallel feedback. In terms of quality of product this model is more effective than the waterfall approach to software development that it immediately be able to meet the needs of customers.

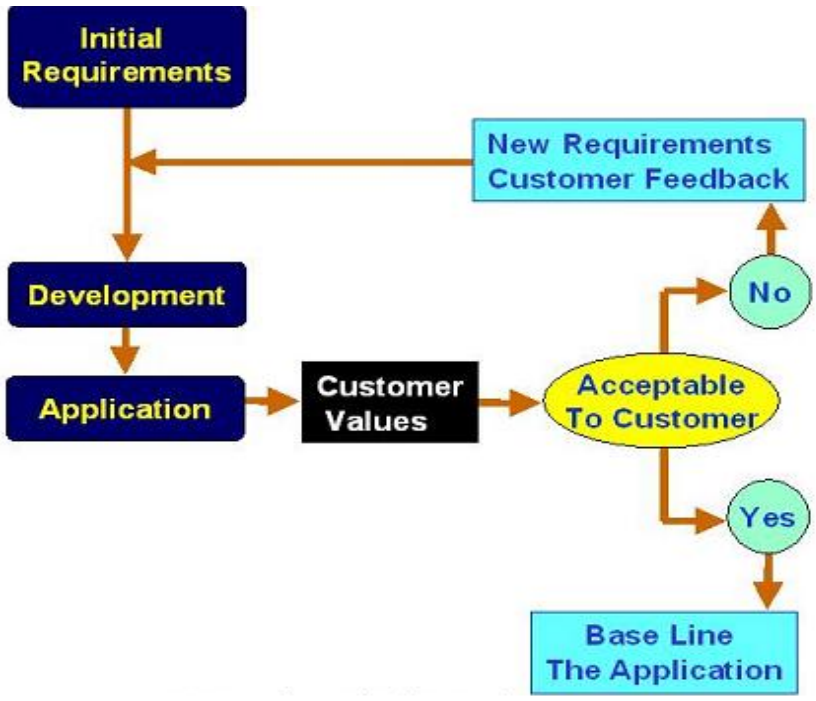

Fig 2: Evolutionary Model

C. Reuse oriented model:

This approach was based on systematic reuse where systems are integrated from existing components systems. In many software projects, the overall cost of software development can be reduced by using a sequence of defined rules which can be

automatically derived from preceding ones. Reusable components enhance the possibility to improve both productivity and quality of software system. Reuse is now the standard approach for building many types of business system.

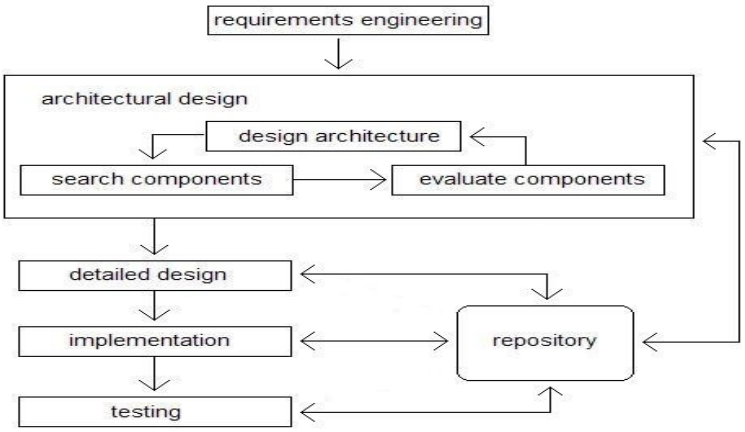

Fig 3: Reuse Oriented Model

D. Spiral model:

spiral model as the name indicates represents each software process activity in a loop wise manner. Starting from innermost loop, each loop defines software feasibility then requirement, design, quality control, risk management and so on. For critical projects, each phase contain more work task that are defined to achieve higher level of formality. In all cases software quality assurance is applied. This model is very good to use for large size projects where small prototypes are designed and developed and can be enhanced to make larger software.

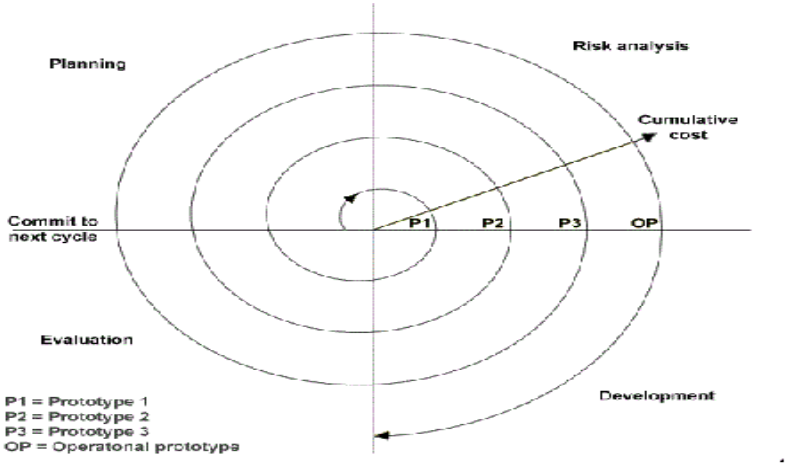

Fig 4: Spiral Model

III. SOFTWARE QUALITY MANAGEMENT

The purpose of quality management is to manage the quality of software & its development processes. An effective system reduces IT risk by preventing problems and detecting defects where they occur. The deliverable from the software process is input to quality management process and is checked to ensure that they are consistent with defined goals and standards. Main benefits of adopting quality management are: save time, reduce compliance cost and build an improved quality culture.

Quality management is a set of 3 different activities:

A. Quality assurance

B. Quality control

C. Quality planning

Fig 5: Quality Management System

Quality of development process directly affects the quality of delivered products. The quality plan is the set of quality related activities that a project plans to do to achieve the quality goal. Quality goals are specified in terms of acceptance criteria. The delivered software should work for all the situations and test cases in the acceptance criteria. The QA process involves selecting standards that can be applied to software process and product.QA process start with planning & conducting inspection and reviews. It is an ongoing process within the software development life cycle (SDLC) that routinely checks the developed software to ensure it meets desired quality measures.

SQA processes tests for quality in each phase of development until the software is complete. With SQA, the software development process moves into the next phase only once the current/previous phase complies with the required quality standards. SQA generally works on one or more industry standards that help in building software quality guidelines and implementation strategies.

Fig 6: Quality Assurance Process

Quality control basically is a set of activities for ensuring quality in software products. The activities focus on identifying defects in the actual products produced. Quality control involves review of software and creating automated software assessment report to carryout technical analysis of product document. It includes activities like control Change management which is supported by verification & how planned changes can influence the quality of created solution and eventually change of test plan. One can be changes in test plan, test cases and scenarios. Verify and evaluate to improve the use of methods, procedures and adopted software tools.

IV. Process and Product Quality Assurance

In order to define the quality of software product, quality assurance is divided into 2 main areas: process assurance, product assurance. Product assurance aims that final product meet its specification, which is usually done by testing the product. While process assurance looks the process (technology) used to design the product. Process quality is quite important in software

development because without using software for a long time, it is difficult to measure its maintainability. Changing a process does not always leads to improved product quality.

Using new innovative application and design process defects could be avoided that will leads to better product. That why it is necessary to monitor development process to keep consistent quality. For software products there are four main factors which can affect product quality: development technology, cost, time, and schedule and people quality.

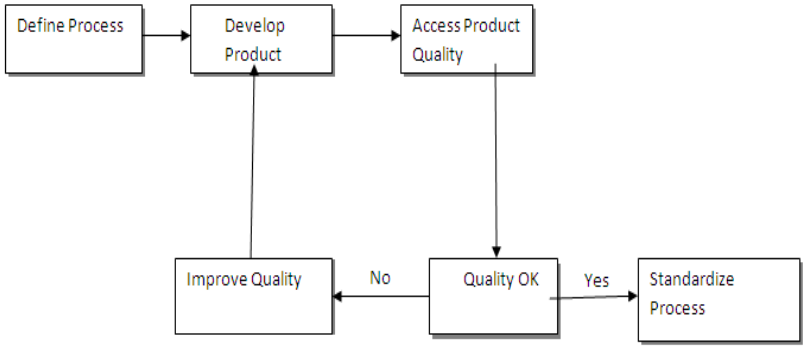

Fig 7: Process Based Quality

The Process and Product Quality Assurance process area collectively supports the delivery of high-quality products and services by providing the project staff and managers at all levels with appropriate visibility into, and feedback on, processes and associated work products throughout the life of the project.

V. QUALITY STANDARDS

Quality standards are concise set of priorities designed to drive measurable quality improvement. Most of the software products have safety standards, and many have usability standards. It is very difficult to enforce standards on actual Program behavior. Standardizing the process can help make sure that no steps are skipped, also Standardizing to an inappropriate process can reduce productivity, and thus leave less time for quality. These standard deals with the many aspects to determine the quality of a software application:

A. Quality model

B. External metrics

C. Internal metrics

D. Quality in use metric

Standards include software safety, usability, reliability, portability, reusability & most important efficiency.

Reliability	Product reliability is calculated in terms of working of system under different working environment conditions. It's a measure of how a product behaves in varying circumstances.
Maintainability	It is the ability of software to adapt to changes, to improve over time, to fix any bugs through preventive maintenance. Maintenance should be cost effective and easy.
Usability	Usability can be measured in terms of ease of use. Software should be user friendly & easy to learn. The system should Provide consistent user interface standards or conventions.
Efficiency	A system should utilize processor capacity, disk space and memory efficiently. If system is using all the available resources then user will get degraded performance failing the system for efficiency. Means we can say it is the ability of the software to do the required processing with least count of hardware.

Portability	Portability defines the ease with which a system can be adapted to run on computers other than the one for which it was designed. It mainly depends on Degree of hardware independence & Implementation language.
Reusability	Reusability is a good cost efficient and time saving development way. It defines the capability for components and its subsystems to be suitable for use in other applications & scenarios.
Security	Security is the ability of the system to remain protected from unauthorized user access. These include both read access and write access. It also prevents system from virus infection.
Testability	Testability is the ability of software to test it completely before putting into production. A System should be easy to test and find bugs. It can be made easy by dividing software in different modules for testing.
Availability	Availability defines the quality of software to keep functioning in spite of problems. It can be measured as a percentage of the total system downtime over a predefined period. It is mainly affected by system errors, infrastructure problems, malicious attacks & system load.

Table 1: Quality Assurance Standards

VI. CONCLUSION

It is crucial to think about quality when we start the project. More quality is not always better, but it is usually is. Quality standards can help to ensure consistent quality for both process and product. Before we deliver a software product it is necessary to review quality assurance plan template and non-compliance report document. Process and product quality assurance report should be prepared for every process area 2-3 times a year as per standards. Product audit should include software requirement specification, quality trends, release notes and evaluation report. This procedure provides a standard for reviewing the efficiency of the execution of the quality assurance processes in an SDLC Model. It should be apply to all SDLC projects as stipulated in the Information System.

References

1. Roger Pressman titled Software Engineering - a practitioner's approach
2. Osterweil, L. (1996). Strategic directions in software quality. ACM Computing Surveys, 28(4), 738–750.
3. Ashrafi, N. (2003). The impact of software process improvement on quality: In theory and practice. Information & Management, 40(7), 677–690.
4. Bessin, G. (2004). The business value of software quality. Retrieved October 10, 2010, from
5. http://www.ibm.com/developerworks/rational/library/dec04/bessin/
6. http://softwaretestingfundamentals.com/software-quality-assurance/
7. http://www.cs.ccsu.edu/~stan/classes/CS530/Slides11/Ch2.pdf
8. http://www.etq.com/quality-management-software/
9. http://airccse.org/journal/acij/papers/0312acij03.pdf
10. Catelani, M. , Ciani, L. , Scarano, V.L. , Bacioccola, A. “A Novel Approach To Automated Testing To Increase Software Reliability” Instrumentation and Measurement Technology Conference Proceedings, 2008. IMTC 2008. IEEE on 12-15 May 2008.

AUTHOR(S) PROFILE

Anu Gupta, received the B.Tech (honors) degree in computer science and engineering from UIET KUK in 2011.Her M.Tech degree is of `A` grade in Information Technology. She has more than 3 years of teaching experience. Right now she is working with Dronacharaya College of Engineering Gurgaon.

