Survey on Software Quality Assurance

Dhaval Jha, Vaishnavi Bhagat, Anjali Bidua
Computer Science and Engineering, Institute of Technology, Nirma University, Ahmedabad, India.
vaishnavi7096@gmail.com

Abstract – Software Quality Assurance (SQA) contains of the complete software development process - observing and improving the process. It completely makes sure that any of the decided standards and measures is followed and make sure that errors are detected and solved. SQA is something aiming for the prevention and if accomplished will result in the making of quality software. SQA has been accepted as an successful approach to quality problems. This paper talks about what is SQA, what are its importance, quality models used and problem that occurs during software production.

I. INTRODUCTION

Even though in today’s date research is going on quality software as because though millions and billions are spent and used demanding to get a quality software but a program or a software is never complete without errors and software bugs which are common. So as we know software is an essential and valuable part of a system, if a software development process quality is insufficient, then definitely developed software quality will be insufficient too. Software Quality Assurance (SQA) contains of the complete software development process - observing and improving the process. It carries out this by inspection of the quality management system below which the software system is created. This inspection is sponsor by one or more standards, mostly ISO 9000[3].

SQA is also responsible for collecting and presenting software metrics. Software quality assurance is interrelated to the practice of quality assurance product development.

II. IMPORTANCE OF SQA

Use of software is increasing in our day to day life. For e.g. electronic devices like computers, laptops, watches, mobile to like banking, medical, ecommerce, etc. and many more. Computers are nowadays universal and all computers run some software it is its basic requirement and thus software is also universal. Due to the widely accepted and use of software systems in various fields and areas, software errors, defects or bugs are seem to be costly and sometimes critical. Bugs have affected many fields and areas like banking systems, stock exchanges, medical institutions, educational institutions and even the Social Security Administration. Mainly bugs, found during software development, can be avoided, by adopting a sound software development process, and having strict software quality control using Software Quality Assurance. The process of SQA is equivalent to Software Testing[1].

ISO/IEC 9126 2. Application or company specific quality models e.g. a. FURPS b. GQM Approach

McCALL’s FACTOR MODEL

Quality Factor signifies a behavioral feature of the system. a. Operation b. Revision c. Transition
Quality Criterion is a characteristic of a quality factor that is interrelated to software manufacture and design.

Quality Metric is a measure that confines some feature of a quality criterion.

THE SIX QUALITY CHARACTERISTICS OF A SOFTWARE (ISO/IEC 9126)

Each attribute is superior to a set of sub-attribute. Each sub-attribute is measured by a set of metrics. Several metrics are common to several sub-attributes.

a. Portability-> Adaptability, Replaceability
   Installability, Conformance.

b. Reliability-> Maturity, Fault tolerance, Recoverability.

c. Usability->Understandability, Learnability, Operability.


e. Efficiency-> Time behaviour, Resource behaviour.


FURPS

Originally FURPS is a quality model which is company specific. Features:

a. Functionality: functions it performs, their generality and Security

b. Usability: aesthetics, consistency, documentation

c. Reliability: frequency and severity of failure, accuracy of output

d. Performance: response time, resource consumption

e. Supportability: it can be extended, adapted, corrected

GQM : GOAL-QUESTION-METRIC

Goals should be set in mind to design a more successful measurement program. Frameworks are been provided by GQM approach with 3 steps:

1. List out the key goals of the development/maintenance of a project.

2. Derive questions from each goal that must be answered to conclude if the goals are being achieved.

3. Come to a decision what must be planned to answer the questions satisfactorily.

IV. PROBLEM DESCRIPTION

There are many problems that can occur during production such as:

DEFECT COUNTS

Lack of software quality leads to software errors\(^4\). Naturally, cause of defects is negatively related to reliability and functionality.

To some degree, defects obstruct other dimensions of quality

FAULTS

Defect within the system is Fault. Faults lead to Errors. Faults could be found during following :

a. System Designing
   - Divide by zero fault
   - Infinite loop fault

b. System implementing
   - Connections fault

c. System testing
   - Platform independency fault
   - Version fault

ERRORS

Derivation of the necessary operation of the system and sub-system is Error.

Human actions that results in defected software are Software Errors.

Faults lead to Failures. There are some faults too that not necessarily leads to errors, they may be warnings with not much effect to your system.

Errors may be transient that once occurred and then disappeared, intermittent that occasionally found or occurred, permanent that once occurred unable to solve it.

FAILURES

When a system fails to perform its necessary functions at that time System Failures occurs.

There are some errors that not necessarily lead to failure, they may be errors but not much effective to lead to failures.
System hanging, giving garbage values instead of correct results, null pointers errors, etc. are all examples that lead to system failures.

SOFTWARE COMPLEXITY

Any of the software, that’s manufacture to provide some useful purpose, is extremely complex which cannot be fully understood by a single person.

CHANGE IN REQUIREMENTS

Software functionality changes leads to change in software functionality. Rapidly changing the requirements and adding additional functionality to the system causes the existing modules in unexpected ways. High level of interdependencies between the modules makes the system error prone.[1].

TIME LIMIT AND DEADLINES

Recently, software development industry is highly competitive, and schedule slippages are not satisfactory. Some projects have impractical schedules, which make the development tactic far from perfect and the manufactured software lacks quality.

V. CONCLUSION

Though we spent millions and billions of money on Software Quality, Software Development with good quality is complex, and is error prone. Many problems that occur during software development can be solved, by approving a good software development process. From this discussion, it’s clear that good processes are vital. Thus software industry is still observing and learning, about good processes for software development. Still SQA makes sure that errors and faults are detected and tackled during software manufacture process and provide good quality software.

VI. REFERENCE

[1] “Software Quality Assurance”, Ramakrishnan Venkitaraman, Graduate Student, Department of Computer Science, The University of Texas at Dallas


[3] “Methods of software quality assurance”, Daniel Osielczak (d.osielczak@gmail.com), Sebastian Mianowski (aeons@wp.pl)