The International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 3, No. 3, Special Issue:
The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13],
San Francisco State University, CA, U.S.A., March 2013
Doi: 10.7321/jscse.v3.n3.26 e-ISSN: 2251-7545

SOFTWARE QUALITY ASSURANCE FOR SAFETY CRITICAL SYSTEMS

A Framework for SCS Quality Assurance

Komal Bashir[1],Faria Kanwal[2],Afnan Bashir[3],Ayesha Haider Ali[4]
*1, LCWU, Email : ko_junaid@yahoo.com
2, LCWU, Email : faria.kanwal@gmail.com
3, COMSATS Institute of Information Technology, Lahore, Email : afnan.bashir@hotmail.com
4, LCWU, Email : ayesha.iqbal@gmail.com

Abstract—**Safety Critical Systems play significant role in almost every domain of technology. All systems ranging from nuclear power stations to cars, in one way or other impact environment or life. Failure or malfunction in such systems may severely harm people's lives and environment. Highest level of accuracy and perfection is required from such system. It is important to ensure quality features and plausible outcomes as a result of the intended role for which that safety critical system was designed for. This paper proposes a framework that is composed of most optimum Software Quality Assurance practices in development of such systems. Paper presents nine phases to solidify the quality assurance perspectives of the said systems.**

Keywords-Safety Critical Softwares, Software Quality Assurance, Framework,Validation, Verification

I. INTRODUCTION

Software plays a vital role in every walk of life these days. From automated washing machines and ovens to office work [1]. It is widely used in almost every task performed by humans and machines. A class of software known as safety critical software (SCS) is responsible for automation of many real time systems. It is a computer system whose failure or malfunction may severely harm people's lives, environment or equipment for example, nuclear power station control, railway systems, aviation control systems etc. [3]. SCS is one that has the potential to cause accidents [9]. It retains the decision making feature, it intervenes when any unsafe condition occurs. Such systems are expected to satisfy a variety of specific qualities including reliability, availability, security and safety [2].

A safety critical system is a system where human safety is dependent upon the correct operation of the system. So the system should be safe, risk free and failure safe [4] [5]. Risk assessment and decision making features should be considered primarily. Beside these features, it is very important to address non-functional requirements of the system. These are quality attributes necessary for the success of software, e.g., reliability, usability, consistency, efficiency etc.

Quality can be defined as "level of user satisfaction", "conformance to the requirements" etc. [7]. According to the IEEE, Software quality is the degree to which a system, component, or process meets specified requirements [8]. Quality Assurance (QA) is an activity concerned with every step of software development. It assures and checks the quality requirements for a software project.

According to the IEEE, Software Quality Assurance (SQA) is planned systematic pattern of all actions necessary to provide adequate confidence that an item or product conforms to established technical requirements [8].

Software directly impacts products success as well as the safety. Quality activates should be conducted throughout the project life cycle to produce quality product [6].

This paper gives a framework for Quality Assurance (QA) of Safety Critical Systems. It covers important QA steps for the stated systems.

II. RELATED WORK

In [6], author gives framework based approach based on the standards of reliability and quality is proposed for Software Reliability and Quality Assurance (SRQA).

[9] States that existing software quality models including McCall's, Boehm's and ISO 9126 are inadequate in addressing the software safety issues of real time safety-critical embedded systems. Also any standard model that widely covers safety factors and metrics of software safety does not exist. [9] Proposes a new model for software safety based on the McCall's software quality model, it explicitly gives the criteria corresponding to software safety for safety critical applications.

[10] Author uses the concept of a quality model to define safety as a quality factor. Paper states that safety is a kind of quality. It deals with three types of requirements for safety critical systems. This work is limited to the requirement phase of SCS.

Paper [11] gives extension to [6]. It suggests comparison between actual and expected level of quality of SCS. According to the author, this activity gives a systematic view for analyzing software's quality. [11] Also suggests use of checklists to identify weak areas.

III. NEED OF FRAMEWORK

In [3], author lists problems occurred in safety critical systems. He further states that the cause of the problems might be due to incorrect supposition about the system units. These units were never checked. If systems units are carefully checked, it can reduce the chance of problem occurrence.

[11] Gives method for comparison between actual and expected level of quality of SCS. It suggests use of some database activities and checklists.

[6] Covers reliability as major concept. It gives a frame work for the SCS quality.

Checking can be performed by implementing proper quality control and testing activities. The QA process should start right from the first phase of the software development till the last phase of development. QA tasks should be performed concurrently with the development life cycle. Each artifact should be tested and quality should be assured against it. Framework proposed by [6] could be extended for the whole project life cycle.

IV. Proposed Framework

This paper gives SQA framework for Safety Critical Software. The proposed framework consists of nine phases; each phase indicates certain QA activities. Figure I show the proposed framework. There is the brief description of each phase of the framework.

A. *Quality Purpose & Management*

1) *Purpose and Scope*

Covers the scope and purpose of the system, it also holds list of software artifacts produced and their uses. It also states the intended purpose of Quality Assurance activity conducted for the particular SCS.

2) *Management*

This segment lists the managerial aspects for the system including the project organization, team hierarchy, tasks and roles of the team members. Organizational structure, step for implementing the project and sequential list of project tasks is made.

B. *Procedures & Standards*

1) *List of Documents*

Describe all the documents to be produced and there review procedure. Make list of all the documents to be built from day one to the closing of the project.

2) *List of Standards*

This section provides the details of various standards applicable to the project. These standards may include documentation standards, coding standards, and organizational standards.

3) *Procedure for Software Quality Programs*

List of Procedures for developing and implementing Software Quality Programs is made.

4) *Software Quality Activity Matrix*

The Software Quality Activity Matrix gives an overview of Software Quality activities that are to be performed during each development phase. This matrix is used as a planning tool to govern the activities and assets that are required to develop and implement a successful Software Quality process.

C. *Process Quality Assurance – Software Verification*

1) *Process Definition*

The system functionality is broken down into subparts to process level. Process is an ongoing activity to produce some functionality. A process can be identified by top down or bottom up approach. [12]

2) *Assure Quality in Process*

Reviews, audits, inspections and testing are carried out to assure process quality. Process metrics can also be used for this purpose.

3) *Use Process Models*

Process models are used to align processes with the standards. Various standards are used including ISO, CMM, CMMI, TICKIT etc.

Two comprehensive suites of standards that can be used context are: [13]

a) CEI/IEC 61508: 1998-2000, Functional safety of electrical/electronic/programmable electronic safety related systems. Parts 1-7 (IEC 1998-2000)

b) ISO/IEC TR 15504, 1-9:1998, Information technology –Software process assessment – Parts 1-9 (ISO/IEC 1998)

4) *Process Refinement*

Process should be continuously monitored to accommodate the changes. These changes may be due to change in technology; change in market trends, there may be change in government regulations for safety, usability, reliability, etc., product up gradation etc.

D. *Checklists for Reviews & Audits*

What reviews will be conducted and how technical and managerial reviews will be performed.

1) *Requirement specification review*

Review to audit SRS (Software Requirement Specification) document, it checks the adequacy of the requirements stated in this document and checks that organizational and other standards are properly followed.

2) Design Review

Review is conducted to validate the design document. The purpose is to validate the design document against requirements.

3) Software Verification Review

To check the testing plan and to ensure that the activities stated in the test plan fully covers all the aspects of the project testing.

4) Functionality Audit

Audit is conducted to confirm that requirements stated in the SRS document are fulfilled. These are the requirements given by the customer. For safety critical systems, this phase is very important as is any issues remain in the requirements, it will cause swear effects.

5) Management Reviews

Reviews conducted by the management to check that all the stated QA activities are properly executed.

E. Documentation Checklist

Check list are prepared for the following documents:

- Software Requirement Specification (SRS)
- Functional Specification Document (FS)
- Data Management Plan
- Configuration Management Plan
- Risk Assessment Plan
- Test Plan
- User Manuals
- Backup and Recovery Plan
- Maintenance Plan

F. Configuration Management

1) Identification of Configuration Items

It is the process of identification and control of configuration items. Status accounting and configuration audit. Configuration items are the factors for which change will be managed; it can be a file, a document or any artifact etc.

2) Control

Items that can undergo some change are identified, they may be some documents or code or modules etc. Change need to be controlled, act of observing and controlling effect of certain change on other items is change management.

3) Status Accounting and Configuration Audit

Change status should be maintained and this information should be available to the stakeholders. For each configuration item, separate logical account is maintained and all transactions are recorded. Information of code where changes are made is also recorded. [7]

Change must be communicated to the stakeholders. Furthermore, it should be audited to verify that all the processes are followed properly and change is effectively managed.

G. Software Product Quality – Software Validation (Testing)

1) MC Call Model / Boehm Model

Models are used for quality factors assessment of product.

2) Ensure Core Functionality

Functionality of the system can be ensured by conducting following testing techniques:

- Boundary Value Analysis
- Intrusive Testing
- Random Testing
- Static Testing
- Thread Testing

3) Risk based Testing

- White Box Testing
- Black Box Testing

4) Ensure Ancillary Requirements

Following testing techniques can be used to check nonfunctional requirements of the system:

- Installation Testing
- Compatibility and Interoperability Testing
- User Interface Testing
- Fault Recovery Testing
- Performance Testing
- Reliability Testing
- Security Testing
- Stress & Load Testing
- Usability Testing

These testing techniques are conducted by development team and auditing teams of the development organization. It is also required to carry out independent verification and validation by third party. This third party is not part of the development team or the organization. These are the separate parties who work independently and provide their service for testing and auditing.

H. Problem Reporting & Corrective Actions

FMEA (Failure Mode and Effects Analysis) is the first step to check system reliability. In case of SCS, reliability is one of the first concerns. FMEA involves reviewing system components, subsystems and relationships between different components to identify failure conditions and effects of these failures on different component and the overall system. These results are recorded in FMEA sheets. These sheets are further used to mitigate the intended risks and hence develop a risk mitigation plan. Hence probability of failure can be reduced in Safety Critical Systems.

This phase of the framework works as defect removal and prevention phase. It describes the system, and safeguards that the software problems are documented and resolved. Problems and corrective actions taken against them are properly documented for future reference. Before implementing the corrective action, it is carefully monitored that it doesn't affect other system operations.

I. Backup & Recovery

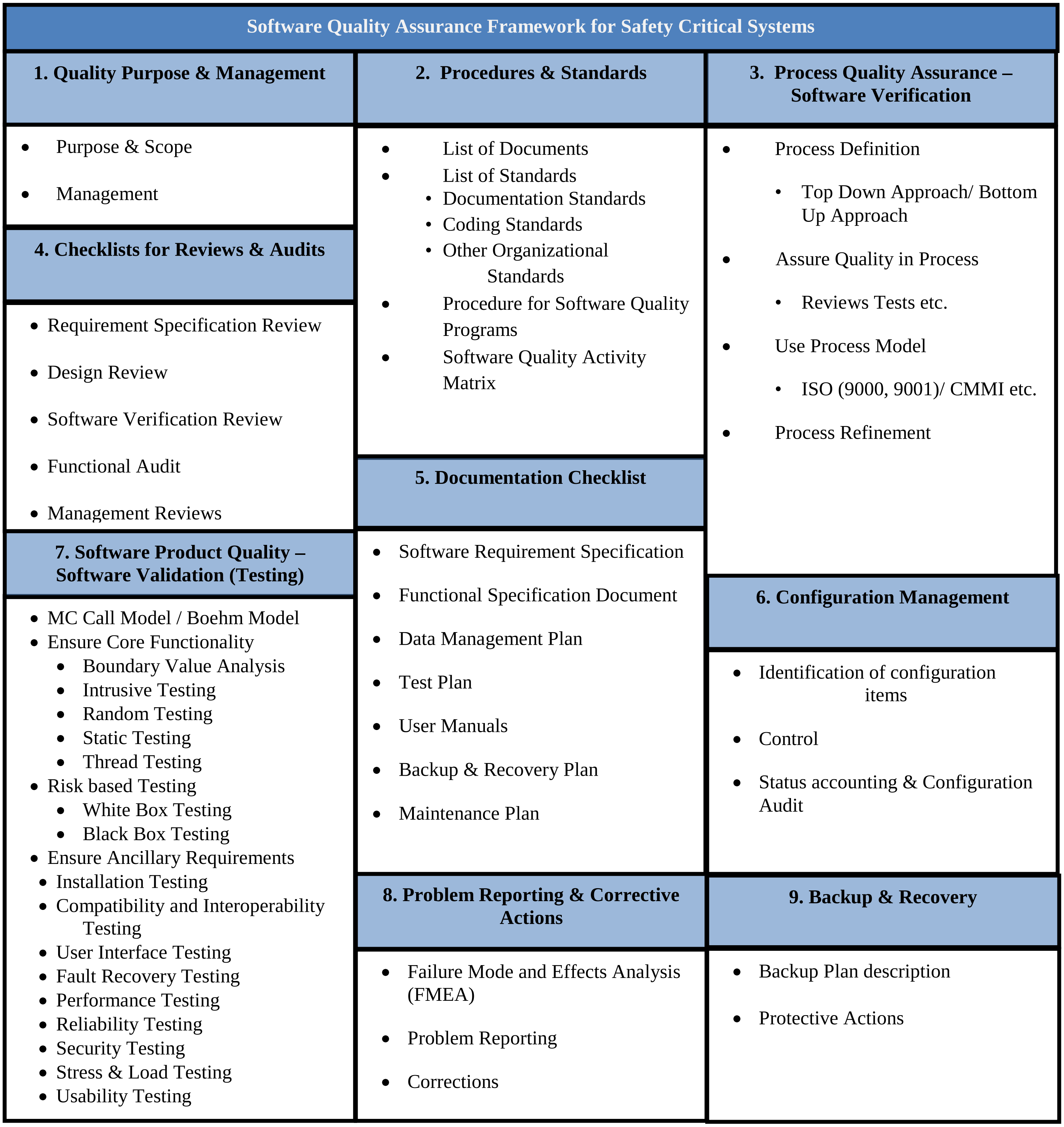
Describe the backup plan how the media will be protected from unauthorized access and hazards. System backup should be stored at some remote safe location. Another plan, disaster recovery plan is developed. Purpose of disaster plan is to enable the system to recover and function properly in case of problematic conditions.

V. CONCLUSION

This paper recommends a framework for assuring the quality in development process of the Safety Critical System. These systems are mission critical and require highest level of accuracy and perfection. Software Quality Assurance is a discipline that checks and ensures such features. This paper proposes a framework that is composed of most optimum Software Quality Assurance practices in development of such systems. Paper presents nine phases to solidify the quality assurance perspectives of the said systems. It starts with aims and objectives of the system, covers functional and nonfunctional requirements, checklists for documentation then it suggests techniques for process verification and product validation. It also suggests backup and recovery planning to deal disaster situations.

VI. FUTURE WORK

As concluded by [6], there is a great disparity in development processes used for the development of the Safety Critical Systems; hence it is nearly impossible to develop a generic solution. This problem can be encountered with the integration of data-ware housing and data mining techniques. That would later serve the purpose of refining the presented approach according to the specific system under development.



Software Quality Assurance Framework for Safety Critical Systems

1. Quality Purpose & Management

- Purpose & Scope
- Management

2. Procedures & Standards

- List of Documents
- List of Standards
  - Documentation Standards
  - Coding Standards
  - Other Organizational Standards
- Procedure for Software Quality Programs
- Software Quality Activity Matrix

3. Process Quality Assurance – Software Verification

- Process Definition
  - Top Down Approach/ Bottom Up Approach
- Assure Quality in Process
  - Reviews Tests etc.
- Use Process Model
  - ISO (9000, 9001)/ CMMI etc.
- Process Refinement

4. Checklists for Reviews & Audits

- Requirement Specification Review
- Design Review
- Software Verification Review
- Functional Audit
- Management Reviews

5. Documentation Checklist

- Software Requirement Specification
- Functional Specification Document
- Data Management Plan
- Test Plan
- User Manuals
- Backup & Recovery Plan
- Maintenance Plan

6. Configuration Management

- Identification of configuration items
- Control
- Status accounting & Configuration Audit

7. Software Product Quality – Software Validation (Testing)

- MC Call Model / Boehm Model
- Ensure Core Functionality
  - Boundary Value Analysis
  - Intrusive Testing
  - Random Testing
  - Static Testing
  - Thread Testing
- Risk based Testing
  - White Box Testing
  - Black Box Testing
- Ensure Ancillary Requirements
- Installation Testing
- Compatibility and Interoperability Testing
- User Interface Testing
- Fault Recovery Testing
- Performance Testing
- Reliability Testing
- Security Testing
- Stress & Load Testing
- Usability Testing

8. Problem Reporting & Corrective Actions

- Failure Mode and Effects Analysis (FMEA)
- Problem Reporting
- Corrections

9. Backup & Recovery

- Backup Plan description
- Protective Actions

FIGURE I: PROPOSED SQA FRAMEWORK FOR SAFETY CRITICAL SYSTEMS

REFERENCES

[1] Greg Rose, “Safety critical software”, CompactPCI Systems April 2003.

[2] Critical System Labs, Inc. 2005-2009. http://www.criticalsystemslabs.com/pgs/What.html

[3] John C. Knight, “Safety Critical Systems: Challenges and Directions” Proceedings of the 24th International Conference on Software Engineering (ICSE), Orlando, Florida, 2002.

[4] M. Ben Swarup and P. Seetha Ramaiah, “An Approach to Modeling Software Safety in Safety-Critical Systems”, Journal of Computer Science 5 (4):311-322, 2009, ISSN 1549-3636, © 2009 Science Publications .

[5] “An introduction to Safety Critical Systems”, IPL Information Processing Ltd.

[6] Ankur Pandit, “A Framework-based approach for Reliability & Quality Assurance of Safety-Critical Software”, IJCSE) International Journal on Computer Science and Engineering Vol. 02, No. 09, 2010, 2874-2879

[7] Nina S. Godbole,”Software Quality Assurance,”, ISBN: 1842651765, EAN: 9781842651766, Chap 1

[8] “Introduction to Software Quality Assurance”,Based on D. Galin Ch1 -4, 5 and R. Patton Ch 1

www.site.uottawa.ca/~ssome/Cours/SEG3203/**introduction**SQA.pdf

[9] M. Ben Swarup and P. Seetha Ramaiah, “A Software Safety Model for Safety Critical Applications” International Journal of Software Engineering and Its Applications, Vol. 3, No.4, October 2009.

[10] Donald Firesmith,” Engineering Safety Requirements, Safety Constraints, and Safety-Critical Requirements” JOURNAL OF OBJECT TECHNOLOGY, Online at http://www.jot.fm. Published by ETH Zurich, Chair of Software Engineering ©JOT, 2004, Vol. 3, No. 3, March-April 2004.

[11] Ankur Pandit, Alka Gulati, Vineet Richhariya,” Enhance Framework for Reliability & Quality Assurance of Safety-Critical Software”, IJCST Vol. 2, Issue 3, September 2011.

[12] Mura li Chemuturi,” SOFTWARE Quality Assurance Best Practices, Tools and Techniques for Software Developers”, ISBN 978-1-60427-032-7. Chap 8, p 178.

[13] O Benediktsson, R B Hunter, A D McGettrick, “Processes for Software in Safety Critical Systems”, Software Process: Improvement and Practice, volume 6, issue 1, pp. 47-62, 2001.