Enhance Framework for Reliability & Quality Assurance of Safety-Critical Software

[1]Ankur Pandit, [2]Alka Gulati, [3]Vineet Richhariya
Dept. of CSE, Lakshmi Narain College of Technology, Bhopal, MP, India

Abstract
Software plays a critical role in almost every facet of our daily life – from cooking in our kitchens, to driving our cars, to working in our offices. Some of these software are safety critical. Failure of software like software used for missile, satellite, cancer radiation therapy machine etc. could cause catastrophic consequences for human life. Safety-Critical Software (SCS) involves high risk in design, development and installation. Also it is responsible for controlling, monitoring number of hardware systems inside a system. Software safety activities occur within the context of system safety, system development, and software development and assurance. The overall complexity and the average size of the software product keep growing; which make it important to assure the reliability and quality of the software. This requires an enhanced reliability and quality framework for SCS. In this paper, we design and develop a framework with different components serve together to assure the reliability and quality of Safety-Critical Software on the basis of International procedures and standards.

Keywords
Safety-Critical Software, quality and reliability assurance, high risk, enhanced framework-based approach, database.

I. Introduction

Within the complex system development throughout the industries, Software has taken on a new, enhanced role and now directly impacts not only product success, but also the safety. Software Reliability & Quality Assurance (SRQA) for Safety-Critical Software (SCS) having the key role in mission success. The term Safety-Critical Software means software systems whose failure may lead to loss of life or severe injury like software used for missile, satellite, cancer radiation therapy machine etc. There is a growing concern in all major industrial nations regarding the legal and ethical obligations of companies and their officers to ensure that software do not violate safety regulations. Every country now a day's emphasize on faster approach for developing mission. SCS involves high risk in design, development and installation. Also it is responsible for controlling, monitoring number of hardware systems inside a system. Thereby making it more important than ever to ensure the reliability and quality of software products. SRQA covers all phases of the software development process, with specific activities to assure both the processes used and the product development.

In this paper an enhanced framework-based approach based on standards of Reliability & Quality Assurance for SCS is proposed. The framework is shown in the Fig. 5. This framework is an enhanced version of already proposed framework [1] with database implementation. This approach provides Software Reliability and Quality activities that should conduct throughout the project life cycle to come out with the quality product. The checklist section of the framework consist of two new column called ALA and ELA i.e. Actual level of achievement and Expected level of achievement. The flow chart to identify the level of quality achieved shown in the Fig. 2. Also database corresponding to the framework is designed and implement using Sql Server 2005, through which we can create and generate report over database.

Note- This paper is only concern with Software Safety not with System Safety.

II. Data Management Plan for Framework

Data Management Plan (DMP) for SQA allows us to store and manage different type of data associated with Software Quality related to the organizational activities. A database management team is required to build and maintain the database and generate reports time to time. The whole database is created corresponding to the framework.

Following are the basic aims behind Data Management Plan for SQA in the organization:

- Develop and maintain a Data Management system for managing both general and proprietary documents in electronic or hardcopy format.
- Identify, collect, manage, and archive internal and external data
- Establish and maintain user entries corresponding to the framework activities.
- Database management system should provide full support to the SQA organization in form of generating reports.

Although number of tables has been generated for framework, the most important are Checklist tables. The structure for checklist is shown in Fig. 1.The relationship between tables is shown below in fig. 2.

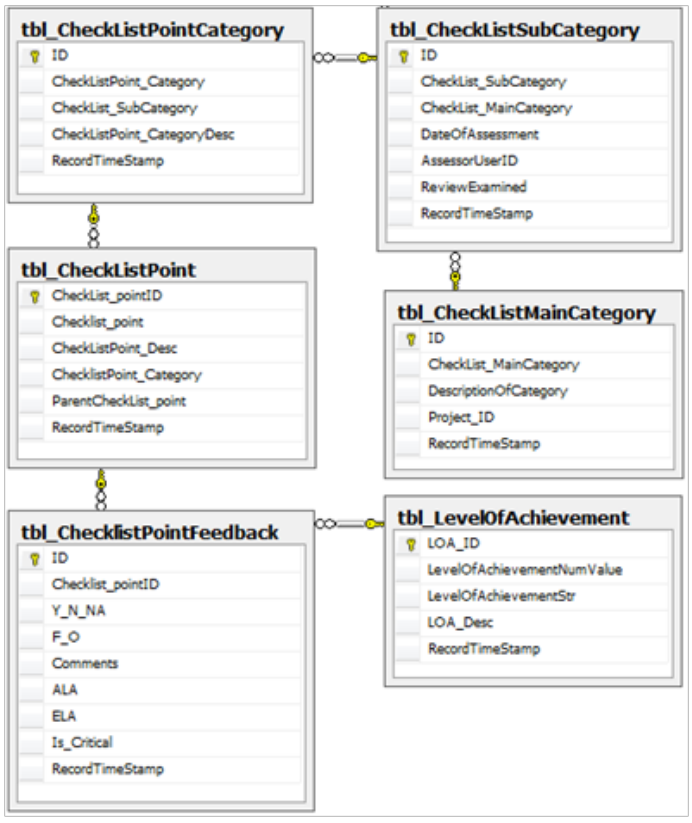


Fig. 1 : Relationship between database tables

		Y , N, NA	F,O	Comments	ALA	ELA
Checklist Point Category						
1.	Checklist point					

Y=Yes, N=No, NA=Not Applicable, F=Finding, O=Observation, ELA- Expected level of Quality Achievement, ALA- Actual Level of Quality Achievement

Fig. 2: Checklist structure

Table 1: Schema of tbl_ChecklistMainCategory

Column	Datatype	Required	Description
ID	INT	YES	Unique auto Id
CheckList_MainCategory	NVARCHAR(50)	YES	Title of Category
Description_of_Category	VARCHAR(128)	NO	Optional. Detail of Category
Project_Id	INT	YES	Foreign key relationship with Table tbl_ProjectDetails
RecordTimeStamp	DateTime	YES	New entry Date time

Table 2 : Schema of tbl_Checklist Sub Category

Column	Datatype	Required	Description
ID	INT	YES	Unique auto Id
Checklist_SubCategory	NVARCHAR(50)	YES	Title of Checklist
CheckList_MainCategory	INT	YES	Foreign key relationship with Table tbl_CheckListMain-Category
DateOfAssessement	DATETIME	YES	Date of checklist assessment
AssessorUserId	INT	YES	CheckList assessed by User id Foreign key relationship with Table tbl_UserManagement
Review Examined	VARCHAR(100)	NO	Optional. Comments
RecordTimeStamp	DateTime	YES	New entry Date time

Table 3: Schema of tbl_Checklist Points Category

Column	Datatype	Required	Description
ID	INT	YES	Unique auto Id
ChecklistPoint_Category	NVARCHAR(50)	YES	Title of Checklist Point Category
Checklist_SubCategory	INT	YES	Foreign key relationship with Table tbl_CheckListSubC-ategory
ChecklistPoint_CategoryDesc	NVARCHAR(MAX)	NO	Optional.
RecordTimeStamp	DateTime	YES	New entry Date time

Table 4: Schema of tbl_Level of Achievement

Column	Datatype	Required	Description
LOA_ID	INT	YES	Unique auto Id
LevelOfAchievementNum	INT	YES	Level of Achievement in Number
LevelOfAchievementStr	NVARCHAR(50)	YES	Title of Level of Achievement
LOA_Desc	NVARCHAR(MAX)	NO	Optional.
RecordTimeStamp	DateTime	YES	New entry Date time

Table 5: Schema oftbl_ChecklistPoint

Column	Datatype	Required	Description
Checklist_pointID	INT	YES	Unique auto Id
Checklist_point	NVARCHAR(MAX)	YES	Title of Checklist point
Checklist_Description	NVARCHAR(MAX)	NO	Optional. Description of checklist point
ChecklistPoint_Category	INT	YES	Foreign key relationship with Table tbl_CheckListPoint Category
ParentChecklist_point	INT	YES	Sub point of Checklist point.0 if parent rest Checklist_pointID
RecordTimeStamp	DateTime	YES	New entry Date time

Table 6: Schema of tbl_Checklist Point Feedback

Column	Datatype	Required	Description
ID	INT	YES	Unique auto Id
Checklist_pointID	INT	YES	Foreign key relationship with Table V.14. Checklist_pointID
Y_N_NA	INT	YES	Value against Checklist point Yes-1, No- 0,NA- -1
F_O	INT	YES	F- Finding, O-Observation
Comments	VARCHAR(1000)	NO	Optional. Comments against Checklist point
ALA	INT	YES	Foreign key relationship with Table tbl_LevelOfAchievement
ELA	INT	YES	Foreign key relationship with Table tbl_LevelOfAchievement
Is_Critical	INT	YES	Whether point is critical? If yes means its ALA should be Max.
RecordTimeStamp	DateTime	YES	New entry Date time

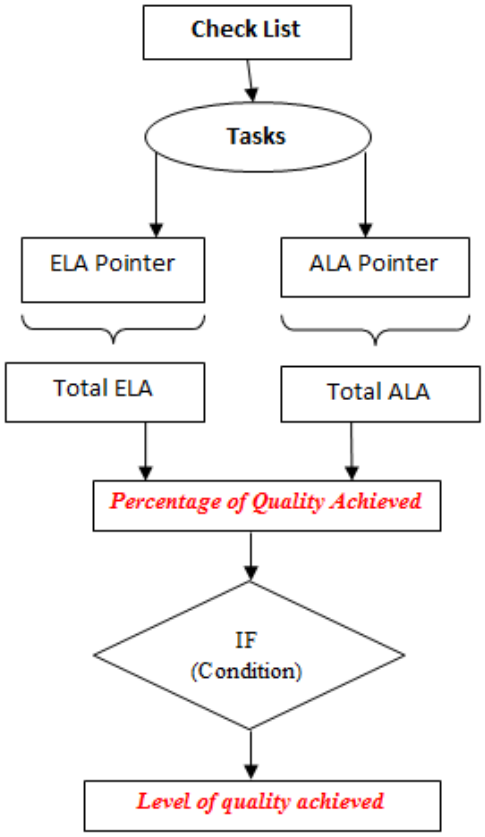


Fig. 3: flow chart for checklist evaluation

Stored procedure for creating graph between ALA & ELA

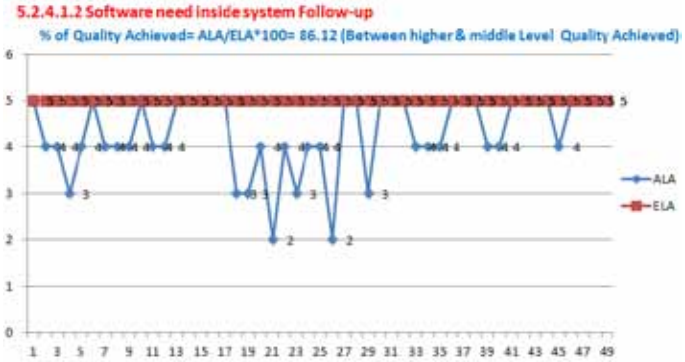
```
CREATE PROC
GetDataForChecklist(@CheckListSubCategoryID AS INT)
AS BEGIN SELECT
CheckListPoint.Checklist_point,
CheckListPointCategory.CheckListPoint_Category,
CheckListSubCategory.CheckList_SubCategory,
ChecklistPointFeedback.Y_N_NA,
ChecklistPointFeedback.F_O,
ChecklistPointFeedback.Comments,
ChecklistPointFeedback.ALA,
ChecklistPointFeedback.ELA,
ChecklistPointFeedback.Is_Critical
```

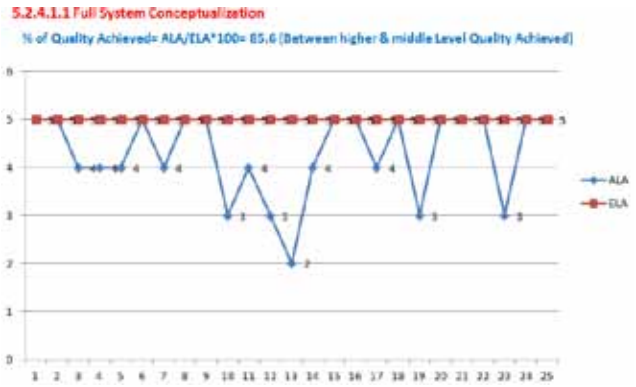
```
FROM
tbl_CheckListPoint AS
CheckListPoint
LEFT JOIN tbl_ChecklistPointFeedback AS
ChecklistPointFeedback ON
CheckListPoint.ChecklistPoint_Categ
ory=ChecklistPointFeedback.ID
LEFT JOIN
tbl_CheckListPointCategory AS
CheckListPointCategory ON
CheckListPointCategory.ID = CheckListPoint.CheckList_
pointID
LEFT JOIN tbl_CheckListSubCategory AS CheckListSubCategory
ON
CheckListSubCategory.ID = CheckListPointCategory.CheckList_
SubCategory


WHERE

CheckListSubCategory.ID = @CheckListSubCategoryID
END
```

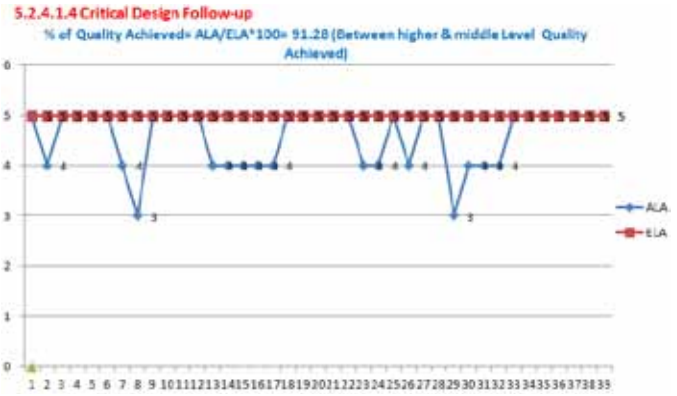
III. RESULTS

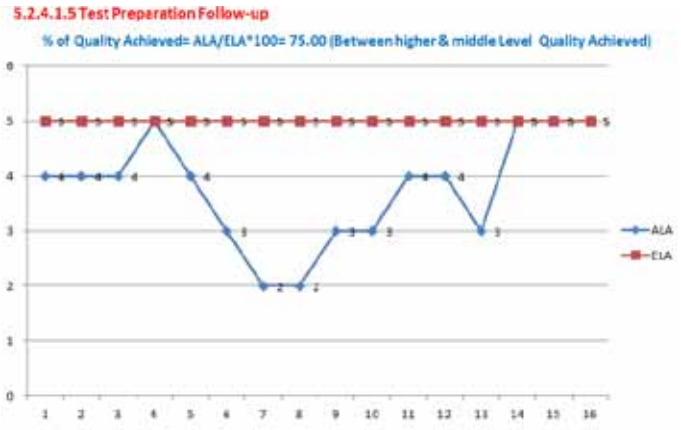
In enhanced framework based approach for reliability and Quality Assurance of Safety Critical Software, a list of results has been successfully generated. The framework includes the different aspects of quality in a single unit. Whenever some Safety critical software has been created, there should be some standards steps should takes place. The results in the form of graphical representation are shown below. Let us discuss about some important outcomes and benefits of this kind of framework.

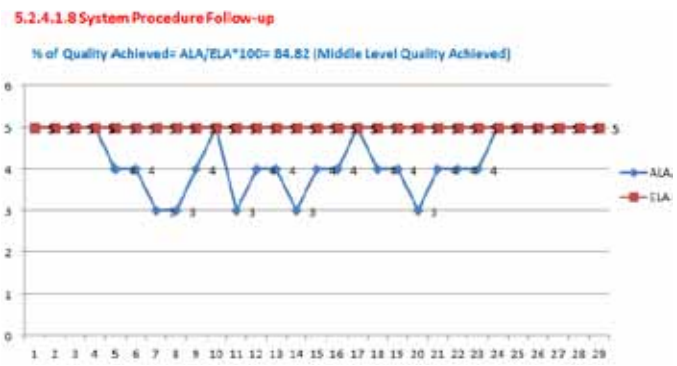
After successful designing and implementation of framework, a list of results has been successfully generated. The framework includes the different aspects of quality in a single unit. Whenever some Safety critical software has been created, there should be some standards steps should takes place. The results in the form of graphical representation are shown below. The results show the comparison between the expected level and actual level of quality achieved (ALA & ELA) and percentage of Quality achieved. We can also plot overall quality achieved for particular checklist group (Software follow-up checklist) plus we can determine the Level of quality achieved through designed algorithm (Refer Fig. 3), taking percentage of quality achieved as input parameter. For example if % of Quality achieved is 85.6 then user can defined its corresponding level of quality on the basis of which one can accept or reject the results. All graphs are generated for particular group of checklist through database. Stored procedure is written to fetch the results.

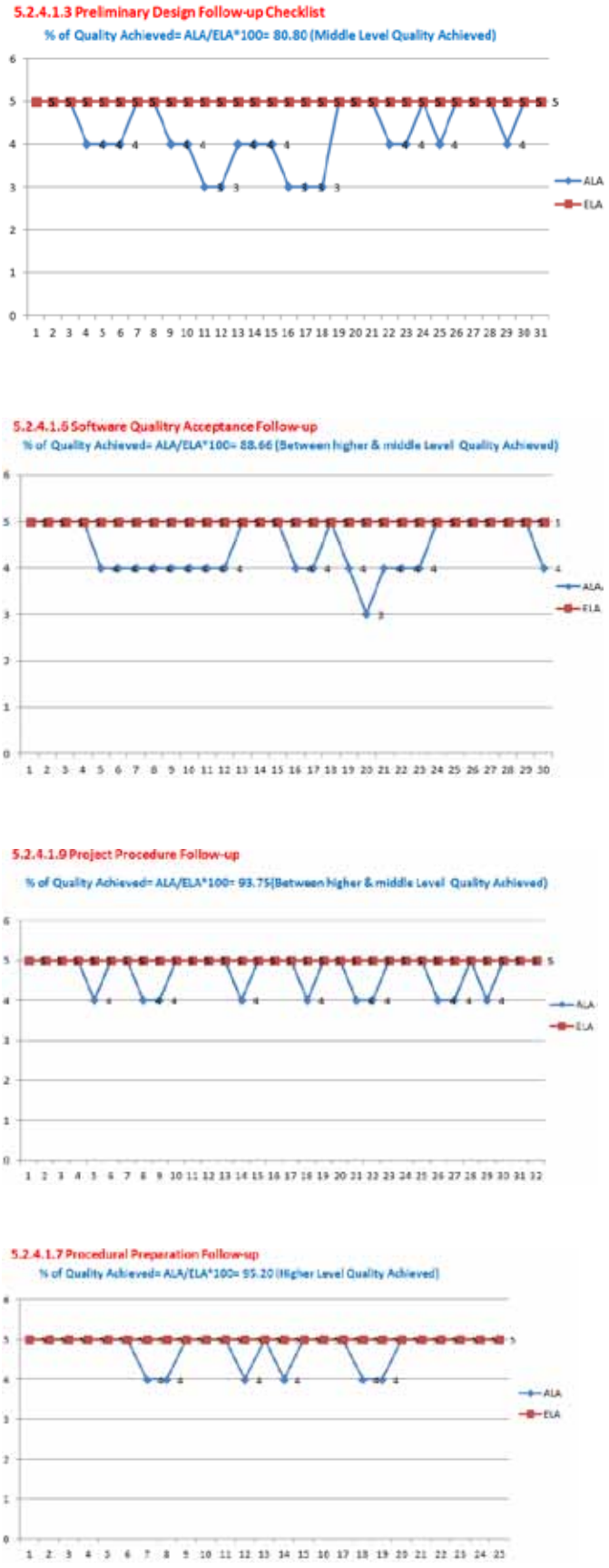












VI. Conclusion and Recommondation

SRQA is very complex and its final goal is to provide the project's success. In addition, safety and reliability of mission critical software is playing key role in success of a project. In this study an enhanced framework for Safety critical software based on International standards and procedure has been developed and implemented via creating database. The framework has to be generic so that it can be used by other organizations seeking to customize their SRQA process. Although every organization has own perception on reliability and quality assurance that still need to identify and incorporate.

The comparison between actual and expected level of quality achieved gives a systematic view to analyze developing software's quality at every stage starting from software planning, designing to delivery. The specific Quality profile is generated for the set of data over particular checklist called System Follow-up Checklist consist of nine different sub checklists. This seems to be reliable at characteristics level. These data has been entered by authorized users with different authentication. Weaker checklist points can be easily identified and more attention can be given to these points to achieve an accepted quality level. The overall profile of System review checklist is show in fig. 4.

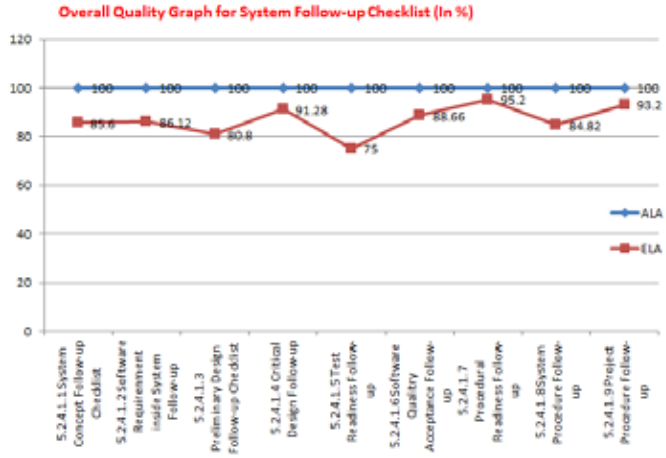


Fig. 4: Profile of System review checklist

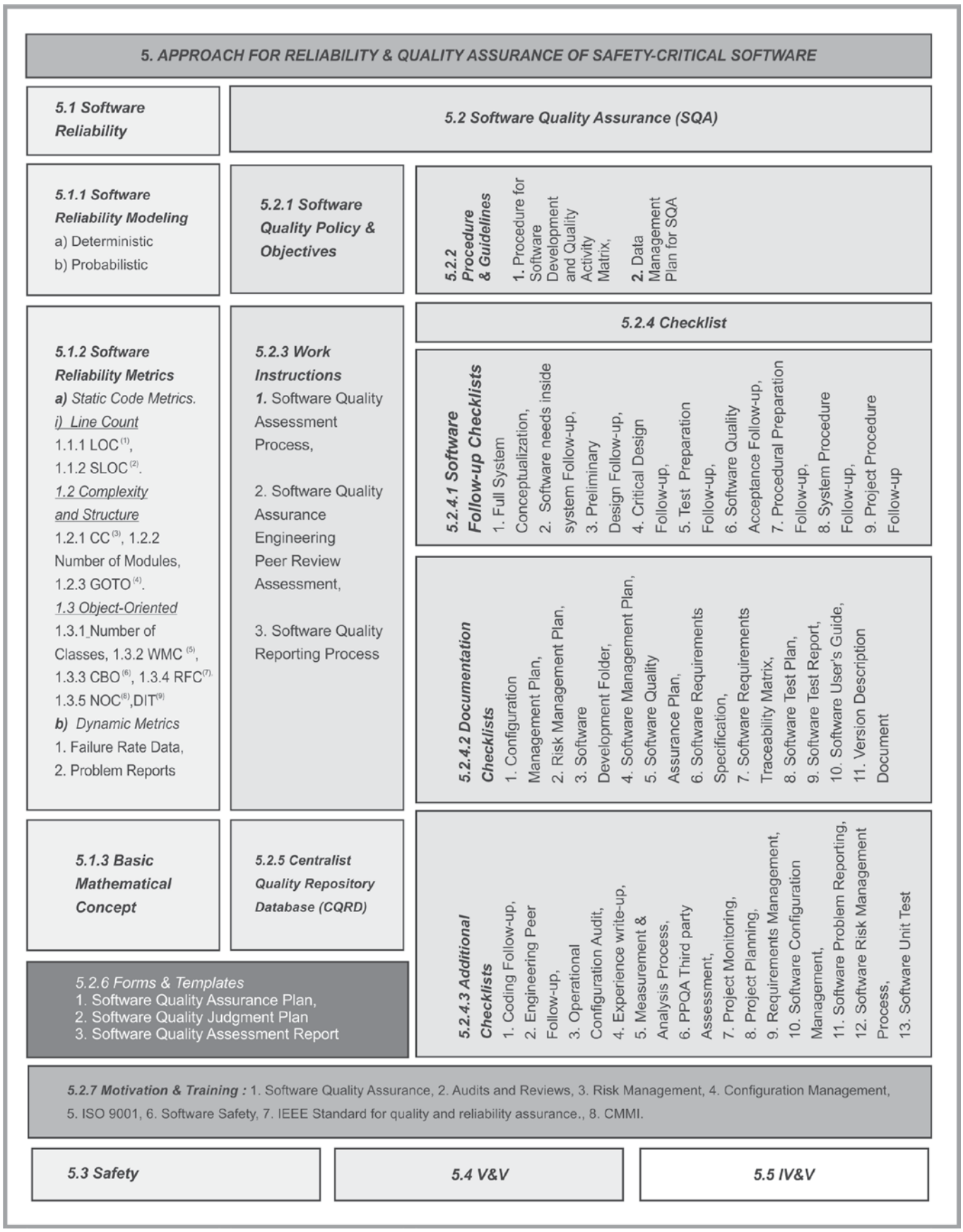
References

[1] 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.

[2] Jeffrey R. Shapiro, 2007, Microsoft SQL Server 2005: The Complete Reference.

[3] Al-Qutaish R., "Measuring the Software Product Quality during the Software Development Life-Cycle: An International Organization for Standardization Standards Perspective", Journal of Computer Science 5 (5), pp. 392-397, 2009.

[4] Mbusi Sibisi, Cornelis Cristo van Waveren, "A Process Framework for Customizing Software Quality Model", IEEE 2007.



Acronyms- Lines of code 2. Source lines of code, 3. Cyclomatic Complexity, 4. Number of Go To Statements, 5. Weighted Methods per Class, 6. Coupling Between Objects, 7. Response for a Class, 8. Number of Child Classes, 9. Depth of Inheritance Tree. Fig. 5: Enhanced Framework for Reliability and Quality Assurance of Safety Critical Software