# Modified Proposal of Life Cycle for Safety Critical Control Systems

Peter Trnovsky, *Member, IACSIT* and Pavol Tanuska

***Abstract*—The main objective of this paper is to modify the life cycle of safety critical control system and to give a brief description of it. The life cycle generally consists of three important phases – design, realization and execution. Each phase consists closer relationship and specified goals, which are briefly described in the paper. Organizations or individuals who have overall responsibility for one or more life stages of safety management have to provide all the technical activities necessary to ensure safety related systems in order to achieve and maintain their required functional safety.**

***Index Terms*—Life cycle, safety, safety critical control system.**

## I. Introduction

Safety critical control system (SCCS) is an emerging area that affecting a wide range of issues: automation in the transport, engineering and chemical industries, nuclear energy, environmentally hazardous chemical operation, etc. [1].

SCCS should be designed and configured so that a) it is reliable (in terms of errors and consequences), b) perform the functions necessary to achieve or maintain a safe condition, or at least reduce the consequences of hazard. There are two groups SCCS; those with the intention to use programmable technology and those without the intention to use programmable electronic devices (such as electromechanical components used).

The main reason for this division is to help the constructor to decide which of the two main standards are applicable for design SCCS: IEC 61508 or EN 954-1. Irrespective of the fact which standard we use, the proposal has to take into account the level of risk reduction, which is required for the system. [2]

## II. The Basic Attributes of Modified Design Life-Cycle Phase

The result of the design phase is a document that contains all the customer requirements, specifies the safety requirements, timetable, cost the developed control system, the conditions for entry into operation and to guarantee the conditions for delivery and service of control system (see Fig. 1[3]). If the developers agree with the customer to use this document as a basis for implementing a control system for testing the conditions for delivery and control system.

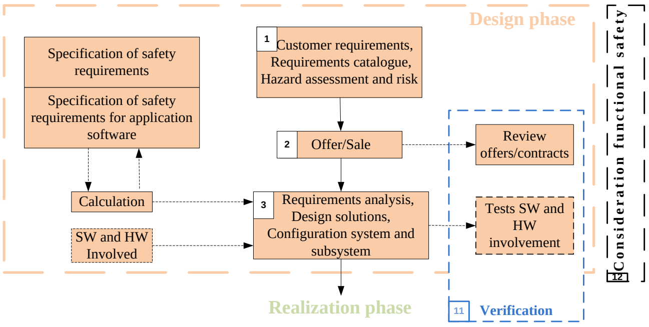
Design phase
Specification of safety requirements
Specification of safety requirements for application software
1
Customer requirements, Requirements catalogue, Hazard assessment and risk
2
Offer/Sale
Review offers/contracts
Calculation
SW and HW Involved
3
Requirements analysis, Design solutions, Configuration system and subsystem
Tests SW and HW involvement
Consideration functional safety
12
Realization phase
11
Verification

Fig. 1. Magnetization as a function of applied field.

### A. Customer Requirements, Requirements Catalogue, Hazard Assessment and Risk

Customer requirements or subcontractor should be complete, legible and technically straightforward. The customer should be treated to:

- Initial documentation of the total life cycle safety,
- Risks assessment (can they handle the engineering organization),
- Table (matrix) causes and consequences (C&E),
- Operating rules to manage operation,
- General requirements for SIF. [4]

At this point the life cycle of the overall safety of control system is necessary to identify hazards and dangerous occurrences (in all modes of operation) for all reasonably foreseeable circumstances, while in terms of defects misuse. Catalog the requirements describes the most common:

- Environment,
- Functional requirements,
- Quality requirements,
- User interface.

### B. Specification Risks

Critical systems are specified on the basis of risks that may occur. This approach is widely used in safety and security critical systems.

The analysis of these risks is based on the following steps:

- Risks identification
- Risks classification
- Decomposing risks
- Assess the possible risk reduction [5]

### C. Specification of Safety Requirements

It is necessary to specify the requirements for each SIS (Safety Instrumented System) instrument with the necessary safety functions and appropriate safety integrity in order to achieve the required functional safety.

Safety requirements:

- Functional requirements - definition of the safety

Manuscript received September 30, 2012; revised November 19, 2012.

The authors are with the Institute of Applied Informatics, Automation and Mathematic at the Slovak University of Technology, Trnava, Slovakia (e-mail: peter.trnovsky@stuba.sk, pavol.tanuska@stuba.sk).

DOI: 10.7763/IJIMT.2013.V4.360

functions of the system.

- Safety integrity requirements - definition of reliability and availability of protective system. The system is classified into one of the classes SIL. The system is the critical value required higher SIL.[3]submission.

### D. Offer/Sale

In this step the conditions of the contract are analyzed, as the system will evolve in order to achieve all the requirements of the contractor. It is also necessary in the contract itself clearly define criterion required for the safety level that the system must meet.

### E. Requirements Analysis, Design Solutions, Configuration System and Subsystem.

Although requirements analysis is the initial phase of project development it is also the most important. The aim of requirements analysis is necessary to define what should be part of the solution. The aim is to determine:

- The area in which the system will be used
- Services that the system will perform
- System boundaries.

The solution design phase which often takes place simultaneously with the collection requirements, defines a framework proposal for the overall solution. Compared with the collection requirements, the proposal is defined as the solution to be implemented. The draft resolution is usually defined:

- The structure of the solution – individual parts and their interactions
- Required hardware and software
- Communication infrastructure
- Compliance with safety standards by specific specifically for the system, etc.

Initial familiarization with the problem area includes identifying the problem. Part of the system configuration contains options and settings related to integrated system devices and related subsystems.

## III. The Basic Attributes of Modified Realization Life-Cycle Phase

The realization phase is programmed by all the functions, links between them based on defined input, performed the integration with other systems and subsystems. It is also creating integration tests and final examination, which are intended to verify the accuracy with programmed validation system and verification (see Fig. 2 [3]). Tests alone should not carry a programmer who is making the control system involved.

### A. HW Project

The project related to its hardware design and also the basic system architecture.

### B. Basic Design, System Architecture

In this step of the process life cycle, we decide on the overall architecture of the project and we work throughout its development. Important input into the project documentation hardware that includes a typical, general hardware configuration, the electronic form of a list of circuits, signals and different parameters of all circuits.

### C. Development of Application SW

In this part of the life cycle of security software applications are made for the original customer requirements, while incorporating the basic system architecture and design. After all the previous steps we proceeded to create a software application itself. New customer requirements that have emerged in this part of the life cycle will make it necessary to review and assess their impact on the overall safety, and then incorporated them into the application. [6]

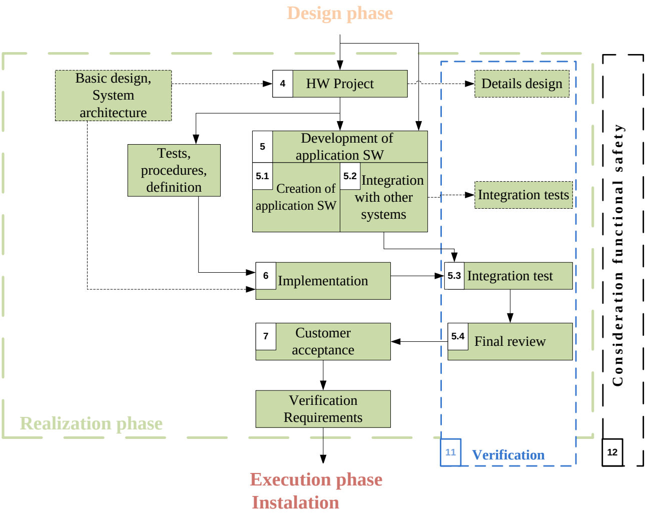
Design phase
Basic design, System architecture
4 HW Project
Details design
Tests, procedures, definition
5 Development of application SW
5.1 Creation of application SW
5.2 Integration with other systems
Integration tests
6 Implementation
5.3 Integration test
7 Customer acceptance
5.4 Final review
Verification Requirements
Realization phase
11 Verification
12
Consideration functional safety
Execution phase
Instalation

Fig. 2. Realization life-cycle phase [3].

### D. Integration with Other Systems

Security software applications are usually associated with other software systems, for example: SCADA, system lock, system management level, etc. Therefore it was necessary to verify and set the communication link between all systems and subsystems according to the principles of safety applications.

### E. Integration Test

The actual testing is very time consuming, and is conducted throughout the development of the life cycle. From the project point of view, this is a test from the system through integration to acceptance tests. Testing also requires a precise documentation format unless otherwise specified. It may be governed by IEEE 829-2008. Documentation of test case can look as follows (see Table I):

TABLE I: The Arrangement of Channels.

| Name of the test | The name of the test case |
|---|---|
| Test description | For what purpose was the test design |
| Target class/component/subsystem | Name of the system to be tested |
| Target operation class/component/subsystem | Name of the operation to be tested |
| Test type | Type of test to be conducted |
| Test level | Unit, integration, system, acceptance |
| Test values | Inputs to be used in the test |
| Verification | Expected results |

After the testing is successful, the integration of functional tests that need to ensure the participation of the system end its users.

Integration test is a verification process of interaction between software components. The objective of this test is to ensure that the individual modules are together (subsystems), working as a test for the whole process starting and ending in the target systems.

### F. Implementation

As part of this stage is implementing a knowledge base according to the proposed structures of knowledge. Successful implementation of this makes the design and implementation of all components, namely: the specification of safety requirements, the basic design and detail design, also from the system architecture, development of application software itself, but also from their own tests as shown in Figure 2.

### G. Customer Acceptance

The customer in controls the meet of its primary requirements and meet the requirements of safety integrity.

## IV. The Basic Attributes of Modified Execution Life-Cycle Phase

At this stage of life cycle control system was held to introduce service delivery and a user training manual. There is ongoing trial operation during which the provider must ensure flawless operation of the control system (see Fig. 3 [3]).

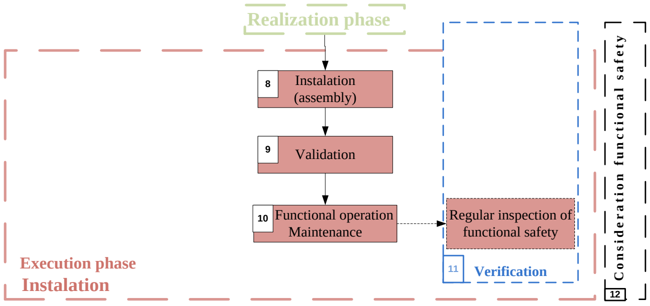
Realization phase
8
Instalation (assembly)
9
Validation
10
Functional operation Maintenance
Regular inspection of functional safety
11
Verification
Consideration functional safety
12
Execution phase
Instalation

Fig. 3. Execution life-cycle phase [3].

### A. Instalation

Installation and commissioning must be carried out in accordance with appropriate programming. All actions taken must also be documented.

### B. Validation

Validation addresses the verification of specifications developed by the system, due to realistic expectations and customer requirements. The fundamental question in the process of validation is whether the system in development is actually the one we want and we develop. This process is usually carried out with the participation of the customer and used for analysis of user requirements (which were defined at the beginning of development), prototyping, etc. [7]

### C. Functional Operation, Maintenance

During the operation and maintenance it is necessary to collect data for possible errors and test requirements and for any accident etc. This data can then be used to verify whether the assumptions made during hazard and operability (HAZOP) are correct. If there is no connection, it must be repeated calculations HAZOP and SIL. [3].

### D. Verification

The task of verification is to verify whether the developed system has been developed properly due to its specification. The process of verification is to detect errors in the system but not their absence, the absence of certain features. This is necessary to describe the system developed in a language with precisely defined syntax and semantics. [6]

### E. Consideration Functional Safety

Is carried out to assess the functional safety and integrity of safety and security of the safety system. Focus on details, which should already include the verification process, or it is not limited validation of the safety system before its launch. Functional safety assessment must include all aspects of the life cycle process. Consideration may be made at any time during the life cycle, but in one case, it is mandatory prior to starting the process. The main objective of the evaluation is to check whether they have made major steps of the life cycle and whether the problems identified are resolved. [3]

## V. Conclusion

Throughout the worlda are high demands, particularly on compliance with world standards, reliability, long life, low cost, and not least safety. Therefore efforts have been made to design a device that would minimize the impact on the control of human factor. The safety screening process is optimal if it is possible to predict safety in the earliest phases of life cycle in order to identify critical system components in terms of safety. After the identification process, it is necessary to adopt such decisions in order to increase ot safety through suitably engineering practices.

## References

[1] M. Franekova, F. Kallay, P. Peniak, and P. Vestenicky, “Komunikačná bezpečnosť priemyselných sietí,” *Žilina: Žilinská Univerzita v Žiline*, pp. 267, 2007.

[2] J. Spalek, “Metodiky modelovania kritick ých procesov,” *Žilina: Žilinská Univerzita v Žiline 2010 Katedra Riadiacich a Informacn ých Syst émov EF ŽU*. Accessed, pp. 26, 2010.

[3] M. Galik, “The safety development of SW industry application lifecycle according IEC 61508 and STN EN 61511,” *Bc. Work. MtF STU Trnava*, pp. 48, 2007.

[4] K. Holla, J. Ristvej, and L. Simak, “Posudzovanie riz ík priemyseln ých procesov,” *Žilina: Žilinská Univerzita v Žiline* 2010.

[5] N. Halbwachs, F. Lagnier, and C. Ratel, “Programming and verifying real-time systems by means of the synchronous data-flow programming language Lustre,” *IEEE Transactions on Software Engineering, Special Issue on the Specification and Analysis of Real-Time Systems*, September 1992.

[6] D. Mudroncik and M. Galik. (2010) Normy pre tvorbu softveru riadiacich systemov. [Online]. Available: http://www.odbornecasopisy.cz/index.php?id_document=38879

[7] P. Schreiber and P. Vazan, “The influence of input parameters on the efficiency of genetic algorithms,” *CIM: Computer Integrated Manufacturing: Advanced Design and Management. - Warszawa: Wydawnictwa Naukowo-Techniczne*, pp. 472-477, 2003.