# A safety requirement engineering method and tool

**Abstract:**

Requirement engineering is one of the most critical system engineering processes, particularly when it deals with the safety requirements which are non-functional requirements and are related to emergent system properties. In fact, safety requirements must be formulated at system level and then be derived at sub-system level. The main objective of this paper is to present a new tool, “SafetyLab”, which implements a method for safety treatment of complex systems. The method allows the definition of the system safety requirements following a risk and hazard analysis, and then their derivation according to a top-down approach. It is based on the famous Failure Mode, Effects, and Criticality Analysis (FMECA) and the use of Fault Trees.

SECTION I.

## Introduction

Modern systems are increasingly complex. The processes and design methods must evolve to reflect this growing complexity [1]. In particular, the management of properties such as security or reliability [2] must evolve accordingly, to ensure and enable the necessary level of confidence [3]. For an effective consideration of safety in the design process, it is necessary to consider safety in overall studies by the engineering system process. Once the safety properties are identified globally, that is to say elicited [4], these ones must be declined locally to be actually realized by the system. The local properties associated with subsystems must be established to ensure the global properties, reaching issues of traceability [5] and requirements engineering.

The work presented in this paper concerns a part of our approach for the integration of safety in system engineering processes. The approach allows taking into account the safety requirements in system engineering process to facilitate their traceability. The paper presents a new tool that allows to derive system safety requirements into sub-systems safety requirements, with the combination of several FMECA and Fault Trees analysis.

SECTION II.

## The Requirement Declination Method

The method contains 4 steps and assumes that a complex system is composed of some subsystems. The Figure 1 summarizes the process associated to the method, illustrates the different steps, and specifies the input and output information.

### A. Step 1: System FMECA

FMECA is a technique used to identify, prioritize, and eliminate potential failures from a system, design or process. For the first step, a system-level FMECA must be performed. For each system function, we identify failure modes, causes of these modes, effects on the system, and then we classify these effects and propose corrective actions. Once this step completed, a set of system causes have been identified and some corrective actions related to the system causes have been defined in order to make the system safe.

### B. Step 2: Fault Trees Analysis

A fault tree provides a simple modeling way to represent the interactions between components from the point of view of reliability. Static fault trees use traditional logic operators (such as AND and OR) to represent combinations of component failures that cause system failures.

From the system causes identified in the first step, the second step is to build fault trees. The top event of each tree corresponds to a system cause and the objective is to determine its causes at sub-system level. The leaves of the fault tree correspond to sub-systems failure modes.

So, these trees will help to link system causes to a set of subsystems failure modes. Once the Fault Trees are obtained, we associate probabilities to the elements of the trees (depending on the available safety data) with respect to the safety goals provided by the corrective actions of the system-level FMECA.

### C. Step 3: Sub-systems FMECA

The third step of the method consists on carrying out several FMECA, applied at the sub-systems level. The subsystems failure modes used in step 2 re-appears through these FMECA. Once this step is complete, useful information for the method comes from the relationship between the subsystems failure modes and the corrective actions (at sub-system level).

### D. Step 4: Derivation Synthesis

The fourth and final step of the method is a synthesis that will provide the derivation of system-level safety requirements in sub-system requirements. This derivation synthesis is based on the following 3 types of relations:

* System causes and system corrective actions,
* System causes and sub-systems failure modes,
* Sub-systems failures modes and sub-systems corrective actions,

Indeed, these various relations allow linking the system corrective actions to the subsystems ones. By translating the corrective actions into safety requirements, we obtain the relationship between system level safety requirements and those at the sub-systems level. The traceability between this requirements is the main purpose of the method, because it enhances the engineering of safer systems and provides the ability to perform impact analysis on safety.

SECTION III.

## The New Safety Tool: “SafetyLab”

A new safety tool implements the method presented in the previous part. It is named “SafetyLab” and it is composed of four different modules.

### A. Defining the System and its Sub-Systems

The first module allows the definition of the system and its sub-systems. The architecture can be update by modifying the name of each element, or adding or removing sub-systems. From this module, all the other modules can be executed following the sequence of the method presented in the previous part.

### B. Performing FMECA

The second module is the one used to perform FMECA. As usual, this analysis is organized in a table. Columns can be personalized, except the one corresponding to the causes, the failure modes and the corrective actions that are protected because of their implication in the method.

### C. Fault Trees Design

The third module deals with the fault trees edition. This module is started after the selection of one system cause, once the “Fault Trees” button of the first module is pushed. It allow the creation of fault trees using different types of events and gates. The desired probabilities (as objectives) can be defined for all the events of the fault tree. Then the computed probabilities of the events are automatically updated based on the desired probabilities of the leaves of the fault tree. The minimal cuts are also automatically generated for all the events. All this data are available in the properties tab of the graphical user interface.

### D. Generating the Derivation Synthesis

The last important module of the “SafetyLab” tool concerns the derivation synthesis. It must be open once the three first steps of the derivation method are achieved, and it generates and shows the derivation of the system safety requirements in subsystem safety requirements (Figure 2).

SECTION IV.

## Conclusion

The new tool “SafetyLab” implements the proposed method which is a support framework to define system safety requirements with an objective of traceability and requirements derivation. The interest is multiple for the safety field: the method deals with the safety elements (failure modes, safety requirements…) and it is done with a comprehensive system engineering (with traceability and requirements derivation) which is a factor contributing to safer systems. This method extends the principle and strengthens the links between failure modes researches and analysis (FMECA) and causes analysis. In this work, only corrective actions for frequency reduction of failure modes are considered. However, the study of corrective actions reducing the severity of the effects caused by the failure modes is in progress. Another current study focused on the definition of requirements concerning with the architecture of the system. These “architectural” requirements, which are linked to the system safety requirements, will be taken into account in the method and the tool in further versions.