Software Requirements for a System to Compute Mean Failure Cost

Anis Ben Aissa  
Faculty of Sciences of Tunisia  
University of Tunis El Manar  
2092 Tunisia  
+216-98-692-415  
anis_enit@yahoo.fr

Robert K. Abercrombie, Frederick T. Sheldon  
Oak Ridge National Laboratory  
Oak Ridge, TN 37831 USA  
+1-865-241-6537/576-1339  
abercrombie@ornl.gov  
sheldonft@ornl.gov

Ali Mili  
College of Computing Sciences  
New Jersey Institute of Technology  
Newark NJ 07102-1982 USA  
+1 973-596-5215  
mili@cis.njit.edu

ABSTRACT

In earlier works, we presented a computational infrastructure that allows an analyst to estimate the security of a system in terms of the loss that each stakeholder. We also demonstrated this infrastructure through the results of security breakdowns for the e-commerce case. In this paper, we illustrate this infrastructure by an application that supports the computation of the Mean Failure Cost (MFC) for each stakeholder.

General Terms


Keywords


1. INTRODUCTION

We envision a system that supports the archival of information pertaining to the Mean Failure Cost (MFC), as well as applications of the MFC metrics (such as: computing the return on investment for a given V&V action, or computing the return on investment for a given architectural enhancement) [1]. Several distinct stakeholders intervene in the determination and entry of relevant information; hence a natural way to structure the requirements is by stakeholder. We identify individual stakeholders, then specify for each, what functionalities that stakeholder expects from the system [1-4]. We document the requirements specification of a system, whose purpose is to compute and use the MFC of systems of interest. In order to avoid confusion, we refer to the system we are specifying as K and to object systems that system K analyzes as S. In this extended abstract, we content ourselves with a summary presentation of the formula for MFC, with minimal explanation, referring the interested reader to previous works [1-4].

\[ MFC = ST \cdot DP \cdot IM \cdot PT \]

Where: MFC is a vector with as many entries as there are system stakeholders, and MFC(H) is the mean failure cost of stakeholder H.

- ST is a matrix with as many rows as there stakeholders and as many columns as there are distinct security requirements, and ST(H,R) is the stake that stakeholder H has in satisfying requirement R. Stakeholders fill this matrix, which we call the Stakes matrix, where each stakeholder fills one row.

- DP is a matrix with as many rows as there are distinct security requirements and as many columns as the system in question has components, and DP(R,C) is the probability that the system fails to meet requirement R if component C is compromised. The architect of system S, who knows the role that each component plays in achieving each requirement, fills this matrix, which we call the Dependability matrix.

- IM is a matrix that has as many rows as the system has components and as many columns as there are security threats under consideration (similar to fault models in reliability analysis), and IM(C,T) is the probability that component C is compromised if threat T has materialized. The Verification and Validation team, who know how the various security threats affect / compromise system components, fill this matrix, which we call the Impact matrix.

- PT is a vector that has as many entries as there are threats in our threat model, and PT(T) is the probability that threat T materializes for a unit of operation time (e.g. one hour of operation). This vector, which we call the Threat vector, is filled by the security team, which knows the threat configuration (i.e. the probability of occurrence of each threat per unit of operation time) within which the system operates.

Matrices DP, IM and vector PT are dimension-less probabilities, but ST entries are quantified in terms of cost per unit of operation time, say $/hour; hence so are MFC entries.
2. STAKEHOLDER REQUIREMENTS

We structure the system requirements by stakeholders: To present the system requirements, we discuss in turn the functionalities offered by the system to each stakeholder. We review in turn the following stakeholders:

- The administrators of system K,
- The requirements engineers of system S,
- The users of system S,
- The architect of system S,
- The security team, and
- The Verification and Validation team.

2.1 The System Administration Team

We begin with the presentation of the System Administration team. This is the most important team with the focus tasks responsible for maintaining information about global ROI (vs. stakeholder ROI). It authorizes and monitors the following login and access privileges:

- Subscribes/un-subscribes stakeholders,
- Manages login information, access rights,
- Initiates an ROI calculation,
- Enters system wide factors,
- Mediates negotiations between stakeholders,
- Moderates cross postings between stakeholders,
- Approves system-wide modifications,
- Displays MFC data and ROI/NPV data for all users, and
- Perform What-If analyses on ROI data.

The user of the system creates an account by filling in the form below (Figure 1).

The user can select his role within the team (optional) as Team reporter or Team recorder. The administrator will validate this request according to the type of connection the "login as" interface (one of four types of interfaces listed).

The user of system can connect by choosing his team and typing his user name and password Figure 2.

2.2 The Requirements Engineer of System S

This stakeholder is responsible for eliciting, organizing, and documenting the requirements of system S. In particular, he has the crucial task of structuring the requirements of system S for the purposes of the stakes matrix. The precision of the MFC calculations is dependent upon the decomposition of the system requirements into orthogonal (i.e. non-overlapping) components, a condition that is virtually impossible to meet in practice, but we maintain it as a criterion. Among the requirements/services that this stakeholder is expected to know and input concerning the system are:

- The ability to represent the clauses of the system requirement specification to be stored. Each clause can be represented by a name, a textual description, a graphic, or some appropriate mathematical notation.
- The ability to modify a previously stored structured requirements/services specification. This will result in an automatic notification, by the system, to all the stakeholders who must be informed (for example, all the users of the system who have previously entered cost information with respect to a decomposition must be informed if the decomposition has been changed).
- The ability to define default cost transfer options when a requirements specification is modified. For example, if a clause is simply renamed, then the cost information is automatically transferred; if a clause is split into two sub clauses, then as a default we can divide associated failure costs in two; etc. (see Figure 3).
2.3 The System User
Each user is responsible for entering the failure costs that he associates with each component of the system requirements, as entered by the requirements engineer. Among the services that this stakeholder is expected to know and input concerning the system are:

- The ability to review clauses of the system (S) requirements specification and to associate failure costs to them, in terms of dollars per hour of operation.
- The ability to address a question regarding a particular clause from the requirements engineer, requesting (for example) a clarification.
- The question as well as the requirement engineer’s reply remain on the public record attached to the clause (so that subsequent users can benefit from the clarification).

The System then:
- Computes and displays stakeholder MFC.
- Computes and displays stakeholder ROI, for a variety of relevant decisions.

This allows informed decision making by the Stakeholder:
- Evaluating, approving, or rejecting possible system wide changes based on calculated ROI/NPV values.
- Performs what-if analyses, without impact on stored data.

2.4 The System Architect
The architect of system S designs the architecture of the system based on the requirements entered by the requirements engineer. He is also responsible for filling in the dependency matrix by entering, for each clause R of the requirements specification and for each component C of the architecture, the probability that system S fails to satisfy clause R given that component C has failed. Note that a component, in our understanding here, may refer to a component or a connector, in the sense of software architectural description languages.
2.5 The Verification and Validation Team
Given a catalog of security threats, the Verification and Validation team is able to determine which components are likely to be affected by each threat, and with what probability each component may be affected once the threat has materialized. This team, in effect, responsible for filling in the impact matrix; they can enter the required information once the security team has entered the threat vector and the architecture team has entered the architecture of the system (see Figure 6). Among the services that this stakeholder is expected to know and input concerning the system are:

- The ability to store and modify entries of the impact matrix.
- The ability to support communication to the system architect, with notification of all relevant parties.
- Assistance with filling in entries of the impact matrix, using common patterns.

2.6 The Security team
The responsibility of this team is to analyze the threat configuration within which the system (S) operates and to derive for it the threat vector PT (see Figure 7).

Among the services that this stakeholder is expected to know and input concerning the are:

- Maintain an active vector of credible threats, possibly from an exhaustive list.
- Assign them occurrence probabilities.
- Assist the security team in computing these probabilities, using some common patterns. (see Figure 7)

3. CONCLUDING REMARKS
In this paper, we have presented an example of software requirements to compute mean failure cost for each stakeholder. We have defined the relationship between the trams and the exchange of the information between them.

4. REFERENCES