A Proposed Technique for Software Development Risks Identification by using FTA Model

Hatem A. Khater, A. Baith Mohamed, and Sara M. Kamel

Abstract—Software Development Risks Identification (SDRI), using Fault Tree Analysis (FTA), is a proposed technique to identify not only the risk factors but also the causes of the appearance of the risk factors in software development life cycle. The method is based on analyzing the probable causes of software development failures before they become problems and adversely affect a project. It uses Fault tree analysis (FTA) to determine the probability of a particular system level failures that are defined by A Taxonomy for Sources of Software Development Risk to deduce failure analysis in which an undesired state of a system by using Boolean logic to combine a series of lower-level events. The major purpose of this paper is to use the probabilistic calculations of Fault Tree Analysis approach to determine all possible causes that lead to software development risk occurrence.

Keywords—Software Development Risks Identification (SDRI), Fault Tree Analysis (FTA), Taxonomy for Software Development Risks (TSDR), Probabilistic Risk Assessment (PRA).

I. INTRODUCTION

IN software development, the possibility of reward is high, but so is the potential for disaster. The need for software risk management is illustrated in Gilb's risk principle. "If you don't actively attack the risks, they will actively attack you" [Gilb-88] [1]. Risk management techniques, when correctly applied, can help ensure the successful outcome of software projects. Risks are potential issues that, if not identified and managed, could unexpectedly surface and cause substantial trouble when least expected. There are many philosophies and approaches for managing risks, including those discussed by Boehm (1989) and Charette (1989). The first step in risk management is to identify and prioritize the risk areas relevant to a project. Each project has different risks due to the unique characteristics that differ from project to project [2].There are several Risk Management models and the most used one is SEI (Software Engineering institute) Risk Management paradigm that consists of five sequential and iterative steps: Identification, Analysis, Planning, Tracking and Control. In parallel, two common activities are performed: Documentation and Communication. The SEI Risk Management paradigm is depicted in Fig. 1 – (Risk Management Model) [3]. The paradigm illustrates a set of functions that are identified as continuous activities through the life cycle of a project. The method was originally

A. Baith Mohamed, Vice Dean for Environmental Affairs & Community Service, is with Arab Academy for Science, Tech. & Maritime Transport, P.O.Box 1029 Alexandria, Egypt (phone : 203-562-2366; e-mail: baithmm@hotmail.com).

developed as a project management method and the element of risk management was later added to the equation.

Fig. 1 Risk Management Model

TABLE I
RISK MANAGEMENT MODEL PROCESS

Function	Description
Identification	Search for and locate risks before they become problems.
Analysis	Transform risk data into decision-making information. Evaluate impact, probability, and time frame; classify risks, and priorities risks.
Planning	Translate risk information into decisions and mitigating actions (both present and future) and implement those actions.
Tracking	Monitor risk indicators and mitigation actions.
Control	Correct for deviations from the risk mitigation plans.
Documentation & Communication	Provide information and feedback internal and external to the project on the risk activities, current risks, and emerging risks.

The remainder of this paper focuses on risk identification and is based on the simple premise that without effective and repeatable risk identification methods, truly effective risk management is impossible; you can't manage what you don't know about. In keeping with this approach, the described

World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:7, No:1, 2013

identification method also begins to address the communication issue central to risk management. In this paper basic concepts of Software Risk Management is introduced; In Section II we will introduce the Software Risk Identification and the Taxonomy for Software Development Risks (TSDS) that are presented by SEI; then in Section III a short introduction of Probabilistic Risk Assessment (PRA); then in Section IV the Fault Tree Analysis model is introduced; finally in Section V applying Fault tree analysis approach to identify all potential causes leading to software development risk occurrence.

II. RISK IDENTIFICATION

Risk Identification in projects means to determine potential risk elements by using a consistent and structured method; this is, probably, the most important step among those that compound the Risk Management activities, due to the fact that without a correct risks determination, it is not possible to develop and to implement in advance proper responses to the problems that could appear in the project [4]. The result of the risks identification is a list that contains the risks that have been identified and their related category. Taxonomies are sorted classifications of elements according to their presumed relationship; they can be used as a very useful tool on different areas of the science and the industry where it is required to organize and to expedite the access to a wide set of related elements [3,5].The SEI risk identification method is based on the following assumptions:

1) Software development risks are generally known by the project's technical staff but are poorly communicated.
2) A structured and repeatable method of risk identification is necessary for consistent risk management.
3) Effective risk identification must cover all key development and support areas of the project.
4) The risk identification process must create and sustain a non-judgmental and non-attributive risk elicitation environment so that tentative or controversial views are heard.
5) No overall judgment can be made about the success or failure of a project based solely on the number or nature of risks uncovered.

The SEI taxonomy of software development maps the characteristics of software development and hence of software development risks.

A. The Software Development Risk Taxonomy

Central to the risk identification method is the software development taxonomy. The taxonomy provides a framework for organizing and studying the breadth of software development issues. Hence, it serves as the basis for eliciting and organizing the full breadth of software development risks both technical and non-technical. The taxonomy also provides a consistent framework for the development of other risk management methods and activities. The software taxonomy is organized into three major classes.

- *Product Engineering*: The technical aspects of the work to be accomplished.

- *Development Environment:* The methods, procedures, and tools used to produce the product.
- *Program Constraints:* The contractual, organizational, and operational factors within which the software is developed but which are generally outside of the direct control of the local management.

These taxonomic classes are further divided into elements and each element is characterized by its attributes [6].
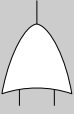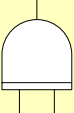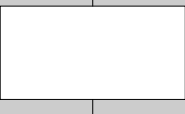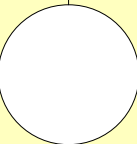
III. PROBABILISTIC RISK ASSESSMENT (PRA)

PRA is a systematic and comprehensive methodology to evaluate risks associated with every life-cycle aspect of a complex engineered technological entity from concept definition, through design, construction and operation, and up to removal from service [11].Fault Tree Analysis (FTA) is one of the most important logic and probabilistic techniques used in PRA and system reliability assessment today. Over the past two decades, probabilistic risk assessment and its underlying techniques, including FTA, has become a useful and respected methodology for safety assessment. Because of its logical and systematic approach, PRA and FTA have been proven capable of uncovering design and operational weaknesses that escaped even some of the best deterministic safety. A foremost strength of PRA and its underlying analysis techniques, including FTA, is that it is a decision support tool. In safety applications, this methodology helps managers and engineers find design and operational weaknesses in complex systems and then help them systematically and efficiently uncover and prioritize safety improvements [10].

IV. FTA MODEL

Fault Tree Analysis (FTA) is one of the oldest, most diffused techniques in industrial applications, for the dependability analysis of critical systems [7].The FTA is a deductive method: beginning with an undesired event (also called the top event) the FTA is used to find the causes for this top event. When determining the causes, a fault tree is constructed from top to bottom. For its construction several symbols are used that indicate the relation between different events. The main symbols are presented in Table II [8].One of the main restrictive assumptions in FTA is that basic events must be assumed to be statistically independent, and their interaction is described by means of Boolean OR/AND gates, so that only the combination of events is relevant, and not their sequence. We refer to this model as Static Fault Tree (SFT) [9]. The gates show the relationships of events needed for the occurrence of a "higher" event. The "higher" event is the output of the gate; the "lower" events are the "inputs" to the gate. The gate symbol denotes the type of relationship of the input events required for the output event [11].

World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:7, No:1, 2013

TABLE II
FAULT TREE SYMBOLS

Symbol	Description
	OR Gate denotes the situation in which an output event occurs if any one or more of the input events occur.
	AND Gate denotes the situation in which an output event occurs only when all the input events occur.
	RECTANGLE denotes an event that results from the combination of fault events through the logic gate.
	CIRCLE denotes a basic fault event.

The Fault Tree Analysis is used for reliability and safety security analyses. The proceeding is very similar to the Reliability Block Diagram RBD. The aim is to determine possible combinations of causes which can lead to certain undesirable events (event), the so called top level events. The job of a FTA is as follows:
- The generation of a graphic / logical tree structure to the understanding of the connections.
- Identification of possible failure causes and their combinations.
- Calculation of the probability of the undesirable event.
- Comparison of variations.

A. Reliability & Failure Probability Relationships

- S = Successes
- F = Failures
- Reliability :

$$R = \frac{S}{(S + F)}$$

- Failure Probability:

$$P = \frac{F}{(S + F)}$$

$$R + P_F = \frac{S}{(S + F)} + \frac{F}{(S + F)}$$

B. Reliability (R) & Failure Probability (PF) through Gates

Gates are the logic symbols that interconnect contributory events and conditions in a fault tree diagram. The AND and OR gates, as well as Voting OR gates in which the output event occurs if a certain number of the input events occur), are the most basic types of gates in classical fault tree analysis.

- *AND Gate:*

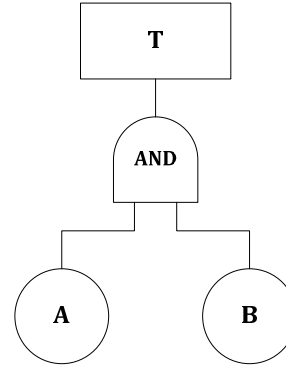Both of two, independent elements must fail to produce system failure.

Fig. 2 AND Gate

$$R_T = R_A + R_B - R_A R_B$$

$$P_F = 1 - R_T$$

$$P_F = 1 - (R_A + R_B - R_A R_B)$$

$$P_F = 1 - [(1 - P_A) + (1 - P_B) - (1 - P_A)(1 - P_B)]$$

$$P_F = P_A P_B \tag{1}$$

- *OR Gate:*

Consider a system with two components: A and B. The system fails if both A and B fail [13].
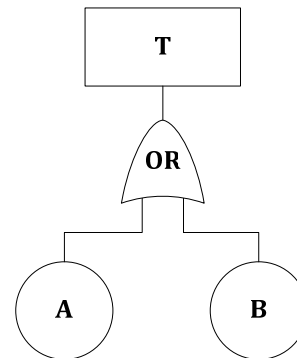

Fig. 3 OR Gate

$$R_T = R_A R_B$$

$$P_F = 1 - R_T$$

$$P_F = 1 - (R_A R_B)$$

$$P_F = 1 - [(1 - P_A)(1 - P_B)]$$

$$P_F = P_A + P_B - P_A P_B \tag{2}$$

World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:7, No:1, 2013

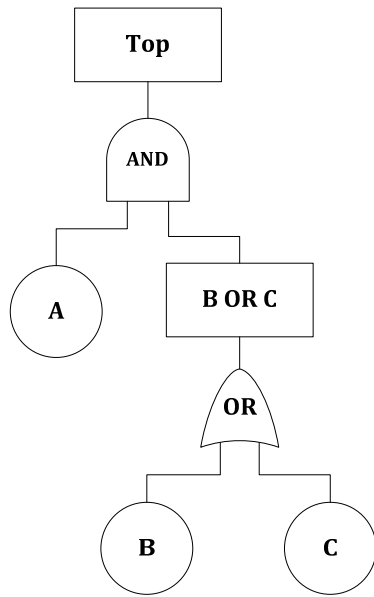The next figure shows a simple Fault Tree

Fig. 4 A Simplified Fault Tree

The probability of top event failure is calculated as shown:

$$P_{top} = P_A \cdot [P_B + P_C - P_B P_C]$$

V. PROPOSED MODEL

The proposed model that described in this paper is based upon the SEI taxonomy of software development risks. The taxonomy provides a framework for organizing and studying the breadth of software development issues and hence provides a structure for surfacing and organizing software development risks. Using Fault Tree Analysis and Probabilistic Risk Assessment is able to find the causes to undesirable events and to evaluate the risk quantitatively; the fault tree is constructed by first identifying the top fault event, which, in this case, is a Programmatic Risks class. The secondary events that are contributed directly to the top fault are the elements that are listed under Programmatic Risks class. These secondary events are further broken down to determine the root causes. We consider that the attributes is the last level of this tree which called the minimal cutsets of this fault tree (leaf events) that are contributed directly to the elements on the Taxonomy. Fig.5 shows the completed fault tree for Programmatic Risks class. A minimal *cutset* is a smallest combination of component events which, if they all occur, will cause the top event to occur.
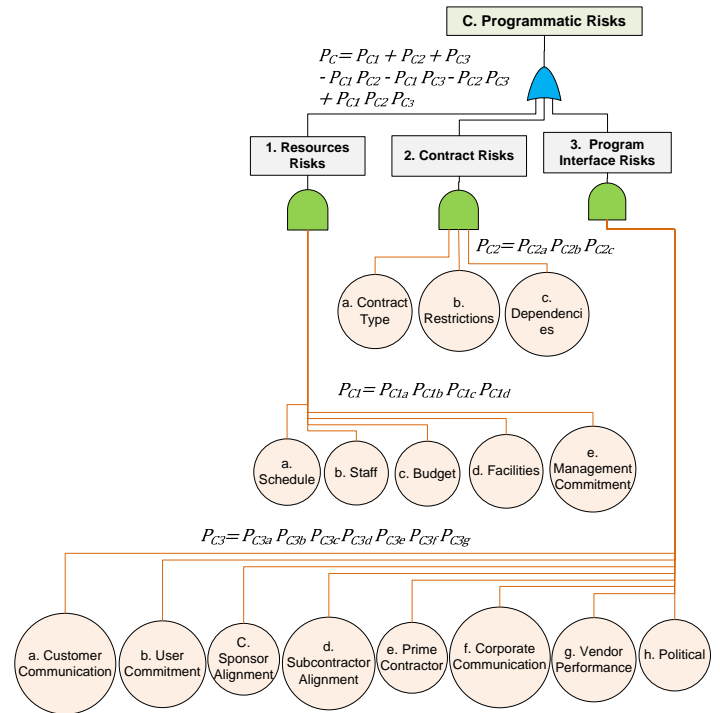

Fig. 5 Fault Tree Analysis for Programmatic Risks class

We assumed the probability values of each risk in attribute level (leaf level) to calculate the top event risk as shown in Table III.

TABLE III
THE PROBABILITY OF RISK AT LEAF LEVEL

Class	Element	Probability of Risk on Attribute level	
C. Program Constraints	1. Resources	a. Schedule	P_{C1a}
		b. Staff	P_{C1b}
		c. Budget	P_{C1c}
		d. Facilities	P_{C1d}
		e. Management Commitment	P_{C1e}
	2. Contract	a. Type of Contract	P_{C2a}
		b. Restrictions	P_{C2b}
		c. Dependencies	P_{C2c}
	3. Program Interfaces	a. Customer	P_{C3a}
		b. Associate Contractors	P_{C3b}
		c. Subcontractors	P_{C3c}
		d. Prime Contractor	P_{C3d}
		e. Corporate Management	P_{C3e}
		f. Vendors	P_{C3f}
		g. Politics	P_{C3g}
		h. political	P_{C3h}

Using equations (1) and (2), we can calculate the probability of Top event failure by the next equation:

$$P_{C1} = P_{C1a} P_{C1b} P_{C1c} P_{C1d} \qquad (3)$$

World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:7, No:1, 2013

$$P_{C2} = P_{C2a}\, P_{C2b}\, P_{C2c} \qquad (4)$$

$$P_{C3} = P_{C3a}\, P_{C3b}\, P_{C3c}\, P_{C3d}\, P_{C3e}\, P_{C3f}\, P_{C3g} \qquad (5)$$

Therefore, the probability of risk of class C can be determined by next equation:

$$P_{C} = P_{C1} + P_{C2} + P_{C3} - P_{C1} P_{C2} - P_{C1} P_{C3} P_{C2} P_{C3} + P_{C1} P_{C2} P_{C3} \qquad (6)$$

The Table IV shows nominal values of basic events' probabilities of occurrence. Then, using equations (3) to (6), the probabilities of intermediate events and at last probabilities of the Top Event should be determined. The Table V shows the results of result of the used model.

TABLE IV
INITIAL DATA

$P_{C1A} = 0.07$	$P_{C2A} = 0.03$	$P_{C3A} = 0.07$
$P_{C1b} = 0.04$	$P_{C2b} = 0.01$	$P_{C3b} = 0.2$
$P_{C1c} = 0.06$	$P_{C2c} = 0.02$	$P_{C3c} = 0.09$
$P_{C1d} = 0.1$		$P_{C3d} = 0.06$
$P_{C1e} = 0.05$		$P_{C3e} = 0.08$
		$P_{C3f} = 0.1$
		$P_{C3g} = 0.05$
		$P_{C3h} = 0.3$

TABLE V
PROBABILITIES CALCULATED BY EQUATIONS

$P_{C1} = 0.84 * 10^{-6}$	$P_{C2} = 6 * 10^{-6}$	$P_{C3} = 0.0091 * 10^{-6}$
$P_{C} = 6.85 * 10^{-6}$		

VI. CONCLUSION

In this paper, we presented the behavioral and probabilistic model of Software risk Identification by using the probabilistic calculations of Fault Tree Analysis, and the following conclusions are drawn:

- Focuses on software risk identification, mainly the SEI-Software Development Risk Taxonomy, because it is more detailed than many other approaches. This Taxonomy is useful for software development organizations as it is very efficient.
- Applying Fault Tree Analysis approach on SEI Software Development Risk Taxonomy, by using Boolean OR/AND gates and assigning probabilities of these risks, we were able to compute an expression for the overall probability of the whole system.
- FTA is a proven method for identifying and evaluating risk in high hazard applications that has the potential failure.
- Identify all the causes that can make the top event occur using fault tree symbols and the logic tree format. More specifically, by using deductive reasoning highlight event that can lead to the occurrence of the top event.

The advantage of this approach is the dynamic monitoring and estimation of various probabilities of risks, because it was appropriate by combining the hierarchical model of the risks (TSDS) and Fault Tree Analysis. Furthermore it can help software risk management process. This probabilistic risk tree structure can apply to some software tools.

World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:7, No:1, 2013

APPENDIX A: FAULT TREE ANALYSIS FOR SOFTWARE DEVELOPMENT RISK
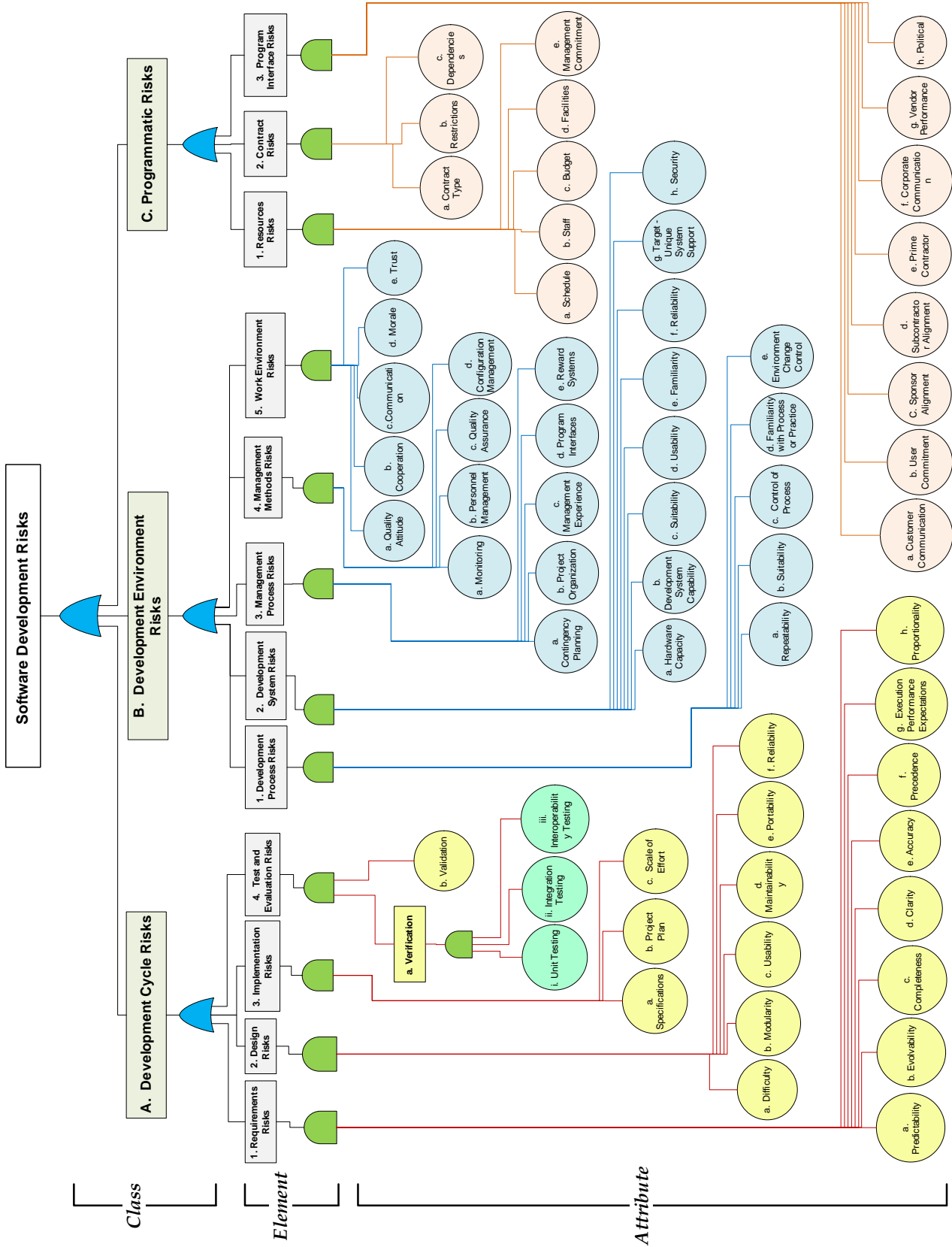
REFERENCES

[1] Linda Westfall, "Software Risk Management" The Westfall Team, PMB 383, 3000 Custer Road, Suite 270Plano, TX 75075, Jan 2002.

[2] GREGORY A. TOTH, "Automated Method for Identifying and Prioritizing Project Risk Factors" USC Center Jbr Software Engineering, University of Southern California, Los Angeles, CA 90089, 1995.

[3] Sebastian Maniasi, Paola Britos and Ramón García-Martínez, "A Taxonomy-Based Model for Identifying Risks" Global Software Group (GSG) Argentina Motorola Argentina Centro de Software .Software & Knowledge Engineering Center. Graduate School. Buenos Aires Institute of Technology 2005.

[4] Futrell, Shafer & Shafer, "Quality Software Project Management" Prentice Hall Editorial. ISBN 0130912972, 2002.

[5] Webster, "Merriam-Webster's Collegiate Dictionary" Merriam-Webster Editorial. ISBN 0877797099. 2004.

[6] Marvin J. Carr, Suresh L. Konda, Ira Monarch, F. Carol Ulric "Taxonomy-Based Risk Identification" Software Engineering Institute Carnegie Mellon University Pittsburgh, Pennsylvania 15213, June 1993.

[7] Guillaume Merle, Jean-Marc Roussel, Jean-Jacques Lesage, "Dynamic Fault Tree Analysis Based On the Structure Function" Annual Reliability and Maintainability Symposium, 2011.

[8] E. Genender, M. Mleczko, O. Doring, H. Garbe, and S. Potthast "Fault tree analysis for system modeling in case of intentional EMI" Advances in Radio Science, 2011.

[9] Guillaume Merle, Jean-Marc Roussel, and Jean-Jacques Lesage, "Improving the Efficiency of Dynamic Fault Tree Analysis by Considering Gates FDEP as Static" European Safety and Reliability Conference, 2010.

[10] Dr. Michael Stamatelatos, "Probabilistic Risk Assessment: What Is It And Why Is It Worth Performing It?" NASA Office of Safety and Mission Assurance, 2000.

[11] Dr. Michael Stamatelatos, Dr. William Vesely, Dr. Joanne Dugan, Mr. Joseph Fragola, Mr. Joseph Minarick, Mr. Jan Railsback, "Fault Tree Handbook with Aerospace Applications" NASA Headquarters Washington, DC 20546,August, 2002.

[12] Pat L. Clemens, "Fault Tree Analysis" Pat L. Clemens and Jacobs Sverdrup, Inc, 4th Edition provided as a free service by www.fault-tree.net, May 1993.

Hatem A. Khater received BSc. in Electrical Engineering, MSc. in Electronic and Communication Engineering and Ph.D. in Computer Engineering, Kent University, United Kingdom, in 2008. He is Dr. at the Arab Academy for Science and Technology (AASTMT), Computer Engineering Department, Electronic Communication Engineering Department, College of Engineering and Technology, College of Computing & Information Technology, Walls University, Electronics and Computer Engineering Department Kent University. In addition, he holds the position of Chief of Naval Research and Development Department. His research interests include Software Engineering, cryptograph, Image Feature Detection, Matching Technique,, Geometric Transformation, Image Registration, Pattern Recognition, Computer Graphics, Web Programming, Automatic Controls, Modern Electronics communication, Acoustics, Voice Identifications, GIS, International and European Business, Economics & Management information Systems. Member and Reviewer at IET. Also Member of Image and Information Engineering Research Group, University of Kent, U.K.

A. Baith Mohamed received the BSc. in Computer Science, Univ. of Vienna, and MSc. and Ph.D. in Computer System Security, Vienna University, in 1992. He is a Professor at the Arab Academy for Science, Technology and Maritime Transport (AASTMT), Computer Engineering Department. In addition, he holds the position of Vice Dean for Training and Community Services, College of Engineering and Technology. His research interests include computer and Network Security, Bioinformatics, Steganography, cryptography, and Genetic Algorithms. He was also a member of an International project team in Europe, for design and implementation and maintenance of subsystems in the environment of peripheral processor controls as part of a large Public Switched Systems (EWSD) in SIEMENS, AG. Austria. Also, he was a scientific researcher in the department of Information Engineering, Seibersdorf Research Institute in Austria for design and implementation of security software system in the domain of railway automation project (VAX/VMS, DEC system), and Austria. He was also a member of software testing for distribution points in an international project in AEG, Vienna, Austria. He is a senior member of IEEE Computer Society since 2001.

Sara M. Kamel received BSc. degree in Computer Engineering from Arab Academy for Science, Technology& Maritime Transport(AASTMT), Alexandria, Egypt in 2007. She is a Master student at Computer Engineering Department (AASTMT).