

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

JULIANO ARAUJO WICKBOLDT

**A Framework for Risk Assessment
Based on Analysis of Historical
Information of Workflow Execution in
IT Systems**

Thesis presented in partial fulfillment
of the requirements for the degree of
Master of Computer Science

Prof. Dr. Lisandro Zambenedetti
Granville
Advisor

Porto Alegre, May 2011

CIP – CATALOGING-IN-PUBLICATION

Wickboldt, Juliano Araujo

A Framework for Risk Assessment Based on Analysis of Historical Information of Workflow Execution in IT Systems / Juliano Araujo Wickboldt. – Porto Alegre: PPGC da UFRGS, 2011.

113 f.: il.

Thesis (Master) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR-RS, 2011. Advisor: Lisandro Zambenedetti Granville.

1. Risk assessment. 2. Risk management. 3. Change management. 4. Project management. I. Granville, Lisandro Zambenedetti. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos Alexandre Netto

Pró-Reitor de Coordenação Acadêmica: Prof. Rui Vicente Oppermann

Pró-Reitora de Pós-Graduação: Prof. Aldo Bolten Lucion

Diretor do Instituto de Informática: Prof. Flávio Rech Wagner

Coordenador do PPGC: Prof. Álvaro Freitas Moreira

Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

*"If I have seen farther than others,
it is because I have stood on the shoulders of giants."*
— SIR ISAAC NEWTON

ACKNOWLEDGMENTS

First, I must express my sincere thanks to my parents and brother for the unconditional support and for being my example of courage and perseverance. If today I am taking one more step ahead, both professionally and personally, this is mostly due to the fact that they have always believed in my potential and encouraged me to move on. To my grandpa Araujo and grandma Carmen, I thank you for teaching me what really matters in life. As a child, when I used to take vacations and enjoyed them with my grandparents in Pelotas, I have learned little things like respect, honesty, and humility, which turned me into better person.

I also thank my advisor Lisandro, who in spite of the distance, has always been incredibly available (*i.e.*, on-line) when I most needed. Lisandro, be sure you have been an excellent professor and also a great friend. Today, I realize how lucky I was to fall almost accidentally into the Computer Networks Group at UFRGS. I could not forget also to thank my unofficial pseudo-co-advisor Luciano, who has always placed much trust on me and was also a great partner in our journeys (deadlines, travels, embarrassing videos, etc.).

To the "gang" from HP's project I thank you guys for your friendship, for your sense of humor, for the endless philosophical discussions (after all, why do we need to manage risks in a system with automated rollback support?), and for long nights of deadlines on campus (with the guards and the dogs). Particularly, to my great friend "Rubão", fellow in exploring Europe and Japan (with only one expression in Japanese, *biru mitsu*); to Weverton (aka Mocarongo) and Machado, my veterans who I bothered to learn something about change management; to my favorite undergrads, Luis (aka Nautilus), the renowned creator of the coffee bot, Fabrício the last active supporter of Juventude, and Alan (aka Lollipop), only those who have already worked as an undergrad intern in a research project know how hard it is; to Ricardo and Bruno, who later joined the group, but fell into place as if we've already known each other for years; to our managers Cristiano and Lincoln, who always found a way to organize our mess (it was not an easy task, I can assure that).

I thank the other members of our group, Flávio "Cockroach", Paulo "Priumo", Rafael SBZ, Clarissa (our brother), Raniery, Jeferson, and all the others, for the friendship and for the gigantic battles fought during events organized by this selective group of which I'm proud to be part of. I could write a funny story involving with each of these guys, but this dissertation would become a real book.

Finally, I thank from the heart to my girlfriend Carolina, who has all these years being there for me, patiently, supporting me when things got tough. She is the only one to know how many days and nights were needed to reach this point. Carol, looking back and realizing that my growth as a person and as a professional is directly associated to the support you have been providing me, makes me want to stand by your side forever. Thank you!

AGRADECIMENTOS

Primeiramente, devo meus sinceros agradecimentos aos meus pais e irmão pelo apoio incondicional e pelo exemplo de garra e perseverança que sempre foram para mim. Se hoje cumpro mais essa etapa com sucesso, tanto profissional quanto pessoal, isso se deve muito ao fato de eles sempre acreditarem no meu potencial e me incentivarem para seguir adiante. Ao vô Araujo e a vó Carmen, agradeço por me ensinarem os reais valores dessa vida. Desde criança, quando eu ainda tirava férias e as desfrutava com eles em Pelotas, aprendi pequenas coisas como respeito, honestidade e humildade, que me fizeram ser hoje uma pessoa melhor.

Agradeço também ao meu orientador Lisandro, que apesar da distância, sempre esteve incrivelmente disponível (*i.e.*, *on-line*) nas horas que eu mais precisei. Lisandro, tenha certeza de que tu foste um excelente professor e também um ótimo amigo. Hoje percebo a sorte que tive ao cair de paraquedas no Grupo de Redes da UFRGS. Não poderia deixar de agradecer também ao meu pseudo-coorientador extraoficial Luciano, que sempre me depositou muita confiança e também foi um grande parceiro nas nossas empreitadas (*deadlines*, viagens, filmagens constrangedoras, etc.).

À "cambada" do projeto da HP agradeço pelo companheirismo, bom humor, pelas intermináveis discussões filosóficas (afinal, para que gerenciar riscos num sistema com *rollback* automático?) e pelas madrugadas de *deadline* no campus (nós, os guardas e os cachorros). Em especial, ao meu grande amigo "Rubão", companheiro desbravador da Europa e do Japão (conhecendo apenas uma expressão em japonês, *biru mutso*); ao Weverton (*aka* Mocarongo) e ao Machado, meus veteranos aos quais tanto incomodei até aprender alguma coisa sobre gerenciamento de mudanças; aos meus bolsistas preferidos, Luis (*aka* Nautilus) o ilustre criador do *bot* do café, Fabrício o último torcedor do Juventude em atividade, e o Alan (*aka* Pirulito), só quem já foi bolsista ou estagiário sabe o trabalho que dá; ao Ricardo e ao Bruno, que se juntaram mais tarde ao grupo, mas se encaixaram como se já nos conhecêssemos há anos; aos nossos gerentes Cristiano e Lincoln, que sempre deram um jeito de organizar essa nossa bagunça (e não foi nada fácil eu garanto).

Agradeço ao restante dos membros do Grupo de Redes, Flávio "Barata", Paulo "Priumo", Rafael SBZ, Clarissa (*é brother*), Raniery, Jéferson e demais integrantes, pela amizade e pelas batalhas campais travadas nos eventos organizados por este seletto grupo do qual me orgulho de fazer parte. Gostaria de escrever apenas uma história divertida que vivi com cada um, mas esta dissertação viraria um livro.

Por fim, devo agradecer de todo o meu coração à minha namorada Carolina, que durante todos esses anos sempre esteve ao meu lado, pacientemente, me apoiando nos momentos mais difíceis. Só ela sabe quantas noites mal dormidas foram necessárias para chegar até este momento. Carol, olhar para trás e perceber que o meu crescimento como pessoa e como profissional está diretamente associado ao teu apoio, me faz desejar continuar ao teu lado para sempre. Muito obrigado!

CONTENTS

LIST OF ABBREVIATIONS AND ACRONYMS	8
LIST OF FIGURES	10
LIST OF TABLES	11
ABSTRACT	12
RESUMO	13
1 INTRODUCTION	14
2 RELATED WORK	17
2.1 Risk Management Framework by M_o_R	18
2.2 Risk Management under IT Change Management Perspective	21
2.3 Risk Management under IT Project Management Perspective	23
3 BACKGROUND	25
3.1 IT Change Management According to ITIL	25
3.2 Project Risk Management as Envisioned by PMBOK	28
4 RISK ASSESSMENT FRAMEWORK	31
4.1 Workflow Information Model	31
4.2 Log Records Information Model	32
4.3 Risk Analyzer Framework Overview	34
4.4 Similarity Calculation	35
4.5 Probability Estimation	36
4.6 Impact Estimation	37
4.7 Risk Classification	40
4.8 Risk Summarization	42
5 EVALUATION	45
5.1 Application to IT Change Management	45
5.1.1 Failure Classification	45
5.1.2 Request for Change Information Model	47
5.1.3 Implementation	49
5.1.4 Scenario and Results	50
5.2 Application to IT Project Management	53
5.2.1 IT Project Life Cycle Information Model	53
5.2.2 Hypothetical Project Structure	55

5.2.3	Comprehensive Risk Reports	56
6	CONCLUSION	58
6.1	Main Contributions and Results Obtained	59
6.2	Final Remarks and Future Work	60
	REFERENCES	61
APPENDIX A	PUBLISHED PAPER – IM 2009	65
APPENDIX B	PUBLISHED PAPER – SBRC 2009	74
APPENDIX C	PUBLISHED PAPER – DSOM 2009	89
APPENDIX D	PUBLISHED PAPER – NOMS 2010	104

LIST OF ABBREVIATIONS AND ACRONYMS

AF	Activity Failure
AMN	Activity Modeling Notation
AR	Absolute Relevance
BN	Bayesian Network
BPEL	Business Process Execution Language
BsR	Business Relevance
BTO	Business Technology Optimization
CAB	Change Advisory Board
CHAMPS	Change Management with Planning and Scheduling
CI	Configuration Item
CIM	Common Information Model
CMS	Configuration Management System
COBIT	Control Objectives for Information and related Technologies
CP	Change Plan
CRUD	Create, Request, Update, and Delete
CV	Constraint Violation
DAO	Data Access Object
DML	Definitive Media Library
DMTF	Distributed Management Task Force
ET	External Trigger
GLM	Generalized Linear Model
HF	Human Failure
HP	Hewlett-Packard
IEEE	Institute of Electrical and Electronics Engineers
IRM	Institute of Risk Management
ISACA	Information Systems Audit and Control Association

ISO	International Organization for Standardization
IT	Information Technology
ITIL	Information Technology Infrastructure Library
IW	Influential Workflow
M_o_R	Management of Risk
NA	No Action
OASIS	Advancing Open Standards for the Information Society
OGC	Office of Government Commerce
PHP	PHP: Hypertext Processor
PMBOK	Project Management Body of Knowledge
PRM	Project Risk Management
PMI	Project Management Institute
RA	Risk Affinity
RF	Resource Failure
RFC	Request for Change
RM	Remediation
SD	Service Disruption
SNMP	Simple Network Management Protocol
SQL	Structured Query Language
TF	Time Failure
WfMC	Workflow Management Coalition
XP	Extreme Programming

LIST OF FIGURES

Figure 2.1: Risk Management processes according to M_o_R (OGC, 2007a)	18
Figure 3.1: Conceptual Architecture of a Change Management System	26
Figure 3.2: Project Risk Management processes according to PMBOK (PMI, 2004)	28
Figure 4.1: Extension from the Workflow Process Definition model	32
Figure 4.2: Information Model to Represent Execution Traces of Workflows .	33
Figure 4.3: Architecture of the Framework for Risk Assessment	34
Figure 5.1: Partial View of the Request for Change Information Model	48
Figure 5.2: Service Disruption Example	50
Figure 5.3: Change Plans for installation/configuration of development and production environments	51
Figure 5.4: IT Project Life Cycle Information Model	54
Figure 5.5: Comprehensive Risk Report in Project Hierarchy View	56
Figure 5.6: Comprehensive Risk Report in Work Plan View	57

LIST OF TABLES

Table 2.1:	Examples of Risk Probability Ranges	19
Table 2.2:	Examples of Risk Impact Ranges	19
Table 4.1:	Classification ranges for probability and impact	40
Table 4.2:	Risks Classification Matrix	41
Table 4.3:	Risks Classification Grid	41
Table 4.4:	Tabular Risk Report	42
Table 4.5:	Summarized Risk Reports	44
Table 5.1:	Risk Reports before the deployment of first phase	52
Table 5.2:	Risk Reports before the deployment of second phase	53

ABSTRACT

Products and services provided by modern organizations are usually designed, deployed, and supported by large-scale Information Technology (IT) infrastructures. In order to obtain the best performance out of provided products and services, it is essential that these organizations enforce rational practices for the management of resources that compose their infrastructures. For this purpose, in recent years a few standards and libraries of best practices for IT infrastructures and services management have been proposed. Among the most widely accepted proposals, in both academy and industry, is worth mentioning the Information Technology Infrastructure Library (ITIL). A common point in most of those standards and libraries is the explicit concern with the risks related to IT activities. Proactively dealing with adverse and favorable events that may arise during everyday operations might prevent, for example: delay on deployment of services, cost overrun in activities, predictable failures of handled resources, and, consequently waste of money.

Although important, risk management in practice usually lacks in automation and standardization in IT environments. Generally, it is performed by stakeholders in interviews and brainstorms, which may be a very time/resource-consuming task and sometimes too imprecise to guide risk related decisions. Therefore, in this dissertation, a framework to support the automation of some key phases of risk management is proposed, aiming to make it simpler, faster, and more accurate. The proposed framework is targeted to workflow-based IT management systems. The main approach is to learn from problems reported in the history of previously conducted workflows in order to estimate risks for future executions. Furthermore, comprehensive and interactive risk reports are proposed aiming to ease the analysis of assessed risks by involved humans.

The proposed framework had its applicability evaluated in two case studies both in IT related areas, namely: IT Change Management and IT Project Management. The results show how the framework is not only useful to speed up the risk assessment process, but also to assist the decision making of project managers and IT operators by organizing risk detailed information in a comprehensive way. In addition, the modular approach employed in the design of the proposed framework allows it to be generic enough to fit in different contexts (changes and projects) and still customizable to adapt to more specific requirements.

Keywords: Risk assessment, risk management, change management, project management.

Um Framework para Estimativa de Riscos Baseado na Análise de Informações Históricas de Sistemas de TI

RESUMO

Produtos e serviços oferecidos pelas organizações modernas são geralmente projetados, implantados e mantidos por meio de infraestruturas de Tecnologia da Informação (TI) de grande escala. A fim de obter o melhor desempenho dos produtos e serviços oferecidos, é essencial que essas organizações façam uso de práticas adequadas para o gerenciamento dos recursos que compõem as tais infraestruturas. Para esse fim, foram propostos recentemente alguns padrões e bibliotecas de melhores práticas para o gerenciamento de infraestruturas e serviços de TI. Entre as mais amplamente aceitas, tanto da academia quanto na indústria, vale destacar o Information Technology Infrastructure Library (ITIL). Um ponto comum na maioria desses padrões e bibliotecas é a preocupação explícita com os riscos relacionados as atividades de TI. Lidar proativamente com eventos adversos e favoráveis que possam surgir durante as operações cotidianas pode evitar, por exemplo: atrasos na a implantação de serviços, extrapolação no custo de atividades, falhas previsíveis nos recursos manipulados, e conseqüentemente, desperdício de dinheiro.

Apesar da sua importância, o gerenciamento de riscos em ambientes de TI, na prática, geralmente sofre pela falta de automação e padronização. Habitualmente, tal gerenciamento é realizado pelos participantes das atividades em entrevistas e *brainstorms*, o que geralmente acaba consumindo tempo/recursos de forma excessiva e sendo muito impreciso para guiar a tomada de decisão. Portanto, nesta dissertação, um *framework* para dar suporte a automação de algumas fases cruciais do gerenciamento de riscos é proposto com o objetivo de tornar tal gerenciamento mais simples, rápido e preciso. O *framework* proposto é direcionado para sistemas de gerenciamento baseados em *workflow*. A ideia básica é aprender a partir de problemas relatados em *workflows* executados no passado, a fim de estimar os riscos para execuções futuras. Além disso, são propostos relatórios de risco compreensivos e interativos com o objetivo de facilitar a análise dos mesmos pelos humanos envolvidos.

O *framework* proposto teve sua aplicabilidade avaliada em dois estudos de caso em áreas relacionadas à TI, a saber: Gerenciamento de Mudanças e Gerenciamento de Projetos de TI. Os resultados demonstram como o *framework* pode ser útil não apenas para acelerar o processo de estimativa de riscos, mas também para auxiliar a tomada de decisão dos gerentes de projetos e operadores de TI, ao organizar as informações relacionadas aos riscos de forma detalhada e ao mesmo tempo compreensiva. Além disso, a abordagem modular empregada na concepção do *framework* proposto permite que este seja genérico o suficiente para ser utilizado em diferentes contextos (mudanças e projetos) e, ainda assim, personalizável para se adaptar às exigências mais específicas de cada contexto.

Palavras-chave: Estimativa de Riscos, Gerenciamento de Riscos, Gerenciamento de Mudanças, Gerenciamento de Projetos.

1 INTRODUCTION

Modern organizations that want to deliver high quality services to their internal and external customers often end up employing large-scale Information Technology (IT) infrastructures. As services are designed, deployed, maintained, and improved, these IT infrastructures become more complex in term of both management and scalability. In order to obtain the best performance out of the provided services, avoiding waste of resources, it is essential that organizations enforce rational practices for the management of their IT infrastructures. For this end, some best practices standards and libraries have been published, aiming at providing guidance for proper management of IT infrastructures and services. Two of the most widely recognized guides are the Information Technology Infrastructure Library (ITIL) (OGC, 2010) – proposed by the British Office of Government Commerce (OGC) – and the Control Objectives for Information and related Technologies (COBIT) (ISACA, 2010) – introduced by the North American Information Systems Audit and Control Association (ISACA).

One key aspect about these guides of best practices is their concern on the necessity of managing the risks within organizations' IT activities. This is emphasized by the fact that both OGC and ISACA have published specific guides for corporative IT risk management, respectively, the Management of Risk (M_o_R) (OGC, 2007a) and the Risk IT (ISACA, 2009). According to M_o_R, to achieve their objectives organizations have to inevitably take a certain amount of risk. Thus, it is the role of Risk Management to help organizations to methodologically deal with the risks associated with their activities.

Commonly, organizations face risks as uncertain future events or conditions that, if happen, might affect the accomplishment of business goals. These events that represent risk to the business should be identified and assessed in terms of probability of occurrence and possible impact to the business objectives. Although the literature recommends tackling risks as both negative (threats) and positive (opportunities) aspects, in practice, the negative side is far more considered, mainly in fields such as safety, construction, and health care. The actual result is that risk management becomes strongly focused on the prevention and mitigation of harm. This observation also holds in the investigations on risks associated to the design and operation of computational systems.

Generally speaking, risk management is divided into four logically sequential and cyclic processes or phases (OGC, 2007a): Identification, Assessment, Response Planning, and Implementation. The first process is focused in the definition of the context of risk management for the activity of the organization being analyzed and identification of possible threats and opportunities to the objectives of this activity. The risk assessment process takes place when the previously indentified risks are

evaluated in terms of probability of occurrence and associated impact (*i.e.*, estimation of possible losses or earnings). Afterwards, based on prioritizations guided by risk assessment, preventive and reactive responses to risks are defined aiming to minimize threats and enhance opportunities. Finally, in the last phase, planned responses are implemented and risks are continuously monitored and controlled in order to evaluate the effectiveness of preventive actions and occasionally dispatching corrective ones.

Along with all these processes, it is important that organizations adopt a common set of internal policies and strategies for risk management to be shared throughout their departments and teams. Some definitions such as tolerance thresholds, roles and responsibilities, scales for estimating probabilities and impact, tools for documenting, reporting, and communication of risks should be common on all organization's activities.

In spite of all these best practices for IT infrastructures and services management, experience of practitioners shows that there is still little evidence that risk management is being applied efficiently as a systematic and repeatable process. Some authors have investigated the actual benefits and shortcomings of different approaches to risk management in real life environments (WYK; BOWEN; AKINTOYE, 2008; BAKKER; BOONSTRA; WORTMANN, 2009; KUTSCH; HALL, 2010). These researches point out many issues found in companies' processes, such as: inadequate documentation about identified risks, making it more difficult to reuse the knowledge acquired; lack of tools for automation in order to assist in reporting, monitoring, and decision making about risks; quality of risk-related decisions is often too much dependent on the experience of stakeholders; and usually risk management processes involve an excessive number of people becoming a time/resource consuming task and sometimes being even counterproductive.

One of the major problems in risk management is the lack of automation enforcement or system-assisted routines. Risk assessment, which is a key process where risks are evaluated in terms of probability and possible impact, for instance, is usually performed in interviews and meetings with stakeholders involved in each activity. In practice, there is little reuse of experiences that have been learned and documented in the past, then the accuracy of estimated values is subject to the empirical knowledge of the involved staff. Moreover, some critical activities, such as incident response, cannot wait for a committee to meet and deliberate about the risks in adopting one or another strategy to solve a problem, because of the immediacy that these activities require. Specially in the management of IT systems and services, where it is already possible to achieve high levels of automation in tasks like configuration, control, and logging, it is certainly possible to also automate procedures or create system-assisted solutions for risk management.

In order to address these issues, in this master's dissertation it is introduced a framework to support the automation of some key phases of risk management, aiming to make it simpler, faster, and more accurate. The target is mainly in risk assessment for workflow-based systems designed for the management of IT infrastructures and services. There are many types of IT management processes that can be modeled in the form of workflows, such as: change management, project management, and incident management. The advantage of using workflows lies in the fact that they define a sequence of fine-grained activities to be executed in a given order, sequential or parallel, and the details of execution of these activities

(including adverse and favorable event reports) may be recorded to a log. The proposed approach to the problem encompasses the investigation of execution records of workflows previously performed, trying to learn from events reported in the past, aiming to help in the design of better workflows for future executions.

Besides the proposed framework itself, some other contributions are outcome of this dissertation, as follows:

1. Firstly, the segregation of events that represent risks following risk classifications that vary depending on the environment being analyzed is discussed. It is important to group events together in such a way that reflects the concerns of each environment and assures that the results of automated risk assessment are meaningful to human managers;
2. Moreover, considering that workflows used in IT management activities are constantly changing, a strategy to compute the similarity among these workflows and to enable the reuse knowledge – even if previous workflows were slightly different – is introduced;
3. Algorithms to allow automated estimation of probability and impact of events based on information retrieved from previously executed and documented workflows are also proposed;
4. Finally, aiming to help in the decision making during risk response planning, comprehensive and interactive risk reports are presented. These reports are organized according to a widely used risk classification and, in addition, a strategy to summarize risk information in different level of details is proposed.

In order to prove concept of the solution proposed in this dissertation, two IT related areas have been selected to be used as case studies, namely: IT Change Management and IT Project Management. The former provides general guidelines for consistently and safely conducting changes over IT infrastructures, from the early specification, planning, deployment, and finally evaluation and review (OGC, 2007b). On the other hand, IT Project Management is focused in the design phase of services aiming to ensure that a project meets its objectives avoiding waste of resources (OGC, 2007c; PMI, 2004). Both projects and changes can be organized in the form of workflows and therefore may have their risks analyzed using the framework proposed in this thesis.

The remainder of this dissertation is organized as follows. In Chapter 2 a review of the available literature on risk management specially related to IT Change Management and IT Project Management is presented. Afterwards, in Chapter 3, some background concepts that are fundamental to understand and motivate the proposal of the framework to automate risk assessment are presented. The conceptual framework itself, as well as algorithms used for impact and probability estimation and strategies for calculating similarity among workflows and risk summarization are introduced in Chapter 4. In Chapter 5 a discussion about results obtained during evaluation of both case studies is presented. Finally, Chapter 6 discusses conclusions, final remarks, and future work.

2 RELATED WORK

Risk management is a cross-discipline that is investigated and employed in several different fields. Risk assessment principles, for example, may be valuable for guiding financial investments (FROOT; SCHARFSTEIN; STEIN, 1993), health care decisions (DANA EI et al., 2005), and strategies of insurance companies (KLÜPPELBERG; KOSTADINOVA, 2008). The literature provides many definitions of what risk is and how it should be managed. For example, Holton (HOLTON, 2003) explains that risk denotes an uncertain event that will affect elements, and may occur in some present or future process. Chicken and Posner (CHICKEN; POSNER, 1998) have conducted a wide research on how people deal with risks across many different areas, including: financial, medical, industry, projects, transport, and sports. In their book named *The Philosophy of Risk*, the authors define that risks should be assessed as a composition of two factors: (i) probability of occurrence of a possibly negative event and (ii) how the object of analysis is affected by this event. This definition seems to be commonsense among the majority of risk management guides and frameworks, especially in regards to corporative risk management.

Nowadays, there are at least four main frameworks and standards for risk management that are well recognized and employed by organizations of any kind. Two of them have been already cited in this dissertation; they are the Management of Risk (M_o_R) from OGC (OGC, 2007a) and the Risk IT from ICASA (ISACA, 2009). Both of them are targeted to risk management for organizations' IT processes. Another two more general purpose standards are worth mentioning, the Risk Management Standard introduced by The Institute of Risk Management (IRM) (IRM, 2002) and the recently released ISO 31000:2009 Risk management – Principles and Guidelines presented by the International Organization for Standardization (ISO) (ISO, 2009).

The basic definitions of how risk management should be properly conducted are quite similar throughout the mentioned frameworks and standards. In this dissertation, the M_o_R framework was selected to serve as guide for the theoretical study since it is well connected to ITIL, which in turn is widely employed by organizations in order to rationally arrange IT infrastructures and services. Therefore, in this chapter, in a first moment, the risk management framework as defined in M_o_R is detailed. Subsequently, some of the most relevant related work lying on the two areas chosen for the case studies – *i.e.*, IT Change Management and IT Project Management – are presented.

2.1 Risk Management Framework by M_o_R

The OGC has published the Management of Risk (M_o_R) framework aiming to help organizations taking decisions in regards to the risks that might affect the performance of any of their activities. This framework is based on four core concepts as depicted in Figure 2.1: M_o_R Principles, M_o_R Approach, M_o_R Processes, and M_o_R Embed and Review.

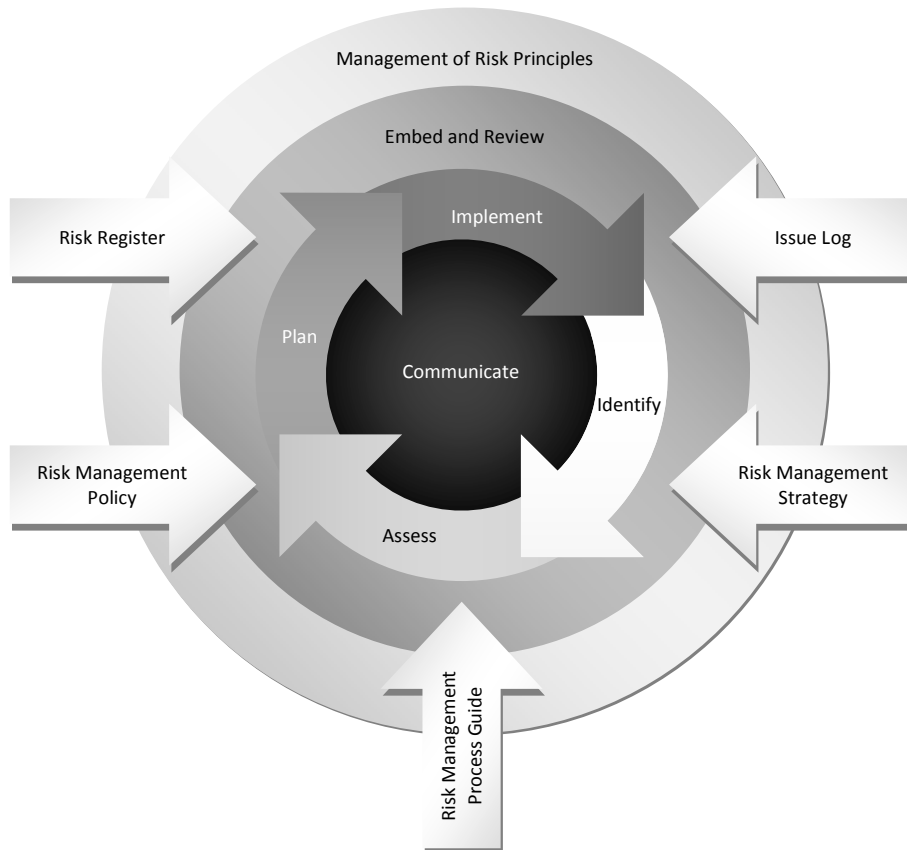


Figure 2.1: Risk Management processes according to M_o_R (OGC, 2007a)

M_o_R Principles: it is important that organizations define principles for risk management as high level directions and guidelines to be applicable for all activities and to enforce the management of risk as a common practice. Examples of principles that should be defined are: organizational context and objectives, stakeholder involvement, roles and responsibilities, early warning indicators, risk reporting, and support structure.

M_o_R Approach: the aforementioned principles have to be customized to be adopted in such a way that suits the activities of each organization. Basically, the defined approaches provide the basis on which risk management practices can be developed. In order to communicate these practices and the way they have to be implemented, five main documents have to be released (represented by five outer arrows in Figure 2.1):

- **Risk Management Policy:** describes how risk management will be implemented throughout the organization or in each specific department;

- **Risk Management Process Guide:** defines a sequence of steps to be followed in order to identify, measure, and treat risks associated with a given activity;
- **Risk Management Strategy:** specifies risk management procedures and parameters for each particular organizational activity. This includes definition of ranges for estimating probabilities and impacts, budgeting or allocation of man-hours for risk management, specific tools and techniques to be employed, and response categories (*e.g.*, reduction, removal, transfer, retention, and share). Tables 2.1 and 2.2 present examples of possible ranges that may be used to estimate probabilities and impacts in risk management;

Table 2.1: Examples of Risk Probability Ranges

Probability Range	Statistical Criteria	Likelihood	Numerical Equivalents	Time Periods
Very High	> 75%	Almost certainly will occur	Less than 1 chance in 10^2	Likely to occur once within three months
High	51 – 75%	More likely to occur than not	Less than 1 chance in 10^3	Likely to occur once within one year
Medium	26 – 50%	Fairly likely to occur	Less than 1 chance in 10^4	Likely to occur once within ten years
Low	6 – 25%	Unlikely to occur	Less than 1 chance in 10^5	Unlikely to occur within ten years
Very Low	0 – 5%	Extremely unlikely	One chance in 10^6	Unlikely to occur within fifty years

Table 2.2: Examples of Risk Impact Ranges

Impact Range	Cost	Time	Requirements
Very High	> £750k	> 25 days	Major shortfall in any of the critical requirements
High	£500k £750k	20 days 25 days	Shortfall in any of the critical requirements
Medium	£250k £500k	10 days 20 days	Shortfall in multiple requirements
Low	£50k £250k	5 days 10 days	Shortfall in ancillary requirements
Very Low	< £50k	< 5 days	Minor shortfall in ancillary requirements

- **Risk Register:** organizes risk related information across all risk management processes of a given organizational activity. It is the purpose of the Risk Register to capture and maintain information about identified threats and opportunities, results of risk assessment, and all kind of information to enable risk response planning and subsequent control of risks;
- **Issue Logs:** maintain in a structured manner information regarding the identified issues that have actually occurred and might require response actions. These logs have to be coordinated with the Risk Register and usually will

describe risks that have been assessed as possible events and turned into real events.

M_o_R Processes: as mentioned before in this dissertation, risk management according to the M_o_R framework is divided into four logically sequential and cyclic processes. These processes describe activities, inputs, and outputs responsible for ensuring that risks are identified, assessed, and controlled in each organizational activity. Also, it is important to communicate risk related information and decisions among involved personnel. The four processes and the communication activity are presented in the center of Figure 2.1 by four rounding arrows plus the Communicate circle:

- **Identify:** in a first moment of this process one should collect information about the context of the activity. This includes, for example, understanding the main objectives, scope limitations, and primary assumptions made. Afterwards, the target is to identify the risks that might either reduce the likelihood of the organization reaching its objectives or enhance performance an organizational activity;
- **Assess:** the main objective of this process is to assess the threats and opportunities previously identified and estimate their associated probabilities and impacts. In real life, this is usually performed by humans involved in the activity (*e.g.*, managers, operators, developers, testers) with no proper support of any automated method or tool; therefore the quality of risk assessment is subject to the experience of responsible personnel. More specific evaluations are recommended only for the most risky events. These evaluations might be obtained through experiments, simulations, or analytical models, which are commonly expensive and time consuming tasks to perform;
- **Plan:** this process is focused on the planning of specific management responses for the threats and opportunities previously identified and assessed. Plans developed are ideally intended to remove or attenuate likeliness and effects of threats and to enhance possible gains or earning from opportunities. Responses for threats are usually classified in reduction, removal, transfer, retention, and share, while classifications of responses for opportunities are realization, enhancement, and exploitation;
- **Implement:** in this process planned risk response actions are implemented in order to guarantee that risks will be effectively dealt in a proactive way. Besides, monitoring procedures have to be started aiming to watch for possible risk events to materialize and also to evaluate efficiency of preventive actions in practice. Corrective actions might have to be triggered in the case responses do not meet expectations;
- **Communicate:** in fact, communication does not take place before or after any of the aforementioned processes. It is indeed an activity that is carried out all the time when performing risk management. Communication plays a key role in risk management since it enforces the participation and engagement of staff across the organization. Effective communication allows easy identification of new threats and opportunities and may also change already identified and assessed risks.

M_o_R Embed and Review: the main objectives of embedding and reviewing management of risks are: *(i)* to introduce the need for risk management in the culture of the organization, *(ii)* to explain how proper risk management can be naturally adopted in all organizational activities, and *(iii)* to enforce the importance of regularly reviewing risk management procedures to ensure that they are being conducted appropriately and successfully across the organization.

As presented in this section, M_o_R provides a complete framework for risk management ranging from very high level guidelines to detailed procedures intended to deal with favorable and adverse events in organizational activities. Among the four processes discussed, the framework proposed in this dissertation is focused specifically in the assessment of risks, aiming to automate estimations of probabilities and impacts in order to help in this process that is usually human performed and time/resource consuming.

2.2 Risk Management under IT Change Management Perspective

The first area upon which the case studies presented in this dissertation are inspired is IT Change Management. In this particular area risk management may become very interesting to enable proactive problem treatment. Since commonly large scale IT infrastructures support services that are essential for business continuity, whenever changes to any of these services are required, risks of many different kinds may arise. According to ITIL, risks should be investigated, measured, and treated before any change is approved (OGC, 2007b). In this dissertation, events seen as risks are possible failures that may happen during the deployment of a change (*e.g.*, failure on the installation of new software, interference of agents external to the changing process, or damage of hardware elements being handled) causing disruption, directly or indirectly, to one or more services provided by the organization.

Before starting with a method proposal for the estimation of possible risks in the context of IT Change Management, it is important to better understand what types of failures should be expected during the execution of changes and how they can be classified. Classification of risky events is interesting to generate more comprehensive results out of risk analysis. Rather than observing isolated events, it may be more intuitive for humans to analyze and draw decisions on risk information grouped or categorized according to a certain classification (more discussions on the comprehensiveness of risk representations are presented in Section 5.2).

Several researches are available on the literature of failure or error representation. Wang *et al.* (WANG; SAHAI; PRUYNE, 2006) explain that there are four requirements to compose a model to well represent errors: *(i)* error categories and hierarchy should be represented, *(ii)* error models should be integrated without modifying models for existing components, *(iii)* component-specific error behaviors should be captured, and *(iv)* error propagations should be handled. Furthermore, the authors propose that an *Error* class may be specialized into two subclasses *Hardware Error* and *Software Error*. Each subclass is still partitioned into categories, respectively: persistent defect, non-persistent defect, performance degradation, and partial failure, for *Hardware Error*; and persistent error/bug, non-persistent error/bug, performance degradation, race condition/timing error/deadlock, and configuration/administration error, for *Software Error*.

Russell *et al.* (RUSSELL; AALST; HOFSTEDE, 2006) have proposed a conceptual framework for classifying the exception handling capabilities of workflow systems and process-aware information systems. The authors classify exceptions into five categories: Work Item Failure, Deadline Expiry, Resource Unavailability, External Trigger, and Constraint Violation. Furthermore, they have proposed strategies to handle exceptions under three considerations: how to handle exceptions on the work item level, how one exception affects other items of the same case, and what kind of recovery actions can be triggered to remediate the situation.

Focusing on Request for Change (RFC) definitions, Keller *et al.* (KELLER et al., 2004; KELLER, 2005) have introduced CHAMPS, which is a system for automating the planning of changes in IT environments. CHAMPS itself did not explicitly address risks because the system assumed that failures would not happen while performing changes in IT systems. However, the authors' formalization of IT changes allowed further advancements. Aiming to deal with failures during change deployment, Machado *et al.* (MACHADO et al., 2008, 2009) proposed a solution that treats change failures in a reactive fashion, undoing the requested changes over a damaged system backwards to its previous consistent state. In spite of the advances, a solution that proactively observes risks to avoid future (and potentially expensive) system rollbacks is still lacking.

Sauvé *et al.* (SAUVÉ et al., 2007) and Rebouças *et al.* (REBOUÇAS et al., 2007) have proposed a solution for risk analysis on the IT Change Management process aiming to automatically determine priorities on the scheduling of various possibly concurrent RFCs. In those researches, it is employed a risk evaluation guided by the business objectives, in order to minimize the impact over the organization's services during the deployment of changes. According to the authors, the elapsed between the submission of an RFC and its implementation causes damage to the services affected by the change, which may suffer from performance degradation or missing functionalities. Moreover, during the deployment of an RFC, the disruption of services and breach of deadlines may cause financial losses or contractual penalties. However, risk analysis proposed in these works has application to the scheduling phase of IT Change Management. In this dissertation, the proposed methods for risk assessment are applicable during the planning of changes, and the objective is to predict likelihood of failures and possible disruption to services.

Also dealing with scheduling of RFCs, Setzer *et al.* (SETZER; BHATTACHARYA; LUDWIG, 2008) have modeled the resources (*e.g.*, hardware, software, and services) of an IT infrastructure as a network of interconnected services. Based on this network, they derive models for analyzing the business impact of change related service downtimes with uncertain length and convert these downtimes into actual financial losses. Employing analytical models, the authors enable decision support for scheduling of single or multiple correlated changes. As well as Sauvé and Rebouça's work, this research focuses on the scheduling phase of IT Change Management; however in this work the authors consider uncertainty in change durations.

Another interesting work was conducted by Marques and Neves-Silva (MARQUES; NEVES-SILVA, 2007) and does not lie in the area of IT Change Management, but deals with probabilities and impacts of adverse events to assess risks. These authors have proposed a method for risk assessment to help in the decision making on complex assembly lines. They propose to compute risks – in terms of both probability and impact of possible incidents – considering information collected

during the system operation. This method was designed to run in an environment where the required parameters for calculating incident's probability and impact have well known values, for a limited set of possible events. For example, when an alarm is fired to indicate that some system variable (*e.g.*, mean time-between-failures) overtook a regular threshold, there are values previously defined for probabilities and impacts of incidents associated to this alarm. In IT Change Management, however, because of the dynamics of IT environments, the amount and diversity of incidents that can happen is likely uncountable. Therefore, solutions able to cope with such diversity are still required.

2.3 Risk Management under IT Project Management Perspective

Earlier in this chapter, four standards and frameworks for risk management were mentioned, namely: Management of Risk (M_o_R), Risk IT, Risk Management Standard, and ISO 31000:2009. More focused in the context of IT projects, there is the Guide to the Project Management Body of Knowledge (PMBOK) introduced by the Project Management Institute (PMI) (PMI, 2004). In order to deal with risks in IT projects, one of the nine so called knowledge areas from PMBOK is focused specifically on Project Risk Management (PRM). The objectives of PRM are: (*i*) to increase the probability and impact of positive events, and (*ii*) to decrease the probability and impact of events adverse to the project. Similarly to the processes presented by the M_o_R framework, the PMBOK divides risk management into six processes, further detailed in Chapter 3: Risk Management Planning, Risk Identification, Quantitative Risk Analysis, Qualitative Risk Analysis, Risk Response Planning, and Risk Monitoring and Control.

As with IT Change Management, also it is important to classify risks in IT Project Management in order to create more readable reports. According to the PMBOK, risks in projects can be faced as events that, if happen, may have positive or negative effects in at least one project objective. These objectives might change according to project's needs. However, there are four objectives are commonly considered in projects and might be used to classify risky events, they are: cost, time, scope, and quality.

Despite the current risk support proposed in the aforementioned frameworks and standards, the adoption of formal procedures in actual projects still demands too much effort, experience, and ability of managers and stakeholders to produce useful results. Kutsch and Hall (KUTSCH; HALL, 2010) have investigated the reasons why IT project managers decide whether or not certain identified risks should be considered relevant against project objectives. By interviewing managers from different IT projects, the authors perceived that behavioral factors influence manager's decisions; therefore the success of risk management is conditioned to their experience. Indeed, when the project manager does not have sufficient experience to effectively prioritize risks, project risk management seems to have little impact on project outcomes, being sometimes even counterproductive. Wyk *et al.* (WYK; BOWEN; AKINTOYE, 2008) have evaluated the risk management methods of a large electricity supplier in South Africa. Although the analyzed company employs best practices for risk management, risk identification, analysis, mitigation, monitoring, and reporting are performed employing no automated tool. As a consequence, the company ends up

involving an excessive number of stakeholders in risk management process. In addition, there is lack of common practices across various divisions, which turns the reuse of knowledge internally to the company more complex.

In order to aid humans in risk management, the automation of certain steps of this process – such as data gathering for risk assessment – could, for example, potentially reduce the time and cost, while increasing the reliability of results. Some authors have employed probabilistic models to predict undesired events as well as estimate metrics for risk management in IT projects. Fewster and Mendes (FEWSTER; MENDES, 2001) have introduced a prediction model using a Generalized Linear Model (GLM) to estimate some Web design and authoring metrics. The paper focuses on the prediction of the effort to build a Web project. Nevertheless, the same GLM has shown to be a powerful tool to create a framework for risk management. The most interesting fact is that the statistical model provides not only a point for the analyzed variable, but a full probability distribution. Instead of estimating the total time for the project execution, it is possible to obtain the probability of not concluding it in, for example, 30 days. With that in mind, the project manager is able to determine a required risk level he/she is willing to deal with.

Bayesian Networks (BNs) have been used in many investigations for similar purposes. Hearty *et al.* (HEARTY *et al.*, 2009), in turn, have designed a model for effort prediction and risk assessment in software development projects that follow the Extreme Programming (XP) methodology. The author's approach is based on the use of Bayesian Networks (BNs), and quantitatively estimates project metrics (*e.g.*, iterations/time to complete) without requiring data about the success of past XP projects. Fenton and Neil (FENTON; OHLSSON, 2000), on the other hand, have applied BNs to predict software defects, while Luu *et al.* (LUU *et al.*, 2009) employ it to estimate the likelihood of time-overrun in construction projects. These works have contributed to the automation of risk assessment and, although relevant, these researches have only considered risks in terms of the probability of occurrence of adverse events; the severity of the impacts that such events might have on the affected projects or businesses has not been taken into account.

3 BACKGROUND

In this chapter, background concepts are presented in order to provide an overview of the areas that have been chosen for the case studies conducted in this research. These concepts are also important to better understand the current shortcoming in risk assessment methods and to motivate, using two real scenarios, the development of a framework to automate risk assessment.

In a first moment, IT Change Management, as envisaged by ITIL Service Transition book and materialized in a prototypical system called CHANGELEDGE, is discussed in Section 3.1. This system has been developed inside the Computer Networks Group of UFRGS and is targeted on the automation of planning and deployment of changes over IT infrastructures. In Section 3.2, definitions of Project Risk Management processes, as proposed by PMI within the PMBOK publication, are presented. In addition, some barriers to the adoption of such processes for risk management in real life projects are also discussed.

3.1 IT Change Management According to ITIL

As mentioned before, being one of the core processes of ITIL, IT Change Management (OGC, 2007b) provides general guidelines for conducting changes over IT infrastructures, from the early specification to the final deployment and evaluation. It defines that all changes should be described in a document called Request for Change (RFC). An RFC specifies, in a declarative way, what must be done and the primary Configuration Items (CIs) affected (*e.g.*, devices, applications, services), but it does not detail how the change should be implemented. In fact, this must be performed by human operators either manually or aided by an automated management system. In addition, RFCs must be reviewed, approved, and scheduled by a Change Advisory Board (CAB). The CAB, usually chaired by a change manager, should be composed of people with extensive knowledge on the organization's processes, often coming from different areas, but not necessarily familiar with the underlying technologies deployed in the IT infrastructure.

A solution to conduct the change process, ranging from the change specification and planning to its deployment, was proposed in previous work (CORDEIRO et al., 2008; MACHADO et al., 2008; CORDEIRO et al., 2009). The components that are part of the conceptual architecture of the change management solution are presented in Figure 3.1. This conceptual architecture was materialized in the CHANGELEDGE system.

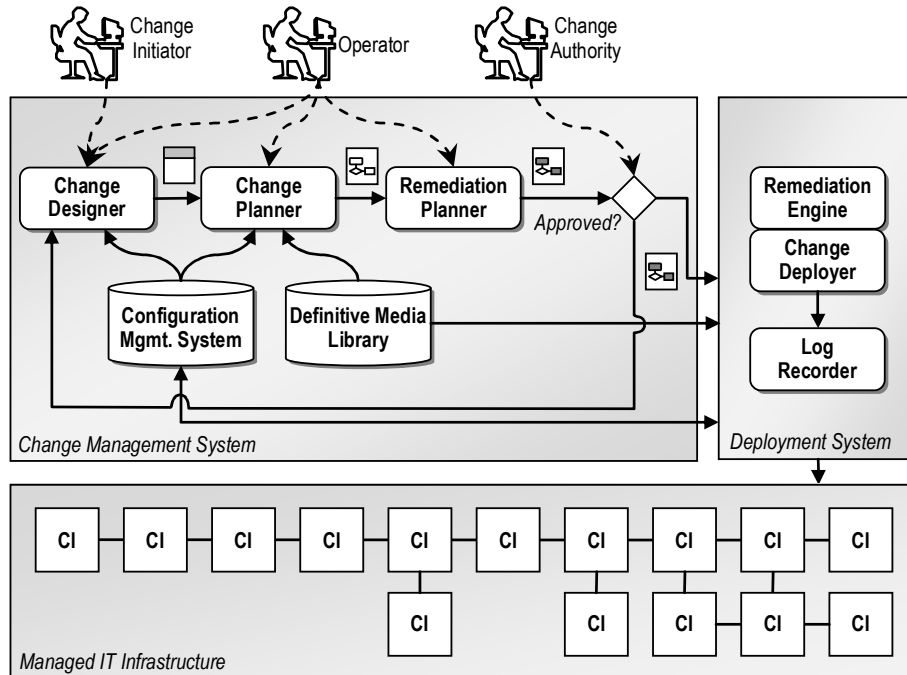


Figure 3.1: Conceptual Architecture of a Change Management System

The *Change Initiator* starts the change process by describing an RFC interacting with the *Change Designer* component. The Configuration Management System (CMS) provides the *Change Initiator* with an updated view of the IT infrastructure and services. Once the RFC is specified, the *Operator* is responsible for designing a preliminary Change Plan (CP) (a workflow of high level activities), also interacting with the *Change Designer*. On the second step of the life cycle of a change, the *Change Planner* is responsible for producing an actionable workflow of finer-grained activities, based on definitions made in the preliminary plan and also gathering information about configuration and software dependencies provided by the *Definitive Media Library* (DML) and the current state of the IT infrastructure (provided by the CMS). The algorithm to generate such refined CP is out of the scope of this dissertation. The interested reader may refer to Cordeiro *et al.*'s work (CORDEIRO *et al.*, 2008).

Each activity of a CP is described adhering to the *Activity Modeling Notation* (AMN) (CORDEIRO *et al.*, 2009). This notation is used to enable the system to identify the elements involved in each operation of the change process and to guarantee that activities are specified avoiding ambiguity. Examples of sentences written in AMN are:

1. install SoftwareElement <sw> at ComputerSystem <h> with <parameters>
2. start Service <sv> at ComputerSystem <h>
3. create DataFile <f> in Directory <d> at ComputerSystem <h>

Example 1 is used to install a given software at a computer system, where `install SoftwareElement <sw>` refers to the software being installed and `at ComputerSystem <h>` indicates the target computational system where the software will be hosted in. Additional parameters for the installation might be specified in the `with <parameters>`

directive. Similarly, to start a given service at a computer system, the sentence shown in Example 2 is employed. Operations such as file manipulations are also possible with the AMN, as presented in the Example 3. All constructs of AMN manipulate elements from the IT infrastructure, logically represented in the CMS, or software packages available for installation retrieved from the DML. A whole set of sentences written in AMN have been presented by Cordeiro *et al.* (CORDEIRO *et al.*, 2009).

Following in IT Change Management process, once the refined CP is completely designed, the *Remediation Planner* automatically computes rollback plans based on remediation marks and groups defined by the *Operator*. Compensation plans may be also specified and attached to the RFC to be executed as an alternative if the primary plan fails. According to ITIL, the definition of remediation plans is required before any change is submitted for deployment. The main objective is to design plans to enable fast recovery of the IT infrastructure's consistency dealing with problems in changes in a reactive fashion. Support for rollback and compensation plans was also covered by previous work (MACHADO *et al.*, 2009) and will not be further discussed in this dissertation.

At this point, an RFC would be ready to be approved by a *Change Authority* (usually in a CAB meeting), scheduled, and deployed. However, these changes may expose the IT services to unnecessary or unknown risks. ITIL recommends risks to be identified, assessed, and treated before any change is approved. Usually, in this context, risk management is conducted in brainstorming or meetings and it is much based on the operators' knowledge. Depending on the urgency of the change, sometimes it is not even possible to wait for a committee to meet and deliberate about the risks involved in a change. That is one reason why in this research a solution to automate risk assessment is proposed. The goal is to make it possible to quickly have an overview about the risks contained in a CP, learning from historical information of past failed changes, giving the operator the opportunity to rearrange a CP before submitting it for approval and subsequent deployment. More discussions on the proposed solution are presented in Chapter 4.

As soon as the RFC analysis and edition processes are completed, the *Change Deployer* will actually apply the changes over the IT infrastructure. The occurrence of failures in any activity during the deployment process will trigger a specific remediation plan. The selection and execution of such plan is responsibility of the *Remediation Engine*. Every time a CI is affected by a change implementation, it is one of the *Deployment System's* roles to update the information on the CMS. This is essential to assure that the CMS has always the latest vision of the IT infrastructure. The *Log Recorder* is responsible for tracing execution records for every change, including execution of rollback and compensation activities. When an operation is performed affecting any element, this component associates activities executed during the change process to the involved CIs. The status of the execution (success or failure) and failure classification are also stored on the CMS for further evaluation. This information is kept on the system to allow the review of every modification performed over a CI.

3.2 Project Risk Management as Envisioned by PMBOK

The PMBOK is a widely used reference of best practices for project management in both academia and industry. Currently, it is recognized by the Institute of Electrical and Electronics Engineers (IEEE) as a standard in its area (IEEE, 2004). The collection of processes and best practices in this guide are organized within nine knowledge areas (PMI, 2004): (i) Project Integration Management, (ii) Project Scope Management, (iii) Project Time Management, (iv) Project Cost Management, (v) Project Quality Management, (vi) Project Human Resource Management, (vii) Project Communications Management, (viii) Project Risk Management, and (ix) Project Procurement Management.

Specifically important in the context of this research, Project Risk Management is a knowledge area that comprises planning, identification, analysis, responses, and monitoring of risks that may affect project objectives. Very aligned with the guidelines previously presented from the M_o_R framework but more focused in the context of projects, PMBOK divides Project Risk Management into six processes, as shown in darker boxes of Figure 3.2.

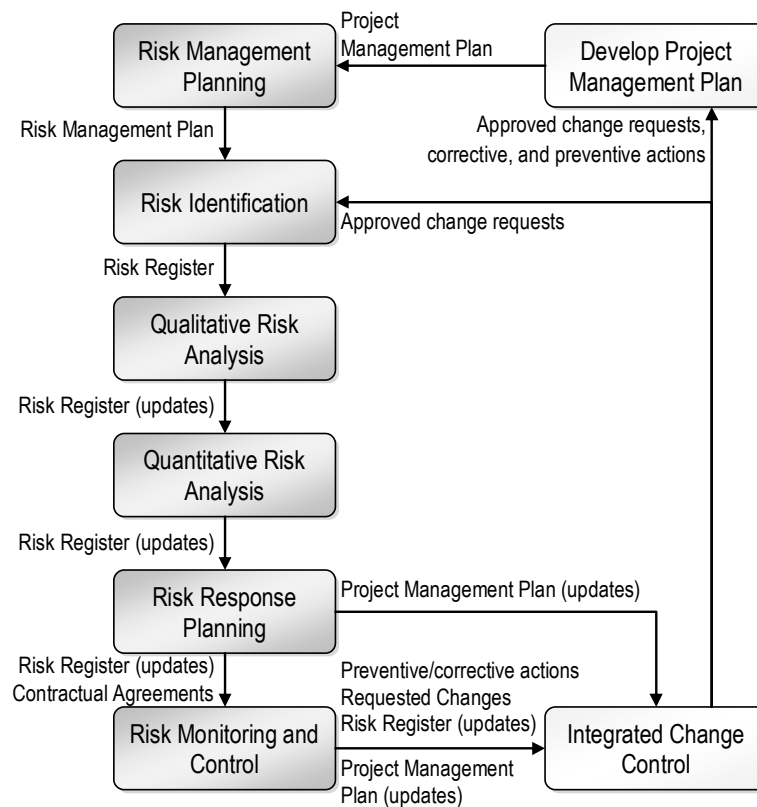


Figure 3.2: Project Risk Management processes according to PMBOK (PMI, 2004)

The set of processes proposed in the Project Risk Management knowledge area from the PMBOK defines a management cycle to address risks throughout the course of a project. As risks are identified and addressed it is necessary to periodically reiterate these processes in order to reevaluate risks and effectiveness of responses adopted. The definitions and links among these six processes are following described:

- **Risk Management Planning:** is the process in which project managers decide how to approach and conduct risk management during the whole project.

This process leads to the specification of a *Risk Management Plan*, which defines methodologies, roles and responsibilities, budgeting, timing, risk categories, and probability/impact matrix for the conduction of risk management in subsequent processes;

- **Risk Identification:** is an iterative process that determines the risks that might affect the project and records their characteristics. Among several techniques, risk identification may be carried out by brainstorming, interviewing, or creating checklists based on historical information that has been accumulated from previous similar projects. The output of this process is the initial entries of the *Risk Register*. The *Risk Register* is a list of identified risks, potential responses, root causes, and risk categories, which is updated during subsequent risk management processes;
- **Qualitative Risk Analysis:** is the process of assigning priorities for treatment of identified risks using their probability of occurrence and corresponding impact on project objectives, such as cost, time, scope, and quality. Probability and impact are assessed, for each identified risk, in interviews or meetings with project team members or other people from outside the project with extensive knowledge on risk assessment. PMBOK itself recognizes that gathering high-quality information for risk assessment is difficult, and usually consumes time and resource beyond the originally planned;
- **Quantitative Risk Analysis:** is the process in which quantitative evaluations are performed for some of the highly relevant risks prioritized in the previous process. Numerical ratings are estimated for the effects of high priority risks aiming to guide the efforts and intensity of response planning. Despite the efforts conducted in this process deeply detailing and measuring risks, it still depends on the knowledge of risk analysis experts. Moreover, it takes a long time to gather useful information about previous projects;
- **Risk Response Planning:** is the process in which project managers, based on qualitative and quantitative analysis, define options and actions to reduce threats (adverse risks) and enhance opportunities (favorable risks). Response actions should be appropriate to each risk (*e.g.*, in terms of cost). As output of this process, risk-related contractual agreements with other parties (*e.g.*, insurance contracts), as well as recommended changes to the *Project Management Plan*, may be established;
- **Risk Monitoring and Control:** is a continuous process that must be executed during the life cycle of the project in order to keep tracking of the identified risks and detect other newly arising. Occasionally, *Preventive Actions* (contingency plans) or *Corrective Actions* (workarounds) planned for risk response result in *Change Requests* to be handled by the *Integrated Change Control* (process from outside the Project Risk Management). All approved changes, workarounds, and contingency plans should be documented and attached, in the *Develop Project Management Plan* process, to the *Project Management Plan*, which, on its turn, should be periodically re-evaluated in terms of risks.

Some shortcomings can be easily identified in PMBOK processes, especially in risk identification and analysis, that could prevent them from being smoothly adopted in real life projects. Firstly, risks are assessed mainly based on human knowledge; hence, the quality of risk management is a function of the experience of stakeholders. The *Qualitative Risk Analysis*, in addition to consuming too much human resources, may propagate errors to the next processes. Since *Quantitative Risk Analysis* is optional for low priority risks, some risks wrongly considered as irrelevant may cause damage to project objectives beyond the expectations.

As exposed in this chapter, risk management is indeed a real concern in both IT Change Management and IT Project Management areas. Based on the analysis of ITIL and PMBOK's guidelines, respectively in Sections 3.1 and 3.2, it is remarkable that in the context of IT projects there is a clear and well defined procedure to deal with risks. On the other hand, for IT changes, risk management is actually mentioned as essential to enhance quality of deployed changes and to avoid disrupting provided services. Nevertheless, there is no specific procedure or method to address risks in this context, the main recommendations are on adopting other general purpose standards, such as the M_o_R framework.

Although all guidelines and process definitions just exposed in this chapter, experience of practitioners shows that lack of automation and standardization are still great barriers for the adoption risk management best practices in real environments. Aiming to tackle this issues, hereafter in this dissertation, the framework proposed to automate some of the key steps of risk management is going to be presented.

4 RISK ASSESSMENT FRAMEWORK

In previous chapters, an overview of the context of IT infrastructures and services management has been presented. In addition, the relevance of risk management in this context and the current shortcomings found specially in risk assessment methods currently applied by organizations that want to deal with risks in their activities have been explored. Such shortcomings motivate the proposal of a framework targeted to support risk management decisions taken by IT operators and managers, more specifically focused in workflow-based IT management processes, such as IT Change Management and IT Project Management. In this chapter, at first, the information models employed for the representation of workflows and their execution traces are explained. Afterwards, the proposed framework itself is detailed in a top-down approach, *i.e.*, its inputs and expected outputs as well as its general behavior are presented; after that, detailed information and algorithms for each component of the framework are described.

4.1 Workflow Information Model

Since the framework proposed in the context of this research applies specifically to workflow-based systems, it is important to have a clear understanding of what workflows are and how they are usually modeled in IT management systems. According to Dumas *et al.* (DUMAS; VAN DER AALST; TER HOFSTEDÉ, 2005) a workflow consists of a coordinated set of activities that have to be executed in order to achieve a predefined goal. These activities may be interconnected in sequence or in parallel and, in the case of parallel branches, transitions might still be subject to conditions. The information model adopted in this dissertation to represent workflows is a subset of the Workflow Process Definition model proposed by the Workflow Management Coalition (WfMC) (WfMC, 2007) as shown in Figure 4.1.

Every workflow is represented by an instance of class the *Workflow Process Definition*, while its activities are represented by instances of the class *Activity*. Activities might be grouped into *Activity Sets* by any kind of criteria defined by the system's operator. Aiming to enable reuse and modular specification of workflows, activities may be specialized into three child classes: *Atomic Activity* that represents finer-grained activities that can be actually executed to perform one specific task, *Block Activity* that express higher-level activities and refer to another set of activities, and *Sub-Process Definition* representing a very high-level activity that actually refers to another workflow that should be executed as part of the current one.

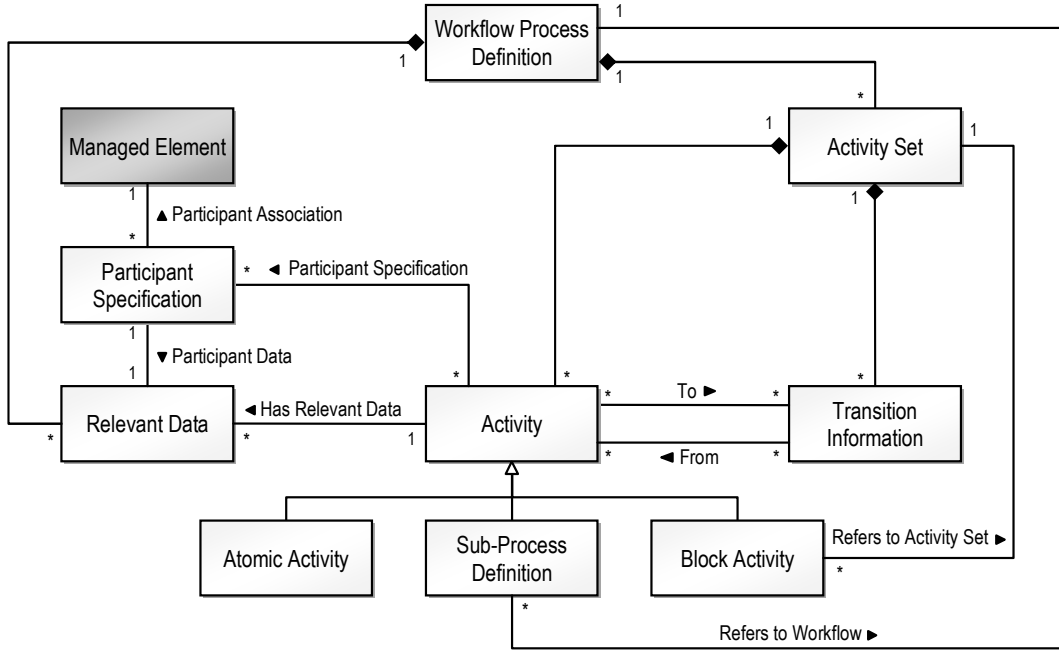


Figure 4.1: Extension from the Workflow Process Definition model

Transitions between activities are represented by instances of the class *Transition Information*. One activity may have transitions *to* many other activities, representing branches (with or without conditions), or transitions *from* another set of activities that represent the joins of the workflow. Usually, activities will have resources associated and the description of the roles and responsibilities of this resources to the execution of the activity is detailed by the *Participant Specification* class. Any kind of resource may be associated to an activity, such as humans, software packages, computer systems, programming languages, libraries, or configuration files. The associated resources are in fact Configuration Items (CIs) available in the IT infrastructure. The connection between the workflow description model and another information model that represents the current state of the IT infrastructure is established by the *Managed Element* class. This is an abstract class that represents the top-level generic element that can be managed and links the workflow's participant resources to the widely used Common Information Model (CIM) proposed by the Distributed Management Task Force (DMTF) (DMTF, 2008).

Finally, all important data produced or consumed during the execution of the activities of a workflow, such as documentation or input parameter, should be captured and stored in instances of the class *Relevant Data*. It is important to keep in mind that workflows specified according to this model are not necessarily attached to any specific vendor or system. It is possible to naturally map them to any workflow description language, such as the Business Process Execution Language (BPEL) (OASIS, 2007).

4.2 Log Records Information Model

The framework for risk assessment proposed in this dissertation bases its analysis on the execution traces of past workflows aiming to estimate risks for executions of further ones. Although the workflow model just described can accurately specify the structure and characteristics of workflows, such as order of execution of activities

and resources involved on them, the actual execution of the workflow may often suffer deviations from what was originally planned depending on decisions made at run-time. For instance, different input parameters may select different branches for many executions of the same workflow or failures during the executions of activities can abnormally interrupt their executions sometimes triggering other workflows as backup plan. Therefore, to allow the representation of execution traces of workflows and future retrieval of these traces for risk assessment, in this dissertation a model that employs a subset of classes from CIM is proposed. Figure 4.2 shows the proposed model where classes in dark grey have been introduced to attach risk related information to the execution records of workflows.

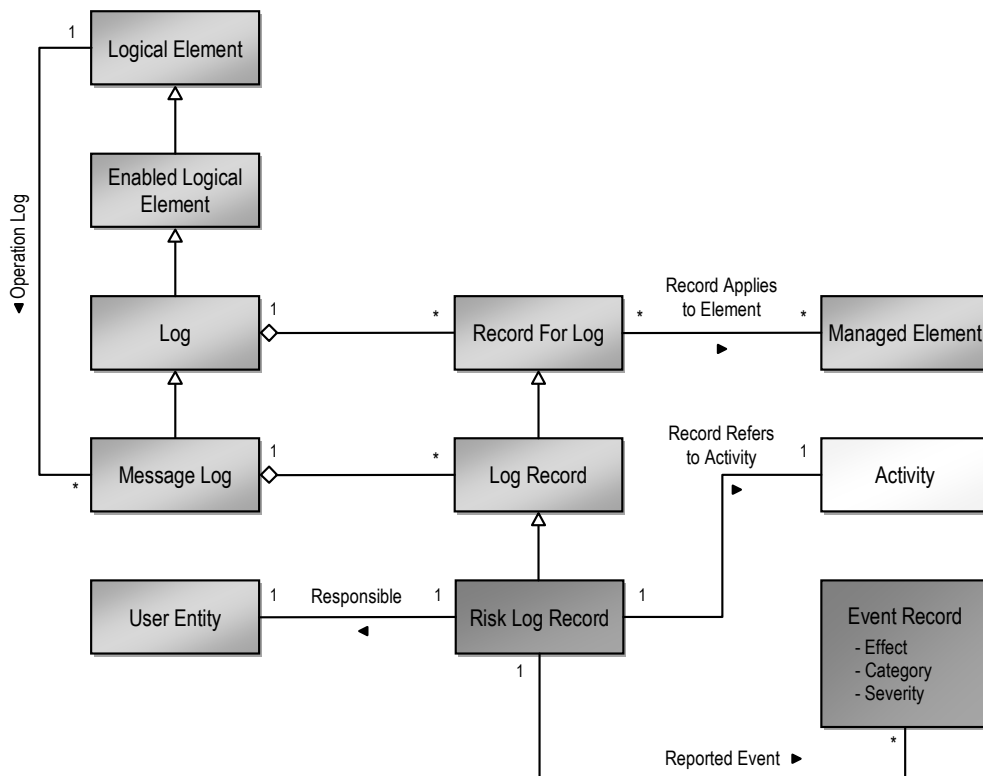


Figure 4.2: Information Model to Represent Execution Traces of Workflows

All classes in light grey are commonly used to represent general purpose log records in systems that use CIM to model any IT infrastructure. Instances of the class *Logical Element* are employed to represent the logical place where log records will be stored and maintained, while instances of the class *Enabled Logical Element*, which extends the former, represent the abstraction that these elements are currently enabled. Typically, these instances of *Logical Element* will represent logical components of a system, such as files or databases. Instances of the *Log* class represent the existence of logs and its characteristics, whereas instances of *Message Log* describe the methods to access, update, or delete log messages. The *Record For Log* class is used to instantiate records that are aggregated to an instance of *Log* and it is its specialization, *Log Record*, that will actually provide definitions of format of entries in a *Message Log* (e.g., recorded date and time, class that created the record, log message expected format, etc.).

Instances of *Log Record* could be used to represent the actual logs of executions of workflows, however the DMTF recommends, as a best practice for CIM usage, to extend the *Log Record* class in order to add semantic information about the stored log entries. Therefore, the *Risk Log Record* class proposed in this work has the purpose of storing risk related information about execution of activities (class *Activity*) of workflows, linking this model back to the workflow description model. Associated to an instance of *Risk Log Record* there will be a *User Entity* that is responsible for the information reported in each record. By inheritance, a *Risk Log Record* is also associated to one or more instances of *Managed Element* (*Record Applies To Element* association). It is useful to associate logs of activities executed involving each specific *Managed Element* from the IT infrastructure. Finally, instances of the *Event Record* class may be associated to a *Risk Log Record* in order to represent all events reported during the execution of an *Activity*. The *Event Record* class also holds relevant information for risk assessment, namely: *Effect* which may be positive or negative (representing favorable or adverse events); *Category* that is useful to segregate risky events, according to a set of categories defined for a specific environment, aiming to enable more human readable risk reporting; and *Severity* that represents the dimension of the damage or advantage caused by the reported event.

4.3 Risk Analyzer Framework Overview

Figure 4.3 presents an overview of the *Risk Analyzer Framework* introduced in the context of this research. The inputs to the framework are: (i) a workflow consistent with the model presented in Figure 4.1, upon which risk analysis should be performed and (ii) a database of log records from previously executed workflows that must be structured according to the model proposed in Figure 4.2.

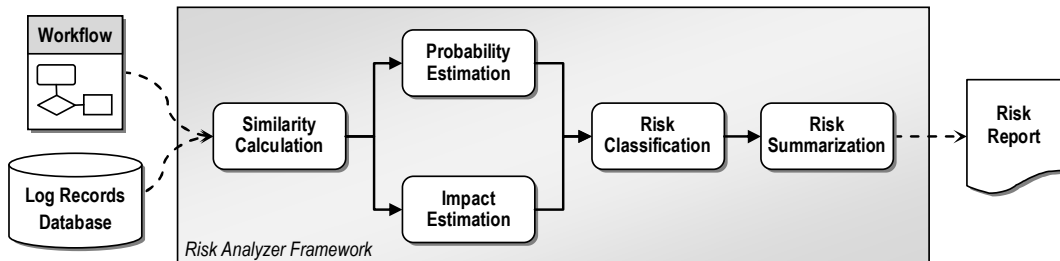


Figure 4.3: Architecture of the Framework for Risk Assessment

By processing these inputs, the *Risk Analyzer Framework* is able to automatically generate a *Risk Report* containing the results of risk assessment and displaying relevant risk related information back to the human operator/manager aiming to help on the decision making for risk response planning process. The possible formats of these reports are presented and discussed in more details in Chapter 5. The following sections present the behavior of each internal component of the proposed framework. In order to ease the understanding of explanations and since negative risks are far more a concern than positive ones, only adverse events will be considered in the remainder of this chapter. However, it is relatively easy to include positive events simply repeating the process.

4.4 Similarity Calculation

Similarity between workflows is a key metric in the context of this research and is currently an open topic of investigation itself (BIANCHIN et al., 2010). The fundamental idea of the proposed framework relies on the fact that it is possible to analyze information documented in previous executions of workflows learning from them in order to avoid risks and to enhance future executions. The first problem that arises in this approach is that there might not be enough precise information about previous executions of the workflow under analysis, simply because it has been recently designed, for example. On the other hand, there can exist an extensive database documenting executions of other workflows, which may not match perfectly the one being analyzed, but can still have similar characteristics, sequences of activities, or involved participants. Hence, the main objective of the *Similarity Calculation* module is to compare the activities of the workflow under analysis with other activities found in a *Log Records Database* identifying a set of workflows that are somewhat similar to former. Based on this calculation, next modules (*i.e.*, *Probability Estimation* and *Impact Estimation*) are able to combine several probabilities and impacts estimated from different activities previously executed and weight these values according to their similarities with the workflow being analyzed.

The *Similarity Calculation* module performs its operations for each activity of the analyzed workflow in three basic steps:

1. **Fetch from the *Log Records Database* activities that are "similar enough" to the ones being analyzed:** this is performed in order to select, from the database, only activities that will have significant similarity values before calculating the similarity for their workflows. Assuming that the *Log Records Database* might be very extensive, this step prevents both the waste of time calculating similarity of workflows that may not be relevant and inclusion of noise caused by adding many activities with very low similarity. In this work, two activities are considered "similar enough" when they have the same basic operation (*e.g.*, install, remove, configure, develop, test, deploy) and have at least one associated participant in common;
2. **Calculate the *Influential Workflow (IW)* for all activities to be compared:** similarity values have to be calculated for pairs of activities, one from the workflow being analyzed by the *Risk Analyzer Framework* and another selected from the *Log Records Database* in the previous step. The *Influential Workflow (IW)* is a subworkflow calculated for both activities, which includes only activities that might potentially influence their executions. In this work, it is stated that an activity b influences an activity a when b is executed either before or in parallel with a in a given workflow. In other words, the IW of an activity a excludes all activities that depend on a 's execution, eliminating from the IW activities that cannot influence the execution of a anymore from the similarity measure;
3. **Calculate the *Risk Affinity (RA)* among selected activities:** in order to capture the similarities between two IWs (obtained in the previous step), a metric called *Risk Affinity (RA)* is employed in this work (as shown in Equation 4.1). RA calculation uses a function θ that returns a value (ranging from zero to one) that represents the highest similarity matching for the k^{th} pair of

activities from the two IW being compared. Internally, this function considers the percentage of coincident participants involved in pairs of activities (*e.g.*, compares involved computers, software, technologies, and humans). However, the θ function respects the same restrictions applied in the first step of this module, which means that it will return more than zero only if both activities have the same basic operation and at least one common participant. The RA metric is computed by a sum of similarities of k pairs of activities up to the size of the smaller workflow, divided by the size of the bigger one. This enables RA to capture not only local differences between activities but also to distinguish workflow sizes.

$$RA(A, B) = \frac{\sum_{k=0}^{\min(|A|, |B|)} \theta_k(A, B)}{\max(|A|, |B|)} \quad (4.1)$$

4.5 Probability Estimation

The procedure for estimating probabilities is performed by the *Probability Estimation* module of the proposed framework and its main function is presented in Algorithm 1. Intuitively, probabilities are calculated by dividing two values: (*i*) the sum of all events occurred and documented in the *Log Records Database* for a given activity (dividend) and (*ii*) the sum of the total executions of the same activity in the same database (divisor). These two values are weighted by the RA between the analyzed workflow and others, previously computed by the *Similarity Calculation* module. The main idea of this division is to take advantage from both workflows that have very similar activities and also prioritize similar activities that have a significant number of previous executions.

Algorithm 1: *Probability Estimation Function*

Input: W : workflow under analysis, A set of activities preselected by similarity

Output: set of tuples containing activity, probability, and event category

1. $S \leftarrow$ set of empty tuples (activity, probability, event category)
2. **for each** Activity $a \in W$
3. **do for each** EventCategory $ec \in$ set of possible event categories
4. **do** $T \leftarrow 0$; $E \leftarrow 0$;
5. **for each** Activity $b \in A$ | b is similar enough to a
6. **do** $R \leftarrow$ RA precomputed between IW_a and IW_b
7. $T \leftarrow T +$ (executions of b in log records $\times R$)
8. $E \leftarrow E +$ (events of category ec for b in log records $\times R$)
9. $\varphi \leftarrow E \div T$
10. $S \leftarrow S \cup \{a, \varphi, ec\}$
11. **return** S

In order to calculate probabilities, Algorithm 1 receives as input the workflow W under analysis and a set A containing the activities that have been selected by the *Similarity Calculation* module because of their similarities against the activities

from W . The algorithm has three main loops that behave as follows: (i) the outermost loop iterates through each activity a of the workflow W (Row 2) for which probabilities are being calculated; (ii) the inner loop iterates every possible event category ec considered in risk assessment process (a set of event categories should be predefined and events classified into one of them as represented in Figure 4.2) (Row 3); and (iii) the innermost loop finds in a set A every activity b that has been preselected by the *Similarity Calculation* module as similar enough to the activity a (Row 5). Following, the algorithm obtains the precomputed *Risk Affinity (RA)* value for the *Influential Workflows (IW)*s of a and b storing it into R (Row 6). Afterwards, the total of executions and events of category ec reported for activity b are weighted and stored in T and E respectively (Rows 7 and 8). After iterating through all activities of set A , probabilities for each activity b (that matched the similar enough criteria of Row 5) are then calculated by dividing E by T and stored in φ (Row 9). The value stored in φ represents an estimation of the probability of an event of category ec happening based on a combination of probabilities of events of the same category that have been found in log records of activities similar to a . Finally, the probability value in φ will be added to the set S (Row 10) along with the activity a and the event category ec . At the end of the algorithm, S is returned as output of the function (Row 11).

4.6 Impact Estimation

As well as probability, impact is another key factor in the proposed framework for risk assessment. In most cases, probability represents a quantification of the likelihood of a certain event that could happen and, in the context of this research, it can be estimated by observing past occurrences of similar events. By contrast, the definition of what impact actually is and how it can be estimated depends very much on the environment under analysis. For example, in the context of IT projects, impact is faced as the effects of events over specific project objectives. Regarding time objective of a project, impact may be faced as unexpected delay during the execution of activities and impact on cost objective can represent that actual expenses have overrun the originally planned budget. On the other hand, when it comes to IT changes, failures that occasionally happen during deployment might have impact to the business by affecting the availability of services currently provided over the IT infrastructure. Indeed, the existence of several ways of estimating impact (just like other factors such as probability or similarity) is not a real problem itself. Since the framework proposed was conceived in a modular approach, what is really important is that there is a way of estimating impact for the activities of a given workflow and this computation is performed by the *Impact Estimation* module. In other words, each module of the framework should work as a black box, receiving a set of inputs, processing them somehow, and providing expected outputs appropriate with the context being analyzed.

The default behavior of the *Impact Estimation* module, presented in Algorithm 2, is very similar to what is proposed for the *Probability Estimation* module. It also iterates through all activities in a workflow W (Row 2), then it traverses all event categories (Row 3) and, finally, goes through a set of activities A preselected by their similarities with the ones in W (Row 5). The main difference is that, instead of counting the number of executions and the occurrence of events of a given category,

the algorithm considers the severity of reported events facing the originally planned for the activity for each specific event category (Rows 7 and 8). As with probability, these values are likewise weighted using RA computed by previous module. This is performed in order to make impact estimated for activities tend to approximate to the activities that were executed in more similar environments. Thus, within this approach, impact estimation results sort of mean historical impact value for events of a certain category reported for activities similar to a .

Algorithm 2 presents an interesting solution for areas like IT Project Management, where risks are addressed as events that affect a set of project objectives and the severity of events reported in the past reflect how harmful they have been facing one of these objectives. For example, assuming a given activity that was planned to take 8 hours of work to conclude. When it is executed an event is reported informing that it took 4 hours more than it should. The impact that this event represents for the project's time objective is the hours that have delayed divided by the hours it was planned to last (in this example, impact would be of 0.5). When dividing the severity (4 hours) by the originally planned for one activity (8 hours) it is expected to generate a normalized value ranging from zero (no impact) to one (great impact). However, it is clearly possible that impact values go beyond the upper bound of this range (in this example, if the delay was of 16 hours, impact would be about 2). Through the *Risk Classification* module the framework deals with these situations by organizing impact into ranges, where values that overcome a given threshold are always considered as highly damaging, as further detailed in Section 4.7.

Algorithm 2: *Impact Estimation Function*

Input: W : workflow under analysis, A set of activities preselected by similarity

Output: set of tuples containing activity, impact, and event category

1. $I \leftarrow$ set of empty tuples (activity, impact, event category)
2. **for each** Activity $a \in W$
3. **do for each** EventCategory $ec \in$ set of possible event categories
4. **do** $T \leftarrow 0$; $E \leftarrow 0$;
5. **for each** Activity $b \in A \mid b$ is similar enough to a
6. **do** $R \leftarrow$ RA precomputed between IW_a and IW_b
7. $T \leftarrow T +$ (sum of expected values of b for $ec \times R$)
8. $E \leftarrow E +$ (sum of severities for ec in logs of $b \times R$)
9. $\lambda \leftarrow E \div T$
10. $I \leftarrow I \cup \{a, \lambda, ec\}$
11. **return** I

In other contexts, such as IT Change Management, the estimation of impact might be conducted differently. Since impact in changes can be measured as business impact caused by unavailability of services affected by deployment failures, initially, a metric that represents the importance of *Configuration Items (CIs)* to business is required. In this work, a metric called *Business Relevance (BsR)*, which is associated to every CI (*e.g.*, software, hardware, or service) that is relevant to the business continuity is proposed. BsR is expressed by a numerical value and, regardless of the scale adopted, it should enable comparisons between relevancies of different CIs. For instance, a possible range of BsR could be: Maximum (1.00), High (0.75), Medium (0.50), Low (0.25), and Not defined (default) (0.00). Moreover, this metric should be assigned before risk assessment, for example by the system's operator, only to

the CIs that have any direct relevance to business. Based on the associations and dependencies between CIs, the Algorithm 3 is able to compute the total impact of each element involved in the activities of a given change.

Algorithm 3: *Impact Estimation Based Elements Relevance*

Input: W : workflow under analysis, A set of activities preselected by similarity

Output: set of tuples containing activity, impact, and event category

1. $U \leftarrow$ empty set of tuples (CI, AR)
2. **for each** ConfigurationItem $ci \in$ set of CIs handled in W
3. **do** $\gamma \leftarrow$ BsR of ci
4. $D \leftarrow$ set of CIs that depend on ci
5. **for each** ConfigurationItem $d \in D$
6. **do** $\gamma \leftarrow \gamma +$ BsR of d
7. $U \leftarrow U \cup \{ci, \gamma\}$
8. $I \leftarrow$ set of empty tuples (activity, impact, event category)
9. $t \leftarrow$ CI that represents the whole IT infrastructure
10. $N \leftarrow extract(t, U)$
11. **for each** Activity $a \in W$
12. **do for each** EventCategory $ec \in$ set of possible event categories
13. **do** $ci \leftarrow$ CI with highest AR handled in a affectable by ec
14. $T \leftarrow$ AR of N
15. $E \leftarrow$ AR of ci
16. $\lambda \leftarrow E \div T$
17. $I \leftarrow I \cup \{a, \lambda, ec\}$
18. **return** I

In a first moment, the algorithm calculates the so called *Absolute Relevance* (AR) of all CIs handled in the workflow W . AR is a metric that indicates the overall perception of relevance of an element to the business continuity, including its BsR and the sum of BsR of all elements that depend on it, directly or indirectly. In this algorithm, for each CI ci handled in W (Row 2), the value of the AR for the element ci (variable γ) is initiated with its own BsR (Row 3). Subsequently, a set D is created and populated with elements that depend, directly or indirectly, on ci (e.g., software that depends on the computer where it is hosted or services that depend on other services) (Row 4). This set is filled in recursively by iterating through dependencies defined between CIs. Following, each element that belongs to D (Row 5) will have its BsR accumulated in variable γ (Row 6). Afterwards, the tuple (CI, AR) is included in the set U (Row 7), which, at the end of all iterations, will hold all CIs handled in W and their respective AR values.

After computing AR values for the CIs handled in W , a normalization of these values is performed in order to associate the actual impact metric to the activities. This metric represents the portion of the IT infrastructure that is compromised by failure of a particular CI, from the business impact point of view. In order to calculate the impact of a CI, an element that represents the IT infrastructure, whose all CIs depend on, is defined in this work. The AR of this element is the sum of all BsRs defined, and it is handled in all workflows. Following, Algorithm 3 initializes a variable t with the element that represents the IT infrastructure (Row 9). Then, it invokes a predefined procedure that locates and extracts the CI t from the set U (Row 10). Subsequently, two loops are employed (analogously to the

ones in Algorithm 2, Rows 2 and 3) in order to iterate through activities in W and event categories (Rows 11 and 12). Inside these loops, one CI handled in activity a among those that can be affected by events of category ec contacting the highest AR value is selected and stored in ci (Row 13). In this algorithm, it is assumed that there is a mapping between the CIs handled in activities and possible event categories. For example, in the context of IT changes, *Activity Failures* can only affect software elements, whereas *Resource Failures* might actually represent hardware damage (further details about failure classifications in IT Change Management are presented in Section 5.1). Following, instead of calculating impact based on log records of similar activities, this is performed by dividing the AR of the selected CI by the total relevance of the IT infrastructure (Row 16). Similarly to Algorithm 2, a set is filled with activities and their estimated impacts for all event categories (Row 17) and, as output of the algorithm, this set is returned (Row 18).

4.7 Risk Classification

So far, the *Risk Analyzer Framework* is able to compute probabilities and impacts of events for every activity of a given workflow for all event categories considered in the risk assessment process. Since one key objective of risk assessment is to aid decision support for further risk response planning, it is important that the proposed framework outputs information about probability and impact values calculated in an organized and comprehensive way. Therefore, it is the role of the *Risk Classification* module to rank these values into risk classification ranges like those presented as best practices in Section 2.1. The Institute of Risk Management (IRM) (IRM, 2002) recommends quantifying probability and impact using the following scales: (i) high (more than 25%), medium (between 25% and 2%), and low (less than 2%), for probabilities, and (ii) high (significant), medium (moderate), and low (insignificant), for impact. The ranges presented in Table 4.1 are employed by default in the proposed framework. However, the boundary values of each range and the number of ranges itself should be customized in order to better meet every environment's needs.

Table 4.1: Classification ranges for probability and impact

	Low	Medium	High
Probability	< 2%	2% - 25%	> 25%
Impact	< 0.02	0.02 - 0.25	> 0.25

After being mapped into one of the aforementioned ranges, the results obtained by previous modules (namely, *Probability Estimation* and *Impact Estimation*), are classified according to the *Risk Classification Matrix* presented in Table 4.2. According to this matrix, each activity of the workflow will be marked with one of nine categories, where Category 1 represents highest risks (high probability and impact) and Category 9 lowest risks (low probability and impact). Also, the dimension of the *Risk Classification Matrix* might be changed in order to better fit the requirements of a specific environment. For example, if two more ranges were included in Table 4.1, Medium-Low (between Low and Medium) and Medium-High (between Medium and High), the matrix would have to be extended to a size of 5x5.

Table 4.2: Risks Classification Matrix

	Probability		
Impact	High Impact High Probability Category 1	High Impact Medium Probability Category 2	High Impact Low Probability Category 3
	Medium Impact High Probability Category 4	Medium Impact Medium Probability Category 5	Medium Impact Low Probability Category 6
	Low Impact High Probability Category 7	Low Impact Medium Probability Category 8	Low Impact Low Probability Category 9

One important fact to notice is that this kind of matrix is widely employed by organizations for risk assessment; however, the association of categories to risky events is usually performed intuitively by humans. Also, it is worth to mention that this matrix tends to emphasize impact rather than probability, since Category 3 (High Impact and Low Probability) represents much higher risk than Category 7 (Low Impact and High Probability), for instance. The prioritization of risks that evidence high impact factors is actually a recommendation in most of the current standards and guides of best practices for risk management. Nevertheless, for larger sets of categories there are other approaches to build risk categorization that will avoid the attenuation of high probabilities. One possible approach is the *Risks Classification Grid* adapted from the M_o_R framework (OGC, 2007a), as presented in Table 4.3. In this grid there is a customizable multiplier factor for each row (probability ranges) and column (impact ranges). By tuning these multipliers the manager/operator is able to increase or decrease relevance for each range as desired. It is possible to visualize that the categories are scattered throughout the grid instead of concentrating categories that represent more risk in high impact ranges.

Table 4.3: Risks Classification Grid

Probability						
Very High	0.9	Category 17 (0.045)	Category 12 (0.09)	Category 8 (0.18)	Category 4 (0.36)	Category 1 (0.72)
High	0.7	Category 19 (0.035)	Category 14 (0.07)	Category 9 (0.14)	Category 5 (0.28)	Category 2 (0.56)
Medium	0.5	Category 21 (0.025)	Category 16 (0.05)	Category 11 (0.10)	Category 7 (0.20)	Category 3 (0.40)
Low	0.3	Category 23 (0.015)	Category 20 (0.03)	Category 15 (0.06)	Category 10 (0.12)	Category 6 (0.24)
Very Low	0.1	Category 25 (0.005)	Category 24 (0.01)	Category 22 (0.02)	Category 18 (0.04)	Category 13 (0.08)
		0.05	0.1	0.2	0.4	0.8
		Very Low	Low	Medium	High	Very High
		Impact				

Although the *Risks Classification Grid* provides a more customizable solution, it also takes more effort from managers/operators in order to tune the right values

for rows and columns. Since three ranges (low, medium, and high) are employed by default in the *Risk Analyzer Framework*, the standard classification used is the *Risk Classification Matrix* of Table 4.2. Similarly to ranges, classification can also be easily adapted to better suit the environment’s needs, as long as their ordering remains; this means that lower categories have to keep representing higher risks.

4.8 Risk Summarization

The final outcome of the *Risk Analyzer Framework* is a *Risk Report* that should contain all relevant risks related information about the workflow being analyzed. As mentioned before, these reports should aid the human manager/operator to quickly identify threats that might be raised during the execution of a given workflow, helping these humans to prioritize efforts for risk mitigation and avoidance. Clearly, a simple tabular report would be enough to present all risk information computed by the framework (probabilities, impacts, and risk categories) for all activities in a workflow yet considering the event categories. Considering, for example, the context of IT projects where risks are analyzed separately for different project objectives (*e.g.*, cost, time, scope, and quality). A detailed tabular report for any random workflow with five activities could be as shown in Table 4.4.

Table 4.4: Tabular Risk Report

Activity		Cost	Time	Scope	Quality
A3	Probability	50.0%	75.0%	5.0%	1.0%
	Impact	0.40	0.30	0.05	0.05
	Category	1	1	5	6
A1	Probability	30.0%	40.0%	1.0%	5.0%
	Impact	0.50	0.30	0.00	0.20
	Category	1	1	9	5
A2	Probability	90.0%	8.0%	50.0%	1.5%
	Impact	0.10	0.30	0.50	0.01
	Category	4	2	1	9
A4	Probability	1.0%	0.0%	1.0%	0.0%
	Impact	0.10	0.00	0.05	0.00
	Category	6	9	6	9
A5	Probability	0.5%	0.0%	1.0%	0.0%
	Impact	0.01	0.00	0.05	0.00
	Category	9	9	6	9

This risk report provides important information about the risks of all activities of the workflow for as many project objectives as needed. The report also brings activities with highest risks (lower categories) to the top, which helps finding the activities that require attention and should have their risks addressed first. However, when it comes to large-scale projects there might be a huge amount of activities spread into several different workflows. Thus, for project managers to tackle the risks of such projects (*i.e.*, composing contingency plans or workarounds), analyzing one activity at a time could still demand too much time and consume excessive resources.

To deal with this type of context, the *Risk Summarization* module is in charge of summarizing risk reports by combining many risk categories into one single value. Summarization takes place by combining groups of risk categories from lower levels activities (*i.e.*, *Atomic Activity*) of workflows, using a given function, into one single risk metric meaningful for evaluation at a higher levels. These summarized values can be displayed in reports taking advantage of the workflow structure, *i.e.*, presenting grouped information for *Activity Sets*, *Block Activities*, *Sub-Process Definitions*, or even for the whole workflow. Thus, the human operator/manager could have a quick overview of the risks contained in a given workflow and zoom in only in the sets of activities that require further attention. Moreover, it is important to keep information apart about the risk categories (*e.g.*, project objectives in the context of IT projects) in all levels detail, in such a way that operators/managers can analyze risks over each category separately.

In this work, a summarization function is proposed in order to perform such combination of risk categories, computing the so called *Average Risk*, as shown in Equation 4.2. This equation actually implements a harmonic mean of risk categories, where n represents the number of categories being summarized (*e.g.*, number of activities in an *Activity Set*). This number is the dividend of the division by the sum of all reciprocals of risk categories (*i.e.*, a_i represents the risk category of the i^{th} activity included in the summarization group). By employing this equation, it is assumed that risk categories will always be represented as values ranging from 1 to any greater positive value and that highest category values represent lower risks.

$$AR = \frac{n}{\sum_{i=1}^n \frac{1}{a_i}} \quad (4.2)$$

One important fact about summarization is that the result of average functions tends to smooth all portions into a mean value. For instance, considering that an *Activity Set* has four activities, being three of them classified in risk category 9 (lowest possible risk) and only one in category 1 (highest possible risk) for one specific risk category. Thus, an arithmetic mean of these values would result in a value of 7, hiding the damage that one of those activities (classified in category 1) could possibly cause if executed. On the other hand, the behavior of Equation 4.2 is quite interesting for risk summarization since it works like a pessimistic approach, making the *Average Risk* tend to approximate to lower values of summarized categories. This helps propagating excessively risky activities, detected by the framework, up into more summarized reports. Using the aforementioned example, the resulting *Average Risk* would assign a value of 3 to the hypothetical workflow, which represents much more risk than the value of 7 obtained with an arithmetic mean.

Yet another possible, and even more pessimistic, approach to risk summarization could be to select always the value that represents the highest risk (lower risk category). Using another example, considering that the five activities in the *Risk Report* of Table 4.4 are grouped into two *Activity Sets*: *AS1* containing activities *A1* and *A2*, and *AS2* containing activities *A3*, *A4*, and *A5*. The results of grouping risk categories into the *Activity Sets* *AS1*, *AS2* and also into the whole workflow (*WF*) are presented in Table 4.5. This table shows the combined risk categories under three different strategies employed for summarization, namely: calculating the *Average Risk* (default behavior of the proposed framework), employing a simple

arithmetic mean, and selecting always the highest risk category. It is quite clear that using an arithmetic mean attenuates much more important risk categories of activities than the other two approaches. Also, Table 4.5 shows that in some situations the strategy employed for summarization might bias efforts of further risk mitigation. For example, by comparing the summarized values of the whole workflow (*WF*) for categories *Cost* and *Time*, when employing an arithmetic mean, one could conclude that more emphasis should be given to address cost related risks; whereas the *Average Risk* tells the other way around. The main problem in the third approach (*Highest Category*) is that it blurs the distinction between categories as summarizations are performed. For instance, looking at the results for the whole workflow, there is no way to tell which of the three risky objectives (*Cost*, *Time*, and *Scope*) requires more attention.

One final consideration about risk summarization is that the *Average Risk* should always be calculated from risk categories of low level activities (*Atomic Activity*), avoiding the use of other averages computed in higher levels of the workflow. This is important to prevent the analysis from losing information about the cardinality of summarized sets (*e.g.*, number of activities in each *Activity Set*). For example, considering a given workflow with two *Activity Sets*, one containing 20 activities and another with only 2, once the *Average Risks* are calculated for both *Activity Sets*, these values will belong to the same range (*i.e.*, from 1 to 9 continuously), and no information is kept about number of activities summarized so far. If an *Average Risk* for the whole workflow was calculated considering summarized information of each *Activity Set*, some important risk categories from the largest one would be attenuated. To tackle this issue, there are two options: (*i*) to calculate the *Average Risk* of the workflow-based on all 22 activities that compose it, or (*ii*) to weight the *Average Risks* from the *Activity Sets* using their cardinals (respectively 20 and 2). Both options produce exactly the same results, although the second is better to avoid recalculation of average values up in the workflow structure. More detailed information on comprehensive and interactive *Risk Reports* are further addressed in Section 5.2.

Table 4.5: Summarized Risk Reports

Average Risk				
	Cost	Time	Scope	Quality
AS1	1.60	1.33	1.80	6.43
AS2	2.35	2.45	5.63	7.71
WF	1.98	1.84	3.04	7.14
Arithmetic Mean				
	Cost	Time	Scope	Quality
AS1	2.50	1.50	5.40	7.00
AS2	5.33	6.33	5.67	8.00
WF	4.20	4.40	5.40	7.60
Highest Category				
	Cost	Time	Scope	Quality
AS1	1.00	1.00	1.00	5.00
AS2	1.00	1.00	5.00	6.00
WF	1.00	1.00	1.00	5.00

5 EVALUATION

In order to evaluate the applicability and technical feasibility of the proposed *Risk Assessment Framework*, in this chapter, two case studies are presented. The first one, introduced in Section 5.1, describes the experiences acquired by implementing the framework as part of the CHANGELEDGE system, enabling automated risk assessment in the context of IT Change Management. The second case study, presented in Section 5.2, was conducted in the context of IT Project Management and its focus is to clarify how comprehensive and interactive reports might help on decision making with regards to risks of IT projects.

5.1 Application to IT Change Management

This first case study was carried out in the context of a project named **CHANGELEDGE: Model Based Change Management for Information Technology Systems**. In this project, a prototype system has been developed (and named after the project) as a proof of concept for the research conducted in the context of IT Change Management. The CHANGELEDGE system, whose conceptual architecture has been already presented in Section 3.1, was conceived to enable some degree of automation in IT change planning and deployment. In this dissertation, the proposed *Risk Assessment Framework* was implemented as a module of the CHANGELEDGE system improving it with automated *Risk Reports* aiming to support the *Change Authority* when deciding whether or not to approve a given change for deployment.

5.1.1 Failure Classification

As mentioned in Section 2.2, in the context of IT Change Management, events that represent risks are regarded as failures that might happen during the deployment of Request for Changes (RFC) affecting the business continuity by disrupting important services provided by means of the managed infrastructure. For the assessment of risks to present more intuitive results, it is interesting that these failures are grouped according to a classification scheme representing failure types that are considered relevant from the point of view of the responsible humans (*i.e.*, *Operator* or *Change Authority*). This classification is further used by the framework to segregate events that represent risks (in this context, failures in changes) during estimations and in risk reporting.

There is no standard classification of failures applied to the context of IT Change Management available in the specialized literature. Therefore, in this dissertation, such classification is proposed considering failures during execution of changes under two aspects, *Source* and *Recovery*, following detailed:

Source of Failures

Some types of failure are recurrent on change deployment and may be either captured by the management system during runtime or informed by an operator in the change review process. Identifying source and classification of failures is important to help the operator understanding the behavior of problematic items and for what reasons they fail. In the proposed failure classification six types of failure are defined inspired on a previous research (RUSSELL; AALST; HOFSTEDE, 2006), as follows:

1. **Activity Failure (AF):** Failures of this type happen for reasons intrinsic to the execution of activities in the Change Plan (CP). Usually, they occur during the installation or configuration of software elements. In addition, they may be triggered by failures on other activities on the workflow (*e.g.*, dependent packages);
2. **Resource Failure (RF):** Resource failures are caused typically by hardware problems that make unavailable the elements where the activities are executed. This could be a physical problem (*i.e.*, the equipment may have been damaged during the deployment process), or a wrong configuration may turn the resource unreachable by the *Change Deployer*. In both cases, an RF is captured;
3. **Human Failure (HF):** Some activities on a CP are performed by humans; therefore, when humans do not behave the way they are supposed to, then HFs are raised. Failures on manual activities should be recorded on the system even if there is no way of automatically capturing them. In these cases, the operator should insert these failure classification records in order to keep the history of changes as well documented as possible;
4. **Time Failure (TF):** When needed, deadlines can be specified to inform when activities should be finished or started. Besides that, time restrictions may be used for synchronization of tasks, for example, to inform that one activity must start before another. Whenever time constraints are breached, a TF should be captured;
5. **External Trigger (ET):** This type of failure occurs when some agent external to the change process interrupts the regular execution of the CP. This could be a signal injected into the system by an external administrator user, informing that the workflow must be interrupted for any reason;
6. **Constraint Violation (CV):** Usually, it happens when an activity in the CP needs to perform an operation that violates any of the organization's policies. Moreover, these failures may be raised by conflicting scheduling of RFCs, for example, when the first change modifies the IT infrastructure implying new conditions not predicted on the next.

Recovery from Failures

Another important aspect in failure classification is what kind of recovery actions can be taken to reestablish system's functionalities after the occurrence of failures. The system should capture remediation actions taken automatically or manually and record them to the logs of changes along with the classification of failures. Although recovery information is not directly used for risk assessment, it is important to keep these records for further analysis, such as estimations of service disruption caused by changes. These recovery actions are classified into two categories, as follows:

1. **No Action (NA):** This class indicates that there was nothing to do after the occurrence of a failure, it could happen because of two reasons: (a) the operator explicitly informs that nothing should be done to recover the system (probably the operation was not significant enough to inspire caution) or (b) in the case of a fatal failure, the system could not take any remediation action to revert the situation;

2. **Remediation (RM):** When the system executes a remediation plan to recover itself, it may be a rollback plan and/or a compensation plan. If there is a rollback plan associated to an activity that fails, this plan will be invoked to undo the changes made by the activity. On the other hand, compensation plans may be a useful instrument to specify alternative ways of reaching the end of the change. For instance, when the installation of a package fails, the system could install another one, that is similar to the first (but not the same), and proceed to the next activities. These compensation plans can be applied along with rollback plans or not; however, they should be used only as a secondary option because they do not guarantee the accomplishment to the original CP requirements. Also, it is important to mention that compensation and rollback plans do not undo failure events. Even if the change process was completed with a remediation plan the system still considers the execution as a failed process, keeping the failure classification and remediation on the record for further analysis.

5.1.2 Request for Change Information Model

In order to specify changes into RFC documents using the CHANGELEDGE system, an information model has been proposed in a previous work (CORDEIRO et al., 2008). This model was conceived based on both the guidelines presented in the ITIL Service Transition book (OGC, 2007b) regarding the change management process and the Workflow Process Definition, proposed by the Workflow Management Coalition (WfMC) (WfMC, 2007). A partial view of the RFC Information Model already extended with classes for logging risk related information is presented Figure 5.1.

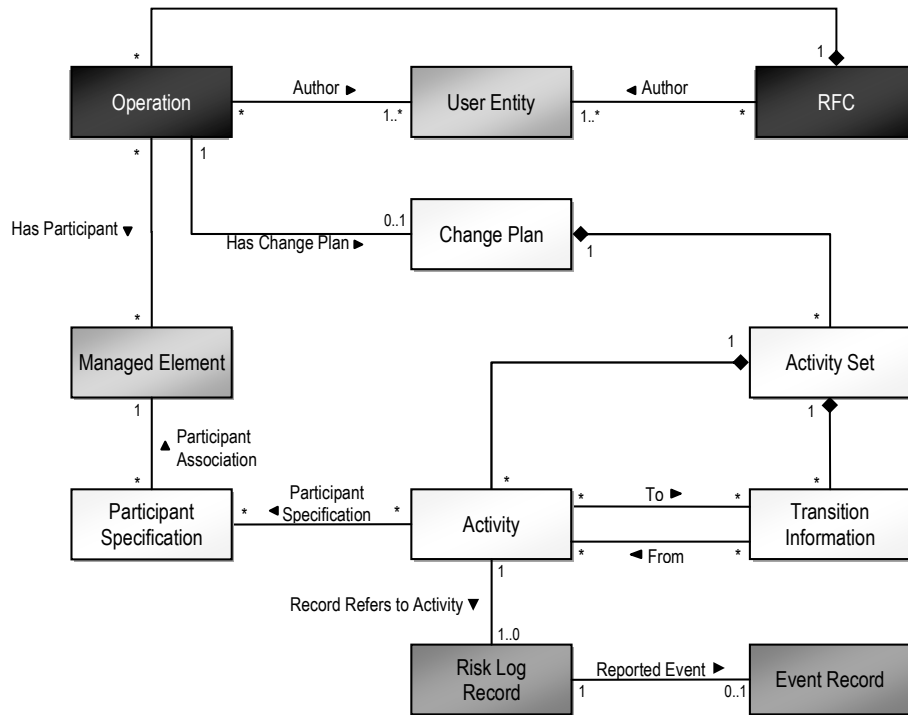


Figure 5.1: Partial View of the Request for Change Information Model

An RFC document (materialized in the *RFC* class) will often contain a description in a very high level of abstraction (usually a textual description) of what changes a *Change Initiator* wants to implement. Also, usually associated to an RFC there will be a list of authors (instances of *User Entity*), an identification number of a problem report in the case the RFC is in response to an incident, time restrictions like deadlines for deployment, among other information. In order to better organize the deployment of an RFC, this document might generate one or more operations (instances of *Operation* class). Each operation has also a list of authors (which might be the same from the RFC or not), some Configuration Items (CIs) from the IT infrastructure (instances of *Managed Element*) that should be involved in the change, and an associated CP (class *Change Plan*). A CP is in fact defined as a workflow of activities interconnected by transitions following all the recommendations from the WfMC in the Workflow Process Definition, already presented in Figure 4.1.

Finally, in order to enable logging in execution of the activities of workflows two classes were added, *Risk Log Record* and *Event Record*, linking back to the model that represents execution traces of workflows (previously presented in Figure 4.2). Each execution of an activity will generate an associated instance of *Risk Log Record*. This instance will contain information related the execution of the activity, such as start and finish time. Whenever an activity fails to conclude, this failure will generate and associate an instance of *Event Record* to the activity's *Risk Log Record*. The *Event Record* will hold information about the effect of the failure (this will always be negative assuming there are no positive failures in changes) and also its classification into one of the six categories presented in Section 5.1.1.

5.1.3 Implementation

In order to evaluate the technical feasibility of the framework in the context of IT Change Management, a prototype has been developed and incorporated into the CHANGELEDGE system. This system employs a subset of classes from the Common Information Model (CIM) (DMTF, 2008) to implement a representation of the managed IT infrastructure. As mentioned in Section 4.6, the CIs of an IT infrastructure should have *Business Relevance (BsR)* values associated that represent their importance to the organization's business. In order to materialize the *BsR* in the prototype, a metric was employed using the CIM *Base Metric Definition* class. This class defines a range of possible values for relevance to be applied to the *Managed Elements*, for example: High (1.00), Average (0.50), and Low (0.25). Elements that have some degree of relevance to the business continuity must have instances of *Base Metric Value* associated with a *BsR* value assigned. If no *BsR* value is assigned for a specific CI, the *Impact Estimation* module will consider that the element is irrelevant for the business (*i.e.*, *BsR* zero).

In order to represent dependencies between CIs, CIM defines several objects that implement relationships between items of an IT infrastructure. Some of these relationships explicitly represent dependencies, such as *Service to Service Dependency* indicating when a service requires features from another service to work properly. Other relationships, although not necessarily representing dependencies, are considered as such by the framework. This is the case of *Installed Software Element*, which implements a dependency of a software element to the computer system where it is hosted in. In the implemented prototype, a list of objects that represent dependencies is iterated by the algorithm in order to calculate impact of ICs.

For the deployment of changes, the CHANGELEDGE system uses of a subsystem called *Deployment System*. It is responsible for translating the CP to be deployed into a BPEL (Business Process Execution Language) document (MACHADO et al., 2008). The generated document is then submitted for execution by a Web service orchestration system called ActiveBPEL (Active Endpoints, 2008), which controls the execution of workflows and captures failures. Each CI of the IT infrastructure should have a management interface via Web services to be invoked by ActiveBPEL in order to implement change activities. After performing each activity, the Web service interface reports to a database: the status of implementation, failures occurred, and time elapsed in the execution of activity. These execution reports compose the *Log Records* of changes and are kept by the *Configuration Management System (CMS)* for further analysis.

For simulation purposes, each Web service implemented by the CIs produces failures pseudo-randomly, according to a uniform probability distribution, during the deployment of changes. Such failures are injected as exceptions and compel the orchestration system to interrupt the regular execution flow starting associated remediation plans. The Web services are customizable to associate different probabilities of failure for different failure types of specific CIs. Although six types of failure are possible, for the sake of simplicity, in this case study only three of the most common failures are randomly generated: Activity Failure (AF), Resource Failure (RF), and Human Failure (HF).

5.1.4 Scenario and Results

In order to evaluate the proposed framework, tests and measurements have been performed on an emulated IT environment. To measure the performance of changes, one of ITIL's recommendations is to use a Service Disruption (SD) metric, which reflects damage to services caused by unsuccessful changes. This metric represents the time elapsed after a failure on change deployment until the system recovers the managed infrastructure, as depicted in Figure 5.2. In addition, SD should consider the impact of failures over the affected services. To this end, in this work the Equation 5.1 is employed to calculate the SD for a given activity i of a CP. The calculation is performed by multiplying three factors: (i) $F_{ft,i}$, which is the total number of failures of a type ft found in the execution records of activity i ; (ii) $t_{ft,i}$ representing the average time to recover the system from a failure of same type in activity i (may be obtained from the execution records of remediation activities); and (iii) $IF_{ft,i}$, which contains the impact factor of the CI affected by the failure of type ft handled in activity i . The sum of these values, for each failure type considered in the risk estimation, results in an SD metric of an activity.

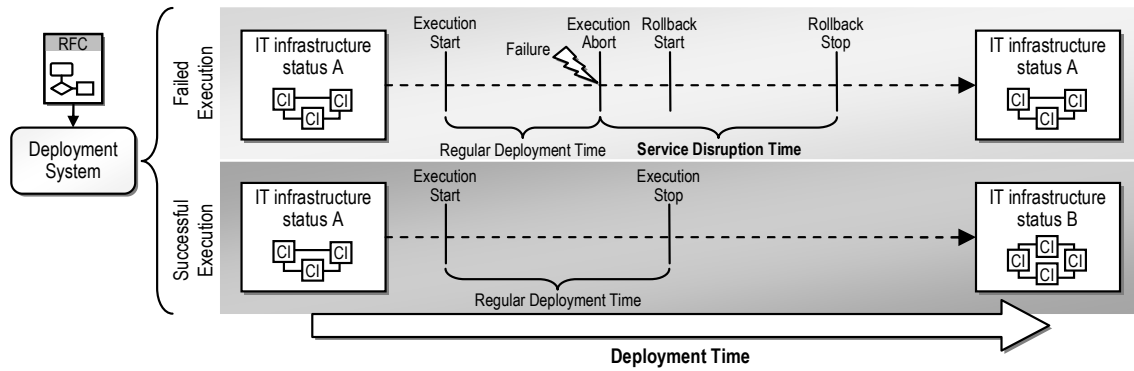


Figure 5.2: Service Disruption Example

$$SD_i = \sum_{ft \in FT} F_{ft,i} * t_{ft,i} * IF_{ft,i} \quad (5.1)$$

Before moving into the case study itself, another point deserves to be mentioned. Since the framework proposed in this research was conceived in a modular approach, minor changes in the *Similarity Calculation* module have been performed in order to make *Risk Affinity (RA)* metric more accurate in the context of IT Change Management. In this case study, the RA calculation considers also the failure type being analyzed. In other words, two activities are only regarded as somewhat similar (RA greater than zero) if they have the same basic operation and one common participant regarding a specific failure type. For example, two activities that install the same software element over two different computer systems will be considered as similar for AF (because they involve the same software participant) and not similar for RF (because they apply to different computational resources).

For this case study’s scenario, it is assumed that a company internally develops an automation software and employs development teams divided into two areas: (i) Web interface and Web services development and (ii) persistency layer and database modeling. The system developed by these two teams has a Web interface written in Flex, Web services written in PHP running on an Apache Web server, and information persisted over a MySQL database. Recently, the company has started developing a new version of this software. Therefore, both teams had their workstations updated using two RFCs, as shown in Figures 5.3 (a) and (b). The former sets up a Web development environment with Apache, PHP, and Flex Builder, while the latter, in addition to the Web server, required for testing purposes, also installs MySQL Server and a Workbench for SQL development. Both RFCs have been executed to deploy these changes over 24 workstations of two development labs (12 successful executions each RFC).

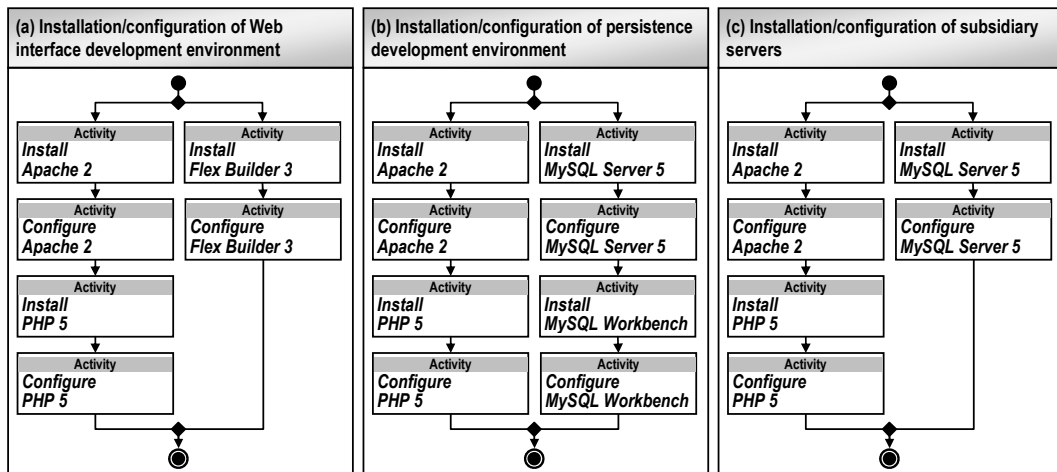


Figure 5.3: Change Plans for installation/configuration of development and production environments

Once the new version of the automation system is ready to be deployed, the IT change management team has to design a new RFC to prepare the 20 servers, each one located on a different subsidiary, to receive this new software. The RFC designed for such change, detailed in Figure 5.3 (c), is supposed to be deployed in all subsidiaries in two phases (being 10 subsidiaries per phase). This RFC describes that Apache, PHP, and MySQL must be installed on each subsidiary’s server. The configuration activities for the three software involved are manual, hence they must have humans associated. In this example, two human roles are defined: the Senior Operator, who performs MySQL and Apache configuration, and the Junior Operator, who is in charge of configuring PHP. Although such RFC has never been executed (therefore it has no execution records for analysis) some of its activities have been performed a number of times in similar RFCs. Intuitively, one may realize that RFC (c) looks more like (b) than it does to (a), since RFCs (c) and (b) have 6 activities in common, while (c) and (a) have only 4. This similarity is captured by the RA calculation (considering software, computers, and humans). For example, activity *Configure PHP* from RFC (c) has a RA of 0.43 comparing to *Configure PHP* from RFC (b) (in regards to AFs), while the RA factor is 0.33 comparing to

the same activity in RFCs (c) and (a).

The *Risk Report* automatically generated for RFC (c) before the deployment of the 10 servers in first phase is presented in Table 5.1. In this report, one may notice that the riskier activities are those performed by humans, being activity *Configure PHP*, which is executed by the Junior Operator, the one that requires special attention. Another important fact to mention is that all categories assigned to activities range between 4 and 6. This basically happens because the impact of changes is measured considering the relevance of services affected by the change. In this case study, all subsidiaries' servers have the same *Business Relevance (BsR)* values resulting always the same impact value; which in this case is medium.

Table 5.1: Risk Reports before the deployment of first phase

Activity		AF	RF	HF
Configure PHP	Probability	5.0%	0.0%	29.5%
	Impact	0.05	0.05	0.05
	Category	5	6	4
Configure Apache	Probability	7.3%	0.0%	4.4%
	Impact	0.05	0.05	0.05
	Category	5	6	5
Configure MySQL	Probability	10.0%	0.0%	1.8%
	Impact	0.05	0.05	0.05
	Category	5	6	6
Install Apache	Probability	8.7%	0.0%	–
	Impact	0.05	0.05	–
	Category	5	6	–
Install PHP	Probability	7.9%	0.0%	–
	Impact	0.05	0.05	–
	Category	5	6	–
Install MySQL	Probability	0.0%	0.0%	–
	Impact	0.05	0.05	–
	Category	6	6	–

Supposing that a *Change Authority* has analyzed the *Risk Report* of Table 5.1 and decided to deploy the RFC as it is, then, in the first deployment phase 10 of the subsidiaries' servers are successfully installed. By the end of this phase, the total SD caused by the change deployment reaches a value of 6.68. This value is mostly influenced by activity *Configure PHP*, which has the worst risk categories. This activity is particularly harmful because it is executed in a later moment on the workflow, hence its failure causes other activities to rollback.

Aiming at reducing SD for the second phase, an *Operator* may suggest modifications in the original CP based on the results generated by the automated risk assessment. For instance, a more experienced human could be reallocated to the riskier activity. Therefore, for the second phase, the RFC was adapted allocating the Senior Operator to configure PHP and the Junior Operator to configure Apache. Table 5.2 shows the *Risk Report* of the RFC with humans reallocated. In this report, it is possible to visualize the reduction of risk categories calculated for the activity

Configure PHP, whereas *Configure Apache* goes the other way around. After the RFC is adjusted, the second phase is deployed, reaching a total SD factor of 4.11. This represents a decrease of 38.47% in the total SD when comparing phases 1 and 2, indicating that the modification of the CP based on automated risk assessment reports has effectively decreased the risks associated to the requested change.

Table 5.2: Risk Reports before the deployment of second phase

Activity		AF	RF	HF
Configure Apache	Probability	6.8%	0.0%	28.8%
	Impact	0.05	0.05	0.05
	Category	5	6	4
Configure PHP	Probability	11.2%	0.0%	8.9%
	Impact	0.05	0.05	0.05
	Category	5	6	5
Configure MySQL	Probability	3.5%	0.0%	8.9%
	Impact	0.05	0.05	0.05
	Category	5	6	5
Install Apache	Probability	6.3%	0.0%	–
	Impact	0.05	0.05	–
	Category	5	6	–
Install PHP	Probability	17.9%	0.0%	–
	Impact	0.05	0.05	–
	Category	5	6	–
Install MySQL	Probability	0.0%	0.0%	–
	Impact	0.05	0.05	–
	Category	6	6	–

5.2 Application to IT Project Management

The second case study, applied to the context of IT Project Management, is targeted to present how it is possible to generate more comprehensive and interactive reports based on the proposed framework. This time, a case study considering a hypothetical software development project was conducted. Also, a database was designed containing synthetic information about workflows from other projects, execution of activities, and documented adverse events. In this section, in a first moment, the model proposed to represent the structure of the life cycle of IT projects is described. Then, the characteristics of the hypothetical project used as example in this case study is presented. Afterwards, comprehensive *Risk Reports* automatically generated by the solution are shown under two different perspectives: Project Hierarchy View and Work Plan View.

5.2.1 IT Project Life Cycle Information Model

In order to enable proper management and reuse of knowledge of IT projects, including management of risk and other aspects, it is important for organizations to document all activities of developed projects employing a single consistent information model. As far as the author of this dissertation is aware of, there is

no widely accepted model for representing management related information of IT projects available in the literature. Therefore, in this work, such a model is proposed – depicted in Figure 5.4 – inspired in a Business Technology Optimization (BTO) software from Hewlett-Packard (HP) called HP Quality Center (HP, 2009).

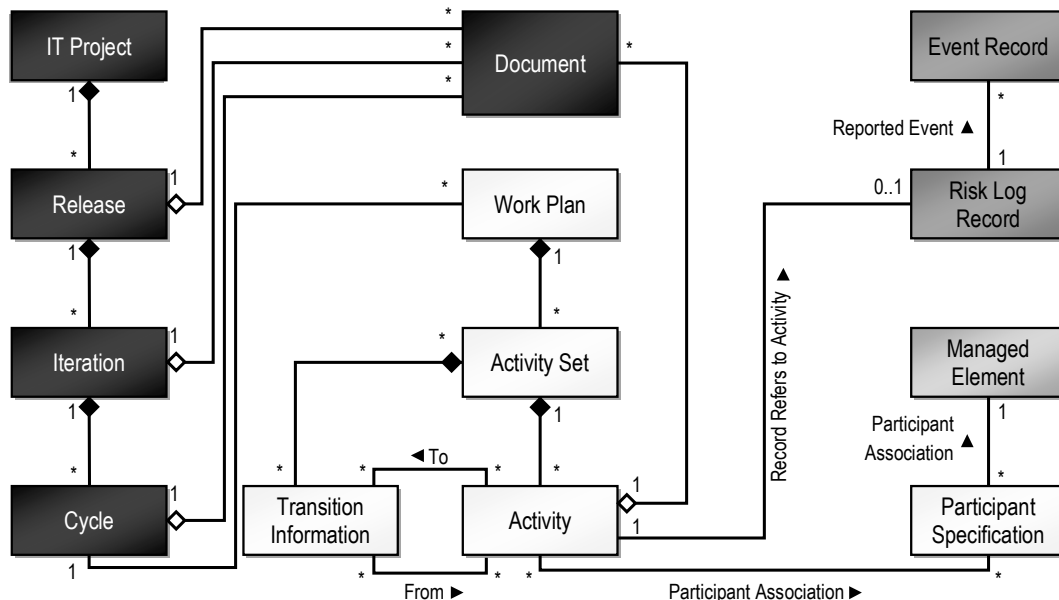


Figure 5.4: IT Project Life Cycle Information Model

Every *IT Project* may be delivered to the final customer through one or more *Releases*. Each *Release* is a partial version of the product or service being designed/developed in the project. It contains a set of functionalities fully developed and tested that may be validated or sometimes put into production by the customer. These functionalities are planned and implemented in one or more *Iterations*. The *Cycles* associated to each *Iteration* will often vary according to the methodology adopted. For example, the *Cycles* of an *Iteration* could be Analysis, Project, Development, and Testing.

In order to organize the *Activities* that have to be performed in each *Cycle*, one or more *Work Plans* have to be defined. Indeed, *Work Plans* are workflows of *Activities* also following the Workflow Process Definition, proposed by the WfMC and presented in Figure 4.1. An *Activity* consumes a certain amount of resources and takes some time to be executed. The *Participant Specification* class associates activities to the allocated resources (*e.g.*, humans or computers). The participants of activities refer to the *Managed Element* class, linking this model with the CIM (used to represent the elements available in the IT infrastructure).

Just like the information model to represent RFCs (presented in Figure 5.1), two classes were introduced particularly to enable logging of activities for automation of risk assessment, namely: *Risk Log Record* and *Event Record*. Every *Activity* performed in a *Work Plan* should have an instance of *Risk Log Record* associated to it in order to indicate the details of its execution. The execution of an *Activity* may trigger events (adverse or favorable). These events are documented in instances of the class *Event Record*, which also contain information about categorization (in the context of IT projects event categories are project objectives, *e.g.*, cost, time, scope, or quality), and the severity measured (*e.g.*, amount of hours delayed in activity).

5.2.2 Hypothetical Project Structure

The goal of the studied project is to develop a system for monitoring, supervision, incident reporting, and problem diagnosis on large-scale corporative networks. The purpose of this system is to provide a company with support for management of an IT infrastructure inventory, monitoring, and supervision of Configuration Items (CIs) (*e.g.*, routers, computers, software packages, and services), and also record incidents involving these CIs, assisting the problem diagnosis process. According to high level definitions of requirements for the project, a project manager split the development efforts into four releases, as follows:

- **Release 1:** Monitoring and supervision basic features;
 - **Iteration 1:** Database modeling to allow composition of IT infrastructure inventory;
 - **Iteration 2:** Development of server-side core module application;
 - **Iteration 3:** Development of client-side core module application;
 - **Iteration 4:** Development of server-side graphical Web interface basic operations;
- **Release 2:** Monitoring and supervision advanced features;
 - **Iteration 1:** Development of server-side advanced reports composer;
 - **Iteration 2:** Development of server-side analytical multivariable graphics module;
- **Release 3:** Monitoring and supervision integration;
 - **Iteration 1:** Development of server-side SNMP support module;
 - **Iteration 2:** Development of server-side Web Services support module;
- **Release 4:** Incident reporting and problem diagnosis;
 - **Iteration 1:** Database modeling for incident reporting;
 - **Iteration 2:** Development of incident reporting Web interface;
 - **Iteration 3:** Development of problem diagnosis tool.

In Release 1, basic functionalities of the system are implemented. In its first iteration, the database to allow representation of CIs from the IT infrastructure is modeled. The core of the system works as a client-server application, where the server requests/receives information about managed clients installed in CIs. The Web interface basic features are also delivered in first release, such as CRUD (Create, Request, Update, and Delete) operations over registered objects. Advanced features, such as reports composition (*e.g.*, availability, network load and latency, and alarms) and graphs for data visualization, are left to the second release. In the third release, modules for integration with Simple Network Management Protocol (SNMP) and Web services are included to enable management of devices that support those management interfaces. Finally, in the fourth release, incident reporting interface and a diagnosis tool are added in order to allow association of reported incidents and problems with corresponding defective CIs. Although not detailed above, every iteration of the project is divided into four cycles: Analysis, Project, Development, and Testing.

5.2.3 Comprehensive Risk Reports

The project analyzed in this case study contains 141 activities disposed in 44 work plans. Since the automated risk assessment calculates four risk categories (one for each affected objective) for all activities of the project, a *Risk Report* as previously shown in Table 4.4 could not be practical to help on decision making for risk response planning. Instead, summarization of these information might be employed in order to generate more comprehensive reports under two perspectives: (i) Project Hierarchy View (Figure 5.5), which gives an interactive overview of risks using the project hierarchical structure, and (ii) Work Plan View (Figure 5.6), useful to investigate particularly risky work plans aiming to understand the sources of risk.

Project Hierarchy View				
Adverse Risks Report	Cost	Time	Scope	Quality
Project	4.84	4.25	5.48	6.66
- Release 1: Monitoring and supervision basic features	3.93	3.11	4.93	6.31
+ Iteration 1: Database modeling to allow composition of IT infrastructure inventory	6.16	3.44	6.08	8.23
+ Iteration 2: Development of server-side core module application	4.46	5.34	6.65	7.45
+ Iteration 3: Development of client-side core module application	7.02	5.76	5.44	8.01
- Iteration 4: Development of server-side graphical Web interface basic operations	2.65	2.06	3.84	4.80
+ Cycle 1: Analysis	6.32	6.62	6.34	3.46
+ Cycle 2: Project	6.30	6.55	5.92	7.10
+ Cycle 3: Development	1.37	1.33	2.40	4.44
+ Cycle 4: Testing	5.90	1.41	5.92	7.10
- Release 2: Monitoring and supervision advanced features	5.25	6.07	6.15	7.48
+ Iteration 1: Development of server-side advanced reports composer	4.31	6.25	6.88	7.02
+ Iteration 2: Development of server-side analytical multivariable graphics module	6.70	5.89	5.55	8.01
- Release 3: Monitoring and supervision integration	5.45	5.55	5.99	7.72
+ Iteration 1: Development of server-side SNMP support module	4.46	5.34	6.65	7.45
+ Iteration 2: Development of server-side Web Services support module	7.02	5.76	5.44	8.01
- Release 4: Incident reporting and problem diagnosis	6.49	5.61	5.81	6.19
+ Iteration 1: Database modeling for incident reporting	6.08	5.39	5.71	5.72
+ Iteration 2: Development of incident reporting web interface	7.02	6.81	5.93	5.17
+ Iteration 3: Development of problem diagnosis tool	6.43	4.94	5.79	8.61

Figure 5.5: Comprehensive Risk Report in Project Hierarchy View

As shown in Figure 5.5, a project manager can interactively choose which part of the project he/she wants to inspect with more details. For example, by expanding (+) an iteration the risks calculated for all of its cycles are displayed. Analyzing this hierarchical report one could notice that, among all releases, the first one holds most of the risks from the hypothetical project analyzed in this case study. Inspecting Release 1, a project manager may figure out that Interaction 4 requires special attention because of its risk factors in all objectives. Observing the cycles of Interaction 4, it is possible to notice that risks of different objectives are mostly distributed among Cycles 1, 3, and 4. Cost and Scope risks are negatively influenced by Cycle 3, Time risks are shared between Cycles 3 and 4, and Quality risks are more evidenced in Cycle 1. A report with these characteristics indicates that, in past similar projects automatically analyzed by the framework, events were reported evidencing poor quality in activities of analysis. That might have caused other adverse events to happen, affecting cost and time of later development and testing cycles.

Whenever a project manager needs to inspect with more details some of the work plans from the project, the Work Plan View may be used. In Figure 5.6, one

work plan from Cycle 3 (Development) of the fourth iteration from the hypothetical project is shown. Similarly to the Project Hierarchy View, in Work Plan View it is also possible to provide more summarized or more detailed visualizations of risk information by exploiting the structure of the workflow depending on the needs of the project manager. In the left part of Figure 5.6, three activity sets describe steps required to implement basic functionalities of the Web interface of the earlier described system. Initially, administration of credentials (*e.g.*, login forms, users names, passwords, and access rights) and system menu structure (*e.g.*, sections and subsections) are developed in *AS1*. In a subsequent moment, two parallel branches are started moving into *AS2* and *AS3*. Both branches develop DAOs (Data Access Objects), for persistence of objects in a relational database, and development of Web forms for CRUD operations of CIs and their categories. In the right part of Figure 5.6, risk classifications automatically assigned to each finer-grained activity are displayed next to them, providing the highest level of detail about each activity set. This visualization helps the identification of problematic activities that might compromise the success of each work plan of a project.

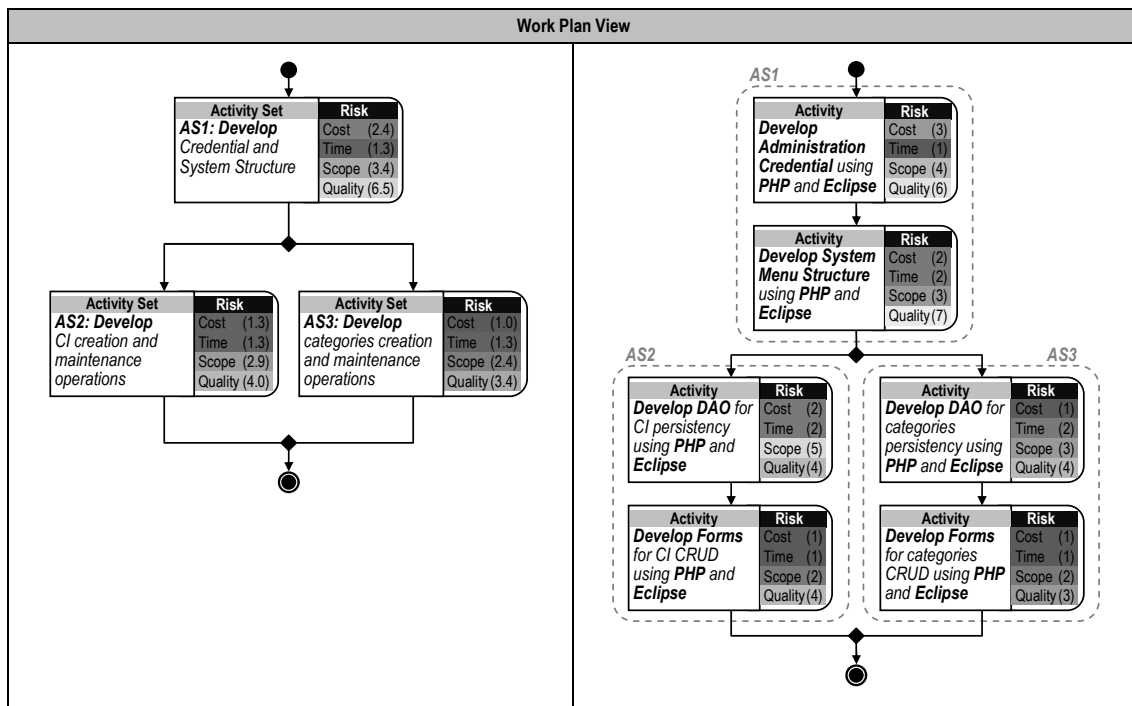


Figure 5.6: Comprehensive Risk Report in Work Plan View

One important fact is that, despite the attenuation caused by the summarization of risk classifications, automatically calculated risks of activities still reflect very well in upper levels of the project. This is clearly visible particularly in Project Hierarchy View (Figure 5.5) used as example in this case study. Some activities from different cycles in Interaction 4 had high risk rating (low categories) and this reflected in high risks for the whole Release 1. Based on these reports a project manager could prioritize risks and establish directions for risk response. For example, one strategy could be addressing risks of a project by iteration. Then, a threshold may be specified defining that preventive actions (contingency plans) are required for iterations with risk factors below 5, and corrective actions (workarounds) for iterations that exceed this value.

6 CONCLUSION

In this dissertation, the current need of organizations to enforce rational practices for IT infrastructures and services management has been discussed. Some standards and libraries of best practices, commonly employed by these organizations in order to better organize their internal processes, have been also presented. Two of them have been highly important to the context of this research and served as reference for most of the concepts developed. The first one is ITIL from OGC, which is a compilation of best practices for the management of IT infrastructures and services in general. The second one is PMBOK from PMI, more focused in the management of IT projects' life cycle.

Among many management aspects covered by these standards, the concern with risk management is so remarkable that some publishers have released specific guides of best practices targeted to manage risks of IT processes. Guidelines from both the M_o_R framework (also from OGC) and the Project Risk Management knowledge area (from PMBOK) head the efforts of many modern organizations that want to tackle their risks. Despite all guidelines and best practices provided by these standards, this research has shown that, in practice, the adoption of risk management procedures is performed in a very *ad hoc* fashion. Lack of automation, standardization, and knowledge reuse are some of the causes that turn risk management inefficient and sometimes counterproductive in actual environments.

In this master's dissertation it was introduced a novel framework with the objective of helping in the risk management process, particularly focusing in workflow-based IT management systems. This objective is pursued, in a first moment, by gathering risk related information from the execution records of past workflows and learning from them in order to assess probability and impact factors of risky events. This kind of data gathering procedure, when performed only based on human experience, tends to be time/resource consuming and sometimes too imprecise to guide decision making. Another relevant contribution of the proposed framework is that risk information is organized in interactive and comprehensive reports. This enables operators/managers to have an overview of the risks automatically assessed in different levels of detail, helping quick identification of threats and efficient directing of risk mitigation efforts.

The proposed framework had its applicability evaluated under two different scenarios: IT Change Management and IT Project Management. These are environments where workflow management systems are commonly employed in order to control and organize operations and resources, usually involving complex IT infrastructures. Although not exhaustive, the case studies presented in each scenario have shown that the framework is generic enough to be applicable to different environ-

ments and how it can still be customized to better reflect specific needs in each situation.

6.1 Main Contributions and Results Obtained

The main contribution of this research is the proposed framework itself and the way risk related information flows through its modules independently of how each module internally performs calculations. As previously mentioned, the framework has the objective of helping on risk management by automating certain procedures, such as data gathering for estimations of probability and impact. This is a too complex problem to tackle with one single and monolithic solution. The approach of creating a modular framework enables breaking the whole problem down into smaller and less complex parts that can be handled individually. Adopting such approach makes it also easier to customize some parts of the framework in order to better reflect the needs of a particular environment, as discussed in the first case study (Section 5.1).

Moreover, there are some other contributions that are worth mentioning. First, classifications of events that represent risks have been proposed for the contexts of the two case studies presented. These classifications have shown to be useful to group events together reflecting the concerns of operators/mangers, thus making the results of risk assessment more meaningful. Additionally, a strategy to calculate similarity among workflows has been proposed enabling knowledge reuse in automated risk assessment even when analyzing newly designed workflows. Different algorithms have been presented to calculate probabilities and impacts of events considering the nuances of the analyzed environment. Finally, strategies to categorize and summarize risk information aiming to present more comprehensive and interactive risk reports have been proposed.

The results obtained, although not exhaustive, have shown how the proposed framework can be adapted and adopted in two different scenarios. Specifically in the context of IT Change Management, the risk reports have shown to be quite interesting to help operators in quickly identifying threats in Change Plans (CPs), enabling proactive problem treatment. Furthermore, a metric of Service Disruption (SD) was employed to compare the performance of different CPs that revealed distinct risk reports. The mitigation of risks in the first case study has caused an improvement in the SD factor, which indicates that risk reports reflect real threats to the services supported.

In the second case study, the focus was to show how comprehensive and interactive reports may help project managers to address risks in the context of IT Project Management. Usually, in IT projects, the risk assessment process is performed in meetings, interviews, and brainstorming with involved stakeholders. By employing the proposed framework it is possible to speed up this process, since analysis are based on information retrieved from a database of previously executed projects, not requiring any human intervention. Remarkably, the generated reports organize information according to the project hierarchical structure, facilitating the identification of risks in each of its phases. This case study also shows that the proposed risk information summarization strategy achieves its objective, since combining risk information from activities of work plans and displaying this information into higher levels of the project hierarchy does not hide relevant risks computed from lower levels.

6.2 Final Remarks and Future Work

In this work, much has been discussed about automation and how the proposed framework is capable of calculating risks of activities in workflows without human intervention. One first substantial consideration should be made about the role of human managers/operators in risk management processes regarding the adoption of the this framework. Firstly, it is important to emphasize that the framework has been proposed with the objective of accelerating risk management processes by supporting the tasks that possibly consume more time/resources especially because of manual work (*e.g.*, gathering data from past projects for risk assessment). Despite the automation levels achieved, skilled operators/managers are still required to provide the best input parameters aligned with each analyzed environment. Definitions like probability and impact ranges, risk classification matrices, and event categorization are essential for the results to be meaningful. Furthermore, since the framework is based on the analysis of information documented in previous executions of workflows, it is also imperative that this data is organized following a well defined model such as the ones presented in Chapter 4.

Above all human responsibilities, the one that might be the most relevant is the ability to draw decisions over results of risk assessment. The role of the proposed framework is limited to support the decision making by presenting comprehensive risk reports. However, it is up to managers/operators to interpret these reports and plan actual responses to these risks. Future investigations could explore further strategies to enhance decision support in the framework, for instance, suggesting modifications on workflows to reduce risks based on previous similar mitigation actions taken.

A very important metric employed by the framework to assess risks is the similarity among the activities of workflows. A first approach to this subject was introduced in this dissertation, but future investigations could explore other strategies to calculate this metric. One known drawback of the *Risk Affinity (RA)* metric proposed in the context of this research is that it does not consider the structure of the *Influential Workflows (IW_s)*; it only calculates the similarity of all activities that compose them. In fact, this research on new ways of calculating similarity has started and already published its preliminary results (BIANCHIN et al., 2010). Furthermore, it is remarkable that similarity may be useful to other situations rather than risk assessment. It could certainly be interesting to estimate possible cost or time of workflows based on the history of previously executed ones, similarly to what has been performed for probability and impact in this work.

Results presented in this dissertation, although not exhaustive, seem to be promising in the contexts of IT Change Management and IT Project Management. Future investigations could extend the framework and apply it to other types of management systems, as long as they are based on workflows. Also as future work, it would be of great value to use data from real life IT management systems in order to better evaluate the applicability of the proposed framework to each specific context. Moreover, it would be interesting to conduct a survey and receive feedback from experienced managers, operators, and other personnel involved in IT operations to evaluate the usability of the proposed risk reports.

REFERENCES

Active Endpoints. **ActiveBPEL Open Source Engine**. Disponível em: <<http://www.activebpel.org>>. Acesso em: November 2008.

BAKKER, K.; BOONSTRA, A.; WORTMANN, H. Does risk management contribute to IT project success? A meta-analysis of empirical evidence. **International Journal of Project Management**, [S.l.], v.In Press, Corrected Proof, p.–, 2009.

BIANCHIN, L. A.; WICKBOLDT, J. A.; SANTOS, R. L.; LUNARDI, R. C.; DALMAZO, B. L.; ANDREIS, F. G.; CORDEIRO, W. L. C.; SOUSA, A. L. R.; GRANVILLE, L. Z.; GASPARY, L. P. Similaridade para Avaliação de Riscos em Planos de Mudança de TI. In: WORKSHOP DE GERÊNCIA E OPERAÇÃO DE REDES E SERVIÇOS (WGRS/SBRC), 2010, Gramado, Brazil. **Proceedings...** [S.l.: s.n.], 2010. p.103–116.

CHICKEN, J.; POSNER, T. **The Philosophy of Risk**. [S.l.]: Thomas Telford, 1998.

CORDEIRO, W. L. C.; MACHADO, G. S.; ANDREIS, F. G.; SANTOS, A. D.; BOTH, C. B.; GASPARY, L. P.; GRANVILLE, L. Z.; BARTOLINI, C.; TRASTOUR, D. ChangeLedge: Change Design and Planning in Networked Systems based on Reuse of Knowledge and Automation. **Computer Networks**, [S.l.], 2009.

CORDEIRO, W. L. C.; MACHADO, G. S.; DAITX, F. F.; BOTH, C. B.; GASPARY, L. P.; GRANVILLE, L. Z.; SAHAI, A.; BARTOLINI, C.; TRASTOUR, D.; SAIKOSKI, K. A Template-based Solution to Support Knowledge Reuse in IT Change Design. In: IEEE/IFIP NETWORK OPERATIONS AND MANAGEMENT SYMPOSIUM (NOMS), 2008, Salvador, Brazil. **Proceedings...** [S.l.: s.n.], 2008. p.355–362.

DANAEI, G.; VANDER HOORN, S.; LOPEZ, A.; MURRAY, C.; EZZATI, M. Causes of cancer in the world: comparative risk assessment of nine behavioural and environmental risk factors. **The Lancet**, [S.l.], v.366, n.9499, p.1784–1793, 2005.

DMTF. **CIM - Common Information Model**. Disponível em: <<http://www.dmtf.org/standards/cim>>. Acesso em: November 2008.

DUMAS, M.; VAN DER AALST, W.; TER HOFSTEDÉ, A. **Process-aware Information Systems**. [S.l.]: Wiley-Interscience, 2005.

- FENTON, N.; OHLSSON, N. Quantitative analysis of faults and failures in a complex software system. **IEEE Transactions on Software Engineering**, [S.l.], v.26, n.8, p.797–814, 2000.
- FEWSTER, R.; MENDES, E. Measurement, prediction and risk analysis for Web applications. In: INTERNATIONAL SOFTWARE METRICS SYMPOSIUM, 2001. METRICS 2001, 2001. **Proceedings...** [S.l.: s.n.], 2001. p.338–348.
- FROOT, K.; SCHARFSTEIN, D.; STEIN, J. Risk management: Coordinating corporate investment and financing policies. **Journal of Finance**, [S.l.], p.1629–1658, 1993.
- HEARTY, P.; FENTON, N.; MARQUEZ, D.; NEIL, M. Predicting Project Velocity in XP Using a Learning Dynamic Bayesian Network Model. **IEEE Transactions on Software Engineering**, [S.l.], v.35, n.1, p.124–137, Jan.-Feb. 2009.
- HOLTON, G. **Value-at-Risk: Theory and Practice**. [S.l.]: Academic Pr, 2003.
- HP. **HP Quality Center**. Disponível em: <https://h10078.www1.hp.com/cda/hpms/display/main/hpms_content.jsp?zn=bto&cp=1-11-127-24_4000_100__>. Acesso em: November 2009.
- IEEE. **IEEE Std 1490-2003 - Guide Adoption of PMI Standard a Guide to the Project Management Body of Knowledge (Revision of IEEE Std 1490-1998)**. Available at: <http://standards.ieee.org/reading/ieee/std_public/description/se/1490-2003_desc.html>. Acesso em: July 2010.
- IRM. **A Risk Management Standard**. United Kingdom: The Institute of Risk Management, 2002.
- ISACA. **The Risk IT Practitioner Guide**. Illinois, USA: Information Systems Audit and Control Association, 2009.
- ISACA. **Control Objectives for Information and related Technologies (COBIT)**. Disponível em: <<http://www.isaca.org/cobitonline/>>. Acesso em: January 2010.
- ISO. **ISO 31000:2009 Risk management – Principles and Guidelines**. Geneva, Switzerland: International Organization for Standardization, 2009.
- KELLER, A. Automating the change management process with electronic contracts. In: IEEE INTERNATIONAL CONFERENCE ON E-COMMERCE TECHNOLOGY WORKSHOPS, 2005, München, Germany. **Proceedings...** [S.l.: s.n.], 2005. p.99–107.
- KELLER, A.; HELLERSTEIN, J.; WOLF, J.; WU, K.; KRISHNAN, V. The CHAMPS system: Change management with planning and scheduling. In: IEEE/IFIP NETWORK OPERATIONS AND MANAGEMENT SYMPOSIUM (NOMS), 2004, Seoul, Korea. **Proceedings...** [S.l.: s.n.], 2004. v.1, p.395–408.
- KLÜPPELBERG, C.; KOSTADINOVA, R. Integrated insurance risk models with exponential Lévy investment. **Insurance Mathematics and Economics**, [S.l.], v.42, n.2, p.560–577, 2008.

KUTSCH, E.; HALL, M. Deliberate ignorance in project risk management. **International Journal of Project Management**, [S.l.], v.28, n.3, p.245–255, 2010.

LUU, V.; KIM, S.; TUAN, N.; OGUNLANA, S. Quantifying schedule risk in construction projects using Bayesian belief networks. **International Journal of Project Management**, [S.l.], v.27, n.1, p.39–50, 2009.

MACHADO, G. S.; CORDEIRO, W. L. C.; DAITX, F. F.; BOTH, C. B.; GASPARY, L. P.; GRANVILLE, L. Z.; BARTOLINI, C.; SAHAI, A.; TRASTOUR, D.; SAIKOSKI, K. Enabling Rollback Support in IT Change Management Systems. In: IEEE/IFIP NETWORK OPERATIONS AND MANAGEMENT SYMPOSIUM (NOMS), 2008, Salvador, Brazil. **Proceedings...** [S.l.: s.n.], 2008. p.347–354.

MACHADO, G. S.; CORDEIRO, W. L. C.; SANTOS, A. D.; WICKBOLDT, J. A.; ROBEN CASTAGNA LUNARDI, F. G. A.; BOTH, C. B.; GASPARY, L. P.; GRANVILLE, L. Z.; TRASTOUR, D.; BARTOLINI, C. Refined Failure Remediation in IT Change Management Systems. In: IFIP/IEEE INTERNATIONAL SYMPOSIUM ON INTEGRATED NETWORK MANAGEMENT (MINI-CONFERENCE OF IM), 2009, New York, NY. **Proceedings...** [S.l.: s.n.], 2009. p.638–645.

MARQUES, M.; NEVES-SILVA, R. Risk Assessment to Support Decision on Complex Manufacturing and Assembly Lines. In: IEEE INTERNATIONAL CONFERENCE ON INDUSTRIAL INFORMATICS, 2007. **Proceedings...** [S.l.: s.n.], 2007. p.1209–1214.

OASIS. **BPEL - Business Process Execution Language - Version 2.0**. Disponível em: <<http://docs.oasis-open.org/wsbpel/2.0/>>. Acesso em: December 2008.

OGC. **Management of risk: guidance for practitioners**. London, UK: Office of Government Commerce, 2007.

OGC. **Information Technology Infrastructure Library: Service Transition Version 3.0**. London, UK: Office of Government Commerce, 2007.

OGC. **Information Technology Infrastructure Library: Service Design Version 3.0**. London, UK: Office of Government Commerce, 2007.

OGC. **Information Technology Infrastructure Library (ITIL)**. Disponível em: <<http://www.ital-officialsite.com/>>. Acesso em: January 2010.

PMI. **A Guide to the Project Management Body of Knowledge: PMBOK Guide – Third Edition**. Pennsylvania, USA: Project Management Institute, 2004.

REBOUÇAS, R.; SAUVÉ, J.; MOURA, A.; BARTOLINI, C.; TRASTOUR, D. A Decision Support Tool to Optimize Scheduling of IT Changes. In: IFIP/IEEE INTERNATIONAL SYMPOSIUM ON INTEGRATED NETWORK MANAGEMENT (IM), 2007, Munich, Germany. **Proceedings...** [S.l.: s.n.], 2007. p.343–352.

RUSSELL, N.; AALST, W. van der; HOFSTEDÉ, A. ter. **Exception Handling Patterns in Process-Aware Information Systems**. [S.l.: s.n.], 2006.

SAUVÉ, J.; SANTOS, R. A.; ALMEIDA, R. R.; MOURA, J. A. B. On the Risk Exposure and Priority Determination of Changes in IT Service Management. In: IFIP/IEEE INTERNATIONAL WORKSHOP ON DISTRIBUTED SYSTEMS: OPERATIONS AND MANAGEMENT (DSOM), 2007, San Jose, CA. **Proceedings...** [S.l.: s.n.], 2007. p.147–158.

SETZER, T.; BHATTACHARYA, K.; LUDWIG, H. Decision Support for Service Transition Management: Enforce Change Scheduling by Performing Change Risk and Business Impact Analysis. In: IEEE/IFIP NETWORK OPERATIONS AND MANAGEMENT SYMPOSIUM (NOMS), 2008, Salvador, Brazil. **Proceedings...** [S.l.: s.n.], 2008. p.200–207.

WANG, L.; SAHAI, A.; PRUYNE, J. **A Model-based Simulation Approach to Error Analysis of IT Services**. Palo Alto, CA: Enterprise Systems and Software Laboratory - HP Laboratories Palo Alto, 2006. (181).

WFMC. **Workflow Process Definition Interface - XML Process Definition Language**. Disponível em: <http://www.wfmc.org/standards/docs/TC-1025_10_xpdl_102502.pdf>. Acesso em: November 2008.

WYK, R. van; BOWEN, P.; AKINTOYE, A. Project risk management practice: The case of a South African utility company. **International Journal of Project Management**, [S.l.], v.26, n.2, p.149–163, 2008.

APPENDIX A PUBLISHED PAPER – IM 2009

In this appendix the paper entitled “A Solution to Support Risk Analysis on IT Change Management” is presented. This was the first deliverable of the research presented in this dissertation which introduced an initial approach to risk assessment in the context of IT Change Management. In addition, a model to represent execution traces of workflows and classification of failures in changes have been proposed. The solution introduced in this paper was implemented as a component of the CHANGE-LEDGE system, named *Risk Analyzer*. A case study much focused in estimations of impact over a hypothetical IT infrastructure has shown that the solution was able to capture the actual impact of a given change over Configuration Items (CIs) affected directly and indirectly. So far, only two failure types were considered in case studies: Activity Failures (AF) and Resource Failures (RF).

- **Title:**
A Solution to Support Risk Analysis on IT Change Management
- **Conference:**
11th IFIP/IEEE International Symposium on Integrated Network Management (IM 2009)
- **URL:**
<http://www.ieee-im.org/2009/>
- **Date:**
1-5 June 2009
- **Venue:**
Hofstra University, Long Island, New York, USA
- **Digital Object Identifier (DOI):**
<http://dx.doi.org/10.1109/INM.2009.5188847>

A Solution to Support Risk Analysis on IT Change Management

Juliano Araújo Wickboldt*, Guilherme Sperb Machado*,
Weverton Luis da Costa Cordeiro*,
Roben Castagna Lunardi*, Alan Diego dos Santos*,
Fabrício Girardi Andreis*, Cristiano Bonato Both*,
Lisandro Zambenedetti Granville*,
Luciano Paschoal Gasparly*

*Institute of Informatics, UFRGS - Brazil

{jwickboldt, gsmachado, wlccordeiro, rclunardi, adsantos,
fgandreis, cbboth, granville, paschoal}@inf.ufrgs.br

Claudio Bartolini[†],
David Trastour[‡]

[†]HP Laboratories Palo Alto, USA

[‡]HP Laboratories Bristol, UK
{claudio.bartolini, david.trastour}@hp.com

Abstract—The growing necessity of organizations in using technologies to support to their operations implies that managing IT resources became a mission-critical issue for the health of the primary companies' businesses. Thus, in order to minimize problems in the IT infrastructure, possibly affecting the daily business operations, risks intrinsic to the change process have to be analyzed and assessed. Risk Management is a widely discussed subject in several areas, although for IT Change Management it is quite a new discipline. The Information Technology Infrastructure Library (ITIL) introduces a set of best practices to conduct the management of IT infrastructures. According to ITIL, risks should be investigated, measured, and mitigated before any change is approved. Even with these guidelines, there is no default automatic method for risk assessment in IT Change Management. In this paper we introduce a risk analysis method based on the execution history of past changes. In addition, we propose a failure representation model to capture the feedback of the execution of changes over IT infrastructures.

I. INTRODUCTION

In the complex IT scenarios of modern companies and organizations, change management is the discipline that organizes how the IT infrastructure evolves in a consistent and safe way. Change management encompasses the request, planning, deployment, and assessment of required changes in IT environments. The Information Technology Infrastructure Library (ITIL) specifically tackles change management by presenting a set of best practices that guide the management of IT infrastructures in an appropriated way [1].

According to ITIL Service Transition book [2], every change that needs to be performed should be described in documents named Requests for Change (RFCs). Such documents describe what needs to be changed, the reasons for changing, the target of the changes (referred as Configuration Items - CIs), responsible personnel, and an identification number of the change request. However, an RFC does not inform how the change should be performed, *i.e.*, the steps to be followed to achieve that. Additionally, all RFCs must be submitted to the Change Advisory Board (CAB) to be analyzed, approved, and scheduled by this council before they are deployed.

After the submission of a new RFC, the next step on the change process is the design of a preliminary Change Plan (CP), which is essentially a workflow of high-level activities (planned by a human operator) that must be executed in order to achieve the change requested in the original RFC. The preliminary CP is then further refined by a change management system, generating as a result another workflow composed of finer-grained activities that are ready to be executed. At the end of such execution the managed infrastructure is supposed to have evolved to a new consistent state complying with the changes requested in the original RFC.

The occurrence of failures during the aforementioned process, however, is clearly possible. In that case, failures should invoke associated remediation plans, whose objective is to treat the occurred problem and avoid the system of evolving to an unknown state. Rollback and compensation plans are examples of remediation strategies that must be designed and approved previously to the RFC deployment.

Since the necessity of changes arises, so do risks associated to it. According to ITIL, risks should be investigated, measured, and treated before a change is approved, in order to reduce as much as possible the chances of harming the business operations. Risks on change processes are of many natures. For instance, the occurrence of a failure on the installation of new software, the interference of agents external to the changing process, and damage of CIs, may lead the IT infrastructure to an inconsistent or undesirable state, and then cause losses to the company's business. Despite the ITIL's recommendations, there is no detailed specification of how to measure or assess risks; the only advice is that risks should be estimated previous to change deployments as a composition of the probability of a possible negative event to happen and its impact over the business in the case it becomes real.

Recently, several authors have been researching the automation of change processes [3]–[6] but the automatic assessment of risks associated to such changes, based on the history of previous executions, has not been investigated. An automated

risk analysis would allow the human operator to more precisely and quickly identify threats in a change plan, prior to its deployment, and thus reacting by either adapting the current change plan or switching to different one, always aiming to reduce the risks of the change process.

Another important ITIL's recommendation is that every change executed over a Configuration Item (CI) should be recorded to a log, in order to keep the history of modifications over the IT infrastructure. Some models can be used to represent IT information, such as the Common Information Model (CIM) [7] from the Distributed Management Task Force (DMTF). CIM is able to model – in addition to devices, services, people, and their relationships – the changes executed over CIs. However, CIM does not model failures within the change process nor their classification. In order to cope with ITIL's recommendation, it is needed a model to represent these information in an appropriate way.

To address the previously mentioned issues, in this paper we introduce a solution to analyze the history of changes deployed over an IT infrastructure, in order to extract information about risks and provide the human operator with support on decisions before an RFC is approved and executed. To achieve that, we first propose a model to represent failures in change processes. With this model populated with data from execution of change deployments, it will be possible to retrieve risk assessment investigating past failures and their impact.

The remainder of this paper is organized as follows. In Section II we review related work and the building blocks for the proposed solution. In Section III the model to represent traces and failures on change processes is presented. The proposed solution to analyze log records and extract risks, as well as a proposal of risk classification and representation are detailed in Section IV. A case study is placed in Section V, in order to evaluate the appliance of the proposed solution. Finally, this paper is closed in Section VI, where final remarks and future work are discussed.

II. RELATED WORK & BACKGROUND

Risk Management is a cross discipline since it applies to several different knowledge areas, such as, Software Engineering, Financial, and Medicine. Each research field has proposed its own risk analysis methods. However, even with all the current research on this subject, risk is a relatively new facet to be explored within the scope of IT change management. Firstly, in this section we will present some of the most relevant related work conducted in recent years. After that, the building blocks for the proposed solution will be introduced.

A. Related Work

Marques and Neves-Silva [8] proposed a risk assessment method to help the decision making on complex manufacturing assembly lines. At that work it was introduced a formula to estimate risk considering the probability of occurrence of a specific incident and the impact of that event. However, the authors assume that these two parameters (probability and impact) are known values defined for each event. This implies

that the system has to be previously configured to monitor a limited set of known events. For example, when an alarm is fired to indicate that some system variable (*e.g.*, mean-time-between-failure) overtook a regular threshold, there is a previously defined probability of the associated incidents and their impact in the case they really happen.

Fewster [9] has introduced a prediction model using a Generalized Linear Model (GLM) to estimate some Web design and authoring metrics. The paper focuses on the prediction of the effort to build a Web project. Nevertheless, the same GLM has shown to be a powerful tool to create a framework for risk management. The most interesting fact is that the statistical model provides not only a point for the analyzed variable, but a full probability distribution. Instead of estimating the total time for the project execution, it is possible to obtain the probability of not concluding it in, for example, 30 days. With that in mind, the project manager is able to determine a required risk level he/she is willing to deal with.

Sauvé *et al.* [4] and Rebouças *et al.* [3] have proposed a risk analysis on the change process to automatically determine priorities on the scheduling of various RFCs. In that work, it is employed a risk evaluation guided by the business objectives, in order to minimize the impact over the organization's services during the deployment of a change. According to the authors, the elapsed between the submission of an RFC and its implementation causes damage to the services affected by the change, which may suffer from performance degradation, for example. Moreover, during the deployment of an RFC, the disruption of services and breach of deadlines may cause financial losses or contractual penalties. However, risk analysis proposed in that work has application to the scheduling of changes, and not to its planning, as discussed in this paper.

On the failure representation subject, several researches are also available. Wang *et al.* [10] explains the four requirements to compose a model to well represent failures: (i) error categories and hierarchy should be represented, (ii) error models should be integrated without modifying models for existing components, (iii) component-specific error behaviors should be captured, and (iv) error propagations should be handled.

Russell and van der Aalst *et al.* [11] have proposed a conceptual framework to classify exception handling on process-aware information systems. The authors classify exceptions into five categories: Work Item Failure, Deadline Expiry, Resource Unavailability, External Trigger, and Constraint Violation. Furthermore, they have proposed strategies to handle exceptions under three considerations: how to handle exceptions on the work item level, how one exception affects other items of the same case, and what kind of recovery actions can be triggered to remediate the situation.

B. Conceptual Solution

A conceptual solution to conduct the change process, ranging from change specification and planning to its deployment, was proposed in previous works [5] [6]. In this section, we will introduce the aforementioned solution that was materialized in the CHANGELEDGE system. In Figure 1 the components

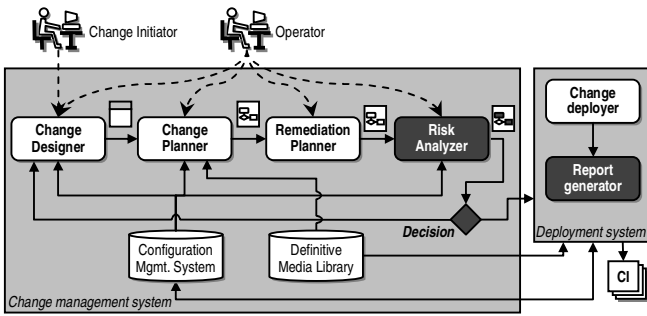


Fig. 1. Elements of the proposed solution

arranged to deliver risk assessment are distinguished by the darker boxes.

The *Change initiator* starts the change process by describing a new RFC interacting with the *Change designer* component. Once the RFC is specified, the *Operator* is responsible for designing a preliminary change plan (a workflow of high level activities), also interacting with the *Change designer*. On the second step of the lifecycle of an RFC, the *Change planner* is responsible for producing an actionable workflow of lower level activities, based on definitions made in the preliminary plan and information about configuration and software dependencies provided by the *Definitive Media Library* (DML). The algorithm to generate such refined change plan is out of the scope of this paper. The interested reader may refer to [5] for additional information. When the refined change plan is completed, the *Remediation planner* automatically computes rollback plans based on marks and groups defined by the *Operator*. Please refer to [6] for further details on the algorithm to support rollback plans on IT change processes.

At this point, in a solution with no automatic risk assessment, the refined change plan would be ready to be submitted to deployment. However, this change could expose the IT infrastructure to unknown risks, and failures might occur during the process affecting the involved CIs. The *Risk analyzer* is the component responsible for investigating the history of executions of the current RFC and classifying the activities in the change plan according to their risk magnitude. By analyzing the risk reports provided by this module, the *Operator* may investigate the source of risks and return the change plan to the *Change designer* in order to modify it aiming to turn it as safe as possible.

When the RFC analysis and edition process is completed, the *Change deployer* will actually apply the changes over the IT infrastructure. Every time a CI is affected by a change implementation, it is one of the *Deployment system's* roles to update the information on the CMS. This is essential to assure that this repository has always the latest vision of the IT infrastructure. The *Report generator* is responsible for tracing execution records for every change. When an operation is performed affecting an item, this component associates activities performed during the change process to the involved CIs. The status of the execution (success or failure) and failure

classification are also stored on the CMS for further evaluation. This information is kept on the system to allow the review of every modification performed over a CI.

The focus of the proposed solution is on the steps that are, somehow, useful for the risk assessment. As mentioned before, every proposed RFC must be approved by the CAB previous to its deployment, in this case, after the risk analysis the final change plan should be submitted for approval and scheduled to be implemented. These steps are out of the scope of this paper and they were omitted in the solution.

III. FAILURE REPRESENTATION MODEL

The risk analysis method proposed in this paper is based on the execution records of activities from an RFC. Therefore, to perform such analysis a structured model is needed to represent the history of changes applied over the items of an IT infrastructure. This model should represent the executions of change plans respecting the real performed workflow, *i.e.*, regarding to possible deviations of the regular flow caused by decisions made during run-time. Furthermore, it is also important to consider alternative remediation plans that could be triggered by the occurrence of failures. The proposed model has the objective of classifying failures on execution of changes under two aspects, *Source* and *Recovery*, as follows:

A. Source

Some types of failure are recurrent on change deployment processes and may be captured by the management system during run-time. Identifying source and classification of failures is important to help the operator understanding the behavior of problematic items and for what reasons they fail. The proposed failure classification is based on [11], and in this section we define six types of failure, as follows:

Activity Failure (AF): Failures of this type happen for reasons intrinsic to the execution of activities in the change plan. Usually, they occur during the installation or configuration of software elements. Also, they may be triggered by failures on other activities on the workflow (*e.g.*, dependent packages).

Resource Failure (RF): Resource failures are caused typically by hardware problems that make unavailable the elements where the activities are executed on. This could be a physical problem (*i.e.*, the equipment may have been damaged during the deployment process), or a wrong configuration may turn the resource unreachable by the change deployer. In both cases a resource failure is captured.

Human Failure (HF): Some activities on a change plan are performed by humans, therefore, when humans do not behave the way they were supposed to, then human failures have to be raised. Failures on manual activities, for example, should be recorded on the system even if there is no way of capturing it automatically. In these cases, the operator should insert the records to keep the history of changes as accurate as possible.

Time Failure (TF): When needed, deadlines can be specified to inform when activities should be finished. Besides that, time restrictions may be used for synchronization of tasks, for example, to inform that one activity must start before another.

Whenever time constraints are breached, a time failure should be captured.

External Trigger (ET): This type of failure occurs when some agent, external to the change process, interrupts the regular execution of the change plan. This could be a signal injected into the system by an external administrator user, informing that the workflow must be interrupted for any reason.

Constraint Violation (CV): Usually, it happens when an activity in the change plan needs to perform an operation that violates any of the organization’s policies. Moreover, these failures may be raised by conflicting scheduling of RFCs, for example, when the first change modifies the IT infrastructure implying new conditions not predicted on the next.

B. Recovery

Another important aspect in this subject is what recovery actions can be taken to reestablish system’s functionalities after the occurrence of failures. The system should capture the remediation actions taken to put the IT infrastructure back to a consistent state. These actions are classified in two categories:

No Action (NA): This class indicates that there was nothing to do after the occurrence of a failure, it could happen because of two reasons: (a) the operator explicitly informs that nothing should be done to recover the system, probably the operation was not significant enough to inspire caution, or (b) in the case of a fatal failure, the system could not take any remediation action to revert the situation.

Remediation (RM): When the system executes a remediation plan to recover itself, it may be a rollback plan and/or a compensation plan. If there is a rollback plan associated to an activity that fails, this plan will be invoked to undo the changes made by the activity. On the other hand, compensation plans may be a useful instrument to specify alternative ways to reach the end of the change plan. For instance, when the installation of a package fails, the system could install another one, that is similar to the first (but not the same), and proceed to the next activities. These compensation plans can be applied along with rollback plans or not; however, they should be used only as a secondary option because they do not guarantee the accomplishment to the original change plan requirements. Also, compensation and rollback plans do not undo failure events. Even if the change process was completed with a remediation plan the system still considers the execution as a failed process, keeping the failure classification and remediation on the record for further analysis.

C. The Conceptual Model

To allow the change management system to represent the execution records of changes, we propose a model that uses a subset of classes from the CIM extending some of them, regarding to represent the classification of failures. Additionally, an RFC and Change Plan representation model, proposed in a previous work [5], is employed to link the execution records to its corresponding RFC. Depicted in Figure 2, the proposed classes are distinguished by the dark boxes.

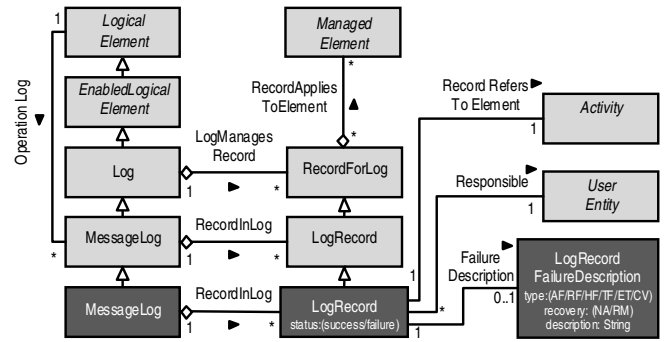


Fig. 2. Execution records representation model

The *Logical Element* class represents any type of process or system, on the proposed model it is employed to describe the process responsible for capturing and storing the log records. *Message Log* describes instances of all change plans executed over the IT infrastructure, while the *Log Record* class, is used to represent each activity performed individually. The records of execution are associated to the *Managed Elements* they apply to, this is essential to represent the elements involved in the change process which generated the log.

On the proposed *Log Record* class (in black) we included an attribute to inform whether instance of log represents a successful or failed activity. In addition, when a *Log Record* represents an unsuccessful operation, a *Log Record Failure Description* should be associated to classify the failure. From the *Log Record* class, we also associate both *User entity* – responsible for the execution – and the performed *Activity* from the change plan.

IV. RISK ANALYSIS METHOD

Before we propose a risk analysis method, it is necessary to define what relevant information is needed to give us subsidy on this process. In the literature, risk is usually seen as a composition of two factors [12]: (a) the probability of occurrence of a possibly negative event, and (b) how exposed the object of analysis is to this event. Marques *et al.* [8] complements this concept by saying that risk denotes a possible event of negative impact that will affect elements, and may occur in some present or future process. ITIL adopts a similar vision of risks, and explains that the risk assessment process has to be made observing the value of the involved items to the organization’s business.

In order to perform failure probabilities estimation, we assume that the subject RFC has been previously executed, and then, there are available log records for analysis. In addition, we should have in mind that remediation plans will be part of the execution records, nevertheless, their activities should not be included in the risk analysis. This is made because any remediation plan is started after a failure, in such case, the primary objective of the RFC is already compromised. Rollback and compensation plans should be always used as an alternative option, because they generally do not guarantee all requirements of the original plan.

In this work the risk analysis will be separated in two procedures: (a) investigate the history of executions of changes to determine the probability of failure of an activity, and (b) evaluate the impact of a failure in the execution of the activity over the business continuity.

A. Probability of Failure

In this work, we assume that previous failures occurred during the change process might be recurrent hereafter, and the rate of these occurrences can be expressed as a probability. For the sake of simplicity, we will consider only two failure classifications on this part of the solution, *Activity Failure* and *Resource Failure*. Also, in this solution, activities are not distinguished by their specific execution parameters, such as, paths for installations, amount of resources to allocate, or period of day that they will be scheduled. For each activity, the probability of failure is estimated examining the execution records of the RFC under two aspects: (a) probability of failure regarding the activity, and (b) probability of failure regarding the target of the operation.

A change plan activity is described complying with the *Activity Modeling Notation* (AMN) [5]. This notation is used to enable the system to identify the elements involved in each operation of the change process. One example of a sentence written in the AMN for installation of a software element at a computer system may be as follows: `install SoftwareElement <package1> at ComputerSystem <host1> with <Parameters>`. Where `install SoftwareElement <package1>` refers to the software been installed, and `at ComputerSystem <host1>` indicates the target computational system where it will be hosted in. When a failure occurs during the execution of an activity, the deployment system will capture the source of it. If the failure is classified as AF we attribute this negative event to the software element manipulation, otherwise, RFs are credited to the involved computer system.

Therefore, two equations were defined to calculate probabilities for each source of failure. The Equations (1) and (2) represent, respectively, the probabilities of failure for both AF and RF on the execution of an activity i .

$$P_i(AF) = \frac{F_{Si}}{E_{Si}} \quad (1)$$

$$P_i(RF) = \frac{F_{Ti}}{M_{Ti}} \quad (2)$$

In the Equation (1), F_{Si} is a counter of failures caused by software manipulation on the history of executions from the activity i , while E_{Si} returns the total of executions of the same activity regardless of target or software failures. The division of these counters will result the probability of a software manipulation failure for this activity. In a similar way, the Equation (2) estimates the probability of a RF by dividing the F_{Ti} , representing the counter of failures involving the target of the activity i , and M_{Ti} that is the total of manipulations of this target on the same RFC.

B. Impact Evaluation

When an RFC is submitted to deployment, failures within this process always have a negative impact over the managed IT infrastructure. In this second step of the risk analysis, the objective is to quantify the impact of failed activities from a change plan focusing on the business continuity. For this purpose, we assume that the configuration items (*e.g.*, systems, services, and computers) have their relevance index assigned according to the organization's guidelines. Therefore, a parameter named *User-defined Relevance* is defined for each relevant CI. This parameter should be expressed by a numerical value, allowing comparison of different elements, regardless of the scale adopted. For example, a possible range of *User-defined Relevance* could be: Maximum (1.00), High (0.75), Medium (0.50), Low (0.25), and Not defined (0.00).

The operator has only the responsibility of assigning relevance to the items that, somehow, are important to the business. This means that this factor should be related to the losses caused by unavailability of a service or system. For example, suppose that a company has a web server that hosts two web applications: an e-Commerce and an institutional web site. The highest relevance value should be associated to the e-Commerce application, because the losses caused by the downtime of this service would be very high. Despite the fact that the web server supports both applications, no relevance should be assigned to it. Note that, unlike the applications, the web server itself has no meaning to the business continuity. However, a failure over this item would affect all hosted services, and then its impact should be observed.

The Equation (3) allows to calculate the *Absolute Relevance* of a configuration item. The result $R(CI)$ of this equation is the accumulated relevance of CI , considering its *User-defined Relevance* $r(CI)$ and the *Absolute Relevance* of other CIs that depend on it. $Dep(CI, j)$ is a generic function that returns the managed element on the list of dependencies of CI indexed by j . The j index ranges from 1 to $x(CI)$, which is the length of the dependencies list of CI . The relevance equation will be recursively invoked for every dependent item of CI .

$$R(CI) = r(CI) + \sum_{j=1}^{x(CI)} R(Dep(CI, j)) \quad (3)$$

The *Absolute Relevance* calculated using the Equation (3) is a portion of the *Total Relevance* of the system, *i.e.*, its result will always be a value ranging from zero (not relevant element) and the sum of all *User-defined Relevancies*. In order to generate an impact scale we use the normalization Equation (4), where $I(CI)$ is the impact of a configuration item CI over the business continuity. Dividing the *Absolute Relevance* $R(CI)$ of a configuration item CI by the *Total Relevance* of the system $R(T)$, we achieve a value between zero and one that will be further used for the impact classification. For the sake of simplicity, we assume that there is a CI T (where $r(T)$ is zero) which has no dependencies and whose all other items depend on. By applying the *Absolute Relevance* calculus to this element T the return is the *Total Relevance* of the system.

$$I(CI) = \frac{R(CI)}{R(T)} \quad (4)$$

It is important to notice that this impact evaluation must be done independently for both software element and the computer system involved in the activity. These values will be used along with their respective probabilities of failure for risk classification, as will be further explained in this work.

C. Risk Classification

One key objective of the risk assessment is to provide decision support to the operator before an RFC is deployed. In order to achieve that, it is mandatory to deliver the information concerning to risks within the IT change process in a clear and concrete way. One ITIL's recommendation is to use a risk categorization matrix like the one represented in the Table I. It is possible to observe that risks are classified into four categories in a crescent scale, where category 1 means highest risk and category 4 the lowest. Another important fact is that greater importance is given to the impact, since the category 2 is before the category 3 on the scale.

TABLE I
ITIL'S RISK CATEGORIZATION MATRIX

Change Impact/Risk Categorization Matrix		
Change Impact	High Impact Low Probability Risk Category: 2	High Impact High Probability Risk Category: 1
	Low Impact Low Probability Risk Category: 4	Low Impact High Probability Risk Category: 3
Probability		

The Institute of Risk Management (IRM) [13] introduces a generic standard for risk management, regardless of area of application. In this standard, the IRM recommends to quantify the probability and impact on the following scales: (i) high (provable), medium (possible), and low (remote) for probabilities; and (ii) high (significant), medium (moderate), and low (insignificant) for impact. Therefore, in this work we introduce the scale to classify impact and probability as shown in Table II. The intervals of values for classification as Low, Medium, and High are displayed as an example, *i.e.*, they could be parameterized by the operator, according to the needed level of confidence and reliability expected for the subject environment.

TABLE II
CLASSIFICATION OF IMPACT AND PROBABILITY SCALE

	Low	Medium	High
Probability	0%-20%	21%-50%	51%-100%
Impact	0.00-0.20	0.21-0.50	0.51-1.00

Evolving from the categorization concept recommended by the ITIL, it is possible to classify risks into nine categories, as demonstrated in Table III. In the proposed classification, the higher impact is prioritized rather than the probability of failure. For instance, adopting the same ITIL's approach, an

event of high impact and low probability would be classified in category 3, representing greater risk than an event of low impact and high probability (category 7).

TABLE III
PROPOSED RISK CLASSIFICATION TABLE

Risk	Impact	Probability
1	High	High
2	High	Medium
3	High	Low
4	Medium	High
5	Medium	Medium
6	Medium	Low
7	Low	High
8	Low	Medium
9	Low	Low

Finally, considering that we have proposed the estimation of probability of failure, and its associated impact under two different aspects (*Activity Failure* and *Resource Failure*), categorization will be also necessary for both of them. However, since the intention of this analysis is to deliver a comprehensive output, thus risks could be easily identified. To achieve this objective, a harmonic mean of the portions is used to compute the final result, as shown in Equation (5).

$$K_i = \frac{2}{\frac{1}{C(AF,i)} + \frac{1}{C(RF,i)}} \quad (5)$$

The K_i represents the *Mean Risk* of executing an activity i over the IT infrastructure. This equation considers all aspects that contribute to the risks on the activity i . As we only consider two risk evaluations, the harmonic mean is calculated between the risk classification of an *Activity Failure* and *Resource Failure*; nevertheless the same equation may be generalized for n portions. The function $C(x, i)$ returns the risk category on which the activity i is classified, where x is the source of failure (AF or RF).

The use of a harmonic mean in this part of the solution makes the *Mean Risk* tend to approach the result to the highest risk portion of the equation (lowest risk category), working as a pessimist analysis. For instance, assuming that an activity j has RF risk classification assigned as 1 and AF as 9, the *Mean Risk* for this activity would be set as 1.8. If we have used an arithmetic mean of the values, then the result would be 5, concealing the hazard of the RF. Another option could be using the lowest category. In that case, the *Mean Risk* of the activity j would be the same of an activity with risk category 1 for both RF and AF, for example, which is not realistic.

In order to clarify the aforementioned concepts, on the remainder of this paper we will present a case study using a real problem and applying risk analysis to identify threads on a complete change plan.

V. CASE STUDY AND ANALYSIS

Our solution for risk assessment is part of the prototypical system called CHANGELEDGE. In this section is introduced a real-life scenario where the proposed concepts are used in order to demonstrate their appliance.

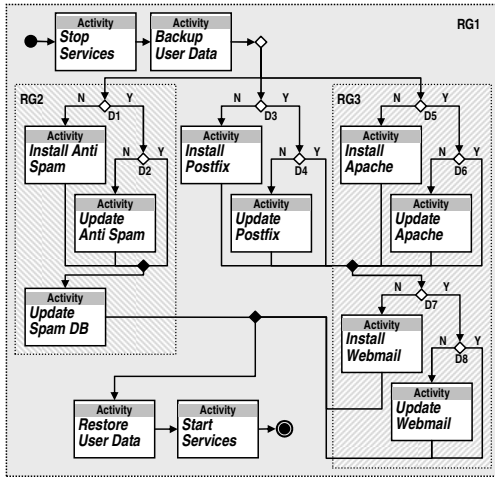


Fig. 3. Workflow of the RFC #32

Suppose that a company has installed an internal mail server for employee's accounts, where most users have mail clients installed on their workstations and access their messages via POP. For those who are outside the office, the company provides a webmail interface as a secondary way to access e-mail. These services are provided by two interacting servers (*Mailsrv* and *WebSrv*). The main objective is to provide the e-mail access via POP/SMTP, so the concern of the company is to keep these services up.

In order to give maintenance on this system we propose a generic RFC that will be applied periodically over the IT infrastructure. This RFC is meant to backup relevant data, install new packages and update software versions every time they are released. The workflow depicted in Figure 3 represents the change plan of an RFC #32, being useful for the installation of new mail delivery systems and to keep these software up-to-date.

The workflow begins by stopping all services involved on the mail system (*Postfix*, *SpamAssassin* and *Apache*). After that, the change management system will backup mailboxes to keep user's personal data safe during the change process. In the next step, the workflow starts three parallel installation/update threads: (i) anti spam system (*SpamAssassin*), (ii) mail delivery system (*Postfix*), and (iii) Web server (*Apache*). Observe that three Reversible Groups were defined (*RG1*, *RG2* and *RG3*). The main group is *RG1* that contains all activities of the workflow. This means that any fail occurring on an activity that belongs to *RG1*, would return the whole mail system to the previous configuration. Other groups (*RG2* and *RG3*) have a secondary objective on this RFC. They are meant only to rollback parts of the workflow (spam filtering and webmail) that are not mandatory for mail delivery system to work.

Additionally, the workflow is able to determine whether a *SoftwareElement* is installed or not (or if it is up-to-date), therefore at run-time it decides which action to perform. Decisions on the workflow are represented by the white filled rhombuses labeled D_n . When n is odd, the decision is whether

the software is already installed or not. If it is not (*N*), the next executed activity will be the installation of a *SoftwareElement*. If n is even, then the system verifies whether the current installed software version is the latest available. In case of the installed version is overtaken by a new one, an update activity will evolve the system to earliest released software version; otherwise no action will be performed over the CI. Finally, the last activities restore user's mailboxes and start all services once again.

Table IV represents execution records of the proposed RFC. By examining the number of the case, we are able to notice that the workflow has been executed three times over the IT infrastructure. Only two failures occurred in the history of executions of this RFC: the installation of the *SpamAssassin* was reverted because of an *AF*, and the *Postfix* update failed due to a *RF* of the *Mailsrv*.

TABLE IV
EXECUTION RECORDS OF THE RFC #32

Case	Activity	Status	Class	Target
1	Stop Services	Success	-	Mailsrv, WebSrv
1	Backup User Data	Success	-	Mailsrv
1	Install Postfix	Success	-	Mailsrv
1	Install Webmail	Success	-	WebSrv
1	Install Anti Spam	Failure	AF/RM	Mailsrv
1	Restore User Data	Success	-	Mailsrv
1	Start Services	Success	-	Mailsrv, WebSrv
2	Stop Services	Success	-	Mailsrv, WebSrv
2	Backup User Data	Success	-	Mailsrv
2	Install Anti Spam	Success	-	Mailsrv
2	Update Spam DB	Success	-	Mailsrv
2	Restore User Data	Success	-	Mailsrv
2	Start Services	Success	-	Mailsrv, WebSrv
3	Stop Services	Success	-	Mailsrv, WebSrv
3	Backup User Data	Success	-	Mailsrv
3	Update Postfix	Failure	RF/RM	Mailsrv

The Figure 4 represents the IT infrastructure which the RFC #32 applies to. Three main servers are maintained by the company to support business operations (*Mailsrv*, *WebSrv* and *Syssrv*), however, only two of them are manipulated by the workflow. For this analysis we considered only two types of dependencies: *InstalledSwEl* indicating that a *SoftwareElement* depends on the host it resides in, and *ServiceDep* used to express when a service depends on other service to work properly. We should notice that not all the elements are manipulated on this change, however, some of them are indirectly affected by it (e.g., e-Commerce and Customer Support). The *User-defined Relevancies* are represented by the numbers on the bottom-right of item's boxes.

On the Table V the results of the risk assessment for all activities of the RFC #32 are displayed as an ordered list. *Start Services* and *Stop Services* have been classified on the risk scale as 3. Despite the fact they have never failed, these activities manipulate a great number of items, and a failure in their execution may have high impact on the IT infrastructure. Another interesting fact is that the *Webmail* has lower relevance to the business than the *Postfix*, for example. However, the host where the *Webmail* resides in supports two other applications of very high relevance (*e-Commerce*

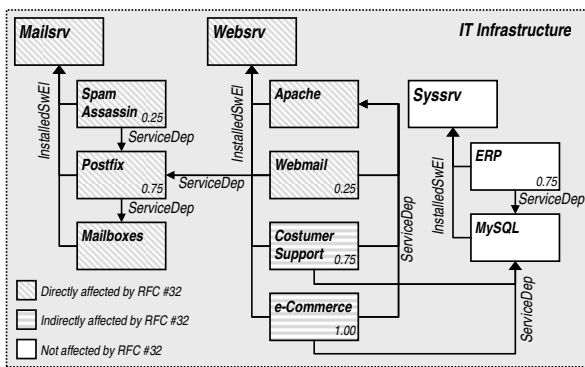


Fig. 4. IT infrastructure representation

and *Costumer Support*), in this case, the manipulation of the *Websrv* increments the risk for this operation. Some of the activities could not be classified, and have been left on the bottom of the list. This is because they have never been executed in this RFC, thus its not possible assign probability of failure for them.

TABLE V
RISK CLASSIFICATION FOR THE ACTIVITIES OF RFC #32

Activity	Mean Risk
Stop Services	3.00
Start Services	3.00
Install Webmail	4.50
Backup User Data	6.00
Install Postfix	6.00
Update Postfix	6.00
Restore User Data	6.00
Install Anti Spam	6.86
Update Spam DB	7.20
Update Anti Spam	N/A
Install Apache	N/A
Update Apache	N/A
Update Webmail	N/A

In this scenario, the risk analysis is important to provide information to the operator about risks on a change plan that could not be easily seen by looking only to its activities. Apparently, one trivial activity such as *Install Webmail* wouldn't require much care. However, its side effects can cause harm to important services. Examining a risk report, one could decide to modify the change plan or the IT infrastructure (e.g., migrate Webmail to another server) in order to reduce the risks of the change process.

VI. CONCLUSION AND FUTURE WORK

In this research we have discussed some of the aspects related to IT Change Management and the importance of risk assessment within the change process. Despite all the efforts been conducted in the aforementioned areas, the estimation of potential risks inside the actions that compose a change plan has been done intuitively by a human operator. However, the large scale modern IT infrastructure makes this kind of analysis too complex, even for experienced operators. Consequently, a superficial risk assessment could expose the environment to unnecessary threads causing losses to the business. The

risk analysis method proposed is based on the history of executions of an RFC and considers the current view of the IT infrastructure. Such method has shown to be useful for classifying the activities on a change plan according to a risk scale, considering their probability of failure and impact to the business continuity.

In this approach we have proposed a model to represent execution records of RFCs, contemplating failure classifications into six categories. Nevertheless, in the risk analysis method only two of them were considered (*Activity Failures* and *Resource Failures*). In a future work the solution should be extended to calculate risks for all proposed classifications, including the attribution of failure probabilities to human resources in the case of *Human Failures*, for instance.

Another future perspective is to apply risk assessment for brand new RFCs. Assuming that when an RFC is successively performed several times, the knowledge over its operations tend to increase, reducing the risk for further executions. On the other hand, for RFCs that have never been executed the uncertainly factors may turn this change into a very risky process. Despite the fact that there is no history of execution of the whole RFC, some activities (or set of activities) may have been reused from another change processes, and for these parts the risk could be analyzed.

REFERENCES

- [1] Office of Government Commerce (OGC), "Information Technology Infrastructure Library (ITIL)," Office of Government Commerce (OGC), 2008, <http://www.itil-officialsite.com/>.
- [2] ITIL, *ITIL Service Transition Version 3.0*. Office of Government Commerce, 2007.
- [3] R. Rebouças, J. Sauvé, A. Moura *et al.*, "A Decision Support Tool to Optimize Scheduling of IT Changes," in *10th IFIP/IEEE International Symposium on Integrated Network Management (IM 2007)*, Munich, Germany, May 2007, pp. 343–352.
- [4] J. Sauvé, R. Santos, R. Almeida *et al.*, "On the Risk Exposure and Priority Determination of Changes in IT Service Management," in *18th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management, DSOM 2007, San Jose, CA, USA, October 29-31, 2007*, 2007, pp. 147–158.
- [5] W. L. C. Cordeiro, G. S. Machado, F. F. Daitx *et al.*, "A Template-based Solution to Support Knowledge Reuse in IT Change Design," in *11th IEEE/IFIP Network Operations and Management Symposium (NOMS 2008)*, Salvador, Brazil, April 2008, pp. 355–362.
- [6] G. S. Machado, W. L. C. Cordeiro, F. F. Daitx *et al.*, "Enabling Rollback Support in IT Change Management Systems," in *11th IEEE/IFIP Network Operations and Management Symposium (NOMS 2008)*, Salvador, Brazil, April 2008, pp. 347–354.
- [7] Distributed Management Task Force, "Common Information Model." [Online]. Available: <http://www.dmtf.org/standards/cim>
- [8] M. Marques and R. Neves-Silva, "Risk assessment to support decision on complex manufacturing and assembly lines," *5th IEEE International Conference on Industrial Informatics*, pp. 1209–1214, June 2007.
- [9] R. Fewster and E. Mendes, "Measurement, prediction and risk analysis for Web applications," *Software Metrics Symposium, 2001. METRICS 2001. Proceedings. Seventh International*, pp. 338–348, 2001.
- [10] L. Wang, A. Sahai, and J. Pruyne, "A model-based simulation approach to error analysis of it services," *Enterprise Systems and Software Laboratory - HP Laboratories, Palo Alto, CA, Tech. Rep. 181*, Dec 2006.
- [11] N. Russell, W. van der Aalst, and A. ter Hofstede, "Exception Handling Patterns in Process-Aware Information Systems," *BPM Center Report BPM-06-04*, 2006.
- [12] J. Chicken and T. Posner, *The Philosophy of Risk*. T. Telford, 1998.
- [13] The Institute of Risk Management (IRM), *A Risk Management Standard*, United Kingdom, 2002.

APPENDIX B PUBLISHED PAPER - SBRC 2009

In this appendix the paper entitled “*Automatizando a Estimativa de Riscos em Sistemas de Gerenciamento de Mudanças em TI*” is presented (in Portuguese). This was the second deliverable of this research still focused in the context of IT Change Management. In this paper, an evolution of the proposed solution has been presented, this time, including a modular specification of the risk assessment procedures and also detailing the behavior of each module in algorithms. More implementations details have been provided including the configuration of test scenarios using an emulated IT infrastructure, the BPEL engine employed, and how random exceptions have been raised to inject artificial failures in change deployments. Moreover, the concept of Service Disruption (SD) was introduced in order to enable the evaluation of performance of change deployments, showing how risk indicators automatically generated can actually help the design of better Change Plans (CPs).

- **Title:**
Automatizando a Estimativa de Riscos em Sistemas de Gerenciamento de Mudanças em TI
- **Conference:**
27th Brazilian Symposium on Computer Networks and Distributed Systems (SBRC 2009)
- **URL:**
<http://www.sbrc2009.ufpe.br/>
- **Date:**
25-29 May 2009
- **Venue:**
Mar Hotel, Recife, Pernambuco, Brazil
- **Digital Library of SBC:**
<http://www.sbc.org.br/bibliotecadigital/download.php?paper=2681>

Automatizando a Estimativa de Riscos em Sistemas de Gerenciamento de Mudanças em TI*

Juliano Araujo Wickboldt, Roben Castagna Lunardi
Guilherme Sperb Machado, Weverton Luis da Costa Cordeiro
Alan Diego dos Santos, Fabrício Girardi Andreis, Cristiano Bonato Both
Lisandro Zambenedetti Granville, Luciano Paschoal Gaspary

Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Porto Alegre, RS – Brasil

{jwickboldt, rclunardi, gsmachado, weverton.cordeiro
adsantos, fgandreis, cbboth, granville, paschoal}@inf.ufrgs.br

Abstract. *Modern organizations take advantage of complex IT infrastructures in order to support their daily operations. Since these environments require special care, whenever changes become necessary, risks associated to them should be investigated. Usually, risk assessment is made by humans based only on their empirical knowledge, which is a very prohibitive task to do, that might lead to inaccurate or incomplete conclusions about risks associated to changes. In this paper, we present a solution for automating the process of risk assessment, based on data collected from past changes in order to identify possible problems for subsequent ones. A prototypical system was developed to evaluate the solution on an emulated IT infrastructure. The results achieved show how the automated solution is capable of raising the quality of the change planning as well as the organization of the managed infrastructure, in this way reducing the chances of disrupting the services delivered by the organization.*

Resumo. *Organizações modernas utilizam infra-estruturas de TI complexas para apoiar suas operações diárias. Por se tratar de um ambiente sensível, sempre que surge a necessidade de se aplicar mudanças é importante estimar quais riscos podem estar associados a elas. Geralmente, estimativas de riscos realizadas por humanos são baseadas apenas em conhecimento empírico, o que, além de acarretar uma quantidade proibitiva de trabalho, muitas vezes, leva a conclusões imprecisas e incompletas sobre os riscos associadas às mudanças. Neste artigo, é apresentada uma solução para automatização do processo de estimativa de riscos, baseando-se em informações de mudanças executadas no passado a fim de identificar possíveis problemas em implantações subsequentes. Foi implementado um protótipo de sistema para avaliação da solução em uma infra-estrutura de TI emulada. Os resultados obtidos indicam que a solução é capaz de elevar a qualidade do planejamento das mudanças bem como da organização da infra-estrutura gerenciada, dessa forma causando menos danos aos serviços prestados pela organização.*

1. Introdução

Nas organizações modernas, a heterogeneidade das infra-estruturas de TI, associada à grande quantidade de dispositivos e aplicações presentes, torna a tarefa de gerenciamento

*Este trabalho foi desenvolvido em colaboração com a HP Brasil P&D.

de TI cada vez mais complexa. Uma infra-estrutura de TI é formada por um conjunto de *itens de configuração* (*Configuration Items* - CIs) que vão desde elementos concretos como servidores, estações de trabalho e roteadores, a elementos lógicos como pacotes de *software* e serviços de rede. Empregar políticas racionais de gerenciamento de TI eleva a qualidade dos serviços oferecidos pelas organizações, além de reduzir os custos de operação. Para manter de forma consistente e segura esse tipo de infra-estrutura, a OGC (*Office Government Commerce*) definiu um conjunto de processos e boas práticas que organizam as atividades de gerenciamento. Tais processos e boas práticas são publicados na biblioteca ITIL (*Information Technology Infrastructure Library*) [ITIL 2008].

A disciplina de *gerenciamento de mudanças*, contemplada no livro *Service Transition 3.0* [ITIL 2007] da ITIL, determina como uma mudança deve ser conduzida sobre uma infra-estrutura de TI, desde a sua solicitação, planejamento e análise até sua implementação. Nesse livro, a ITIL recomenda que toda mudança a ser realizada deve ser descrita em uma *requisição de mudança* (*Request for Change* - RFC). Uma RFC deve definir, de forma declarativa, dentre outros parâmetros, os motivos da mudança requisitada, os CIs envolvidos e o que deve ser alterado. Não é função de uma RFC, porém, indicar quais atividades de mais baixo nível devem ser executadas para que uma mudança seja realizada; isso é de fato tratado, por exemplo, por sistemas de gerenciamento automatizados, ou até mesmo por operadores humanos. Adicionalmente, todas as RFCs devem ser submetidas à análise, aprovação e agendamento por parte de um comitê denominado *Change Advisory Board* (CAB). Esse comitê, presidido geralmente por um *gerente de mudanças*, deve ser formado por pessoas com conhecimento amplo sobre os processos da organização, provenientes de diversas áreas, e não necessariamente terem domínio sobre as tecnologias utilizadas na infra-estrutura de TI.

Sabendo que as infra-estruturas de TI suportam serviços fundamentais para a continuidade do negócio das organizações, sempre que a necessidade de se realizar uma mudança nessas infra-estruturas é iminente, os riscos associados à mudança requisitada precisam ser considerados. Segundo a ITIL, riscos devem ser investigados e mensurados antes que uma mudança seja aprovada. Além disso, contramedidas devem ser estabelecidas para minimizar a possibilidade dos riscos se materializarem em problemas reais, causando deste modo danos para a continuidade do negócio. Alguns exemplos de eventos que caracterizam riscos aos quais uma infra-estrutura de TI fica exposta durante a implantação de mudanças são: falhas durante a instalação de *softwares*, configurações incorretas de equipamentos como *firewalls* ou roteadores e defeitos nos CIs manipulados. As ocorrências desses eventos podem fazer com que a infra-estrutura de TI evolua para um estado indesejável ou desconhecido.

Uma das recomendações apresentadas pela ITIL é que os riscos devem ser vistos como uma combinação da probabilidade da ocorrência de um evento possivelmente negativo e o impacto dessa ocorrência sobre os negócios da organização [ITIL 2007]. Porém, estimativas de risco são realizadas normalmente por operadores humanos, baseadas apenas no conhecimento empírico adquirido pelos mesmos ao longo de suas carreiras. No entanto, devido ao grande número de CIs envolvidos nas mudanças e a quantidade de variáveis que se deve considerar (*e.g.*, histórico de falhas e impacto dos CIs afetados), esse tipo de análise pode acabar sendo superficial ou imprecisa demais para que se possa usar como base para tomada de decisões.

Apesar das boas práticas introduzidas pela ITIL, essa biblioteca não define um método claro de análise de riscos no processo de mudança. Recentemente, alguns autores propuseram soluções para a automação do gerenciamento de mudança em suas diversas etapas [Cordeiro *et al.* 2008] [Machado *et al.* 2008] [Rebouças *et al.* 2007] [Sauvé *et al.* 2007]. Porém, tais trabalhos não propõem uma metodologia automatizada para investigação de riscos no planejamento de mudanças. Um método padronizado de análise de riscos pode fornecer ao operador subsídios para rapidamente identificar ameaças na mudança requisitada, antes de submetê-la para implantação. Com base nas informações da análise de riscos, o operador poderia fazer alterações na mudança original ou até mesmo promover modificações na infra-estrutura de TI, objetivando reduzir as possibilidades da mudança em questão causar danos às operações normais da organização.

A fim de atacar o problema previamente exposto, este trabalho propõe um método de análise automatizada de riscos em processos de gerenciamento de mudanças. A solução proposta baseia-se no histórico de execuções de mudanças sobre uma infra-estrutura de TI, analisando a ocorrência de falhas em implantações passadas para identificar possíveis problemas para as próximas execuções. Dessa forma, é possível munir um sistema de gerenciamento de mudanças com informações para tratamento de incidentes de forma proativa. Isso significa fornecer ao operador humano a oportunidade de minimizar a possibilidade de falhas ajustando as mudanças solicitadas e, conseqüentemente, elevar a qualidade dos serviços suportados pela infra-estrutura de TI.

No seguimento deste trabalho serão discutidos, na Seção 2, alguns dos principais trabalhos relacionados ao gerenciamento de riscos e gerenciamento de mudanças. Um detalhamento da solução de análise de riscos proposta é apresentado na Seção 3, enquanto que detalhes da implementação do protótipo desenvolvido para validação são descritos na Seção 4. Na Seção 5 é exibida uma avaliação experimental utilizada para mensurar os resultados da solução e, por fim, na Seção 6 são discutidos conclusões e trabalhos futuros.

2. Trabalhos Relacionados

Gerenciamento de riscos é uma disciplina transversal, ou seja, que se aplica a diferentes áreas do conhecimento. De uma forma genérica, pode-se dizer que o gerenciamento de riscos é o processo pelo qual as organizações avaliam os riscos associados as suas atividades, com objetivo de identificar e tratar ameaças obtendo um nível máximo sustentável de benefício [IRM 2002]. Os riscos em si, podem ser vistos como potenciais eventos com conseqüências que podem constituir oportunidades ou ameaças ao sucesso. Apesar de o risco vir sendo considerado na literatura sob os dois aspectos (positivo e negativo), em áreas como a de segurança, por exemplo, dificilmente se encontrará um lado positivo para eles. Nos últimos anos alguns trabalhos foram publicados abordando tópicos relacionados ao gerenciamento de riscos e gerenciamento de mudanças. Porém, são raras as iniciativas que levam em conta os riscos ocasionados pela necessidade de mudança.

Marques e Neves-Silva [Marques e Neves-Silva 2007] propuseram um método de avaliação de riscos para ajudar na tomada de decisão em linhas de montagem de grande porte. Os autores propõem quantificar o risco considerando a probabilidade da ocorrência de incidentes e os impactos que esses eventos teriam caso acontecessem. No entanto, esse método é aplicável para um ambiente onde os parâmetros necessários para o cálculo (probabilidade e impacto) possuem valores conhecidos para um conjunto limitado de eventos

possíveis. Por exemplo, quando um alarme é acionado indicando que uma variável monitorada ultrapassou um determinado limite (*e.g.*, tempo médio entre falhas). A partir de valores de probabilidade e impacto predefinidos é feita a estimativa automática de riscos para cada um dos incidentes possíveis.

Fewster e Mendes [Fewster e Mendes 2001] introduziram um *framework* para análise de riscos em editoração e desenvolvimento de sistemas Web utilizando um Modelo de Generalização Linear (*Generalized Linear Model* - GLM). O GLM se mostrou bastante eficaz na previsão de riscos, tais como, ultrapassar orçamento ou prazo previsto de término de um projeto. Utilizando um modelo estatístico não se fornece apenas um ponto máximo ou mínimo para a variável analisada, mas sim, uma distribuição de probabilidade. De posse dessas informações um gerente de projeto poderia estimar a probabilidade de não terminar um projeto dentro de um determinado tempo (por exemplo, 30 dias). Apesar disso, apenas a probabilidade de ocorrência dos eventos negativos são estimadas pelo GLM, o impacto que esses eventos possam ter para o projeto não são considerados.

Sauvé *et al.* [Sauvé *et al.* 2007] e Rebouças *et al.* [Rebouças *et al.* 2007] apresentaram um processo de análise de riscos durante a fase de agendamento de mudanças com a intenção de determinar as prioridades de execução de RFCs potencialmente concorrentes. Os métodos propostos são fortemente baseados em estimativas de tempo para implantação de RFCs e na maneira como elas podem ser agendadas em momentos diferentes, alterando assim o impacto dessas implantações sobre os objetivos do negócio. Segundo os autores, o tempo que transcorre desde a submissão de uma RFC até a sua implementação causa danos aos serviços afetados pela mudança, que podem, por exemplo, sofrer por degradação de desempenho. Além disso, durante a fase de implantação de uma RFC, a interrupção dos serviços alterados e eventuais descumprimentos de prazos podem acarretar perdas financeiras ou penalizações contratuais. No entanto, a análise de riscos proposta nesses trabalhos possui aplicação para a fase de agendamento de mudanças, e não para o seu planejamento, como abordado neste artigo.

De acordo com o conhecimento dos autores deste artigo, não existe um método de estimativa de riscos padronizado para gerenciamento de mudanças na fase de planejamento de RFCs. A importância dessa estimativa reside no fato de que a infra-estrutura gerenciada suporta os serviços prestados pela organização. Sendo assim, problemas durante a implantação de mudanças podem ocasionar indisponibilidade desses serviços, afetando a continuidade do negócio. A ITIL reforça essa importância afirmando que mesmo mudanças aparentemente inofensivas do ponto de vista de sua complexidade, ainda que indiretamente, podem causar danos significativos a serviços relevantes para o negócio.

3. Estimativa de Riscos Automatizada

Para que uma estimativa de riscos no processo de mudança possa ser automatizada, essa estimativa deve ser baseada em informações sobre execuções de mudanças coletadas do próprio ambiente de TI. A partir dessas informações, uma metodologia padronizada seria capaz de quantificar os riscos aos quais a infra-estrutura de TI estará exposta durante a implantação de uma mudança e servir de guia para a especificação de mudanças mais prudentes. Apesar das diversas abordagens adotadas para gerenciamento de riscos nas mais variadas áreas do conhecimento, riscos são geralmente tratados como uma combinação de dois fatores: (i) a possibilidade da ocorrência de um evento potencial-

mente negativo e (ii) o prejuízo que esse evento é capaz de causar sobre o objeto de análise [Chicken e Posner 1998]. A ITIL adota uma visão similar para riscos no gerenciamento de mudanças em TI ressaltando que estes devem ser avaliados levando em consideração os objetivos do negócio da organização.

Neste trabalho assume-se que falhas durante a implantação de mudanças são recorrentes, isto é, ao se observar o histórico de execuções de uma RFC é possível analisar falhas ocorridas no passado e estimar probabilidades de novas ocorrências das mesmas. Assume-se também que os itens da infra-estrutura de TI possuem uma relevância para os objetivos do negócio, direta ou indiretamente, e que essa relevância é definida para cada CI. Sendo assim, falhas que afetem esses CIs têm um impacto sobre a continuidade do negócio da organização, portanto, tal impacto deve ser investigado pela análise de riscos.

Nesta seção será apresentada a solução para automatização da análise de riscos no processo de gerenciamento de mudanças, considerando dois fatores: (i) a probabilidade de falha na implantação de uma RFC e (ii) o impacto dessas falhas para a continuidade do negócio da organização. Em um primeiro momento será revisado o ciclo de vida regular de uma RFC, desde a sua emissão até a implantação da mudança requerida. Posteriormente, será apresentado o componente do sistema gerenciamento de mudanças responsável por realizar a análise de riscos.

3.1. Arquitetura do Sistema de Gerenciamento de Mudanças

Uma vez que uma RFC é submetida ao sistema de gerenciamento de mudanças, um operador humano fará a especificação de um plano de mudança (*Change Plan* - CP) preliminar. Basicamente, este plano consiste em um *workflow* de atividades de alto nível que descrevem os passos a serem seguidos para materializar a mudança solicitada na RFC. O CP preliminar passa então por um processo de refinamento, gerando assim um *workflow* de atividades de baixo nível que podem ser efetivamente executadas sobre os CIs [Cordeiro *et al.* 2008]. Ao término da execução do CP refinado, a infra-estrutura de TI deve ter evoluído para um novo estado consistente. Como falhas podem ocorrer durante esse processo, devem ser previstos planos de remediação que serão executados para minimizar os danos causados por tais falhas. Planos de remediação podem tanto retornar a infra-estrutura de TI ao estado anterior à mudança (*rollback*) quanto executar atividades que compensem as falhas ocorridas [Machado *et al.* 2009].

Em um sistema sem suporte a análise de riscos, ao término das definições do plano de mudança e dos planos de remediação, uma RFC estaria pronta para ser aprovada e executada. Porém, sem uma avaliação apropriada de riscos, essa mudança poderia expor a infra-estrutura de TI a riscos desconhecidos. Falhas durante a execução do CP ocasionariam interrupções nos serviços por um tempo indeterminado, até que os planos de remediação fossem postos em prática. Por esse motivo, neste trabalho é introduzido o componente *Risk Analyzer* (Figura 1) que contempla a etapa de estimativa automatizada de riscos no planejamento da mudança. Esse componente recebe como entrada a RFC que se pretende executar, os registros de execuções anteriores da mesma e uma visão da infra-estrutura de TI. A partir dessas entradas, são feitas estimativas de forma automática sobre os riscos aos quais a infra-estrutura de TI estará exposta durante a execução da RFC e, ao final, um relatório de riscos será apresentado ao operador do sistema. Esse relatório pode ser utilizado como base para possíveis alterações na RFC original de forma a mitigar os riscos nela contidos, ou então, permitir que o operador promova modificações em

pontos críticos da infra-estrutura de TI a fim de minimizar o impacto da mudança sobre os CIs manipulados.

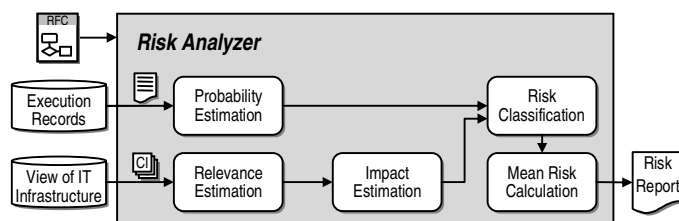


Figura 1. Disposição dos elementos do componente de análise de riscos

Os registros de execução (*Execution Records*) de uma RFC são definidos segundo um modelo proposto em um trabalho anterior [Wickboldt *et al.* 2009]. Esses registros representam os traços de execução do *workflow* da mudança respeitando a ordem em que as atividades foram realizadas. Além disso, são incluídas informações sobre o *status* da execução (sucesso ou falha) e, em caso de falha, tais registros compreendem ainda a classificação da falha e as medidas de remediação utilizadas. No modelo proposto, as falhas poderiam ser classificadas em seis categorias: *Activity Failure* (AF), *Resource Failure* (RF), *Human Failure* (HF), *Time Failure* (TF), *External Trigger* (ET) e *Constraint Violation* (CV). Porém, neste trabalho consideramos para fins de avaliação apenas duas classificações: AF, que representa as falhas intrínsecas às atividades do CP (*e.g.*, falhas em instalação de *software*) e RF, que representa falhas dos recursos manipulados durante a mudança (*e.g.*, defeitos em equipamentos alterados).

Para que se possa manter uma visão consistente da infra-estrutura de TI (*View of IT Infrastructure*) se faz necessário o uso de um modelo que represente, de maneira adequada, os CIs nela contidos. Neste trabalho, utilizou-se um subconjunto de classes do CIM (*Common Information Model*) [DMTF 2008], proposto pelo DMTF (*Distributed Management Task Force*). Esse modelo permite representar todos os tipos de CIs, sejam eles, elementos de *hardware*, *software*, serviços ou configurações, assim como as definições de relações e dependências entre esses elementos. A estimativa automatizada de riscos requer que, para cada CI representado, seja atribuído um valor de *relevância para o negócio* (*Business Relevance - BsR*). Esse parâmetro deve refletir a importância de um elemento (CI) da infra-estrutura de TI para a continuidade do negócio e deve ser expressado por um valor numérico qualquer, permitindo comparar relevâncias de diferentes elementos, independentemente da escala adotada. A BsR deve ser atribuída apenas aos elementos que possuem alguma relevância para o negócio e será útil para o cálculo de impacto dos CIs a ser apresentado no seguimento deste artigo.

3.2. Algoritmos para Análise de Riscos

Internamente, o *Risk Analyzer* procede a estimativa das probabilidades de falha através do módulo *Probability Estimation*. Esse módulo realiza o cálculo segundo uma função descrita no Algoritmo 1. Como entradas, são recebidos os registros de execução e o plano de mudança da RFC em questão. Para cada atividade do plano de mudança (Linha 2), a função encontra, entre os registros de execução, o número total de vezes que tais atividades foram realizadas (Linha 3). Em seguida, para cada tipo de falha (neste trabalho

são considerados AF e RF, Linha 4) a função procura nos registros de execução a quantidade de falhas de um determinado tipo para uma atividade (Linha 5). De posse dessas informações, a probabilidade de falha é calculada dividindo o número total de falhas encontradas pelo total de execuções da atividade (Linha 6). Cada probabilidade forma uma tupla, juntamente com a atividade e o tipo de falha, que será inserida em um conjunto (Linha 7), que ao final do cômputo será retornado pela função (Linha 8).

Algoritmo 1: *Função de Cálculo de Probabilidade*

Entrada: R : conjunto de registros de execução de uma RFC, CP : plano de mudança

Saída: conjunto de tuplas contendo atividade, probabilidade de falha e tipo de falha

1. $S \leftarrow$ conjunto vazio de tuplas (atividade, probabilidade de falha, tipo de falha)
2. **for each** $i \in$ conjunto de atividades do CP
3. **do** $T \leftarrow$ total de execuções de i **in** R
4. **for each** $j \in$ tipos possíveis de falha
5. **do** $F \leftarrow$ total de falhas da atividade i para o tipo de falha j **in** R
6. $\varphi \leftarrow F \div T$
7. $S \leftarrow S \cup \{i, \varphi, j\}$
8. **return** S

A segunda funcionalidade do *Risk Analyzer* é estimar o impacto de uma mudança sobre os elementos (CIs) da infra-estrutura de TI. Em um primeiro momento, o módulo *Relevance Estimation* calculará a *relevância absoluta* (*Absolute Relevance* - AR) dos elementos manipulados no plano de mudança através da função apresentada no Algoritmo 2. A AR é um fator que indica a relevância total de um elemento para a continuidade do negócio, incluindo sua BsR e a de todos os elementos que dependem dele, direta ou indiretamente. Nesse algoritmo, para cada CI (variável ci) envolvido no plano de mudança (Linha 2), o algoritmo inicia o valor de AR do elemento (variável γ) com a sua própria BsR (Linha 3). Logo após, é criada uma lista (D) contendo os elementos dependentes, direta ou indiretamente, de ci (e.g., *softwares* que dependem dos computadores em que estão instalados ou serviços que dependem de outros serviços) (Linha 4). Essa lista é preenchida recursivamente, percorrendo as dependências definidas entre os CIs. No entanto, esse procedimento não é apresentado neste artigo por medida de simplificação. Feito isso, para cada elemento contido na lista D (Linha 5), é acumulada sua BsR na variável γ (Linha 6). Após percorrer todos os elementos de D , a tupla (CI, AR) é incluída no conjunto U (Linha 7), que ao final da função será retornado (Linha 8).

Algoritmo 2: *Função de Cálculo de Relevância Absoluta*

Entrada: V : visão da infra-estrutura de TI, CP : plano de mudança

Saída: conjunto de tuplas contendo CIs e suas relevâncias absolutas

1. $U \leftarrow$ conjunto vazio de tuplas (CI, relevância absoluta)
2. **for each** $ci \in$ conjunto de CIs manipulados pelo CP
3. **do** $\gamma \leftarrow$ BsR de ci
4. $D \leftarrow$ lista de todos os elementos dependentes diretos e indiretos de ci
5. **for each** $d \in D$
6. **do** $\gamma \leftarrow \gamma +$ BsR de d
7. $U \leftarrow U \cup \{ci, \gamma\}$
8. **return** U

Uma vez calculadas as ARs dos elementos, será função do módulo *Impact Estimation* fazer a normalização desses valores para uma escala de impacto. O *fator de impacto* (*Impact Factor* - IF) de um elemento representa a parcela da infra-estrutura de TI que é afetada pela falha de um determinado CI, no que diz respeito ao prejuízo causado para a continuidade do negócio. A função de cálculo do IF, detalhada no Algoritmo 3, recebe como entrada a saída da função de cálculo de AR realizado pelo Algoritmo 2. Para que seja possível calcular o impacto dos CIs em relação à infra-estrutura de TI, existe um elemento que representa a infra-estrutura gerenciada, do qual todos os outros CIs dependem. Esse elemento terá como AR a soma de todas as BsRs definidas e será manipulado em todas as RFCs. Inicialmente, o algoritmo instancia na variável t , o elemento que representa a infra-estrutura de TI como um todo (Linha 2) e a seguir utiliza um procedimento que localiza e extrai tal CI do conjunto R (Linha 3). Para cada tupla do conjunto R (Linha 4), é dividida a AR do CI contido na tupla i pela AR total do sistema contido na tupla T (Linha 5). Um conjunto I receberá os resultados dessas divisões (Linha 6) e será retornado ao final da função (Linha 7).

Algoritmo 3: *Função de Cálculo de Fator de Impacto*

Entrada: R : conjunto de tuplas contendo CIs e suas relevâncias absolutas

Saída: conjunto de tuplas contendo CIs e seus fatores de impacto

1. $I \leftarrow$ conjunto vazio de tuplas (CI, fator de impacto)
2. $t \leftarrow$ CI que representa a infra-estrutura de TI
3. $T \leftarrow extract_ci(t, R)$
4. **for each** $i \in$ conjunto tuplas R
5. **do** $\lambda \leftarrow$ AR de $i \div$ AR de T
6. $I \leftarrow I \cup \{ci, \lambda\}$
7. **return** I

Os resultados obtidos através dos cálculos das probabilidades de falha e dos impactos dos CIs servirão de base para a classificação dos riscos das atividades do plano de mudança a ser feita pelo módulo *Risk Classification*. O objetivo, ao se realizar estimativas de riscos automatizadas, é auxiliar o operador a compreender os riscos contidos em uma requisição de mudança. Por isso, os resultados precisam ser apresentados de forma clara e objetiva. O IRM [IRM 2002] recomenda que se quantifique a probabilidade e o impacto nas seguintes escalas: (i) alta (provável), média (possível) e baixa (improvável) para probabilidades e (ii) alto (significante), médio (moderado) e baixo (insignificante) para impacto. Os valores obtidos nos passos anteriores serão então mapeados nessas escalas, sendo que os índices de probabilidade e impacto (alto, médio e baixo) podem ser parâmetros do sistema e variar conforme a exigência do ambiente. Uma matriz de classificação de riscos, como a apresentada na Tabela 1, costuma ser utilizada pelas organizações modernas no gerenciamento de riscos. Finalmente, cada atividade do plano de mudança receberá uma classificação em uma das nove categorias da matriz para cada tipo de falha considerado. O algoritmo que classifica as atividades segundo as categorias de risco é trivial e não será apresentado neste artigo.

Na última etapa da análise de riscos será calculado o *risco médio* (*Mean Risk* - MR) de cada atividade do plano de mudança através do módulo *Mean Risk Calculator*. A entrada para esse módulo será o conjunto de atividades do CP classificadas segundo a matriz da Tabela 1 para cada tipo de falha considerado na análise. No entanto, apresentar

Tabela 1. Matriz de classificação de riscos

		Probabilidade de Falha		
Fator de Impacto	Impacto Alto Probabilidade Baixa Categoria 3	Impacto Alto Probabilidade Média Categoria 2	Impacto Alto Probabilidade Alta Categoria 1	
	Impacto Médio Probabilidade Baixa Categoria 6	Impacto Médio Probabilidade Média Categoria 5	Impacto Médio Probabilidade Alta Categoria 4	
	Impacto Baixo Probabilidade Baixa Categoria 9	Impacto Baixo Probabilidade Média Categoria 8	Impacto Baixo Probabilidade Alta Categoria 7	

a um operador diversas classificações de risco para cada atividade do CP pode acabar gerando uma quantidade de dados impraticável para avaliação, dependendo do número de atividades e de tipos de falha considerados. Por esse motivo, o *Mean Risk Calculator* calcula uma média harmônica dos valores das categorias de risco obtidos para cada tipo de falha, resultando em um valor de MR (em uma escala de 1 a 9) por atividade. Por exemplo, supondo que uma atividade do CP seja de instalação de um *software sw* sobre um sistema computacional *cs*. Onde a probabilidade de AF é Média com impacto Baixo (Categoria 8) e a probabilidade de RF é Baixa com impacto Alto (Categoria 3). Sendo assim, o MR da atividade de instalação de *software* classificada nas categorias 3 e 8 teria um valor de 4,36. A utilização da média harmônica funciona como uma abordagem pessimista para a estimativa de riscos, uma vez que esse cálculo sempre aproxima o resultado final da menor parcela, tendendo assim a priorizar a categoria com maior risco. O relatório de riscos exibido ao final da análise apresenta as atividades do CP ordenadas pelos seus valores de MR de forma crescente, levando as atividades com maior fator de risco para o topo da lista.

4. Protótipo

A fim de comprovar a funcionalidade da solução proposta para automatização da análise de riscos, foi desenvolvido um protótipo e incorporado ao sistema de gerenciamento de mudanças CHANGEEDGE, concebido em um esforço conjunto entre a HP e a UFRGS. Nesse sistema é utilizado um subconjunto de classes do CIM para representação da infraestrutura gerenciada e uma extensão do modelo de *workflow* proposto pelo WfMC (*Workflow Management Coalition*) [WfMC 2007] para expressar os planos de mudança. A seguir, serão descritos alguns detalhes técnicos do protótipo.

Conforme mencionado anteriormente neste artigo, os CIs da infra-estrutura de TI devem receber valores de BsR ajustados à sua importância frente ao negócio da organização. Para representar a BsR no protótipo foi definida uma métrica através da classe `BaseMatrixDefinition` do CIM. Essa métrica define uma faixa de valores de relevância possíveis para serem aplicados aos elementos gerenciados, por exemplo: Alta (1,00), Média (0,50) e Baixa (0,25). Para os elementos relevantes devem ser associadas instâncias de `BaseMatrixValue` contendo o valor de BsR atribuído ao CI. Caso não seja definida uma BsR a um CI, a função de cálculo de AR considerará o elemento irrelevante do ponto de vista do negócio (*i.e.*, BsR igual a zero).

Para representar dependências entre os CIs, o CIM define uma série de objetos que mapeiam relações entre itens de uma infra-estrutura de TI. Algumas dessas relações re-

presentam dependências explícitas como, por exemplo, `ServiceServiceDependency`, que indica quando um serviço depende de outro serviço para funcionar. Outras relações, apesar de não necessariamente representarem dependências, são consideradas como tal para a análise de riscos. Esse é o caso da relação `InstalledSoftwareElement`, que mapeia a dependência de um *software* para o sistema computacional onde ele está instalado. No protótipo, é utilizada uma lista de dependências, a qual é percorrida pelo algoritmo para calcular as ARs dos CIs.

A fim de aplicar as mudanças sobre a infra-estrutura de TI, o sistema CHANGELEDGE faz uso de um subsistema de implantação de mudanças (*deployment system*) que faz a tradução de um *workflow* de mudança em um documento BPEL (*Business Process Execution Language*) [Machado *et al.* 2008]. O documento BPEL gerado é então submetido para execução pelo sistema de orquestração de *Web services* ActiveBPEL [Active Endpoints 2008] que fará o controle da execução do *workflow* e tratamento de falhas. Cada CI da infra-estrutura de TI deve possuir uma interface de gerenciamento por *Web services* a ser invocada pelo ActiveBPEL para execução das atividades de mudança. Ao término de cada atividade o *Web service* de gerenciamento reporta a uma base de dados o *status* da execução, eventuais falhas ocorridas, tempo transcorrido, entre outros dados para popular os registros de execução da RFC.

Para fins de simulação, os *Web services* fornecidos pelos CIs introduzem falhas de forma pseudo-aleatória, segundo uma distribuição de probabilidades uniforme, durante a execução das atividades de mudança. Tais falhas, inseridas na forma de exceções, fazem com que o sistema de orquestração interrompa o fluxo normal do *workflow* e ative os planos de remediação associados. É possível definir diferentes probabilidades de falha para os diferentes tipos de atividade ou para falhas na manipulação de CIs específicos.

5. Avaliação Experimental

Com o objetivo de comprovar a usabilidade da solução de estimativa de riscos proposta neste trabalho, foi criado um ambiente de TI emulado, sobre o qual foram realizados testes e medições com uso do sistema CHANGELEDGE. A RFC definida para avaliação possui o objetivo de manter atualizado um sistema de envio e recebimento de e-mails de um domínio corporativo, incluindo funções como fazer *backup* dos dados dos usuários e atualizar filtros de lixo eletrônico (*spam*). Esse sistema utiliza em conjunto quatro elementos de *software*: Postfix para os servidores POP e SMTP, SquirrelMail para o serviço de Webmail, Apache como servidor HTTP para hospedagem do Webmail e um SpamAssassin para filtragem de *spam*. No cenário estabelecido, o serviço que possui maior relevância é o de troca de mensagens via POP/SMTP, uma vez que a maioria dos colaboradores utilizará programas cliente de e-mail. Sendo assim, o serviço de Webmail serve como uma opção de acesso alternativo às mensagens. O plano de mudança descrito na Figura 2 foi desenvolvido para ser executado periodicamente e manter todos os *softwares* envolvidos no fornecimento do serviço de e-mail atualizados.

Na ocasião da instalação do serviço de e-mail, a infra-estrutura de TI já continha um sistema de *Enterprise Resource Planning* (ERP), o qual utiliza um servidor dedicado (*System Server*) e um banco de dados próprio (*MySQL*). Porém, com o passar do tempo novos serviços foram sendo incorporados, evoluindo a infra-estrutura gerenciada para um novo estado, conforme representado na Figura 3. A organização, que antes for-

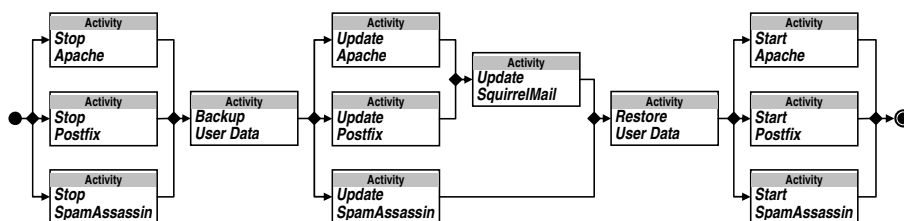


Figura 2. Workflow do plano de mudança

neia produtos através de venda direta ou televentas, passou a oferecê-los também por comércio eletrônico, assim como o suporte aos consumidores passou a ser prestado por uma aplicação de suporte *on-line*. As dependências entre os objetos mapeadas na Figura 3 são representadas pelas setas, indicando, por exemplo, que os novos serviços de vendas e suporte *on-line* dependem do serviço prestado pelo *software* Apache para funcionarem. A BsR dos elementos é representada pelos números posicionados na parte inferior-direita das caixas, sendo que a escala de relevâncias utilizada foi: Máxima (1,00), Alta (0,75), Média (0,50), Baixa (0,25) e Nenhuma (0,00).

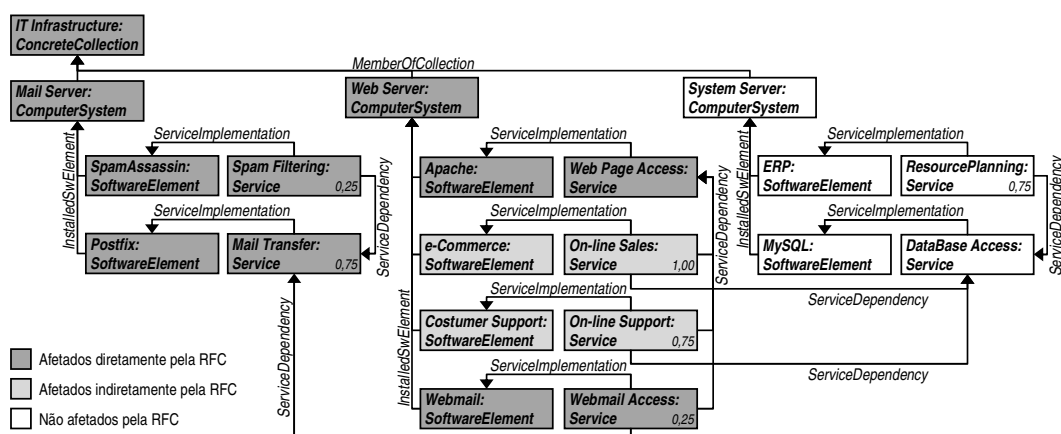


Figura 3. Representação atual da infra-estrutura de TI

No momento da criação do plano de mudança da RFC descrita, os índices de riscos eram baixos. Porém, com as mudanças na infra-estrutura, esse plano de mudança deve ser repensado, para que os riscos sejam reduzidos a níveis aceitáveis. O resultado da análise de riscos realizada sobre esse CP, considerando a infra-estrutura de TI da Figura 3, é mostrado na Tabela 2-(a). Observando esse resultado, fica claro que as quatro atividades que possuem maior risco são executadas sobre o mesmo servidor (*Web Server*). Isso acontece devido aos efeitos indiretos que a mudança tem sobre elementos que não fazem parte da RFC, mas que dependem dos serviços alterados por ela. Por exemplo, das atividades com maior risco no plano de mudança, três manipulam o *Apache*, que ainda provê serviço para outras três aplicações, enquanto uma atualiza o *SquirrelMail*, o qual dentro do serviço de e-mail possui um papel secundário.

Frente a esse cenário, uma alteração sobre a infra-estrutura de TI foi promovida, a fim de minimizar o impacto das atividades de maior risco. Essa alteração contemplou a migração do *SquirrelMail* para o *Mail Server*, combinada com a instalação de um servidor HTTP exclusivo para o serviço de Webmail nessa mesma máquina. Ajustando o plano de

mudança à nova realidade, a análise de riscos apresenta novos resultados conforme a Tabela 2-(b). É notório que houve uma redução dos índices de risco das atividades que manipulam o *Apache* e o *SquirrelMail*, logo, a modificação feita na infra-estrutura obteve êxito em reduzir os impactos do plano de mudança sobre o negócio da organização.

Tabela 2. Resultados da análise de riscos antes e depois da mudança promovida

(a) Resultado no cenário atual		(b) Resultado após a mudança	
Atividade	Mean Risk	Atividade	Mean Risk
Update Apache	2,40	Update Postfix	5,45
Start Apache	3,00	Stop Postfix	6,00
Stop Apache	3,00	Backup User Data	6,00
Update SquirrelMail	4,50	Restore User Data	6,00
Update Postfix	5,45	Start Postfix	6,00
Restore User Data	6,00	Update SpamAssassin	6,86
Backup User Data	6,00	Update Apache	6,86
Stop Postfix	6,00	Update SquirrelMail	7,20
Start Postfix	6,00	Stop Apache	7,20
Update SpamAssassin	6,86	Stop SpamAssassin	7,20
Stop SpamAssassin	7,20	Start SpamAssassin	7,20
Start SpamAssassin	7,20	Start Apache	7,20

A redução dos indicadores de riscos não comprova, por si só, uma efetiva melhora na qualidade do plano de mudança. A ITIL recomenda que sejam utilizadas medidas para analisar o desempenho das mudanças implementadas em uma infra-estrutura de TI. Uma dessas medidas é um fator de indisponibilidade dos serviços (*Service Disruption - SD*) originado por mudanças mal sucedidas. O SD depende do tempo que transcorre após uma falha em uma mudança até que o sistema seja capaz de recuperar a consistência da infra-estrutura gerenciada, como demonstrado na Figura 4. Além disso, o SD deve levar em consideração o impacto do serviço afetado. Neste trabalho, é utilizada a Equação 1 para o cálculo do SD para uma dada atividade i de um plano de mudança. O cálculo é feito multiplicando três parcelas: $(F_{x,i})$ total de falhas de um tipo x encontradas nos registros de execução da RFC para a atividade i , $(t_{x,i})$ tempo médio de recuperação do sistema para uma falha do tipo x em uma atividade i (pode ser obtido analisando os registros de execução das atividades de remediação) e $(IF_{x,i})$ fator de impacto do elemento afetado pela falha do tipo x da atividade i . Esses produtos são somados para cada tipo de atividade considerado (nesta simulação utilizou-se AF e RF).

$$SD_i = (F_{AF,i} * t_{AF,i} * IF_{AF,i}) + (F_{RF,i} * t_{RF,i} * IF_{RF,i}) \quad (1)$$

Para avaliar o fator de SD da RFC de atualização do serviço de e-mail, foi criado um ambiente de TI emulado onde foram reproduzidos os dois planos de mudança e a infra-estrutura de TI apresentados nesta seção. Os planos de mudança foram submetidos para implantação 30 vezes cada (representando uma execução semanal durante pouco mais de 6 meses) e falhas foram inseridas de forma pseudo-aleatória em suas atividades. Os percentuais de falha inseridos foram: 20% para AF de *update*, 5% para AF de *start/stop*, 1% para AF de *backup/restore* e 5% para RF de qualquer atividade. Considerando as falhas injetadas durante a emulação e os tempos de recuperação do sistema, o plano de mudança original executado sobre a infra-estrutura da Figura 3 obteve um valor total de SD (somando o SD_i de todas as atividades) de 19,43. Enquanto que a execução do plano modificado com base na análise de riscos atingiu um valor total de SD de 14,31,

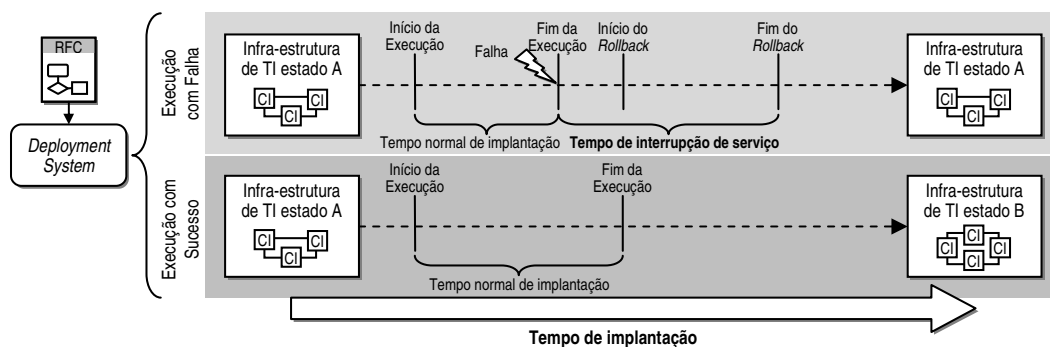


Figura 4. Tempo de interrupção de serviços devido a falhas em implantação de mudanças

o que representa uma redução de aproximadamente 26% na interrupção dos serviços e, conseqüentemente, uma melhora no desempenho do plano de mudança.

6. Conclusões e Trabalhos Futuros

Neste trabalho, foi discutida a necessidade das organizações em utilizar políticas racionais de gerenciamento de mudanças para suas infra-estruturas de TI. Foi visto que falhas durante a implantação dessas mudanças são uma realidade, e que as mesmas podem ter efeito direto na continuidade do negócio. Sendo assim, é fundamental que os riscos associados às mudanças sejam estimados e mitigados, porém, esse processo de avaliação de riscos geralmente fica sob responsabilidade de humanos. Por esse motivo, neste artigo foi proposta uma solução para automatização da estimativa de riscos em gerenciamento de mudanças, visando auxiliar os administradores a minimizar as possibilidades das mudanças causarem danos aos serviços suportados pela infra-estrutura de TI.

Os resultados obtidos demonstraram, em um primeiro momento, que a estimativa proposta neste trabalho é capaz de gerar indicadores de riscos para planos de mudança com base nas informações contidas no sistema de gerenciamento, analisando o histórico de execuções de uma RFC e a visão da infra-estrutura de TI. Essa estimativa se mostrou útil para identificar ameaças em um plano de mudança, servindo como base para criação de medidas de tratamento dos riscos e para tomada de decisões estratégicas durante o planejamento de mudanças. Além disso, uma medida de indisponibilidade de serviços foi utilizada para comparar os diferentes planos de mudança, que revelaram riscos distintos entre si. A redução dos índices de riscos, que ocasionou em uma melhora no fator SD, indica que os relatórios da estimativa de riscos automatizada refletem ameaças reais aos serviços prestados.

Na estimativa de riscos proposta neste artigo foram considerados apenas dois tipos de falha dentre os seis previstos pela classificação adotada. Porém, a solução se mostrou perfeitamente ajustável para contemplar outras classificações. Em trabalhos futuros podem ser analisadas, por exemplo, probabilidades de falhas de humanos alocados para as atividades manuais do plano de mudança. Essa análise poderia auxiliar na alocação de recursos humanos de forma mais adequada considerando as falhas ocorridas em execuções anteriores. Além disso, seria interessante considerar outras possibilidades de combinação dos valores de probabilidade de falha e impacto (além da classificação da Tabela 1), procurando entender as diferenças entre os resultados obtidos.

Referências

- Active Endpoints (2008). ActiveBPEL Open Source Engine. <http://www.activebpel.org>.
- Chicken, J. e Posner, T. (1998). *The Philosophy of Risk*. Thomas Telford.
- Cordeiro, W. L. C., Machado, G. S., Daitx, F. F., *et al.* (2008). A Template-based Solution to Support Knowledge Reuse in IT Change Design. In *11th IEEE/IFIP Network Operations and Management Symposium (NOMS 2008)*, pages 355–362, Salvador, Brazil.
- DMTF (2008). Common Information Model. <http://www.dmtf.org/standards/cim>.
- Fewster, R. e Mendes, E. (2001). Measurement, prediction and risk analysis for Web applications. *7th International Software Metrics Symposium, 2001. METRICS 2001*, pages 338–348.
- IRM (2002). *A Risk Management Standard*. The Institute of Risk Management, United Kingdom.
- ITIL (2007). *ITIL - Information Technology Infrastructure Library: Service Transition Version 3.0*. Office of Government Commerce (OGC).
- ITIL (2008). ITIL - Information Technology Infrastructure Library (ITIL). <http://www.itil-officialsite.com/>.
- Machado, G. S., Cordeiro, W. L. C., Santos, A. D., *et al.* (2008). Algoritmo para Geração Automática de Ações de Rollback em Sistemas de Gerenciamento de Mudanças em TI. In *Brazilian Symposium on Computer Networks (SBRC 2008)*, Rio de Janeiro, Brazil.
- Machado, G. S., Wickboldt, J., Cordeiro, W. L. C., *et al.* (2009). Refined failure remediation in it change management systems. In *Mini-conference of 11th IFIP/IEEE International Symposium on Integrated Network Management (IM 2009)*, New York, USA.
- Marques, M. e Neves-Silva, R. (2007). Risk assessment to support decision on complex manufacturing and assembly lines. *5th IEEE International Conference on Industrial Informatics*, pages 1209–1214.
- Rebouças, R., Sauv e, J., Moura, A., *et al.* (2007). A Decision Support Tool to Optimize Scheduling of IT Changes. In *10th IFIP/IEEE International Symposium on Integrated Network Management (IM 2007)*, pages 343–352, Munich, Germany.
- Sauv e, J., Santos, R. A., Almeida, R. R., Moura, A., *et al.* (2007). On the Risk Exposure and Priority Determination of Changes in IT Service Management. In *18th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management (DSOM 2007)*, pages 147–158, San Jose, CA, USA.
- WfMC (2007). Workflow Process Definition Interface - XML Process Definition Language. http://www.wfmc.org/standards/docs/TC-1025_10_xpdl_102502.pdf.
- Wickboldt, J. A., Machado, G. S., Cordeiro, W. L. C., *et al.* (2009). A Solution to Support Risk Analysis on IT Change Management. In *Mini-conference of 11th IFIP/IEEE International Symposium on Integrated Network Management (IM 2009)*, New York, NY, USA.

APPENDIX C PUBLISHED PAPER - DSOM 2009

In this appendix the paper entitled “Improving IT Change Management Processes with Automated Risk Assessment” is presented. In this paper, for the first time similarity between workflows was employed in the context of this research in order to calculate the risks of Change Plans (CPs). So far, it was only possible to assess risks of changes that were implemented recurrently several times. By employing the similarity calculation proposed in this paper, it became possible to perform automated risk assessment for changes that had been just designed. Furthermore, in this paper human allocation for changes was considered, therefore Human Failures (HF) were also possible in the case study. The results showed how Service Disruption (SD) could be decreased by reallocating the same humans in a given Change Plan base on the results of automated risk assessment.

- **Title:**
Improving IT Change Management Processes with Automated Risk Assessment
- **Conference:**
20th International Workshop on Distributed Systems: Operations and Management (DSOM 2009)
- **URL:**
<http://www.manweek.org/2009/dsom/>
- **Date:**
26-30 October 2009
- **Venue:**
Telecom Italia Future Centre, Venice, Italy
- **Digital Object Identifier (DOI):**
http://dx.doi.org/10.1007/978-3-642-04989-7_6

Improving IT Change Management Processes with Automated Risk Assessment

Juliano Araujo Wickboldt¹, Luís Armando Bianchin¹, Roben Castagna Lunardi¹,
Fabrício Girardi Andreis¹, Weverton Luis da Costa Cordeiro¹, Cristiano Bonato
Both¹, Lisandro Zambenedetti Granville¹, Luciano Paschoal Gaspar¹, David
Trastour² and Claudio Bartolini³

¹ Federal University of Rio Grande do Sul, Porto Alegre, Brazil

² Hewlett Packard Laboratories, Bristol, UK

³ Hewlett Packard Laboratories, Palo Alto, USA

Abstract. The rational management of IT infrastructures is a goal of modern organizations that aim to deliver high quality services to their customers in an affordable way. Since changes are imminent in such a dynamic environment, failures during this process may directly affect business continuity. Hence, risk assessment is a key process in IT change management. Despite its importance, risks are usually assessed by humans based on empirical knowledge, leading to inaccurate basis for decision making. In this paper, we present a solution for automating the risk assessment process, which combines historical data from previous changes and analyzes impact of changes over affected elements. A prototypical system was developed to evaluate the solution on an emulated IT infrastructure. The results achieved show how the automated solution is capable of raising the quality of changes, therefore reducing service disruption caused by changes.

1 Introduction

Modern organizations take advantage of information technology (IT) resources and services to add value to their businesses. The heterogeneity of these technologies, which together constitute an IT infrastructure, makes the task of IT management increasingly complex. In this scenario, the rational management of IT infrastructures improves the quality of provided services and reduces operational costs. For consistent and secure maintenance of these infrastructures, the Office Government Commerce (OGC) has introduced the Information Technology Infrastructure Library (ITIL) [1], which is a set of processes and best practices that provides guidance for the proper management of IT resources and services.

Being one of the core processes of ITIL, change management [2] provides general guidelines for conducting changes over IT infrastructures, from the early specification to the final deployment and evaluation. It defines that all changes should be described in a document called Request for Change (RFC). An RFC specifies, in a declarative way, what should be done and the primary Configuration Items (CIs) affected (devices, applications, services, etc.), but not detailing how the change should be implemented. This must be indeed performed by human operators or even by an automated management system. Subsequently, RFCs must be reviewed, approved, and scheduled by the Change

Advisory Board (CAB). This committee, usually chaired by a change manager, should be composed of people with extensive knowledge on the organization's processes, often coming from different areas, but not necessarily familiar with the underlying technologies deployed in IT infrastructure.

IT infrastructures support services that are essential for business continuity. Hence, when changes to the managed infrastructure are required, the risks associated with it should be considered. According to ITIL, risks should be measured and treated before a change is approved. Risk mitigation aims to reduce the possibility of changes causing unnecessary disruption to changed services. Risks in IT change management should be observed as a combination of the probability of occurrence of potentially negative events and their impact to business continuity [1]. Examples of such events include: failure on software installation, incorrect configurations, and physical defects in CIs.

Risk assessment has been typically performed by human operators, often based only on empirical knowledge. However, due to the large number of CIs associated with a change request and the amount of variables that should be considered (*e.g.*, history of failures and impact of affected CIs), the adoption of such approach may end up presenting superficial and/or inaccurate results to serve as basis for decision making. Despite the recommendations proposed by ITIL, it does not present a practical method for risk assessment in change management. Recently, some authors have proposed solutions for the automation of change management in its several phases [3] [4] [5]. Nevertheless, no previous work proposed an automated approach for the risk assessment in the planning phase of change management. By employing a proper method for risk assessment, an automated system could aid the human operator to quickly identify threats in a requested change before deploying it to the IT infrastructure. Based on risk reports, the operator would be able to modify the original RFC or even adapt the IT infrastructure, in order to reduce the possibility of occurrence of change related incidents.

To address the aforementioned issue, we propose in this paper a solution for automating risk assessment in IT change management. Our solution is based on the history of executions of changes over an IT infrastructure, observing the occurrence of previous deployment failures and identifying potential issues for future executions. With this solution we aim to provide a change management system with support for proactive treatment of incidents, enabling operators to redesign changes in order to reduce occurrence deployment failures upon change executions.

The remainder of this paper is organized as follows. Section 2 briefly reviews some of the most prominent research initiatives in risk and change management. Section 3 details our solution for automated risk assessment, whereas Section 4 describes the prototypical implementation developed. In Section 5 we present an experimental evaluation conducted to measure the results of the solution. Finally, Section 6 closes this paper with concluding remarks and prospective directions for future work.

2 Related Work

Risk management is a cross-discipline that has been investigated and employed in several different areas. Risk assessment, for example, can be a tool for guiding financial investments [6], health care decisions [7], and the strategies of insurance companies

[8]. According to the Institute of Risk Management (IRM)¹, the risk management discipline defines the process whereby organizations methodologically address the risks associated with their activities, aiming at achieving sustained benefits [9].

The literature usually defines risks as events whose potential consequences may be either positive or negative to the successful accomplishment of a goal. However, in practice, the negative aspect is far more considered, mainly in critical areas such as health care. The actual result is that risk management becomes strongly focused on the prevention and mitigation of harms. This observation also holds in the investigations on risks associated to the design and operation of computational systems.

Some authors have employed probabilistic models to predict undesired events as well as estimate metrics for risk management in IT. Fewster and Mendes [10] have proposed a framework that, using a Generalized Linear Model (GLM), is able to analyze the risks associated with the development of Web-based systems. The authors showed that GLM was effective in predicting the risks of, for example, overcoming project budget or violating final deployment deadlines. Hearty *et al.* [11], in turn, have designed a model for effort prediction and risk assessment in software development projects that follow the Extreme Programming (XP) methodology. The author's approach is based on the use of Bayesian Networks (BNs), and quantitatively estimates project metrics (*e.g.*, iterations/time to complete) without requiring data about the success of past XP projects. Fenton and Neil [12], in another research, have shown that BNs are also an effective mechanism for predicting software defects. Although relevant, these researches have only considered risks in terms of the probability of occurrence of adverse events; the severity of the impacts that such events might have on the affected projects or businesses has not been taken into account.

On the other hand, Marques and Neves-Silva [13] have proposed a method for risk assessment to help in the decision making on complex assembly lines. The authors propose to compute risks – in terms of both probability and impact of possible incidents – considering information collected during the system operation. This method was designed to run in an environment where the required parameters for calculating incident's probability and impact have well known values, for a limited set of possible events. In IT change management, however, due to the dynamics of IT environments, the amount and diversity of incidents that can happen is likely uncountable. Solutions able to cope with such a diversity is then still required.

In the context of IT change management, Sauv e *et al.* [5] have proposed a risk analysis method to support the scheduling of Request for Changes (RFCs). Their primary objective was to determine priorities for the implementation of potentially concurrent RFCs over a common managed IT infrastructure. The proposed method is heavily based on estimates of deployment time of RFCs and the way they can be scheduled at different moments, affecting the impact of change deployment to business objectives. Their work, however, applies to the scheduling phase of change management, and does not consider the risks associated to improper planning of RFCs, thus leaving no room for possible RFC adjustments. Aiming to deal with failures during change deployment, Machado *et al.* [4] proposed a solution that treats change failures in a reactive fashion, undoing the requested changes over a damaged system backwards to its previous con-

¹ <http://www.theirm.org/>

sistent state. In spite of the advances, a solution that proactively observes risks to avoid future (and potentially expensive) system rollbacks is still lacking.

The importance of risk assessment in IT change management lies in the fact that failures on change implementation may cause disruption of services that are relevant to business. This is underscored by the fact that some changes may look innocuous and, even indirectly, cause harm beyond their apparent complexity [2]. Oppenheimer *et al.* [14] have investigated several component failures in large-scale Internet services, concluding that human operator error is the leading cause of failures in these services. Automation of maintenance and operation of large-scale systems is a key factor to enhance service availability. In this context, as far as the authors of this paper are aware of, there is no automated method for estimating risks in the planning phase of change management. In the next sections, we envisage a solution for risk assessment in change planning and the way it may act as a tool to help operators designing better RFCs.

3 Automated Risk Assessment Solution

In order to support risk assessment in the context of IT management, we have introduced, in a previous work [15], a new component – called *Risk Analyzer* – in the conceptual architecture of the CHANGELEDGE system [3]. In this paper, we introduce and detail mechanisms, algorithms, and equations for (i) processing information collected from the IT environment and (ii) estimating risks based on metrics to quantify probability of failures and impact on affected elements. Early in this section, we review the traditional change management process, as envisaged by ITIL and materialized by the CHANGELEDGE system; whose architecture is depicted in Figure 1. Afterwards, we present how automatic risk assessment is performed in this context.

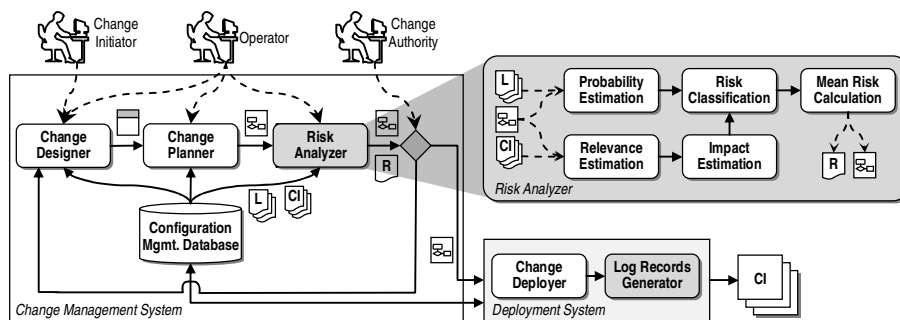


Fig. 1. Architecture of a change management system with risk assessment support

The change process starts when the *Change Initiator* specifies an RFC by interacting with the *Change Designer* component. Subsequently, the *Operator* sketches a preliminary Change Plan (CP), which consists of a workflow of high-level activities that describe the steps required to deliver the requested change. This workflow will be

further refined by the *Change Planner* component. The outcome of this refinement is a CP composed of finer-grained activities that can be actually deployed over CIs [3].

In a system without risk assessment support, at this point an RFC would be ready to be approved by a *Change Authority*, scheduled, and deployed. However, these changes may expose the provisioned services to unnecessary or unknown risks. Therefore, the *Risk Analyzer* component (detailed in Figure 1 top right) automatically estimates risks in the refined CP. As input, the *Risk Analyzer* receives, from the *Change Planner* the change that will be the subject of analysis. This component also consumes the (i) execution records (list of logs L) of previous change deployments and (ii) the updated view of the IT infrastructure (list of CIs), both available from the *Configuration Management Database (CMDB)*. By processing these inputs, the *Risk Analyzer* automatically generates a *Risk Report (R)*. Analyzing this report, the *Change Authority* could then decide whether the risks of deploying the original RFC are acceptable or not. If not accepted, the RFC is returned to be redesigned, aiming at mitigating the reported risks. This could be done, for instance, by modifying the original workflow or the CIs affected by the chance. If the risks are considered acceptable, the CP is then scheduled and finally submitted to be deployed by the *Deployment System*.

As mentioned, in order to estimate the probability of failures, the *Risk Analyzer* processes information from the execution records of RFCs. These records (following the information model proposed in a previous work [15]) are produced by the *Log Records Generator* during the deployment of RFCs. These records represent the execution traces of CPs obeying the original sequence of activities performed. Moreover, they include information about succeeded and failed executions; in the case of unsuccessful deployments, they include the failure classification and remediation actions taken. In the adopted information model, failures are classified into six categories: Activity Failure (AF), Resource Failure (RF), Human Failure (HF), Time Failure (TF), External Trigger (ET), and Constraint Violation (CV). In this work, however, we focus our evaluation on three of these categories: AF represents failures inherent to the activities of the CP (e.g., software installation failure); RF represents failures on the resources handled in activities (e.g., hardware damage during deployment); and HF represent the failures caused by incorrect actions taken by human operators. Some types of failures may not be easily caught by a failure detection system (especially HF). In these cases, the operator that reviews and closes RFCs should insert these records to enable future risk estimation.

For impact estimation, the automated risk assessment process requires a metric that represents the importance of the CIs to business. In this work, we propose a metric called Business Relevance (BsR), which is associated to every CI that is relevant to the business continuity. BsR is expressed by a numerical value and, regardless of the scale adopted, it should enable comparisons between relevancies of different CIs. Along with the BsR, relationships and dependencies between CIs are collected from the CMDB and used for impact estimation in the risk assessment process.

3.1 Probability of Failures Estimation

In order to present the behavior of the *Probability Estimation* module, we first introduce the definitions and metrics that support this module. One key aspect in probability estimation is the way several probabilities of failure from different RFCs compose a single

probability weighted by a metric which we call Risk Affinity (RA). The goal of RA is to capture the similarities between two workflows according to a given failure type, as shown in Equation 1. This equation uses a function θ that returns a value (ranging from zero to one) that represents the *likeness* of the k^{th} pair of activities of the two workflows according to the failure type ft . In other words, the θ function considers the percentage of coincident CIs involved in pairs of activities (*e.g.*, compares involved computers, software, and humans). However, in the case of ft been an RF, θ only returns more than zero if the activities' actions and resources are the same (*e.g.*, same computer). The RA metric is computed by a sum of *likeness* of k pairs of activities up to the size of the smaller workflow, divided by the size of the bigger one. This enables RA to capture not only local differences of activities but also to distinguish workflow sizes.

$$RA(A, B, ft) = \frac{\sum_{k=0}^{\min(|A|, |B|)} \theta_k(A, B, ft)}{\max(|A|, |B|)} \quad (1)$$

Three other functions are still required by the *Probability Estimation* module. The first one, called *influences*, returns a subworkflow of activities that influence a given activity a in the scope of a CP. We say that an activity b influences an activity a when b is executed either before or in parallel with a in the CP. The second function, *alike_enough*, returns *true* when an activity is found to be similar to another in the context of a failure type. For example, for AFs, activities that perform the same action over the same software element are regarded as similar. The third function, *possible_failure_types*, returns a set of possible failure types that may happen, given and activity a (*e.g.*, HF can only happen if a is a manual activity).

The process of estimating probabilities is performed by the Algorithm 1. Intuitively, probabilities are calculated by dividing two values: (i) the sum of failure occurrences of a given activity in a set of RFCs (dividend) and (ii) the sum of the total executions of the same activity in the same set of RFCs (divisor). These two values are weighted by the RA between the analyzed CP and others extracted from the execution records. The idea is to reuse the logs from RFCs that have very similar CPs, prioritizing also similar activities that have a significant number of historical executions.

In order to calculate these probabilities, the Algorithm 1 receives as input the *CP* of the RFC under analysis and a set of all execution records of RFCs available in the CMDDB (for performance matters, this set is previously filtered matching RFCs having the same set of affected CIs as the one under analysis). Then, for each activity a of *CP* (Line 2), a subworkflow CP' containing a and the activities that influence its execution is defined (Line 3). Following, for every possible failure type ft (Line 4), the algorithm iterates through all CPs from set R (Line 6) searching for activities that meet the *alike_enough* criteria (Line 8). After that, RCP' will be a subworkflow with the activity b and all activities that influence it in cp (Line 9). Based on CP' , RCP' , and a failure type the RA between both subworkflows is computed (Line 10). The result of the RA (stored in A) acts as a weight for prioritizing failure probabilities of RFCs that have similar workflows. Following, the executions and failures of b are weighted and stored in T and F respectively (Lines 11 and 12). Probability of failures for each

activity and failure type are then calculated by dividing F by T (Line 13) and added to the set S (Line 14). At the end S is returned as output of the function (Line 15).

Algorithm 1: *Probability of Failures Calculation Function*

Input: R : set of CPs with their execution records (logs), CP : change plan

Output: set of tuples containing activity, failure probability, and failure type

```

1.  $S \leftarrow$  set of empty tuples (activity, failure probability, failure type)
2. for each Activity  $a \in CP$ 
3.   do  $CP' \leftarrow influences(a, CP)$ 
4.     for each FailureType  $ft \in possible\_failure\_types(a)$ 
5.       do  $T \leftarrow 0; F \leftarrow 0;$ 
6.         for each ChangePlan  $cp \in R$ 
7.           do for each Activity  $b \in cp$ 
8.             do if  $alike\_enough(a, b, ft)$ 
9.               then  $RCP' \leftarrow influences(b, cp)$ 
10.                  $A \leftarrow RA(CP', RCP', ft)$ 
11.                  $T \leftarrow T + (\text{executions of } b \text{ in logs of } cp * A)$ 
12.                  $F \leftarrow F + (\text{failures of type } ft \text{ for } b \text{ in logs of } cp * A)$ 
13.            $\varphi \leftarrow F \div T$ 
14.            $S \leftarrow S \cup \{a, \varphi, ft\}$ 
15. return  $S$ 

```

3.2 Impact Estimation

Another functionality of the *Risk Analyzer* is to estimate the impact of a change on the CIs. Initially, the *Relevance Estimation* module computes the Absolute Relevance (AR) of the items handled in the CP, by means of the Algorithm 2. AR is a metric that indicates the overall perception of relevance of an element to the business continuity, including its BsR and the sum of BsR of all elements that depend on it, directly or indirectly. In this algorithm, for each CI ci handled in the CP (Line 2), the value of the AR for the element ci (variable γ) is initiated with its own BsR (Line 3). Subsequently, a set D is created and populated with elements that depend, directly or indirectly, on ci (e.g., software that depend on the computer where it is hosted or services that depend on other services) (Line 4). This set is filled in recursively by iterating through dependencies defined between CIs. Following, each element that belongs to D (Line 5) will have its BsR accumulated in the variable γ (Line 6). Finally, the tuple (CI, AR) is included in the set U (Line 7), and at the end of calculation U is returned (Line 8).

After AR computation, the *Impact Estimation* module will proceed with the normalization of these values to a metric we call Impact Factor (IF). This metric represents the portion of the infrastructure that is compromised by failure of a particular CI. The IF calculation function (Algorithm 3) receives as input the output of Algorithm 2. In order to calculate the IF of CIs, we define an element that represents the IT infrastructure, whose all CIs depend on. The AR of this element is the sum of all BsRs defined, and it is handled in all RFCs. Firstly, the algorithm instantiates the variable t with the element

that represents the IT infrastructure (Line 2). Then, it invokes a pre-defined procedure that locates and extracts the CI t from the set R . (Line 3). For each tuple of the set R (Line 4), the AR from the CI contained this tuple is then divided by the AR of the whole infrastructure contained in tuple T (Line 5). Finally, a set I receives the results of these divisions (Line 6), which is returned as output of the function (Line 7).

Algorithm 2: *Absolute Relevance Calculation Function*

Input: V : updated representation of IT infrastructure, CP : change plan

Output: set of tuples containing CIs their Absolute Relevancies

1. $U \leftarrow$ empty set of tuples (CI, AR)
2. **for each** ConfigurationItem $ci \in$ set of handled CIs of CP
3. **do** $\gamma \leftarrow$ BsR of ci
4. $D \leftarrow$ set of CIs that depend on ci
5. **for each** ConfigurationItem $d \in D$
6. **do** $\gamma \leftarrow \gamma +$ BsR of d
7. $U \leftarrow U \cup \{ci, \gamma\}$
8. **return** U

Algorithm 3: *Impact Factor Calculation Function*

Input: R : set of tuples containing CIs and their Absolute Relevancies

Output: set of tuples containing CIs and their Impact Factors

1. $I \leftarrow$ empty set of tuples (CI, IF)
2. $t \leftarrow$ CI that represents the whole IT infrastructure
3. $T \leftarrow extract_ci(t, R)$
4. **for each** Tuple $i \in R$
5. **do** $\lambda \leftarrow$ (AR of i) \div (AR of T)
6. $I \leftarrow I \cup \{ci, \lambda\}$
7. **return** I

3.3 Classifying and Reporting Risks

The results obtained with Algorithms 1 and 3 (respectively, probability of failure and impact of change) serve as input for the classification of risks of the activities belonging to the CP under analysis, which is performed by the *Risk Classification* module. The main objective when automating risk assessment is to provide support for decision making on the approval of RFCs. Therefore, the results must be presented in clear and objective way. IRM [9] recommends quantifying probability and impact using the following scales: (i) high (more than 25%), medium (between 25% and 2%), and low (less than 2%), for probabilities, and (ii) high (significant), medium (moderate), and low (insignificant), for impact. The results obtained in previous steps are then mapped to these scales according to the risk classification matrix presented in Table 1. According to this matrix, each activity of the change plan may be then classified in one of nine categories.

Table 1. Risks Classification Matrix

	Probability of Failure		
Impact Factor	High Impact Low Probability Category 3	High Impact Medium Probability Category 2	High Impact High Probability Category 1
	Medium Impact Low Probability Category 6	Medium Impact Medium Probability Category 5	Medium Impact High Probability Category 4
	Low Impact Low Probability Category 9	Low Impact Medium Probability Category 8	Low Impact High Probability Category 7

In the last step of risk assessment process, the Mean Risk (MR) of activities is calculated by the module *Mean Risk Calculator*. The input of this module is a set of activities classified according to the matrix from Table 1, considering each possible failure type. However, a report with several risk classifications for each activity of a CP may not be practical for a human to analyze and draw conclusions over it. For this reason, in this step, a harmonic mean of the categories of risk is calculated, resulting in a value of MR (ranging from 1 to 9 continuously) for each activity. For instance, assuming an activity of a CP that installs a software *sw* on a computer system *cs*. This activity has Activity Failure (AF) probability medium with low impact (Category 8), and Resource Failure (RF) probability low with high impact (Category 3). In this example, the MR of this activity results in a value of 4.36. The use of harmonic mean approximates the MR to the lowest risk category value, therefore working as a pessimistic approach, and prioritizing categories with highest risk. A *Risk Report (R)* is shown at the end of the automated risk assessment process, displaying activities sorted descending by MR values, having riskier activities at the top of the list.

4 Prototype Implementation

In order to evaluate the technical feasibility of our solution, a prototype has been developed and incorporated into a change management system, designed by our research group, called CHANGELEDGE. This system uses a subset of classes from the Common Information Model (CIM), proposed by the Distributed Management Task Force (DMTF) [16], to implement a representation of the managed IT infrastructure. The RFC and change plan documents are formalized using an extension of a model proposed by the Workflow Management Coalition (WfMC) [17], which was introduced and detailed in a previous work [3].

As mentioned earlier in this paper, the CIs of an IT infrastructure should have BsR values associated that represent their importance to the organization's business. To materialize the BsR in the prototype, a metric was employed using the CIM *Base Matrix Definition* class. This class defines a range of possible values for relevance to be applied to the managed elements, for example: High (1.00), Average (0.50), and Low (0.25). Elements that have some degree of relevance to the business continuity must have instances of *Base Matrix Value* associated with a BsR value assigned. If no BsR value is

assigned for a specific CI, the AR calculation will consider the element as irrelevant for the business (*i.e.*, BsR zero).

In order to represent dependencies between CIs, CIM defines several objects that implement relationships between items of an IT infrastructure. Some of these relationships explicitly represent dependencies, such as *Service Service Dependency* indicating when a service requires features from another service to work properly. Other relationships, though not necessarily representing dependencies, are considered as such by the risk analysis. This is the case of *Installed Software Element*, which implements a dependency of a software element to the computer system where it is hosted. In our prototype, a list of objects that represent dependencies is employed, which is iterated by the algorithm that calculates ARs of ICs.

For deployment of changes, CHANGELEDGE makes use of a subsystem called *Deployment System*. It is responsible for translating the CP to be deployed into a BPEL (Business Process Execution Language) document [4]. The generated document is then submitted for execution by a Web services orchestration system called ActiveBPEL [18], which controls the execution of workflows and captures failures. Each CI of the IT infrastructure should have a management interface via Web services to be invoked by ActiveBPEL in order to implement change activities. After performing each activity, the Web service interface reports to a database: the status of implementation, failures occurred, and time elapsed in the execution of activity.

For simulation purposes, each Web service implemented by the CIs produces failures pseudo-randomly, according to a uniform probability distribution, during the deployment of changes. Such failures are injected as exceptions and compel the orchestration system to interrupt the regular execution flow starting associated remediation plans. The Web services are customizable to associate different probabilities of failure for different failure types of specific CIs.

5 Experimental Evaluation

In order to evaluate our solution, tests and measurements have been performed on an emulated IT environment. To measure the performance of changes, one of ITIL's recommendation is to use a Service Disruption (SD) metric, which reflects damage to services caused by unsuccessful changes. This metric represents the time elapsed after a failure on change deployment until the system recovers the managed infrastructure. In addition, SD should consider the impact of failures over the affected services. To this end, we propose Equation 2 to calculate the SD for a given activity i of a CP. The calculation is performed by multiplying three factors: (i) $F_{ft,i}$ which is the total number of failures of a type of ft found in the execution records of activity i ; (ii) $t_{ft,i}$ representing the average time to recover the system from a failure of same type in activity i (may be obtained from the execution records of remediation activities); and (iii) $IF_{ft,i}$ which contains the impact factor of the CI affected by the failure of type ft handled in activity i . The sum of these values for each failure type considered in the risk estimation results in an SD metric of an activity.

$$SD_i = \sum_{ft \in FT} F_{ft,i} * t_{ft,i} * IF_{ft,i} \quad (2)$$

For the case study, we assume a company that internally develops an automation software and that employs development teams divided into two areas: (i) Web interface and Web services development and (ii) persistency layer and database modeling. The system developed by these two teams has a Web interface written in Flex, Web services written in PHP running on Apache Web server, and information persisted over a MySQL database. Recently, the company has started developing a new version of this software. Therefore, both teams had their workstations updated using two RFCs, as shown in Figures 2 (a) and (b). The former sets up a Web development environment with Apache, PHP, and Flex Builder, while the latter, in addition to the Web server, required for testing purposes, also installs MySQL Server and a Workbench for SQL development. We assume that both RFCs have been executed to deploy these changes over 24 workstations of two development labs (12 successful executions each RFC).

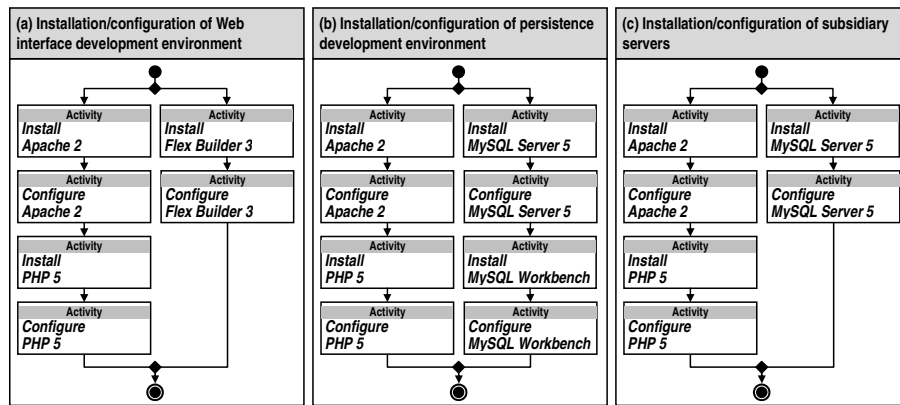


Fig. 2. Change plans of installation/configuration of environments

Once the new version of the automation system is ready to be deployed, the IT change management team has to design a new RFC to prepare the 20 servers, on each subsidiary, to receive this new software. The RFC designed for such change, detailed in Figure 2 (c), is supposed to be deployed in all subsidiaries in two phases (being 10 subsidiaries per phase). This RFC describes that Apache, PHP, and MySQL must be installed on each subsidiary's server. The configuration activities for the three software involved are manual, hence they must have humans associated. In this example, we define two human roles: the Senior operator, who performs MySQL and Apache configuration, and the Junior operator, who is in charge of configuring PHP. Although such RFC has never been executed (therefore it has no execution records for analysis) some of its activities have been performed a number of times in similar RFCs. Intuitively, one may realize that RFC (c) looks more like (b) than it does to (a), since RFCs (c) and (b) have 6 activities in common, while (c) and (a) have only 4. This similarity is captured by the *Risk Affinity (RA)* calculation (considering software, computers, and humans). For example, activity *Configure PHP* from RFC (c) has a RA of 0.43 comparing to

Configure PHP from RFC (b) (in regards to Activity Failures), while the RA factor is 0.33 comparing to the same activity in RFCs (c) and (a).

The Risk Report automatically generated for RFC (c) is illustrated in Table 2 (a). In this report, one may notice that the riskier activities are those performed by humans. Considering this report, activity *Configure PHP*, which is executed by the Junior operator, requires special attention. Another fact is that all MR values are between 4 and 6; this happens because all subsidiaries' servers have medium impact. Supposing that a *Change Authority* has analyzed this risk report and decided to deploy the RFC as it is. In the first deployment phase, 10 of the subsidiaries' servers have been successfully installed. By the end of this process, the total SD caused by the change deployment reaches a value of 6.68. This value is mostly influenced by activity *Configure PHP*, which has the worst MR. This activity is specially harmful because it is executed in a later moment on the workflow, hence its failure causes other activities to rollback. Aiming at reducing SD for the second phase, the *Operator* may suggest modifications in the original CP based on the results generated by the automated risk assessment. For instance, a more experienced human could be reallocated to the riskier activity. Therefore, for the second phase, the RFC was adapted allocating the Senior operator to configure PHP and the Junior operator to configure Apache. Table 2 (b) shows the risk report of the RFC with humans reallocated. In this report, one may notice the reduction of MR in the activity *Configure PHP*, whereas the MR of *Configure Apache* increases. After the RFC is adjusted, the second phase is deployed, reaching a total SD factor of 4.11. This represents a decrease of 38.47% in the total SD when comparing phases 1 and 2, indicating that the modification of the CP based on risk assessment reports has effectively decreased the risks associated to the requested change.

Table 2. Risk Reports before and after the modification of the Change Plan

(a) Results before 1st phase		(b) Results before 2nd phase	
Activity	Mean Risk	Activity	Mean Risk
Configure PHP	4.86	Configure Apache	4.86
Configure Apache	5.29	Configure PHP	5.29
Configure MySQL	5.29	Configure MySQL	5.29
Install Apache	5.45	Install Apache	5.45
Install PHP	5.45	Install PHP	5.45
Install MySQL	6.00	Install MySQL	6.00

6 Conclusion and Future Work

In this work, we discussed the organization's need for rational IT management. Since changes are imminent in such a dynamic environment, failures during this process may have direct effect on business continuity. Therefore, risks associated to changes should be investigated and mitigated. However, risk assessment has been usually left under the responsibility of humans operators, which may lead to inaccurate basis for decision

making. Thus, in this paper, we proposed a solution for automating the risk assessment in IT change management, aiming to aid administrators to design better changes improving quality of change management and managed services.

The results obtained, although not exhaustive, have shown that the automated risk assessment was able to combine several probabilities of failures from similar RFCs into a single probability weighted by a *Risk Affinity* factor. Moreover, the impact of affected CIs was considered along with probability of failures to classify activities of a CP according to a risk scale. The risk reports have shown to be useful to identify threats in a CP enabling proactive treatment of risks. Furthermore, a metric of Service Disruption was employed to compare the different CPs which revealed distinct risks reports. The mitigation of risks has caused an improvement in the SD factor, which indicates that risk reports reflect real threats to supported services.

In future work, we intend to investigate how to take advantage of other probability combination strategies, such as Bayesian Networks, as proposed by Hearty and Fenton. By employing such a technique an administrator could inject other factors into probabilities, such as uncertainty for RFC having very low historical information available. In addition, the case study presented in this paper has shown that human allocation to manual activities may definitely affect risks associated with changes. This leads to another question: What are the tradeoffs between different human allocations, in regard to costs, deployment time, and risks?

References

1. Office of Government Commerce (OGC): ITIL - Information Technology Infrastructure Library, 2008. <http://www.itiil-officialsite.com/>
2. Office of Government Commerce (OGC): ITIL - Information Technology Infrastructure Library: Service Transition Version 3.0, 2007.
3. Cordeiro, W. L. C.; Machado, G. S.; Andreis, F. G.; Santos, A. D.; Both, C. B.; Gaspary, L. P.; Granville, L. Z.; Bartolini, C.; Trastour, D.: ChangeLedge: Change Design and Planning in Networked Systems based on Reuse of Knowledge and Automation. *Computer Networks* (2009), doi: 10.1016/j.comnet.2009.07.001.
4. Machado, G. S.; Cordeiro, W. L. C.; Daitx, F. F.; Both, C. B.; Gaspary, L. P.; Granville, L. Z.; Sahai, A.; Bartolini, C.; Trastour, D.; Saikoski, K.: Enabling Rollback Support in IT Change Management Systems. In: 11th IEEE/IFIP Network Operations and Management Symposium (NOMS), Salvador, Brazil, pp. 347–354, 2008.
5. Sauv e, J. P.; Santos, R. A.; Almeida, R. R.; Moura, J. A. B.: On the Risk Exposure and Priority Determination of Changes in IT Service Management. In: 18th IFIP/IEEE Distributed Systems: Operations and Management (DSOM), San Jose, USA, pp. 147–158, 2007.
6. Froot, K. A.; Scharfstein, D. S.; Stein, J. C.: Risk management: Coordinating corporate investment and financing policies. *Journal of Finance*, pp. 1629–1658, 1993. American Finance Association.
7. Danaei, G.; Hoorn, S. V.; Lopez, A. D.; Murray, C. J. L.; Ezzati, M.: Causes of cancer in the world: comparative risk assessment of nine behavioural and environmental risk factors. *The Lancet*, vol. 366, no.9499, pp. 1784–1793, 2005. Elsevier.
8. Kl uppelberg, C.; Kostadinova, R.: Integrated insurance risk models with exponential Levy investment *Insurance Mathematics and Economics*, vol.42, no.2, pp.560–577, 2008. Elsevier
9. Institute of Risk Management (IRM): A Risk Management Standard, United Kingdom, 2002.

10. Fewster, R.; Mendes, E.: Measurement, prediction and risk analysis for Web applications. In: 7th IEEE International Software Metrics Symposium, pp.338–348, 2001.
11. Hearty, P.; Fenton, N.; Marquez, D.; Neil, M.: Predicting Project Velocity in XP Using a Learning Dynamic Bayesian Network Model. IEEE Transactions on Software Engineering, vol.35, no.1, pp.124–137, 2009.
12. Fenton, N. E.; Neil, M.: A critique of software defect prediction models. IEEE Transactions on Software Engineering, vol.25, no.5, pp.675-689, 1999.
13. Marques, M.; Neves-Silva, R.: Risk Assessment to Support Decision on Complex Manufacturing and Assembly Lines. In: 5th IEEE International Conference on Industrial Informatics, pp.1209-1214, 2007.
14. Oppenheimer, D.; Ganapathi, A.; Patterson, D. A.: Why do Internet services fail, and what can be done about it? In: 4th USENIX Symposium on Internet Technologies and Systems (USITS), Seattle, USA, 2003.
15. Wickboldt, J. A.; Machado, G. S.; Cordeiro, W. L. C.; Lunardi, R. C.; Santos, A. D.; Andreis, F. G.; Both, C. B.; Granville, L. Z.; Gaspary, L. P.; Bartolini, C.; Trastour, D.: A Solution to Support Risk Analysis on IT Change Management. In: 11th IFIP/IEEE International Symposium on Integrated Network Management (IM), New York, NY, 2009 (*to appear*).
16. Distributed Management Task Force (DMTF): CIM - Common Information Model, 2009. <http://www.dmtf.org/standards/cim>.
17. Workflow Management Coalition (WfMC): Workflow Process Definition Interface - XML Process Definition Language, 2009. <http://www.wfmc.org/xpdl.html>.
18. Active Endpoints: ActiveBPEL Open Source Engine, 2008. <http://www.activebpel.org>.

APPENDIX D PUBLISHED PAPER - NOMS 2010

In this appendix the paper entitled “Computer-Generated Comprehensive Risk Assessment for IT Project Management” is presented. This was the fourth deliverable of this research and the first work focused in the context of IT Project Management. The solution previously proposed to assess risks of changes was adapted to this new context and have shown to be generic enough to be used as a framework. In this paper, a model to represent management information of project’s life cycle was introduced. Also, a new strategy to summarize risk information and composition of more comprehensive and interactive Risk Reports have been proposed.

- **Title:**
Computer-Generated Comprehensive Risk Assessment for IT Project Management
- **Conference:**
12th IEEE/IFIP Network Operations and Management Symposium (NOMS 2010)
- **URL:**
<http://www.ieee-noms.org/2010>
- **Date:**
19-23 April 2010
- **Venue:**
Osaka International Convention Center, Osaka, Japan
- **Digital Object Identifier (DOI):**
<http://dx.doi.org/10.1109/NOMS.2010.5488498>

Computer-Generated Comprehensive Risk Assessment for IT Project Management

Juliano Araujo Wickboldt, Luís Armando Bianchin,
 Roben Castagna Lunardi, Fabrício Girardi Andreis,
 Ricardo Luis dos Santos, Bruno Lopes Dalmazo,
 Weverton Luis da Costa Cordeiro, Abraham
 Lincoln Rabelo de Sousa, Lisandro Zambenedetti
 Granville, Luciano Paschoal Gaspar
 Institute of Informatics
 Federal University of Rio Grande do Sul, Brazil
 Email: {jwickboldt, labinchin, rclunardi, fgandreis,
 rlsantos, bldalmazo, wlccordeiro, rabelo,
 granville, paschoal}@inf.ufrgs.br

Claudio Bartolini
 HP Laboratories Palo Alto, USA
 Email: claudio.bartolini@hp.com

Abstract—Information Technology (IT) products and services provided by modern organizations are designed in projects that often involve large amount of resources (*e.g.*, humans, hardware, and software). It is essential that organizations enforce rational practices for project management, in order to successfully conclude projects and avoid waste of substantial resources. In this context, Risk Management is fundamental to guarantee the accomplishment of project's objectives by dealing with adverse and favorable events. Although important, risk assessment in IT projects is usually performed by stakeholders in interviews and brainstorming which may be a very time/resource-consuming task. Therefore, in this paper, we introduce a solution to automate the risk assessment process, based on the history of previously conducted projects. Furthermore, comprehensive and interactive risk reports are proposed in order to ease the analysis of automatically generated reports. The results show that our solution is not only useful to speed the risk assessment process, but also to assist the decision making of project managers by organizing risk information according to the project structure.

I. INTRODUCTION

Many organizations provide services to their customers by means of information technology (IT) infrastructures. The design and deployment of these services are usually made in projects that involve large amount of resources (*e.g.*, hardware, software, and people). Due to the complexity of large IT projects, rational practices for project management should be enforced to ensure that each project will fulfill its requirements avoiding waste of resources. For that end, some libraries and standards on best practices for IT projects have been published, such as the IT Infrastructure Library (ITIL) – proposed by the Office of Government Commerce (OGC) [1] – and the Guide to the Project Management Body of Knowledge (PMBOK) – introduced by the Project Management Institute (PMI) [2].

Every project has risks associated to it, such as exceeding the established schedule or overcoming the initially planned budget. Risks in projects can be faced as events that, if happen, may have positive or negative effects in at least one

project objective. These objectives might change according to project's needs. In this work, we consider a common set of objectives of a project, which are: cost, time, scope, and quality. In order to tackle risk in IT projects, one of the nine so-called knowledge areas from PMBOK is focused specifically on Project Risk Management, whose objectives are: (i) to increase the probability and impact of positive events, and (ii) to decrease the probability and impact of events adverse to the project.

Commonly, in IT projects, the risk assessment process is performed by humans that gather risk information from the stakeholders through meetings, interviews, or brainstorming. Depending on the size of the project, the amount of variables that should be considered for proper risk assessment might turn this process into a very time/resource-consuming task. In addition, the final results may not be as accurate as required, leading project managers to take inappropriate actions to mitigate risks. In this context, risk management must be able to cope with large amount of risk-related variables, and still be intuitive and meaningful for these managers to analyze.

Imprecise and expensive risk assessment based in human knowledge is indeed an issue also in other areas, such as IT change management discussed in ITIL Service Transition book [3]. Recently, some researches have already tackled this problem by automating the risk assessment process [4] [5]. The already proposed methods for automation could be naturally adapted to Project Risk Management. With such an automation, a system would be able to collect information about previous deployed projects and estimate risks for the project under analysis. This would significantly reduce the time spent in gathering information and, moreover, would increase the reliability and accuracy of the results. Still, depending on the project's size, the resulting amount of information could prevent humans from understanding the risks automatically calculated. We believe, however, that a

complementary solution for summarizing risk information, considering different levels of details, can help project managers to better understand risks in a more interactive fashion.

In this paper we introduce a novel risk information summarization strategy aiming at creating a comprehensive representation of risks in IT projects. Our approach encompasses hierarchically organizing risk information upon different levels of detail, presenting more valuable reports according to the interests of the observer. We assume that IT projects have their risks observed in a hierarchy of six levels: activities, plans, cycles, interactions, releases, and project. In this hierarchy, summarization takes place to combine risk information from one lower level and present it in another more abstract level. Our approach enables project managers to observe risks firstly at a higher level, to then zoom in when they become interested in specific parts of the project.

The remaining of this paper is organized as follows. Section II describes some research efforts that have been carried out to manage risks in IT project management. In Section III we introduce the method for automated risk assessment in IT project management, while in Section IV the strategies proposed for comprehensive risk assessment are detailed. A case study is presented in Section V along with discussions about the results obtained. Finally, the paper is closed with conclusions and future work in Section VI.

II. RELATED WORK

Risk management is subject of research in several different fields. One general purpose standard for organizational risk management is published by the Institute of Risk Management (IRM) [6]. According to this standard, the risk management discipline defines the process whereby organizations methodologically address the risks associated with their activities, aiming to achieve sustained benefits. ITIL encourages the use of a framework for risk management, also proposed by OGC, called Management of Risk (M_o_R) [7]. This framework defines systematic repeatable processes for risk identification and assessment, in a first moment, and subsequent planning and implementation of responses for those identified risks. More focused in the context of IT projects, PMBOK with its Project Risk Management knowledge area, divides risk management into six processes, further detailed in Section III: Risk Management Planning, Risk Identification, Quantitative Risk Analysis, Qualitative Risk Analysis, Risk Response Planning, and Risk Monitoring and Control [2].

Despite the current risk support proposed in the aforementioned frameworks and standards, the adoption of formal procedures in actual projects still demands too much effort, experience, and ability of managers and stakeholders to produce useful results. Kutsch and Hall [8] have investigated the reasons why IT project managers decide whether or not certain identified risks should be considered relevant against project objectives. By interviewing managers from different IT projects, the authors perceived that behavioral factors influence manager's decisions; therefore the success of risk management is conditioned to their experience. Indeed, when

the project manager does not have sufficient experience to effectively prioritize risks, project risk management seems to have little impact on project outcomes, being sometimes even counterproductive. Wyk *et al.* [9] have evaluated the risk management methods of a large electricity supplier in South Africa. Although the analyzed company employs best practices for risk management, risk identification, analysis, mitigation, monitoring, and reporting are performed employing no automated tool. As a consequence, the company ends up involving an excessive number of stakeholders in risk management process. In addition, there is lack of common practices across various divisions, which turns the reuse of knowledge internally to the company more complex.

In order to aid humans in risk management, the automation of certain steps of this process – such as data gathering for risk assessment – could, for example, potentially reduce the time and cost, while increasing the reliability of results. Probabilistic models are commonly employed in risk management in order to predict undesirable events and also for estimation of metrics, such as cost and time of projects. Fewster and Mendes [10] have proposed a framework, based on a Generalized Linear Model that is capable of estimating probabilities for some project's negative events (*e.g.*, overcoming project budget or violating deployment deadlines). Bayesian Networks (BNs) have been used in many investigations for similar purposes. Hearty *et al.* [11] have designed a model, based on BNs, for effort prediction and risk assessment in Extreme Programming (XP) software development projects. Also, Fenton and Neil [12] have applied BNs to predict software defects, while Luu *et al.* [13] employ it to estimate the likelihood of time-overrun in construction projects. These works have contributed to the automation of risk assessment. However, they concentrate on the prediction of adverse events; the impacts that such events might have over the project objectives are not considered.

Solutions for automation and decision support for risk assessment in IT change management systems have already been proposed in some previous investigations. Based on estimates of time, Sauv e *et al.* [14] have proposed a risk analysis method to determine priorities for scheduling potentially concurrent Request for Changes (RFCs). Also dealing with scheduling of RFCs, Setzer *et al.* [15] have modeled the resources of an IT infrastructure as a network of interconnected services; then, risk is quantified by analyzing the impact of changes over affected services.

One particular solution for automation of risk assessment in the planning phase of IT change management, proposed in a previous work of our research group [4] [5], has been used as basis for the contributions in this paper. In that solution, probabilities of failure were estimated analyzing historical execution traces of *Change Plans* (workflows of activities to perform a change over an IT infrastructure [16]). Besides, the impact of changes was automatically calculated based on the definitions of relevancies of affected elements and their dependencies/relationships. That solution has shown promising results to help on decision making and risk mitigation of changes; hence, it could certainly be adapted to the

context of IT projects. However, the generated risk reports show information about risks of every activity of a *Change Plan*. Considering that projects might have many activities in their several phases and that risks may affect specific objects of a project, the amount of data in risk reports tend to be too extensive, preventing proper human comprehension. Therefore, in the following sections, we will introduce a novel approach for risk assessment in the context of IT projects.

III. AUTOMATED IT PROJECT RISK ASSESSMENT

In this section we present, in a first moment, the recommendations of the PMBOK for Project Risk Management, emphasizing in which moment automation is needed. Afterwards, a model conceived to represent project management information is detailed, highlighting important classes required to store events that constitute risks. Finally, the solution, adapted from the context of IT change management, to estimate risks in IT projects is presented.

A. Project Risk Management Process

Project Risk Management is a knowledge area that comprises planning, identification, analysis, responses, and monitoring of risks that may affect project objectives. PMBOK divides this process into six processes, as shown in Figure 1 (darker boxes).

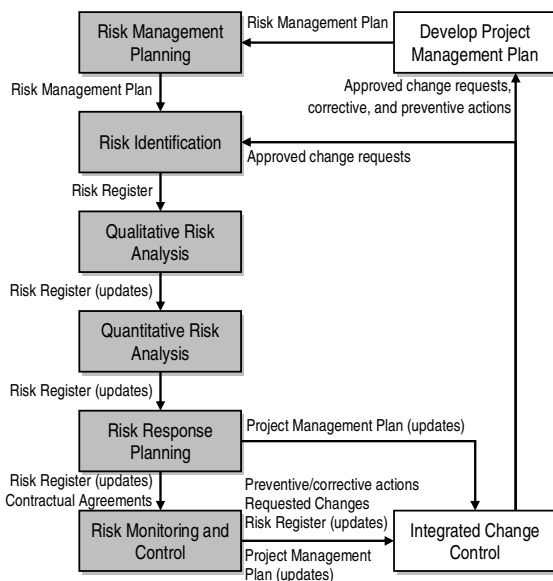


Fig. 1. Project Risk Management processes according to PMBOK [2]

Risk Management Planning is the process in which project managers decide how to approach and conduct risk management during the whole project. This process leads to the specification of a *Risk Management Plan*, which defines of methodology, roles and responsibilities, budgeting, timing, risk categories, and probability/impact matrix for the conduction of risk management in subsequent processes.

Risk Identification is an iterative process that determines the risks that might affect the project and records their characteristics. Among several techniques, risk identification may be

carried out by brainstorming, interviewing, or creating checklists based on historical information that has been accumulated from previous similar projects. The output of this process is the initial entries of the *Risk Register*. The *Risk Register* is a list of identified risks, potential responses, root causes, and risk categories, which is updated during subsequent risk management processes.

Qualitative Risk Analysis is the process of assigning priorities for treatment of identified risks using their probability of occurrence and corresponding impact on project objectives (such as, cost, time, scope, and quality). Probability and impact are assessed, for each identified risk, in interviews or meetings with project team members or other people from outside the project with extensive knowledge on risk assessment. PMBOK itself recognizes that gathering high-quality information for risk assessment is difficult, and usually consumes time and resource beyond the originally planned.

Quantitative Risk Analysis is the process in which quantitative evaluations are performed for some of the risks prioritized in the previous process. Numerical ratings are estimated for the effects of high priority risks aiming to guide the efforts and intensity of response planning.

Risk Response Planning is the process in which project managers, based on qualitative and quantitative analysis, define options and actions to reduce threats (adverse risks) and enhance opportunities (favorable risks). Response actions should be appropriate to each risk (e.g., in terms of cost). As output of this process, risk-related contractual agreements with other parties (e.g., insurance contracts), as well as recommended changes to the *Project Management Plan*, may be established.

Risk Monitoring and Control is a continuous process that must be executed during the life cycle of the project in order to keep tracking of the identified risks and detect other newly arising. Occasionally, *Preventive Actions* (contingency plans) or *Corrective Actions* (workarounds) planned for risk response result in *Change Requests* to be handled by the *Integrated Change Control* (process from outside the Project Risk Management). All approved changes, workarounds, and contingency plans should be documented and attached, in the *Develop Project Management Plan*, which, in turn, should be periodically re-evaluated in terms of risks.

Some problems can be easily identified in PMBOK processes, especially in risk identification and analysis. Firstly, risks are assessed mainly based on human knowledge; hence, the quality of risk management is a function of the experience of stakeholders. The Qualitative Risk Analysis, in addition to consuming too much human resources, may propagate errors to the next processes. Since Quantitative Risk Analysis is optional for low priority risks, some risks wrongly considered as irrelevant may cause damage to project objectives beyond the expectations.

B. IT Project Life Cycle Information Model

In order to enable proper management and reuse of knowledge of IT projects, including management of risk and other

aspects, it is important for organizations to document all activities of developed projects employing a single consistent information model. As far as the authors of this paper are aware of, there is no widely accepted model for representing data of IT projects available in the literature. Therefore, in this work, we propose such a model – depicted in Figure 2 – inspired in a Business Technology Optimization (BTO) software from Hewlett-Packard (HP) called HP Quality Center[©].

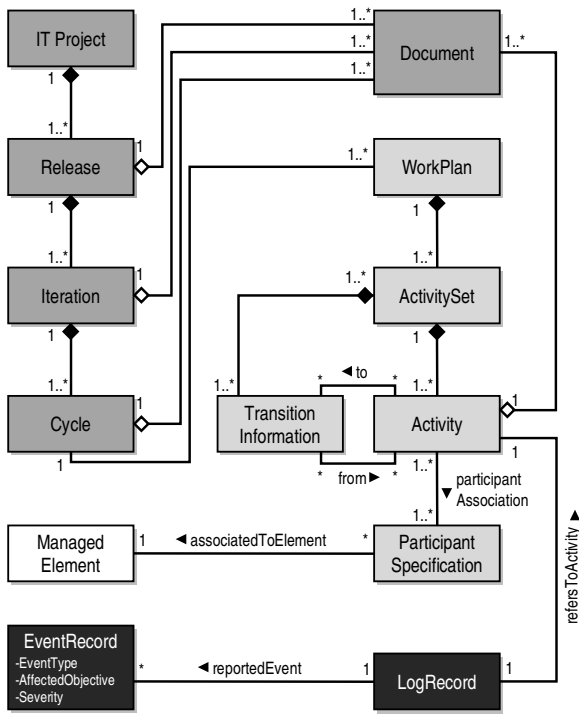


Fig. 2. IT Project Life Cycle Information Model

Every *IT Project* may be delivered to the final customer through one or more *Releases*. Each *Release* is a partial version of the product or service being designed/developed in the project. It contains a set of functionalities fully developed and tested that may be validated or sometimes put into production by the customer. These functionalities are planned and implemented in one or more *Iterations*. The *Cycles* associated to each *Iteration* will often vary according to the methodology adopted. For example, the *Cycles* of an *Iteration* could be Analysis, Project, Development, and Testing.

In order to organize the *Activities* that have to be performed in each *Cycle*, one or more *WorkPlans* have to be defined. Indeed, *WorkPlans* are workflows of *Activities* following the workflow process definition, proposed by the Workflow Management Coalition (WfMC) [17]. Instances of *TransitionInformation* are used in order to define the logical flow of *Activities* to be executed. These transitions may also represent branches (conditionals or parallelism) or joins. An *Activity* consumes a certain amount of resources and takes some time to be executed. The *ParticipantSpecification* class associates *Activities* to the allocated resources (e.g., humans or computers).

The participants of activities refer to the *ManagedElement* class, linking this model with the Common Information Model (CIM) (often used to represent IT infrastructures) from the Distributed Management Task Force (DMTF) [18].

Two classes (bottom of Figure 2) are particularly important for automation of risk assessment: *LogRecord* and *EventRecord*. Every *Activity* performed in a *WorkPlan* must have an associated *LogRecord* to it in order to indicate the details of its execution. The execution of an *Activity* may trigger events (adverse or favorable). These events are documented in instances of the class *EventRecord*, along with information about *EventType* (e.g., adverse or favorable), *AffectedObjective* (e.g., cost, time, scope, or quality), and *Severity* (e.g., amount of hours delayed in activity).

C. Automated Risk Assessment Solution

Risk assessment in IT projects is often performed by combining two factors: (i) the probability of occurrence of events (positive or negative) and (ii) the impact that these events might have on the project objectives. A computational system could facilitate this process – usually performed by humans – by calculating both factors using information from a database of previously executed projects. Also, it is important to organize this information in a comprehensive report assisting the managers on decision making concerning to risk mitigation. In a previous work we have proposed a solution for the automation of risk assessment in IT change management area [4] [5]. Hence, in this subsection we briefly explain this solution, emphasizing how it is adapted to the context of IT projects.

Firstly, it is important to keep in mind that risks in IT projects are addressed according to their impacts facing different objectives of the project. Thus, probability and impact have to be assessed separately for as many objectives as managers want to consider. Since the negative side of risks requires more attention of project managers, only adverse events will be considered in the following explanations. However, it is relatively easy to include positive events simply repeating the process.

Intuitively, the probability of occurrence of negative events is calculated per activity in each work plan of a project in four steps:

First step - Search for executions of similar activities in the database of previously executed work plans: Since it is a good practice to reuse knowledge in IT projects, activities or even complete work plans might be repeated (sometimes with small modifications) across several projects. Then, in this first step, our solution selects activities that are considered similar to the activity being analyzed. This similarity is calculated by matching the type of activity (e.g., planning, development, analysis, test, or deployment) and its associated participants (e.g., people, computers, or technologies).

Second step - Calculate Risk Affinity among activities: Risk Affinity (RA) is a concept introduced in a previous work [5]. Basically, it represents an affinity index, with respect to risks, between two activities from different workflows (the one

contained in the work plan being analyzed and another selected from a project database). It is computed for all activities in both work plans considering their types, participants, and eliminating from the work plans activities that are executed after the pair which the affinity regards to. This elimination has the purpose of removing from the affinity calculation activities that do not affect the activities being analyzed.

Third step - Count number of executions and events affecting objectives caused by activities: In this step, our solution investigates, in the *LogRecords*, the number of executions of each activity selected in the first step and number of events affecting each objective caused by these activities.

Fourth step - Calculate probabilities of occurrence of adverse events for selected activities weighted by their RA values: Finally, the probability of occurrence of adverse events is calculated by dividing two values: (i) the sum of occurrences of these events for every selected activity (dividend) and (ii) the sum of executions of the same set of activities (divisor). These values are weighted by the respective RA calculated for each activity.

Impact estimation is calculated following the same steps as proposed for probabilities, except that, instead of counting the number of executions and the occurrence of adverse events (third step), the severity of the event facing the originally planned for the activity is considered. For example, assuming a given activity that was planned to take 8 hours of work to conclude. When it is executed an event is reported informing that it took 4 hours more than it should. The impact that this event represents for the activity's time objective is the hours that have delayed divided by the hours it was planned to last (in this example impact on time objective is 0.5). The impacts are also weighted using the RA in order to make impact analysis tend to approximate its value to the activities that were executed in more "similar" environments, just as happens to probabilities in step four.

So far, our solution is able to calculate probabilities and impacts of adverse events for every activity in the work plans of a project considering their effects over different objectives. Nevertheless, it is important to display these results in such a way that project managers can actually analyze them. IRM [6] recommends mapping probability and impact values to the following scales: (i) high (more than 25%), medium (between 25% and 2%), and low (less than 2%), for probabilities, and (ii) high (significant), medium (moderate), and low (insignificant), for impact. After that, risks may be classified in one of the nine categories from the matrix presented in Table I. Clearly, the ranges for mapping of probabilities and impacts as low, medium, and high may be adapted to fit the requirements of each project. Also, the risks classification matrix might be extended from 3x3 to a 5x5, for example, to provide more punctual results. These definitions should be made on the Risk Management Plan, before the start of risk assessment.

It is important to notice that, at the end of this classification, each activity of all work plans of the analyzed project will have assigned a risk category facing each different project objective. Depending on the size of the project, it may include a large

TABLE I
RISKS CLASSIFICATION MATRIX

		Probability		
Impact	High Impact Low Probability Category 3	High Impact Medium Probability Category 2	High Impact High Probability Category 1	
	Medium Impact Low Probability Category 6	Medium Impact Medium Probability Category 5	Medium Impact High Probability Category 4	
	Low Impact Low Probability Category 9	Low Impact Medium Probability Category 8	Low Impact High Probability Category 7	

amount of work plans and, consequently, many activities. To present a risk report to a project manager with hundreds or thousands of lines, taking riskier activities to the top, but possibly losing their context in the project, might not be intuitive enough to help on risk mitigation. The approach then, which is further explained in the following section, is to improve these reports by grouping the risks of activities into higher levels of abstractions (work plans, cycles, iterations, or releases).

IV. COMPREHENSIVE RISK REPRESENTATION

In previous works, risk reports for IT change management have been typically presented to operators in tabular format, displaying all activities of a certain change ordered by their risk factor. Those reports have shown to be very interesting in pointing risks of failure in activities of change, helping operators to prioritize efforts for risk mitigation by adjusting changes before deployment. Considering that risks in IT projects are analyzed separately for different objectives, a detailed tabular report for any random work plan with five activities could be as shown in Table II.

TABLE II
TABULAR RISK REPORT

Activity		Cost	Time	Scope	Quality
A1	Probability	25.0%	25.0%	1.0%	5.0%
	Impact	0.50	0.30	0.00	0.20
	Category	1	1	9	5
A5	Probability	0.5%	50.0%	1.0%	0.0%
	Impact	0.01	0.30	0.05	0.00
	Category	9	1	6	9
A3	Probability	12.0%	3.0%	6.0%	0.00
	Impact	0.10	0.30	0.05	0.00
	Category	5	2	5	9
A2	Probability	15.0%	8.0%	1.0%	2.0%
	Impact	0.10	0.30	0.05	0.01
	Category	5	2	6	9
A4	Probability	1.0%	0.0%	1.0%	0.0%
	Impact	0.10	0.00	0.05	0.00
	Category	6	9	6	9

This risk report is not only correct but also provides important information about the risks of all activities of the work plan for as many project objectives as needed. However, large-scale projects might include a huge amount of activities in its several work plans. Thus, for project managers to address the risks of those projects (*i.e.*, composing contingency plans or

workarounds), analyzing one activity at a time could demand too much time and consume excessive resources. Therefore, in this section we introduce a novel strategy to summarize risk reports in different levels of the project (*i.e.*, activities, work plans, phases, interactions, and project). Moreover, risk reports are displayed graphically in two different perspectives: (i) Project Hierarchy View, providing a general project risk overview, and (ii) Work Plan View, displaying more detailed information about risk in specific work plans. Graphical representation of risk reports are further clarified in Section V.

The basic approach of summarizing risk information is to combine a group of values from lower levels of project hierarchy, using a given function, into one single risk metric meaningful for evaluation at a higher level. Furthermore, it is important to keep information apart about the affected objectives in all levels of the project, in such a way that managers can analyze risks over each objective separately. We propose then to use a function to calculate an Average Risk of all risk categories for activities of a work plan, and display this information as the risk metric of the whole work plan. Another important fact is that the result of an average functions tends to smooth all portions into a mean value. For instance, considering that a work plan has four activities, being three of them classified in risk category 9 (lowest possible risk) and only one in category 1 (highest possible risk) for the cost objective. Thus, an *arithmetic mean* of these values would result in an Average Risk of 7, hiding from the report the damage that one of those activities (classified in category 1) could possibly cause to the project. Therefore, in this work, in order to calculate the Average Risk, we employ a *harmonic mean*, as shown in Equation 1.

$$AR = \frac{n}{\sum_{i=1}^n \frac{1}{a_i}} \quad (1)$$

In Equation 1, n represents the number of risk values being summarized (*e.g.*, number of activities in a work plan, or number of work plans in a cycle). This number is the dividend of the division by the sum of all reciprocals of risk values (*i.e.*, a_i is the i^{th} risk value being summarized). In this equation, we assume that risk categories will always be represented as values ranging from 1 to any greater positive value. Using the aforementioned example (three activities with risk category 9 and one with risk category 1), the resulting Average Risk (AR) would assign a value of 3 to the hypothetical work plan. The employment of this function works as a pessimistic approach to risk summarization, propagating very high risk activities, detected by the automated solution, up in the project hierarchy.

One final consideration about risk summarization is that Average Risk should always be calculated from risk categories of activities, avoiding the use of other averages computed in higher levels of the project. This is important to prevent the analysis from losing information about the cardinality of summarized sets (*e.g.*, number of activities in work plans, or number of iterations in a release). For example, considering a

given cycle with two work plans, one containing 20 activities and another with only 2. Once Average Risks are calculated for both work plans, these values will belong to the same range (*i.e.*, from 1 to 9 continuously), and no information is kept about work plans amount of activities. If an Average Risk for the cycle is calculated considering the computed average from its two work plans, some risky activities from the largest plan could be attenuated. To tackle this problem, there are two options: (i) to calculate the Average Risk of the cycle from all 22 activities from both work plans, or (ii) to use a *weighted harmonic mean* of the Average Risks from work plans, where the weights are their cardinals (respectively 20 and 2). Both options produce exactly the same results, although the second is better to avoid recalculation of average values up in the hierarchy of the project.

V. CASE STUDY

Aiming to prove concept and technical feasibility of the proposed solution, we have conducted a case study considering a hypothetical software development project. Also, a database was designed containing synthetic information about work plans from other projects, execution of activities, and documented adverse events. In this section, in a first moment, we briefly present the hypothetical project's structure. Afterwards, comprehensive risk reports automatically generated by the solution are shown under two different perspectives: Project Hierarchy View and Work Plan View.

A. Hypothetical Project Structure

The goal of the studied project is to develop a system for monitoring, supervision, incident reporting, and problem diagnosis on large-scale corporative networks. The purpose of this system is to provide a company with support for management of an IT infrastructure inventory, monitoring, and supervision of Configuration Items (CIs) (*e.g.*, routers, computers, software packages, and services), and also record incidents involving these CIs, assisting the problem diagnosis process. According to high level definitions of requirements for the project, a project manager split development efforts into four releases, as follows:

- **Release 1:** Monitoring and supervision basic features;
 - **Iteration 1:** Database modeling to allow composition of IT infrastructure inventory;
 - **Iteration 2:** Development of server-side core module application;
 - **Iteration 3:** Development of client-side core module application;
 - **Iteration 4:** Development of server-side graphical Web interface basic operations;
- **Release 2:** Monitoring and supervision advanced features;
 - **Iteration 1:** Development of server-side advanced reports composer;
 - **Iteration 2:** Development of server-side analytical multi-variable graphics module;
- **Release 3:** Monitoring and supervision integration;

- **Iteration 1:** Development of server-side SNMP support module;
- **Iteration 2:** Development of server-side Web Services support module;
- **Release 4:** Incident reporting and problem diagnosis;
 - **Iteration 1:** Database modeling for incident reporting;
 - **Iteration 2:** Development of incident reporting Web interface;
 - **Iteration 3:** Development of problem diagnosis tool.

In Release 1, basic functionalities of the system are implemented. In its first iteration, the database to allow representation of CIs from the IT infrastructure is modeled. The core of the system works as a client-server application, where the server requests/receives information about managed clients installed in CIs. The Web interface basic features are also delivered in first release, such as CRUD (Create, Request, Update, and Delete) operations over registered objects. Advanced features, such as reports composition (*e.g.*, availability, network load and latency, and alarms) and graphs for data visualization, are left to the second release. In the third release, modules for integration with SNMP and Web Services are included to enable management of devices that support those management interfaces. Finally, in the fourth release, incident reporting interface and a diagnosis tool are added in order to allow association of reported incidents and problems with corresponding defective CIs. Although not detailed above (due to space limitation), every iteration of the project is divided into four cycles: Analysis, Project, Development, and Testing.

B. Comprehensive Risk Reports

The project analyzed in this case study contains 141 activities disposed in 34 work plans. Since the automated risk assessment solution calculates four risk categories (one for each affected objective) for all activities of the project, a risk report as shown in Table II could not be practical to help on decision making for risk response planning. Instead, the new approach proposed in this paper generates more comprehensive reports under two perspectives: (i) Project Hierarchy View (Figure 3-a), which gives an interactive overview of risks using the project hierarchical structure, and (ii) Work Plan View (Figure 3-b), useful to investigate particularly risky work plans aiming to understand the sources of risk.

As shown in Figure 3-a, a project manager can interactively choose which part of the project he/she wants to inspect with more details. For example, by expanding (+) an iteration the risks calculated for all of its cycles are displayed. Analyzing this hierarchical report one could notice that, among all releases, the first one holds most of the risks from the hypothetical project analyzed in this case study. Inspecting Release 1, a project manager may figure out that Interaction 4 requires special attention due to its risk factors in all objectives. Observing the cycles of Interaction 4, it is possible to notice that risks of different objectives are mostly distributed among Cycles 1, 3, and 4. Cost and Scope risks are negatively influenced by Cycle 3, Time risks are shared between Cycles

3 and 4, and Quality risks are more evidenced in Cycle 1. A report with these characteristics indicates that, in past similar projects automatically analyzed by our solution, events were reported evidencing poor quality in activities of analysis. That might have caused other adverse events to happen affecting cost and time of later development and testing cycles.

Whenever a project manager needs to inspect with more details some of the work plans from the project, the Work Plan View may be used. In Figure 3-b, one work plan from Cycle 3 (Development) of the fourth iteration from the hypothetical project is shown. The detailed work plan defines six development activities necessary to implement basic functionalities of the Web interface of earlier described system. Initially, administration of credentials (*e.g.*, login forms, users names, passwords, and access rights) and system menu structure (*e.g.*, sections and subsections) are developed. In a subsequent moment, two parallel branches are started. Both branches develop DAOs (Data Access Objects), for persistence of objects in a relational database, and development of Web forms for CRUD operations of CIs and their categories. The risk classifications automatically assigned to the activities are displayed next to each of them in the work plan structure. This visualization helps the identification of problematic activities that might compromise the success of the work plan.

One important fact is that, despite the attenuation caused by the summarization of risk classifications, automatically calculated risks of activities still reflect very well in upper levels of the project. This is clearly visible in the hierarchical view (Figure 3-a) used as example in this case study. Some activities from different cycles in Interaction 4 had high risk rating (low categories) and this reflected in high risks for the whole Release 1. Based in these reports a project manager could prioritize risks and establish directions for risk response. For example, one strategy could be addressing risks of a project by iteration. Then, a threshold may be specified defining that preventive actions (contingency plans) are required for iterations with risk factors below 5, and corrective actions (workarounds) for iterations that exceed this value.

VI. CONCLUSION & FUTURE WORK

In this work, the need for rational IT project management have been discussed, emphasizing how it is supposed to help organizations to conduct successful projects saving substantial resources. In this context, risk management plays a key role in the successful accomplishment of project objectives. Nevertheless, we have argued that current risk assessment practices, in addition to consuming too much human resources/time of the project, still do not present clear results in practice. Thus, in this paper, we have introduced a solution to automate the risk assessment process in IT projects by investigating the database of events documented in past projects. Also, more comprehensive and interactive reports have been designed to aid the decision making on risk response planning.

The automated solution proposed has shown to be useful to speed the process of risk assessment, since it is based on information retrieved from a database of previously executed

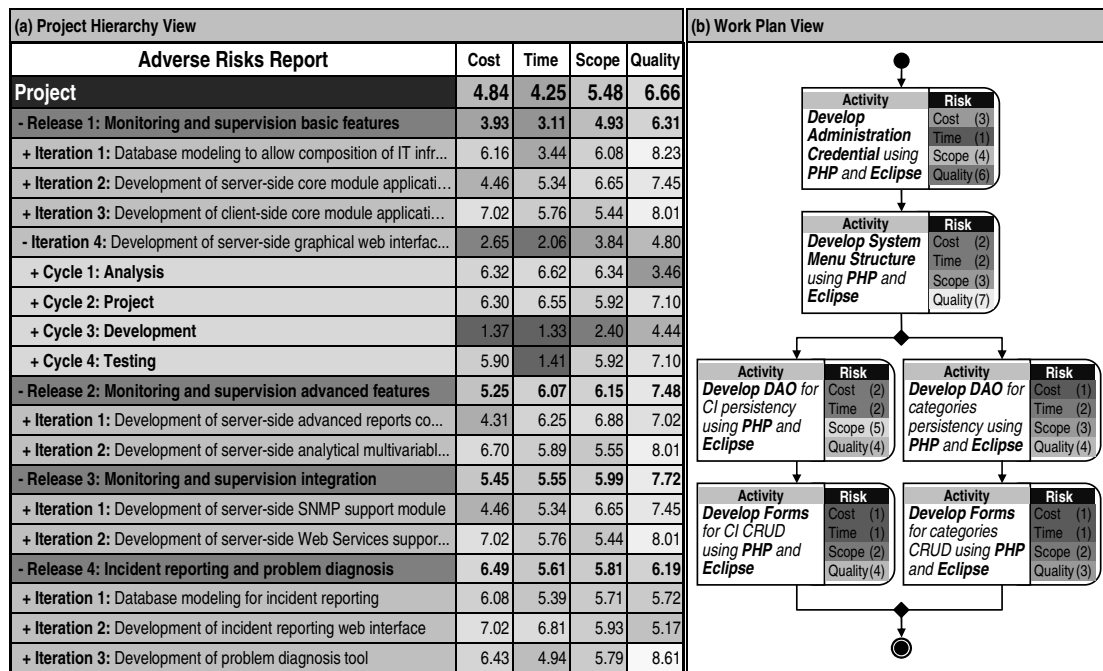


Fig. 3. Automatically Generated Risk Reports

projects and does not require any human intervention. Remarkably, the reports generated organize information according to the project hierarchical structure, facilitating the identification of risks in each phase of the project. The proposed risk information summarization strategy achieved its objective, which was to combine risk information from activities of work plans displaying this information into higher levels of the project without hiding risks from lower level.

In future work, we intent to use data from real life projects to better evaluate the applicability of the proposed solution. Also, it would be interesting to conduct a survey and receive feedback from experienced project managers and stakeholders to evaluate the usability of the proposed risk reports.

ACKNOWLEDGMENT

This result was achieved in cooperation with Hewlett-Packard Brasil Ltda. using incentives of Brazilian Informatics Law (Law n^o 8.2.48 of 1991).

REFERENCES

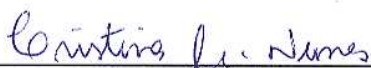
- [1] *ITIL – Information Technology Infrastructure Library: Service Design Version 3.0*. London, UK: Office of Government Commerce, 2007.
- [2] *A Guide to the Project Management Body of Knowledge: PMBOK Guide – Third Edition*. Pennsylvania, USA: PMI - Project Management Institute, 2004.
- [3] *ITIL – Information Technology Infrastructure Library: Service Transition Version 3.0*. London, UK: Office of Government Commerce, 2007.
- [4] J. A. Wickboldt, G. S. Machado, W. L. C. Cordeiro *et al.*, “A Solution to Support Risk Analysis on IT Change Management,” in *Mini-conference of 11th IFIP/IEEE International Symposium on Integrated Network Management (IM)*, New York, NY, 1-5 June 2009, pp. 445–452.
- [5] J. A. Wickboldt, L. A. Bianchin, R. C. Lunardi *et al.*, “Improving IT Change Management Processes with Automated Risk Assessment,” in *20th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management (DSOM), LNCS 5841*, Venice, Italy, 27-28 October 2009, pp. 71–84.
- [6] IRM, *A Risk Management Standard*. United Kingdom: The Institute of Risk Management, 2002.
- [7] *Management of risk: guidance for practitioners*. London, UK: Office of Government Commerce, 2007.
- [8] E. Kutsch and M. Hall, “Deliberate ignorance in project risk management,” *International Journal of Project Management*, 2009.
- [9] R. van Wyk, P. Bowen, and A. Akintoye, “Project risk management practice: The case of a South African utility company,” *International Journal of Project Management*, vol. 26, no. 2, pp. 149–163, 2008.
- [10] R. Fewster and E. Mendes, “Measurement, prediction and risk analysis for Web applications,” 2001, pp. 338–348.
- [11] P. Hearty, N. Fenton, D. Marquez, and M. Neil, “Predicting project velocity in xp using a learning dynamic bayesian network model,” *IEEE Transactions on Software Engineering*, vol. 35, no. 1, pp. 124–137, Jan-Feb. 2009.
- [12] N. Fenton and N. Ohlsson, “Quantitative analysis of faults and failures in a complex software system,” *IEEE Transactions on Software engineering*, vol. 26, no. 8, pp. 797–814, 2000.
- [13] V. Luu, S. Kim, N. Tuan, and S. Ogunlana, “Quantifying schedule risk in construction projects using Bayesian belief networks,” *International Journal of Project Management*, vol. 27, no. 1, pp. 39–50, 2009.
- [14] J. Sauv e, R. A. Santos, R. R. Almeida *et al.*, “On the Risk Exposure and Priority Determination of Changes in IT Service Management,” in *18th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management (DSOM)*, San Jose, CA, October 2007, pp. 147–158.
- [15] T. Setzer, K. Bhattacharya, and H. Ludwig, “Decision support for service transition management,” in *11th IEEE/IFIP Network Operations and Management Symposium (NOMS)*, April 2008, pp. 200–207.
- [16] W. L. C. Cordeiro, G. S. Machado, F. G. Andreis *et al.*, “ChangeLedge: Change Design and Planning in Networked Systems based on Reuse of Knowledge and Automation,” *Computer Networks*, 2009.
- [17] WfMC, “Workflow process definition interface - xml process definition language,” 2007, http://www.wfmc.org/standards/docs/TC-1025_10_xpd_102502.pdf.
- [18] DMTF, “Common Information Model,” 2008, <http://www.dmtf.org/standards/cim>.

“*A Framework for Risk Assessment Based on Analysis of Historical Information of Workflow Execution in IT Systems.*”

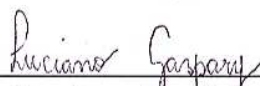
por

Juliano Araujo Wickboldt

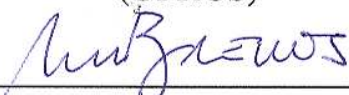
Dissertação Apresentada aos Senhores:



Profa. Dra. Cristina Moreira Nunes
(PUCRS)



Prof. Dr. Luciano Paschoal Gaspar
(UFRGS)



Prof. Dr. Antonio Marinho Pilla Barcellos
(UFRGS)

Vista e permitida a impressão.
Porto Alegre, 31/08/2010



Prof. Dr. Lisandro Zambenedetti Granville
Orientador


Prof. Alvaro Freitas Moreira
Coordenador do Programa de
Pós-Graduação em Computação - PPGC
Instituto de Informática - UFRGS