Computer Networks 55 (2011) 2954–2975

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

A framework for risk assessment based on analysis of historical information of workflow execution in IT systems

Juliano Araujo Wickboldt [a,*], Luís Armando Bianchin [a], Roben Castagna Lunardi [a], Lisandro Zambenedetti Granville [a,*], Luciano Paschoal Gaspary [a], Claudio Bartolini [b]

[a] Institute of Informatics, Federal University of Rio Grande do Sul, Av. Bento Goncalves, 9500, CEP 91.509-900, Porto Alegre, RS, Brazil
[b] Hewlett Packard Laboratories, Palo Alto, USA

ARTICLE INFO

Article history:
Received 23 December 2010
Received in revised form 17 May 2011
Accepted 29 May 2011
Available online 24 June 2011

Keywords:
Infrastructures and services management
Risk management
Change management
Project management
Networks operations and management

ABSTRACT

Services provided by modern organizations are usually designed, deployed, and supported by large-scale IT infrastructures. In order to obtain the best performance out of these services, it is essential that organizations enforce rational practices for the management of the resources that compose their infrastructures. A common point in most guides and libraries of best practices for IT management – such as ITIL or COBIT – 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. Therefore, in this article, we introduce a framework to support the automation of some key steps of risk management. Our goal is to organize risk information related to IT activities providing support for decision making thus turning risk response planning simpler, faster, and more accurate. The proposed framework is targeted to workflow-based IT management systems. The fundamental approach is to learn from problems reported in the history of previously conducted workflows in order to estimate risks for future executions. We evaluated the applicability of the framework 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.

1. Introduction

In order to deliver high quality services to customers, modern organizations often end up employing large-scale Information Technology (IT) infrastructures, typically composed of physical and logical heterogeneous resources such as routers, firewalls, servers, end-user hosts, network protocols, and software packages. As IT services are designed, deployed, maintained, and improved, organizations can run into problems, for example, of scalability and complexity of management. To achieve better outcome from provided services and avoid waste of substantial resources, rational practices in the management of IT infrastructures must be enforced. To this end, some best practice standards and libraries have been published, aiming to provide guidance for proper IT management. Two of the most widely recognized guides are the Information Technology Infrastructure Library (ITIL) [1] – proposed by the Office

* Corresponding authors. Tel.: +55 (51) 9861 1336; fax: +55 (51) 3316 7308 (J.A. Wickboldt).
E-mail addresses: jwickboldt@inf.ufrgs.br (J.A. Wickboldt), labianchin@inf.ufrgs.br (L.A. Bianchin), rclunardi@inf.ufrgs.br (R.C. Lunardi), granville@inf.ufrgs.br (L.Z. Granville), paschoal@inf.ufrgs.br (L.P. Gaspary), claudio.bartolini@hp.com (C. Bartolini).

1389-1286/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2011.05.025

of Government Commerce (OGC) – and the Control Objectives for Information and related Technologies (COBIT) [2] – introduced by the Information Systems Audit and Control Association (ISACA).

An explicit concern of the IT management guides is related to the necessity of managing risks associated with an organization's IT activities. This is emphasized by the fact that both OGC and ISACA have published specific documents for corporative IT risk management: the Management of Risk (M_o_R) [3] from OGC, and the Risk IT [4] from ISACA. According to M_o_R, to achieve their objectives, organizations must necessarily take a certain amount of risk. It is thus the role of the risk management discipline to help organizations to methodologically deal with risks associated with their activities.

Usually, organizations take risks as uncertain events or conditions that, if happen, may affect the accomplishment of business goals. Those events, along with the conditions that represent risks 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 both negative (threats) and positive (opportunities) effects of risks, in practice, negative effects are far more considered in real IT environments. This results in current risk management practices being in fact strongly focused on the prevention and mitigation of harm.

The risk management discipline is based on four logically sequential and cyclic processes [3]: (i) *identification* of possible threats and opportunities to the objectives of a given organizational activity, (ii) *assessment* of identified risks in terms of probability of occurrence and associated impact (*i.e.*, estimation of possible losses or earnings), (iii) *response planning* for preventive and reactive responses to identified risks, aiming to minimize threats and enhance opportunities, and (iv) *implementation and monitoring* of the planned responses in order to tackle risks, evaluate the effectiveness of preventive actions, and occasionally dispatch corrective ones. Along all these processes, it is important that organizations adopt a common set of internal policies and strategies for risk management to be shared among their departments and teams. Some of these policies and strategies, for example, may define tolerance thresholds, scales for estimating probabilities and impacts, and tools for documenting, reporting, and communicating risks.

Despite all best practices and recommendations, the experience of practitioners shows that there is little evidence that risk management is being efficiently applied in a systematic and repeatable way. In fact, standard guides like ITIL or COBIT only provide high level guidelines for general purpose risk management in a textual descriptive form. Very few information is given about how to actually implement these standards in practice and most of the proposed processes are assumed to be manual. Recently, some authors have investigated the actual benefits and shortcomings of different approaches for risk management in real-life environments [5–7]. These investigations expose many issues of current risk management actual practices, such as inadequate documentation, little knowledge reuse, and lack of tools to automate, report, monitor, and support decision making. In the end, the quality of risk-related decisions is often too much dependent on the experience of IT managers. The current practice on risk management usually encompasses an excessive dependency on people, thus becoming a time/resource consuming, occasionally counterproductive task.

Considering the today's actual risk management scenario, we emphasize that one of the major problems in risk management is the lack of automation and system-assisted routines. In this research, we pay special attention to problems in the *risk assessment* process, in which risks are tackled in terms of probability of occurrence and possible impact. The risk assessment process is usually based on interviews and brainstorms with involved stakeholders, in a very *ad hoc* fashion. Since the quality of risk related decisions and response planning depends directly on the accuracy of risk assessment, the employment of automated tools to assist IT managers in achieving more precise estimations becomes a key factor for the success of risk management as a whole.

Several authors have been investigating ways to support risk management in specific contexts or situations [8–13]. In previous investigations of our research group, we have also proposed punctual solutions to enable some degree of automation in estimating probabilities and impacts in risk assessment [14–16]. Our main goal in this article is to consolidate the approaches previously proposed into one single unified framework to support the automation of key processes of risk management, aiming to make it simpler, faster, and more accurate. The proposed framework is based mostly on best practices proposed in the aforementioned standards and libraries (*e.g.*, ITIL and M_o_R). In this work, we focus on 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, portfolio 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 and the details of the execution of these activities (including reports of adverse and favorable events) can be recorded to logs for further analysis. Our approach encompasses the automated analysis of logs of previously executed workflows in order to learn from events reported in the past, aiming to help in the design of better workflows for future execution.

In order to prove the concept of our solution, two case studies are taken from two IT related areas, namely *IT change management* and *IT project management*. The former presents general guidelines for consistently conducting changes over IT infrastructures, from the early specification, planning, and deployment, towards evaluation and review [17]. The latter is focused on the design phase of services, aiming to ensure that a project meets its objectives avoiding waste of resources [18,19]. These areas are relevant in the context of IT infrastructures and services management since they have received much attention from both academy and industry in recent years. Moreover, both projects and changes can be organized in the form of workflows and therefore may have their risks assessed using the unified framework proposed in this work.

The remainder of this article is organized as follows. In Section 2 a review of the available literature on risk management, specially related to IT Change Management and IT Project Management, is presented. In Section 3, some background concepts that are fundamental to understanding and motivating the proposal of our framework are presented. The information models employed to represent workflows and their executions are presented in Section 4. The conceptual framework itself, algorithms used for impact and probability estimation, strategies for calculating similarity among workflows, and risk summarization are introduced in Section 5. In Section 6 a discussion on the results from the evaluation of both case studies is presented. Finally, in Section 7 the article is concluded with 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 [20], health care decisions [21], and strategies of insurance companies [22]. The literature provides many definitions of what risk is and how it should be managed. For example, Holton [23] explains that risk denotes an uncertain event that will affect elements, and may occur in some present or future process. Chicken and Posner [24] have conducted a research on how people deal with risks across many different areas, including finances, medical, industry, projects, transport, and sports. In their book *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. Two of them have been already mentioned in this article, namely Management of Risk (M_o_R) [3] and Risk IT [4]. Both of them target risk management for organizations' IT processes. Two other more general purpose standards are the Risk Management Standard, introduced by The Institute of Risk Management (IRM) [25], and the ISO 31000:2009 Risk management – Principles and Guidelines, proposed by the International Organization for Standardization (ISO) [26].

2.1. Risk management from the IT change management perspective

Since large scale IT infrastructures support services that are essential for the business continuity of organizations, whenever changes to any of these services are required, IT operators should be aware of risks that may arise. According to ITIL, risks should be investigated, measured, and treated before any change is approved [17]. In this context, risk management can be considered an effort towards proactive problem treatment. In this article, we consider events that pose risks to changes as possible failures that might happen during their deployment (*e.g.*, failure on the installation of new software or damage of hardware being handled). Such failures have the potential to cause disruption, directly or indirectly, to one or more services supported by the IT infrastructure.

Before start proposing methods for the estimation of risks in IT changes, it is important to better understand what types of failures should be expected during change deployment and how this failures can be classified. Classification of risky events is important in order to generate more comprehensive results out of risk assessment. Rather than observing isolated events, it may be more intuitive for humans to analyze and draw decisions on grouped or categorized risk information (further discussions on the comprehensiveness of risk representations are presented in Section 6.2).

Several studies exist on failure or error representation. Wang et al. [27] explain that there are four requirements to compose a model that properly represents errors: (i) error categories and hierarchy should be represented, (ii) error models should be integrated without modifying the models of existing components, (iii) component-specific error behaviors should be captured, and (iv) error propagations should be handled. Russell et al. [28] have proposed a conceptual framework for classifying the exception handling capabilities of workflows and process-aware information systems. The authors classify exceptions into five categories: Work Item Failure, Deadline Expiry, Resource Unavailability, External Trigger, and Constraint Violation. This work inspires the classification of failures in changes employed in our first case study, further discussed in Section 6.1.

Focusing on Request for Change (RFC) definitions, Keller et al. [29] have introduced CHAMPS, which is a system for automating the planning of changes in IT environments. CHAMPS itself does 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. [30,31] 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 these advancements, a solution that proactively observes risks to avoid future (and potentially expensive) system rollbacks is still lacking.

Sauvé et al. [9] and Rebouças et al. [32] have proposed a solution for risk analysis in changes aiming to automatically determine priorities on the scheduling of various possibly concurrent RFCs. In those studies, a risk evaluation guided by the business objectives, in order to minimize the impact over the organization's services during the deployment of changes is employed. Also dealing with scheduling of RFCs, Setzer et al. [8] have modeled the resources of an IT infrastructure (*e.g.*, hardware, software, and services) 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 looses. Employing analytical models, the authors enable decision support for scheduling of single or multiple correlated changes. However, the risk analysis proposed in these investigations has application to the scheduling phase of IT change management. In this article, we apply the proposed framework for risk assessment to the planning of changes. Therefore, our objective is to predict likeliness of failures and possible disruption of services while changes are still being designed.

2.2. Risk management from the IT project management perspective

An important reference in the context of IT projects is the Guide to the Project Management Body of Knowledge (PMBOK), introduced by the Project Management Institute (PMI) [19]. In order to deal with risks in 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. PMBOK divides risk management into six processes, further detailed in Section 3: Risk Management Planning, Risk Identification, Quantitative Risk Analysis, Qualitative Risk Analysis, Risk Response Planning, and Risk Monitoring and Control. Similarly to IT change management, it is also important to classify risks in IT projects. PMBOK defines that risks in projects can be taken 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, which may be used to classify risky events: cost, time, scope, and quality.

Despite the current support of 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 [7] 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. [5] 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 the risk management process.

Some authors have employed probabilistic models to predict undesired events as well as estimate metrics for risk management in IT projects. Fewster and Mendes [11] have introduced a prediction model using a Generalized Linear Model (GLM) to estimate some Web design and authoring metrics. That research 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. Bayesian Networks (BNs) have been used in many investigations for similar purposes. Hearty et al. [33] have designed a model for effort prediction and risk assessment in software development projects that follow the Extreme Programming (XP) methodology. The authors' approach is based on the use of Bayesian Networks (BNs) and quantitatively estimated metrics (*e.g.*, iterations/time to complete) without requiring data about the success of past XP projects. Fenton and Neil [12], on the other hand, have applied BNs to predict software defects, while Luu et al. [13] employ it to estimate the likelihood of time-overrun in construction projects. Those investigations have contributed to the automation of risk assessment and, although relevant, these studies 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.

In this section we have mentioned some of the most used guides of best practices for risk management in IT. Although these guides provide general purpose guidelines for corporative risk management, they do not provide many details on how to implement these practices in daily operations. Usually, risk related processes are assumed to be manual and end up becoming time/resource consuming or inefficient tasks. In order to provide some degree of automation in specific contexts, we have shown that some authors proposed solutions to aid risk management and help on decision making. The main difference between previous approaches and our proposal is that, instead of focusing in one particular context, we propose a framework for risk assessment specifically for workflow-based management systems. Since this kind of system can be applied to many different environments, our framework has the potential to be adapted and employed in a variety of situations.

3. Background

In this section, in a first moment, we discuss IT change management, as envisaged by ITIL Service Transition book and materialized in a prototypical system called CHANGELEDGE developed by our research group. Afterwards, we present definitions of Project Risk Management processes, as proposed within the PMBOK, along with some of the barriers to the adoption of such processes in real life projects.

3.1. IT change management according to ITIL

Being one of the core processes of ITIL, IT change management [17] provides general guidelines for conducting changes over IT infrastructures, from the early specification and planning 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

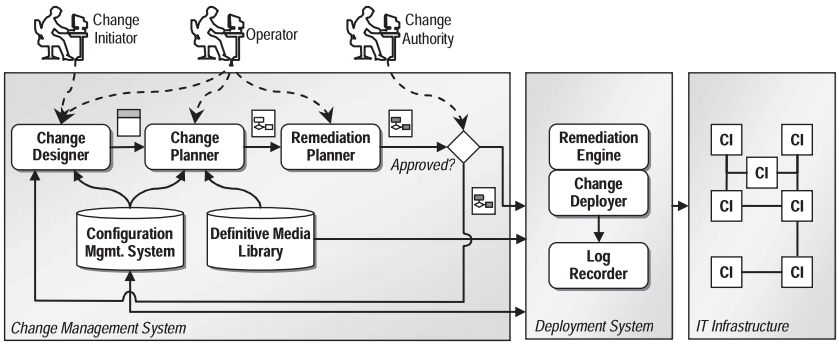


Fig. 1. Conceptual architecture of a change management system.

Configuration Items (CIs) affected (*e.g.*, devices, applications, services), but it does not to 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, has been proposed in previous work [30,31,34,35]. The components that are part of the conceptual architecture of the change management solution (*i.e.*, the CHANGELEDGE system) are presented in Fig. 1.

The *Change Initiator* starts the change process describing an RFC by 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*. After that, the *Change Planner* is responsible for producing an actionable workflow of finer-grained activities, based on definitions made in the preliminary plan, on information about configuration and software dependencies provided by the *Definitive Media Library* (DML), and on the current state of the IT infrastructure (provided by the CMS). The algorithm to generate such refined CP has been proposed in a previous research [34].

Each activity of a CP is described adhering to the *Activity Modeling Notation* (AMN) [35]. This notation is used to enable systems 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>`

For instance, sentence 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 might be specified in the `with <parameters>` directive. All constructs of AMN manipulate elements from the IT infrastructure, logically represented in the CMS, and/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. [35].

Following in the change 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 deployed. 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 has also been covered by previous work [30,31].

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 provided 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 brainstorms or meetings and it is much based on operators' knowledge. Depending on the urgency of a change, sometimes it is not 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 we propose a solution to automate risk assessment. Our goal is quickly have an overview about the risks in a CP, learning from historical information of past failed changes, giving the operator the opportunity to rearrange the CP before submitting it for approval and deployment.

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 a plan is responsibility of the

Remediation Engine. Every time a CI is affected by a change, 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 (*i.e.*, success or failure) and failure classification are also stored on the CMS for further evaluation.

3.2. Project risk management as envisioned by PMBOK

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 [36]. Specifically important in the context of this research, Project Risk Management is one of nine knowledge areas of PMBOK that comprises planning, identification, analysis, responses, and monitoring of risks that may affect project objectives. PMBOK divides Project Risk Management into six processes, as shown in darker boxes of Fig. 2.

The set of processes of the Project Risk Management knowledge area from 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 vs. 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 resources beyond those 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 taken in this process of detailing and measuring risks, it still depends on the knowledge of experts in risk analysis. 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 whole life cycle of a 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 other processes of other knowledge areas.

Some barriers to the adoption of PMBOK processes can be easily identified, especially in risk identification and analysis. First, in real projects, 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 resources, may propagate errors of qualitative

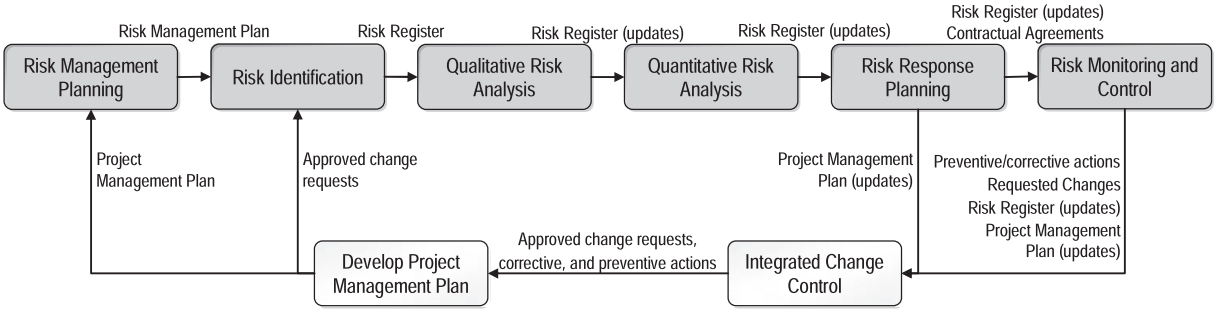


Fig. 2. Project Risk Management processes according to PMBOK [19].

assumptions 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 section, risk management is indeed a real concern in both IT change management and IT project management areas. In the context of IT projects there are well defined procedures to deal with risks. On the other hand, for IT changes, risk management is actually mentioned as essential to avoid excessive disruption of provided services; however there are no specific procedures or methods recommended. Aiming to help IT operators and managers on efficiently tackling risks, the framework proposed to automate some of the key steps of risk management will be presented.

4. Information models

In previous sections, the relevance of risk management in IT operations and the current shortcomings present in this context have been explored. In this section, we begin to introduce our solution to help on the automation of risk assessment for workflow-based management systems by describing the information models employed for the representation of workflows (Section 4.1) and their execution traces (Section 4.2) are explained.

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. According to Dumas et al. [37] 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 article to represent workflows is a subset of the Workflow Process Definition model proposed by the Workflow Management Coalition (WfMC) [38] as shown in Fig. 3.

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.

Transitions between activities are represented by instances of the class *Transition Information*. One activity may have transitions *to* or *from* (branches or joins) many other activities. Usually, activities will have resources associated and the description of the roles and responsibilities of this resources is detailed by instances of 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) [39].

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) [40].

4.2. Log records information model

Our framework bases its analysis on the execution traces of past workflows aiming to estimate risks for a future execution of a given workflow in hand. Although the model just described in Fig. 3 can accurately specify the structure and characteristics of workflows, their actual execution 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 in many executions or failures in performing activities can abnormally interrupt their executions sometimes triggering other workflows as backup plans. Therefore, to allow the representation of execution traces of workflows and future retrieval of these traces for risk assessment, in this article a model that employs a subset of classes from CIM is employed. Fig. 4 shows the proposed model where classes in dark gray have been introduced to attach risk related information to the execution records of workflows.

All classes in light gray are commonly used to represent general purpose log records in systems that use CIM to model any IT infrastructure. Instances of the class *Log* 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 is introduced for storing risk related information

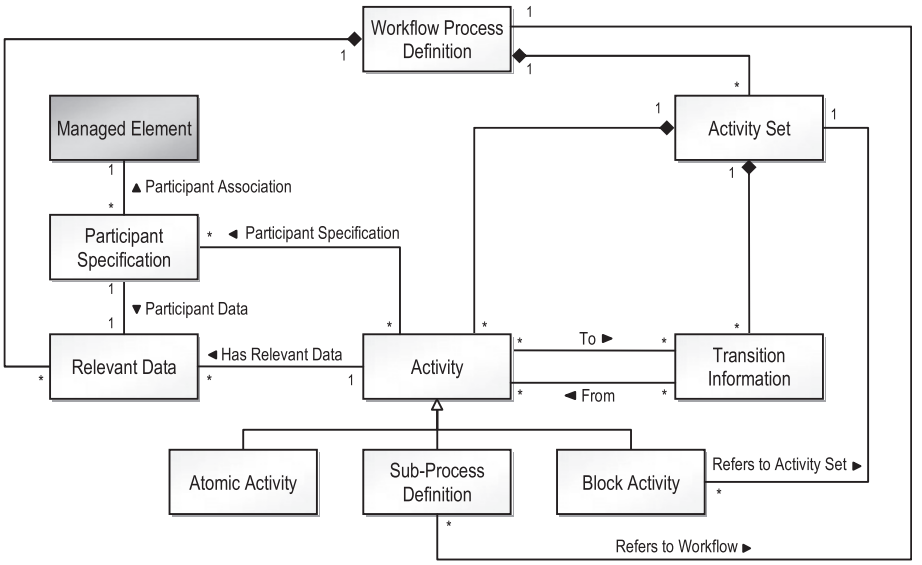


Fig. 3. Extension from the Workflow Process Definition model.

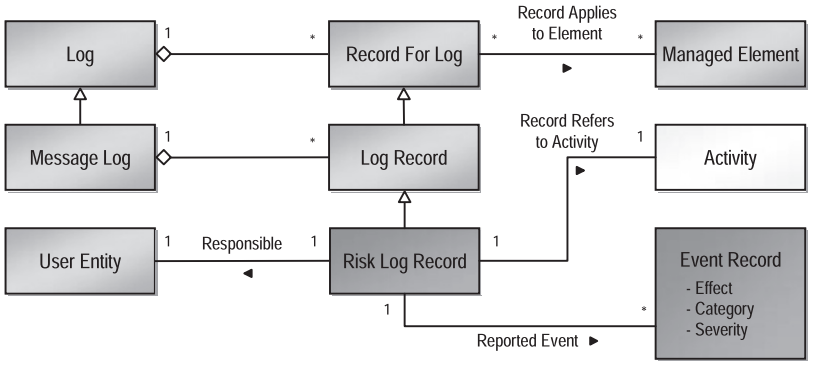


Fig. 4. Information model to represent execution traces of workflows.

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 events reported during the execution of a given *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 events according to a set of categories defined for a specific environment; and *Severity* that represents the dimension of the damage or advantage caused by the reported event.

5. Risk assessment framework

The current shortcomings of risk assessment in IT environments previously exposed in this article, motivate the proposal of a framework targeted to support risk related decisions taken by IT operators and managers, more specifically focused in workflow-based IT management processes. In this section, 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.

Fig. 5 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 Fig. 3, 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 Fig. 4.

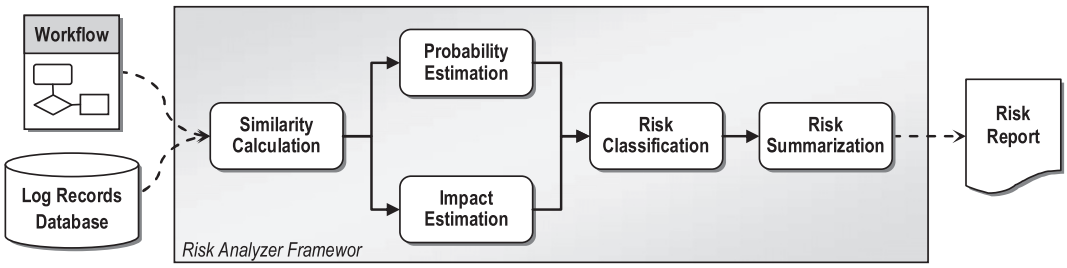


Fig. 5. Architecture of the framework for risk assessment.

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. The possible formats of these reports are presented and discussed in more details in Section 6. The following sections present the behavior of each internal component of the proposed framework. In order to ease the understanding of explanations, since negative risks are far more a concern then positive ones, only adverse events will be considered in the remainder of this section. However, it is relatively easy to include positive events simply repeating the process.

5.1. Similarity calculation

Similarity between workflows is a key metric in the context of this research and is currently an open topic of investigation itself [41]. 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 in 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. 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 will be able to combine several probabilities and impacts estimated from different activities previously executed and weight these values according to their similarities to the workflow being analyzed.

The *Similarity Calculation* module performs its operations for each activity of the analyzed workflow in three 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, develop, test, deploy) and have at least one associated participant in common;
2. **Calculate *Influential Workflows (IWs)***: similarity values have to be calculated for pairs of activities, one from the workflow being analyzed and another selected from the *Log Records Database* in the previous step. The 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 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 from the similarity measure;
3. **Calculate *Risk Affinity (RA)***: in order to capture the similarities between two IWs, we employ the RA metric (as shown in Eq. (1)). RA calculation uses a function θ that returns a value (ranging from zero to one) representing 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. For example, if two activities from different workflows have been used for the configuration of a given software element, performed by the same human role, but executed over different servers, they will reach a similarity factor of 75%, since three elements (*i.e.*, configuration, software, and human) out of four are the same (server is the 4th but different element). 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.

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

5.2. Probability estimation

The procedure for estimating probabilities is performed by the *Probability Estimation* module 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. This approach enables to take advantage of 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 $b \in A \mid b$ 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 are similar to *W* selected by the *Similarity Calculation* module. 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 be calculated; (ii) the inner loop iterates every possible event category *ec* considered in risk assessment process (a set of categories should be predefined and events segregated into them as presented in Fig. 4) (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 (IWs)* 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).

Probability is a very important metric used in the framework proposed in this article. In order to estimate probabilities that accurately reflect the reality of each environment, it is essential that log records are likewise accurately maintained. It means that operators and managers should invest time documenting the execution of workflows or creating automated means of doing so. Evaluating the accuracy of probabilities estimated is not an objective of this study, in fact the quality of documentation of log records and its impact over the accuracy of probabilities estimated by this framework is an avenue for future research.

5.3. Impact estimation

Impact is another key factor in the proposed framework for risk assessment. Probability, in contrast to impact, represents a quantification of the likelihood of a certain event that could happen. On the other hand, the definition of what impact is and how it can be estimated depends 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 objectives, impact may be faced as unexpected delay during the execution of activities. 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 for our solution. Since the proposed framework was conceived in a modular approach, what is really matters 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 preselected activities *A* (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 represents an interesting approach for areas like IT project management, where risks are addressed as events that affect project's objectives and can have their severities documented during project review. For example, assuming a given activity that was planned to take 8 h of work and when it was actually executed an event was reported to inform a time overrun of 4 h. In this case, the impact over time objective would be of 0.5. When dividing the severity (4 h) by the originally planned for one activity (8 h) it is expected to generate a normalized value ranging from zero (no impact) to one (great impact). However, it is clearly possible to have impact values beyond the upper bound of this range (in this example, a delay of 16 h, would produce an impact of 2). Through the *Risk Classification* module we deal 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 5.4.

impact in changes can be measured as business impact caused by unavailability of services, initially, a metric that represents the importance of *Configuration Items (CIs)* (*e.g.*, software, hardware, or service) to business is required. In previous work [14], we have introduced a metric called *Business Relevance (BsR)*, which is associated to every CI that is somehow relevant to the business. BsR is expressed by a numerical value and, regardless of scale adopted, it should enable comparisons between relevancies of 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 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 activities of workflows.

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) (Row 4). Following, each element that belongs to *D* (Row 5) will have its BsR accumulated in variable γ (Row 6). Afterwards, the tuple (*ci*, γ) is included in the set *U* (Row 7), which, at the end of all iterations, will contain 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, from the business impact point of view, the portion of the IT infrastructure that is compromised by failure of a particular CI. In order to calculate the impact of

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

a CI, an element that represents the IT infrastructure, whose all CIs depend on, is initialized in a variable *t*

Algorithm 3. Impact estimation based elemens 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

(Row 9). The AR of *t* is the sum of all BsRs defined, and it is handled in all workflows. Following, a predefined procedure is invoked 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* containing 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 are discussed in Section 6.1). Following, impact is calculated 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).

5.4. 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 risks in an organized and comprehensive way. Therefore, it is the role of the *Risk Classification* module to rank calculated probabilities and impacts into risk classification ranges, like those in Table 1. The Institute of Risk Management (IRM) [25] recommends quantifying probability using the following scale: high (more than 25%), medium (between 25% and 2%), and low (less than 2%). For impact, the IRM does not state general purpose threshold values, neither does any other guide, because impact depends very much on the context being analyzed. Therefore, we adopted the sample ranges presented in Table 1 as default values in the proposed framework. However, the threshold values of each range and the number of ranges itself should be customized in order to better match environment's needs.

After being mapped into one of the aforementioned ranges, the calculated probabilities and impacts are classified according to the *Risk Classification Matrix* presented in Table 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

Table 1
Classification ranges for probability and impact.

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

Table 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

Table 3
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

impact). Similarly to the classification ranges, the dimension of the *Risk Classification Matrix* might also be changed in order to better fit the requirements of a specific environment. For example, if two more ranges were included in Table 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 5×5.

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* described in the M_o_R framework [3], which employs 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.

5.5. Risk summarization

The final outcome of the proposed 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 on the prioritization of risk mitigation efforts. 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 3.

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 in all levels of 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 Eq. (2). This equation in fact 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 ith 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.

$$AR = \frac{n}{\sum_{i=1}^{n} \frac{1}{a_i}}. \tag{2}$$

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 Eq. (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.

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 6.2.

At this point we have presented the behavior of all modules of our risk assessment framework. It is important to remember that our solution is mostly based on the definitions of risk management processes found in the most popular guides for IT management, such as ITIL and PMBOK. These guides, although widely accepted and employed by organizations, do not provide detailed information on how to implement best practices for risk management in everyday operations. Moreover, most risk management task are assumed to be manually performed. With this framework we intend to enable some degree of automation in the risk assessment process by specifying the inputs, well defined by the information models previously presented, and establishing the steps (as modules of the framework) to perform measurements, analysis, and reporting of risks.

6. Evaluation

In order to evaluate the applicability and technical feasibility of the proposed framework, we present in this section two case studies in two IT related areas: change management and project management. Both of these areas recently considered as relevant by the research community of IT infrastructures and services management. This is highlighted by the amount of investigations published in last years covering issues on these topics in the main conferences and journals of this community (*e.g.*, IEEE/IFIP IM/NOMS, Manweek, CNSM, Springer JNSM, and IEEE TNSM). In addition, the industry is also showing interest in these areas expressed mainly by projects funded, like the CHANGELEDGE project mentioned in this work.

Our first case study, presented in Section 6.1, 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, described in Section 6.2, 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.

6.1. 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 that 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. Our proposed framework has been implemented as a module of this system, improving it with automated *Risk Reports*, and some preliminary results of this case study have been previously published [15]. In the remainder of this section, this case study is described along with some more discussions on the results achieved.

As mentioned in Section 2.1, in the context of IT changes, events that represent risks are regarded as failures that might happen during the deployment of Request for Changes (RFC) affecting the business 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*). There is no standard classification of failures applied to IT changes available in the specialized literature. Therefore, in this article, we employ a classification proposed by our research group considering failures during execution of changes under two aspects, *Source* and *Recovery* [14].

Regarding the source of the failure, six types are defined, as follows: (i) Activity Failure (AF) is directly related to a problem in the execution of an activity, usually related to software installation or configuration issues; (ii) Resource Failure (RF) is typically caused by hardware problems that make unavailable the elements where the activities are executed; (iii) Human Failure (HF) happens when humans associated to an activity do not behave the way they are supposed to; (iv) Time Failure (TF) is raised when deadlines are crossed or sometimes by issues in the synchronization of activities; (v) External Trigger (ET) occurs when some agent external to the change process interrupts or interferes in the regular execution of a

change; and (vi) Constraint Violation (CV) usually is related to an activity in a CP that needs to perform an operation that violates any of the organization's policies or by conflicting changes.

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. Although recovery information is not directly used for our framework, it is important to keep track of these records for further analysis, such as estimations of service disruption caused by changes. These recovery actions are classified into two categories, as follows: (i) No Action (NA) indicating that there was nothing to do after the occurrence of a failure (*e.g.*, the operator explicitly informs that nothing should be done or in the case of a fatal failures); and (ii) Remediation (RM) when the system executes a remediation plan to recover itself, it may be a rollback plan and/or a compensation plan.

In order to specify changes into RFC documents using the CHANGELEDGE system, an information model has been proposed in a previous work [34]. That model was conceived based on both the guidelines presented in the ITIL Service Transition book [17] regarding the change management process and the Workflow Process Definition, proposed by the Workflow Management Coalition (WfMC) [38] previously presented in Section 4.1. The main difference comparing the RFC model proposed by Cordeiro et al. and the Workflow Process Definition model is the addition of classes *RFC*, employed to textually describe the change, and *Operation*, which groups workflows of change (*i.e.*, Change Plans) into an RFC. Logging in execution of activities of changes are recorded according the model previously presented in Fig. 4.

6.1.1. Scenario and results

In order to evaluate the proposed framework, tests and measurements have been performed on an emulated IT environment. Each CI of this emulated IT infrastructure, for simulation purposes, implement a web service that produces failures pseudo-randomly during the deployment of changes in this case study, according to a uniform probability distribution. 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.

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 Fig. 6. In addition, SD should consider the impact of failures over the affected services. To this end, in this work Eq. (3) 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 assessment, results in an SD metric of an activity.

$$SD_i = \sum_{ft \in FT} F_{ft,i} * t_{ft,i} * IF_{ft,i}. \quad (3)$$

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 hardware 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

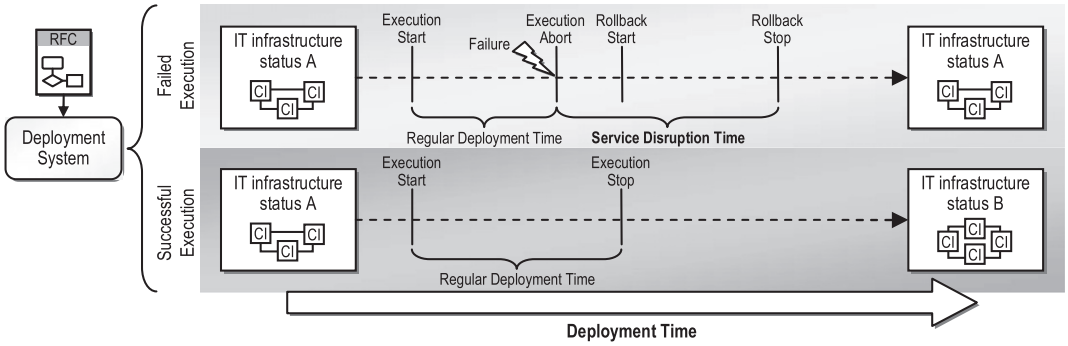


Fig. 6. Service disruption example.

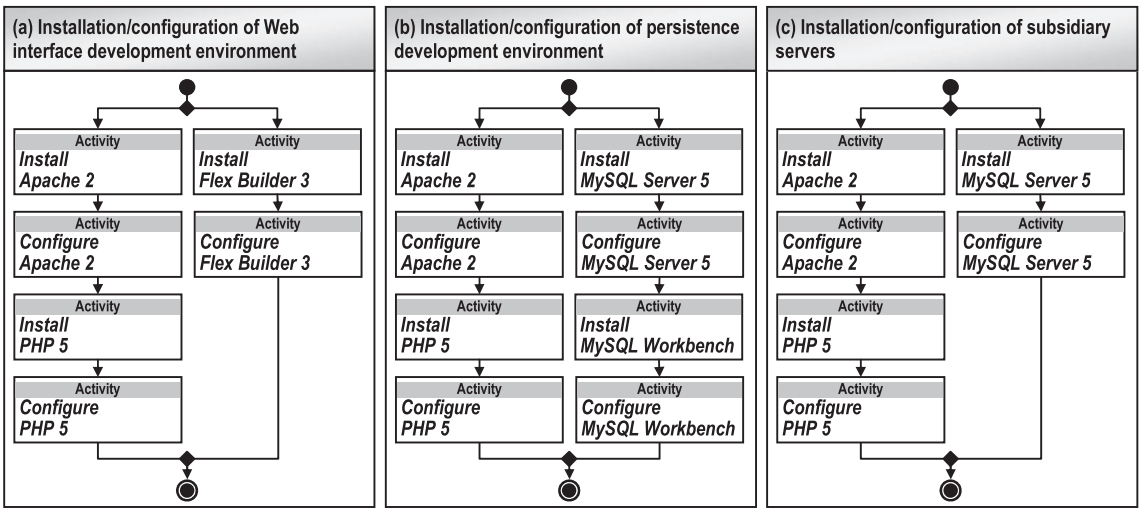


Fig. 7. Change plans for installation/configuration of development and production environments.

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 Fig. 7(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).

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 Fig. 7(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 six activities in common, while (c) and (a) have only four. 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 4(a). 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.

Supposing that a *Change Authority* has analyzed the *Risk Report* of Table 4(a) 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 4(b) 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 4
Risk reports before the deployment of first (a) and second (b) phases.

(a) 1st phase	AF	RF	HF	(b) 2nd phase	AF	RF	HF	
Configure	5.0%	0.0%	29.5%	Configure	6.8%	0.0%	28.8%	Probability
PHP	0.05	0.05	0.05	Apache	0.05	0.05	0.05	Impact
	5	6	4		5	6	4	Category
Configure	7.3%	0.0%	4.4%	Configure	11.2%	0.0%	8.9%	Probability
Apache	0.05	0.05	0.05	PHP	0.05	0.05	0.05	Impact
	5	6	5		5	6	5	Category
Configure	10.0%	0.0%	1.8%	Configure	3.5%	0.0%	8.9%	Probability
MySQL	0.05	0.05	0.05	MySQL	0.05	0.05	0.05	Impact
	5	6	6		5	6	5	Category
Install	8.7%	0.0%	–	Install	6.3%	0.0%	–	Probability
Apache	0.05	0.05	–	Apache	0.05	0.05	–	Impact
	5	6	–		5	6	–	Category
Install	7.9%	0.0%	–	Install	17.9%	0.0%	–	Probability
PHP	0.05	0.05	–	PHP	0.05	0.05	–	Impact
	5	6	–		5	6	–	Category
Install	0.0%	0.0%	–	Install	0.0%	0.0%	–	Probability
MySQL	0.05	0.05	–	MySQL	0.05	0.05	–	Impact
	6	6	–		6	6	–	Category

6.2. Application to IT project management

The second case study, applied to the context of IT project management, is intended to present how it is possible to generate comprehensive and interactive reports based on the proposed framework. Preliminary results of this case study considering a hypothetical software development project have been published in a previous work of our research group [16]. In order to obtain the results a database containing synthetic information about workflows of projects, execution of activities, and documented adverse events has been created. In this section, after characterizing the studied scenario, comprehensive *Risk Reports* automatically generated by our solution are shown under two different perspectives: Project Hierarchy View and Work Plan View.

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. To our knowledge, there is no widely accepted model for representing management related information of IT projects available in the literature. Therefore, in a previous work [16], such a model has been proposed inspired in a Business Technology Optimization (BTO) software from Hewlett–Packard (HP) called HP Quality Center [42]. In that model, IT projects were hierarchically organized into three levels: (i) *Releases* that are a partial version of the product or service being designed/developed in the project, (ii) *Iterations* intended to group together implementation efforts of the set of functionalities of a given *Release*, and (iii) *Cycles* that are associated to each *Iteration* and will often vary according to the methodology adopted (*e.g.*, one iteration can have four *Cycles* analysis, project, development, and testing). Moreover, activities of projects are organized into *Work Plans*, which are in fact workflows of that follow the Workflow Process Definition of WfMC. Logging of activities of *Work Plans* is also possible through the log records information model already presented in Fig. 4; where adverse and favorable events are recorded along with their categorization (in this context, categories are project objectives, *e.g.*, cost, time, scope, and quality), and the severity measured (*e.g.*, amount of hours delayed in activity).

6.2.1. Scenario and evaluation of risk reports

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 the 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 depicted in Fig. 8.

In *Release* 1, basic functionalities are implemented, such as database instantiation, client–server core modules, and web interface for basic CRUD (Create, Request, Update, and Delete) operations. Advanced features, such as reports composition (*e.g.*, availability, network load and latency, and alarms) and graphs for data visualization, are left to *Release* 2. In *Release* 3, 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 *Release* 4, 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 in Fig. 8, every iteration of the project is divided into four cycles: Analysis, Project, Development, and Testing.

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

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

Fig. 8. Comprehensive risk report in project hierarchy view.

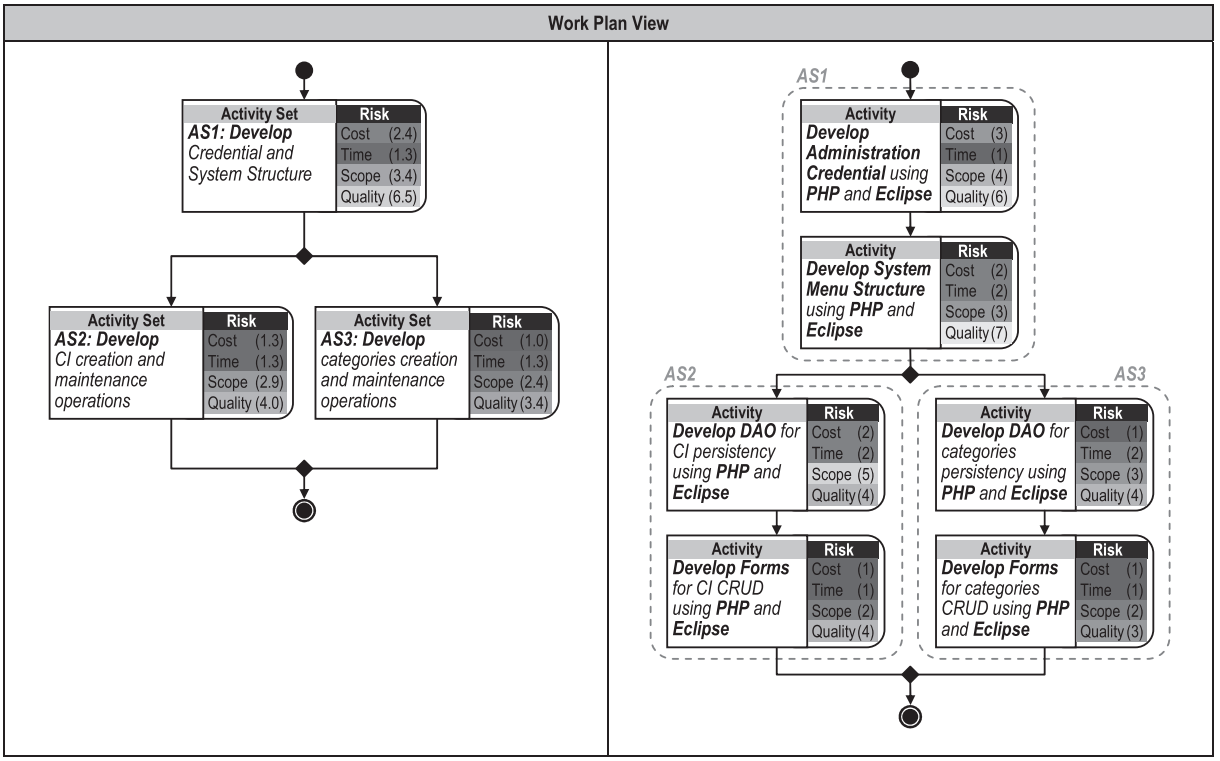


Fig. 9. Comprehensive risk report in work plan view.

each affected objective) for all activities of the project, a *Risk Report* as previously shown in Table 3 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 (Fig. 8), which gives an interactive overview of risks using the project hierarchical structure, and (ii) Work Plan View (Fig. 9), useful to investigate particularly risky work plans aiming to understand the sources of risk.

As shown in Fig. 8, 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 this studied project. 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, 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* of a project, the Work Plan View may be used. In Fig. 9, one *Work Plan* from *Cycle* 3 of *Iteration* 4 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 Fig. 9, 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 Fig. 9, 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.

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 (Fig. 8) 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.

Both case studies presented in this section have shown the applicability of the proposed framework to each specific scenario and that promising results can be achieved by adopting it. In Section 2, we have presented other studies that have dealt with similar problems, usually in specific situations. The main difference between our approach and those others is that, instead of focusing in a specific environment for application, we aim to develop a framework for workflow-based management systems; which in turn can be employed in several different environments. Unfortunately, there is no widely accepted framework for risk assessment available for use that we could simply apply to our case studies and quantitatively compare results against it. In addition, it is not possible to accurately reproduce the results of those cited investigations in order to perform a fair comparison of results.

7. Conclusion

In this article, the current need of organizations to enforce rational practices for IT infrastructures and services management has been discussed. Among many aspects covered by widely employed standards of best practice for IT management, such as ITIL introduced by OGC and PMBOK presented by PMI, the concern with risk management is remarkable. Guidelines from both the M_o_R framework (also from OGC) and the Project Risk Management knowledge area of PMBOK head the efforts of many modern organizations that want to rationally deal with their risks. Despite all guidelines and best practices provided by theses 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.

Therefore, we have 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 case studies presented in two different scenarios, namely IT change management and IT project management, have shown that the framework can be applicable to at least two different environments and also that it can be customized to better reflect specific needs in each situation. Although not exhaustive, the results indicate that the proposed framework is generic and may be applied a wider range of environments.

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 6.1).

Moreover, there are some other contributions in the context of this research that are worth mentioning. First, classifications of events that represent risks have been proposed, as presented in our two case studies. 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 introduced, which enabled 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.

Future investigations could extend the framework and apply it to other scenarios, such as incident management or portfolio management, as long as they employ workflow-based management systems. In both case studies presented in this article we have performed risk analysis over randomly generated workflow execution records and emulated IT environments. In order to evaluate the accuracy of predictions performed by our framework, it would be of great value to use data from real life IT management systems and compare the results with the actual measurements of these systems. 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

[1] OGC, Information Technology Infrastructure Library (ITIL), 2010. Available at: <http://www.itil-officialsite.com/. (accessed January 2010).

[2] ISACA, Control Objectives for Information and related Technologies (COBIT), 2010. Available at: <http://www.isaca.org/cobitonline/> (accessed January 2010).

[3] OGC, Management of Risk: Guidance for Practitioners, Office of Government Commerce, London, UK, 2007.

[4] ISACA, The Risk IT Practitioner Guide, Information Systems Audit and Control Association, Illinois, USA, 2009.

[5] R. van Wyk, P. Bowen, A. Akintoye, Project risk management practice: the case of a South African utility company, International Journal of Project Management 26 (2) (2008) 149–163, doi:10.1016/j.ijproman.2007.03.011.

[6] K. Bakker, A. Boonstra, H. Wortmann, Does risk management contribute to it project success? a meta-analysis of empirical evidence, International Journal of Project Management 28 (5) (2010) 493–503, doi:10.1016/j.ijproman.2009.07.002.

[7] E. Kutsch, M. Hall, Deliberate ignorance in project risk management, International Journal of Project Management 28 (3) (2010) 245–255, doi:10.1016/j.ijproman.2009.05.003.

[8] T. Setzer, K. Bhattacharya, H. Ludwig, Decision support for service transition management: enforce change scheduling by performing change risk and business impact analysis, in: 11th IEEE/IFIP Network Operations and Management Symposium (NOMS), Salvador, Brazil, 2008, pp. 200–207. doi:10.1109/NOMS.2008.4575135.

[9] J. Sauvé, R.A. Santos, R.R. Almeida, J.A.B. Moura, 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, USA, 2007, pp. 147–158.

[10] M. Marques, R. Neves-Silva, Risk assessment to support decision on complex manufacturing and assembly lines, in: 5th IEEE International Conference on Industrial Informatics (INDIN), Vienna, Austria, 2007, pp. 1209–1214. doi:10.1109/INDIN.2007.4384903.

[11] R. Fewster, E. Mendes, Measurement, prediction and risk analysis for web applications, in: 7th IEEE International Software Metrics Symposium (METRICS), London, UK, 2001, pp. 338–348. doi:10.1109/METRIC.2001.915541.

[12] N. Fenton, N. Ohlsson, Quantitative analysis of faults and failures in a complex software system, IEEE Transactions on Software Engineering 26 (8) (2000) 797–814, doi:10.1109/32.879815.

[13] V. Luu, S. Kim, N. Tuan, S. Ogunlana, Quantifying schedule risk in construction projects using Bayesian belief networks, International Journal of Project Management 27 (1) (2009) 39–50, doi:10.1016/j.ijproman.2008.03.003.

[14] J.A. Wickboldt, G.S. Machado, W.L.C. Cordeiro, R.C. Lunardi, A.D. dos Santos, F.G. Andreis, C.B. Both, L.Z. Granville, L.P. Gaspary, C. Bartolini, D. Trastour, 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, USA, 2009, pp. 445–452. doi:10.1109/INM.2009.5188847.

[15] J.A. Wickboldt, L.A. Bianchin, R.C. Lunardi, F.G. Andreis, W.L.C. Cordeiro, C.B. Both, L.Z. Granville, L.P. Gaspary, D. Trastour, C. Bartolini, Improving IT change management processes with automated risk assessment, in: 20th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management (DSOM), LNCS, vol. 5841, Venice, Italy, 2009, pp. 71–84. doi:10.1007/978-3-642-04989-7_6.

[16] J.A. Wickboldt, L.A. Bianchin, R.C. Lunardi, F.G. Andreis, R.L. dos Santos, B.L. Dalmazo, W.L.C. Cordeiro, A.L.R. Sousa, L.Z. Granville, L.P. Gaspary, C. Bartolini, Computer-generated comprehensive risk assessment for IT project management, in: 12th IEEE/IFIP Network Operations and Management Symposium (NOMS), Osaka, Japan, 2010, pp. 400–407. doi:10.1109/NOMS.2010.5488498.

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

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

[19] PMI, A Guide to the Project Management Body of Knowledge: PMBOK Guide, third ed., PMI – Project Management Institute, Pennsylvania, USA, 2004.

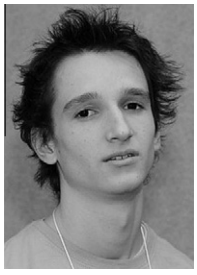
[20] K. Froot, D. Scharfstein, J. Stein, Risk management: coordinating corporate investment and financing policies, Journal of Finance 48 (5) (1993) 1629–1658.

[21] G. Danaei, S.V. Hoorn, A.D. Lopez, C.J.L. Murray, M. Ezzati, Causes of cancer in the world: comparative risk assessment of nine behavioural and environmental risk factors, The Lancet 366 (9499) (2005) 1784–1793, doi:10.1016/S0140-6736(05)67725-2.

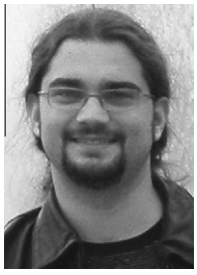
[22] C. Klnppelberg, R. Kostadinova, Integrated insurance risk models with exponential Lévy investment, Insurance: Mathematics and Economics 42 (2) (2008) 560–577, doi:10.1016/j.insmatheco.2007.06.002.

[23] G. Holton, Value-at-risk: Theory and Practice, Academic Press – Elsevier, 2003.
[24] J. Chicken, T. Posner, The Philosophy of Risk, Thomas Telford, 1998.
[25] IRM, A Risk Management Standard, The Institute of Risk Management, United Kingdom, 2002.
[26] ISO, ISO 31000:2009 Risk management – Principles and Guidelines, International Organization for Standardization, Geneva, Switzerland, 2009.
[27] L. Wang, A. Sahai, J. Pruyne, A model-based simulation approach to error analysis of it services, in: 10th IFIP/IEEE International Symposium on Integrated Network Management (IM), Munich, Germany, 2007, pp. 805–808. doi:10.1109/INM.2007.374718.
[28] N. Russell, W. van der Aalst, A. ter Hofstede, Exception Handling Patterns in Process-aware Information Systems, Tech. Rep., Eindhoven University of Technology and Queensland University of Technology, Eindhoven, The Netherlands, BPM Center Report BPM-06-04, BPMcenter.org, 2006.
[29] A. Keller, J. Hellerstein, J. Wolf, K. Wu, V. Krishnan, The CHAMPS system: change management with planning and scheduling, in: 9th IEEE/IFIP Network Operations and Management Symposium (NOMS), Seoul, Korea, 2004, pp. 395–408. doi:10.1109/NOMS.2004.1317679.
[30] G.S. Machado, F.F. Daitx, W.L.C. Cordeiro, C.B. Both, L.P. Gaspary, L.Z. Granville, C. Bartolini, A. Sahai, D. Trastour, K. Saikoski, Enabling rollback support in IT change management systems, in: 11th IEEE/IFIP Network Operations and Management Symposium (NOMS), Salvador, Brazil, 2008, pp. 347–354. doi:10.1109/NOMS.2008.4575154.
[31] G.S. Machado, W.L.C. Cordeiro, A.D. Santos, J.A. Wickboldt, F.G.A. Roben Castagna Lunardi, C.B. Both, L.P. Gaspary, L.Z. Granville, D. Trastour, C. Bartolini, Refined failure remediation in it change management systems, in: Mini-conference of 11th IFIP/IEEE International Symposium on Integrated Network Management (IM), New York, USA, 2009, pp. 638–645. doi:10.1109/INM.2009.5188872.
[32] R. Reboutas, J. Sauvé, A. Moura, C. Bartolini, D. Trastour, A decision support tool to optimize scheduling of IT changes, in: 10th IFIP/IEEE International Symposium on Integrated Network Management (IM), Munich, Germany, 2007, pp. 343–352. doi:10.1109/INM.2007.374799.
[33] P. Hearty, N. Fenton, D. Marquez, M. Neil, Predicting project velocity in XP using a learning dynamic Bayesian network model, IEEE Transactions on Software Engineering 35 (1) (2009) 124–137, doi:10.1109/TSE.2008.76.
[34] W.L.C. Cordeiro, G.S. Machado, F.F. Daitx, C.B. Both, L.P. Gaspary, L.Z. Granville, A. Sahai, C. Bartolini, D. Trastour, K. Saikoski, A template-based solution to support knowledge reuse in IT change design, in: 11th IEEE/IFIP Network Operations and Management Symposium (NOMS), Salvador, Brazil, 2008, pp. 355–362. doi:10.1109/NOMS.2008.4575155.
[35] W.L.C. Cordeiro, G.S. Machado, F.G. Andreis, A.D. dos Santos, C.B. Both, L.P. Gaspary, L.Z. Granville, C. Bartolini, D. Trastour, ChangeLedge: change design and planning in networked systems based on reuse of knowledge and automation, Computer Networks 53 (16) (2009) 2782–2799, doi:10.1016/j.comnet.2009.07.001.
[36] IEEE Std 1490-2003, IEEE Guide Adoption of PMI Standard a Guide to the Project Management Body of Knowledge, (Revision of IEEE Std 1490-1998), 2004. doi:10.1109/IEEESTD.2004.94565.
[37] M. Dumas, W. Van Der Aalst, A. Ter Hofstede, Process-aware Information Systems, Wiley-Interscience, 2005.
[38] WfMC, Workflow Process Definition Interface – Xml Process Definition Language. <http://www.wfmc.org/standards/docs/TC-1025_10_xpdl_102502.pdf> (accessed November 2009).
[39] DMTF, CIM – Common Information Model, 2009. <http://www.dmtf.org/standards/cim> (accessed November 2009).
[40] OASIS, BPEL – Business Process Execution Language – Version 2.0, 2007. <http://docs.oasis-open.org/wsbpel/2.0/> (accessed November 2009).
[41] L.A. Bianchin, J.A. Wickboldt, L.Z. Granville, L.P. Gaspary, C. Bartolini, M. Rahmouni, Similarity metric for risk assessment in IT change plans, in: 6th International Conference on Network and Service Management (CNSM), Niagara Falls, Canada, 2010, pp. 25–32.
[42] Hewlett-Packard Development Company, HP Quality Center. Available at: <https://h10078.www1.hp.com/cda/hpms/display/main/hpms_content.jsp?zn=bto&cp=1-11-127-24_4000_100__> (accessed November 2009).

Juliano Araujo Wickboldt is a Ph.D. student at the Institute of Informatics (II) of Federal University of Rio Grande do Sul (UFRGS), in Brazil. He achieved his B.Sc. degree in Computer Science at Pontifical Catholic University of Rio Grande do Sul (PUCRS) in 2006. He also holds a M.Sc. degree in Computer Science from the PPGC/UFRGS conducted in a joint project with HP Labs Bristol and Palo Alto. His current research interests include network management, management of Future Internet networks, and IT infrastructures and services management.



Luís Armando Bianchin is an undergraduate student in Computer Science at the Institute of Informatics of the Federal University of Rio Grande do Sul (UFRGS), Brazil. He is currently participating at an joint internship program between the II/UFRGS and the Technische UniversitSt Berlin. His research interests include computer networks, IT services management, and workflow similarity.



Roben Castagna Lunardi is a Ph.D. student at the Institute of Informatics (II) of Federal University of Rio Grande do Sul (UFRGS), in Brazil. He holds a B.Sc. degree in Computer Science from the Federal University of Santa Maria (UFSM) and a M.Sc. degree in Computer Science from the PPGC/UFRGS, concluded respectively in 2008 and 2010. His current research interests include IT infrastructures and services management and people in IT.

Lisandro Zambenedetti Granville is an Associate Professor at the Institute of Informatics of the Federal University of Rio Grande do Sul (UFRGS), Brazil. He received his M.Sc. and Ph.D. degrees, both in computer science, from UFRGS in 1998 and 2001, respectively. He has served as a TPC member (2003–2008), General Co-Chair (2004), and Steering Committee member (2005–2008) for the Brazilian Symposium on Computer Networks (SBC/LARC SBRC). Currently, he is member of the Brazilian Internet Committee (CBG.br). He has served as a TPC member for many important events in the area of computer networks, such as IM, NOMS, and CNSM. His main areas of interest include policy-based network management, management using/of Web services, and P2P-based services and applications.

Luciano Paschoal Gaspary received the Ph.D. degree in Computer Science from the Institute of Informatics of the Federal University of Rio Grande do Sul (UFRGS), Brazil, in 2002. In 2006, he joined the same institute, where he is now an associate professor. In addition to his appointment at UFRGS, Prof. Gaspary is currently Director of Technical and Scientific Council of the National Laboratory on Computer Networks (LARC) and vice chair of the Special Interest Group on Information and Computer System Security of the Brazilian Computer Society (SBC). Furthermore, he has been directly involved in the organization and served as technical program committee member of several IEEE, IFIP and ACM conferences including IM, NOMS, DSOM, IPOM, GLOBECOM and ICC.

Claudio Bartolini manages a research team HP laboratories in Bristol and Palo Alto working on collaborative Cloud services supporting decisions in IT strategy (in particular IT project and portfolio management). His background is on architecture and design of software systems and frameworks. Claudio obtained his Ph.D. in Information Engineering from the University of Ferrara, Italy, and has an M.Sc. in Electronic Engineering (hons) from the University of Bologna, Italy. He has published over twenty papers on international journals, conferences and workshop, and contributed to book chapters. He is a co-author of the W3C WSCL specification. He holds a number of patents in various countries. He is a frequent speaker at conferences, and chaired a number of conferences and workshops.