Improving IT Change Management Processes with Automated Risk Assessment

Juliano Araujo Wickboldt[1], Luís Armando Bianchin[1], Roben Castagna Lunardi[1], Fabrício Girardi Andreis[1], Weverton Luis da Costa Cordeiro[1], Cristiano Bonato Both[1], Lisandro Zambenedetti Granville[1], Luciano Paschoal Gaspary[1], David Trastour[2] and Claudio Bartolini[3]

[1] Federal University of Rio Grande do Sul, Porto Alegre, Brazil
[2] Hewlett Packard Laboratories, Bristol, UK
[3] Hewlett Packard Laboratories, Palo Alto, USA

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

1 Introduction

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

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

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

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

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

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

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

2 Related Work

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

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

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

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

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

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

[1] http://www.theirm.org/

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

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

3 Automated Risk Assessment Solution

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

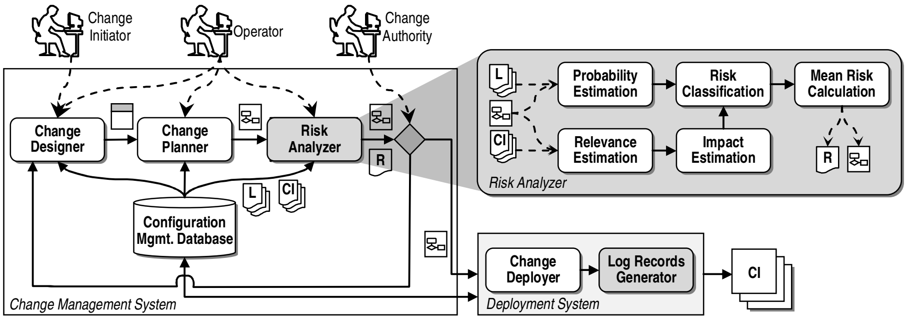

Fig. 1. Architecture of a change management system with risk assessment support

The change process starts when the *Change Initiator* specifies an RFC by interacting with the *Change Designer* component. Subsequently, the *Operator* sketches a preliminary Change Plan (CP), which consists of a workflow of high-level activities that describe the steps required to deliver the requested change. This workflow will be further refined by the *Change Planner* component. The outcome of this refinement is a CP composed of finer-grained activities that can be actually deployed over CIs [3].

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

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

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

3.1 Probability of Failures Estimation

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

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

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

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

In order to calculate these probabilities, the Algorithm 1 receives as input the CP of the RFC under analysis and a set of all execution records of RFCs available in the CMDB (for performance matters, this set is previously filtered matching RFCs having the same set of affected CIs as the one under analysis). Then, for each activity a of CP (Line 2), a subworkflow CP' containing a and the activities that influence its execution is defined (Line 3). Following, for every possible failure type ft (Line 4), the algorithm iterates through all CPs from set R (Line 6) searching for activities that meet the $alike\_enough$ criteria (Line 8). After that, RCP' will be a subworkflow with the activity b and all activities that influence it in *cp* (Line 9). Based on CP', RCP', and a failure type the RA between both subworkflows is computed (Line 10). The result of the RA (stored in A) acts as a weight for prioritizing failure probabilities of RFCs that have similar workflows. Following, the executions and failures of b are weighted and stored in T and F respectively (Lines 11 and 12). Probability of failures for each activity and failure type are then calculated by dividing F by T (Line 13) and added to the set S (Line 14). At the end S is returned as output of the function (Line 15).

Algorithm 1: *Probability of Failures Calculation Function*
Input: R: set of CPs with their execution records (logs), CP: change plan
Output: set of tuples containing activity, failure probability, and failure type

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

3.2 Impact Estimation

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

After AR computation, the *Impact Estimation* module will proceed with the normalization of these values to a metric we call Impact Factor (IF). This metric represents the portion of the infrastructure that is compromised by failure of a particular CI. The IF calculation function (Algorithm 3) receives as input the output of Algorithm 2. In order to calculate the IF of CIs, we define an element that represents the IT infrastructure, whose all CIs depend on. The AR of this element is the sum of all BsRs defined, and it is handled in all RFCs. Firstly, the algorithm instantiates the variable t with the element that represents the IT infrastructure (Line 2). Then, it invokes a pre-defined procedure that locates and extracts the CI t from the set R. (Line 3). For each tuple of the set R (Line 4), the AR from the CI contained this tuple is then divided by the AR of the whole infrastructure contained in tuple T (Line 5). Finally, a set I receives the results of these divisions (Line 6), which is returned as output of the function (Line 7).

Algorithm 2: *Absolute Relevance Calculation Function*
Input: V: updated representation of IT infrastructure, CP: change plan
Output: set of tuples containing CIs their Absolute Relevancies

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

Algorithm 3: *Impact Factor Calculation Function*
Input: R: set of tuples containing CIs and their Absolute Relevancies
Output: set of tuples containing CIs and their Impact Factors

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

3.3 Classificating and Reporting Risks

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

Table 1. Risks Classification Matrix

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

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

4 Prototype Implementation

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

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

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

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

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

5 Experimental Evaluation

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

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

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

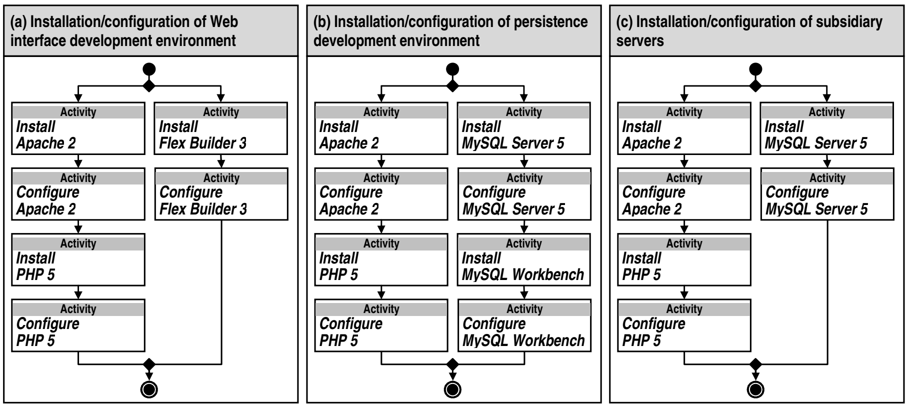
(a) Installation/configuration of Web interface development environment
Activity Install Apache 2
Activity Install Flex Builder 3
Activity Configure Apache 2
Activity Configure Flex Builder 3
Activity Install PHP 5
Activity Configure PHP 5
(b) Installation/configuration of persistence development environment
Activity Install Apache 2
Activity Install MySQL Server 5
Activity Configure Apache 2
Activity Configure MySQL Server 5
Activity Install PHP 5
Activity Install MySQL Workbench
Activity Configure PHP 5
Activity Configure MySQL Workbench
(c) Installation/configuration of subsidiary servers
Activity Install Apache 2
Activity Install MySQL Server 5
Activity Configure Apache 2
Activity Configure MySQL Server 5
Activity Install PHP 5
Activity Configure PHP 5

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

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

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

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

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

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

6 Conclusion and Future Work

In this work, we discussed the organization's need for rational IT management. Since changes are imminent in such a dynamic environment, failures during this process may have direct effect on business continuity. Therefore, risks associated to changes should be investigated and mitigated. However, risk assessment has been usually left under the responsibility of humans operators, which may lead to inaccurate basis for decision making. Thus, in this paper, we proposed a solution for automating the risk assessment in IT change management, aiming to aid administrators to design better changes improving quality of change management and managed services.

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

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

References

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

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