On the Risk Exposure and Priority Determination of Changes in IT Service Management

Jacques P. Sauvé, Rodrigo A. Santos, Rodrigo R. Almeida, and J. Antão B. Moura

Federal University of Campina Grande, Brazil
{Jacques,almeida,rodrigor,antao}@dsc.ufcg.edu.br

Abstract. This paper deals with the Change Management process within IT Service Management. Change Management includes several activities, some of which need to evaluate the risk exposure associated with changes to be made to the infrastructure and services. We present a method by which risk exposure associated with a change can be evaluated and the risk exposure metric is applied to the problem of automatically assigning priorities to changes. A formal model is developed to this end; the model captures the business perspective by using financial metrics in the evaluation of risk. A case study, performed in conjunction with a large IT service provider, is reported and provides good results when compared to decisions made by human managers.

Keywords: Change Management, Risk, Change Prioritization, IT Service Management, Business-Driven IT Management.

1   Introduction

The context. Information Technology Service Management (ITSM) has been the object of concentrated study over the past decade due to the ever-growing importance of IT to corporate activity. As a result, best practice collections for ITSM such as the Information Technology Infrastructure Library (ITIL)[2] and Control Objectives for Information and related Technology (COBIT) [1] are becoming popular. In this paper, ITIL vocabulary is used, although the work applies in general settings. ITIL divides IT management processes in several areas, one of which is Service Support, which includes such processes as Service Desk, Incident Management, Problem Management, Configuration Management, Change Management and Release Management. This paper focuses on the Change Management (CM) process.

Changes are made to *Configuration Items* (CIs) that are part of the IT infrastructure. CIs include servers, communication equipment, systems software, embedded software in, for example, routers, middleware, application software and so on. The CM process aims to ensure that efficient and prompt handling of all changes to the IT infrastructure is performed using standard procedures, in order to minimize the impact of changes on the services supported by the infrastructure. Change Management is a very important process as is attested to by the following assessment by Stephen Elliot, Research Manager, IDC: "Over 80% of business-critical service disruptions can be attributed to poor change control processes including flawed

change impact assessment." It is for this reason that, when an enterprise initiates the implementation of ITIL management processes, one of the first to be included is CM.

The CM process includes several activities including change initiation where a Request for Change (RFC) describing the required change is registered, change filtering, priority allocation, categorization ("minor", "significant", "major"), planning, testing, implementation and review. Significant or major changes must go through the Change Advisory Board (CAB) for analysis and approval. The CAB is a group of people capable of analyzing changes from a technical as well as from a business point of view. This paper looks into the *prioritization of change* activity in greater detail. When prioritizing a change, the change manager (or the CAB) must evaluate the *impact* on the business of *not* implementing the change as well as the *urgency* to the business. In ITIL terms, impact is usually associated with a degradation of service levels and urgency is associated with a business perspective of the change. The urgency can be partially estimated by examining the deadline specified by the business by which it needs the change to be implemented. Penalties are frequently paid by the service provider if the deadline is overstepped. For each change, the change manager must therefore ask: "How long can I delay to handle this change?" The answer is given as a priority level, say one of: "Immediate", "High priority", "Medium priority" or "Low priority". These priority levels and the semantics behind them are company policy.

Observe that assigning priorities does not schedule changes. Scheduling is an activity that is performed further down the line when plans are ready, changes have been tested and when changes must be allocated to *change windows*, chosen according to business convenience. Change prioritization is performed very early in the CM process, before plans are ready and before very much is known about change implementation.

How is prioritization done? Many dimensions must be taken into account by the change manager, including the business impact of service down time, the business urgency (deadline), the complexity of the change and risks associated with the change implementation. Risk itself is a complex dimension that includes change complexity, whether the activities have been performed before, the probability of a change being unsuccessful, etc. Uncertainty in the time needed to perform the change activities causing possible delays and service disruption, crossing deadlines with consequent penalties are a major source of risk.

The problem. Whether performed by the change manager or by the CAB, the change prioritization activity is difficult. At that early stage in the process, little information is known about the change, business impact must somehow be evaluated and risk must be accounted for. All of these difficulties are compounded by the sheer scale of the problem (the number of changes to be dealt with); for example, cases are known where a large service provider must deal with hundreds of changes *per week* for a single customer. How can be prioritization activity be performed adequately?

Our objective. This paper provides a method through which priorities can be automatically (or semi-automatically) assigned to changes. We use a Business-Driven IT Management (BDIM) approach to capture the business impact of service disruption due to changes. The impact is estimated by evaluating the risk exposure of service disruption due to delays caused by uncertainties in time when performing change activities. In an ITIL context, the resulting method can be packaged as a tool,

used by a change manager whenever change prioritization need to be calculated. This would typically be done whenever inputs to the tool change, for example, whenever a new Request For Change is submitted.

The rest of the paper is structured as follows: section 2 proposes a new risk-based impact model; this model is used in section 3 to automate the assignment of priorities to changes; section 4 discusses a real case study that validates the approach; section 5 summarizes related work; finally, section 6 offers a brief summary, conclusions and discussion of further possible work.

2   Estimating Change Impact Through Risk

Our objective in this section is to calculate the business impact associated with a change; risk will be used in the formulation. Let us first describe what we intend to do informally. A formal treatment will follow. We want to estimate the business impact caused by a set of changes to be applied to IT infrastructure. Since a BDIM approach is being used, one wants to estimate this impact using a metric understood by business people and a financial measure of impact is thus chosen. Assume that these changes affect a single IT service; it is straightforward to extend the formal treatment to cover several services affected by the changes. Several sources of impact are considered:

1. As soon as the RFC is submitted, there is already a need felt for the change to be implemented. The business is somehow suffering until the change has been implemented; in other words, there is a business impact right from the start. This may be due to a service being down, for example, as would happen if the change were requested as a result of a problem that disrupted the service. There may be other impact causes, say lost opportunities such as would occur for a change meant to bring up a completely new service.
2. While the change is being implemented, assume that the service is down. Thus the impact due to service unavailability will be captured. Change windows negotiated with the business, during which unavailability is not counted against the provider, are not considered here for the sake of brevity; they can easily be accommodated.
3. When the deadline associated with the change is crossed, a penalty may be paid by the service provider and new, more severe, business impact is felt until the change is implemented.

The change implementation time is subject to statistical variations and the resulting uncertainty in the time needed to complete a change can affect the impact. Also, the more one waits to *start* change implementation, the higher the business impact will be, since the probability of completing change implementation before the deadline will decrease. Since the impact is closely tied to the time-related risks associated with changes, we will call the numerical impact calculated for a change the (financial) *risk exposure* of that change. The risk exposure of a change is defined as the expected value (in a probabilistic sense) of the business losses accrued as a result of waiting to implement the change and then implementing it using activities of uncertain time duration (but with known distribution). Observe that we are borrowing a fairly standard definition of risk (see section 6): risk is typically calculated by taking into account the probability of occurrence of certain events and the impact resulting from each event. The expected value of impact is the risk exposure value.

Let us now formalize these concepts: our goal is to find an expression for the risk exposure, $R_n(t)$, associated with the nth change if the change implementation starts at time t. Time $t=0$ is *now*, the time at which the change manager is performing the risk calculation; time t thus indicates how far in the future one expects to start implementing the change. Let the set of changes for which one needs to calculate risk exposure be $C = \{c_1, \dots, c_n, \dots, c_{|C|}\}$.

Several parameters will be associated with change c_n:

- The change is initiated at time t_n^I (superscript I means "Initial"). Choosing this value is obviously of critical importance and will be considered in the next section when change prioritization is analyzed.
- The duration of implementation for the change is the random variable $\tilde{t}_n$ with cumulative probability distribution: $F_n(t) = P[\tilde{t}_n \le t]$. Let $M_n(t) = 1 - F(t)$ be the complementary cumulative distribution function.
- A deadline t_n^D (superscript D means "Deadline") exists before which the change must be implemented. Different impact calculations may be performed before and after the deadline. Although a change can have its own deadline, frequently, the deadline will be imposed by a Service Level Agreement (SLA) associated with the service affected by the change; in that case, the change simply inherits the SLA-specified deadline.
- Before a change is implemented, non-obedience to contractual clauses or the loss of business opportunities may cause business impact. Assume that, while the change remains unimplemented, financial loss is accrued at a rate $\varphi_{n,BD}^c$, before the deadline (in the interval $[0,t_n^D)$) and at a rate $\varphi_{n,AD}^c$ after the deadline (in the interval $[t_n^D, \infty)$). To help the reader, observe that superscript c means "Change" and subscripts BD and AD mean "Before Deadline" and "After Deadline", respectively.
- Recall that the service is assumed to be down during change implementation activities. Let $\varphi_{n,BD}^S$ (superscript S means "Service") be the rate of financial loss due to service unavailability during the time interval $[t_n^I, t_n^D)$, when $t_n^I < t_n^D$ (unavailability before the deadline). After the deadline, the cost of service unavailability may be higher and we let financial loss accrue at a rate of $\varphi_{n,AD}^S$ during the interval $[t_n^I, \infty]$, when $t_n^I \ge t_n^D$.

The crucial observation in calculating the expected value of loss can now be stated: if, at time t, change c_n has not yet been completed and loss is accumulated at rate φ, then, over the time interval $[t, t + dt]$, the accumulated loss is $\varphi \cdot dt$. Now, consider two time instants t_1 and t_2 occurring *before* a change has started; the total loss over the period is simply $\int_{t_1}^{t_2} \varphi \cdot dt = (t_2 - t_1)\varphi$. If, however, the change implementation has already started, loss will only accumulate until the implementation is completed. The probability distribution of $\tilde{t}_n$ must therefore be taken into account to calculate loss. At time t, loss accumulates with probability $P[\tilde{t}_n > t] = 1 - F_n(t) = M_n(t)$. Thus, between time instants t_1 and t_2, the accumulated financial loss will be $\int_{t_1}^{t_2} M_n(t) \cdot \varphi \cdot dt$.

We are now ready to combine these concepts to calculate risk exposure (expected value for impact). The expression depends on whether implementation starts before or after the deadline:

$$R_n(t_n^I) = \begin{cases} t_n^I \cdot \varphi_{n,BD}^c + \int_0^{t_n^D - t_n^I} (\varphi_{n,BD}^c + \varphi_{n,BD}^s) M_n(t) dt + \int_{t_n^D - t_n^I}^{\infty} (\varphi_{n,AD}^c + \varphi_{n,AD}^s) M_n(t) dt & \text{if } t_n^I < t_n^D \\ \\ t_n^D \cdot \varphi_{n,BD}^c + (t_n^I - t_n^D)\varphi_{n,AD}^c + \int_0^{\infty} (\varphi_{n,AD}^c + \varphi_{n,AD}^s) M_n(t) dt & , \text{if } t_n^I \geq t_n^D \end{cases} \qquad (1)$$

Figures 1 and 2 show which loss rate is in effect at which time and will help the reader understand the above equation. Figure 1 applies when the change implementation starts before the deadline and Figure 2 when it starts after the deadline.
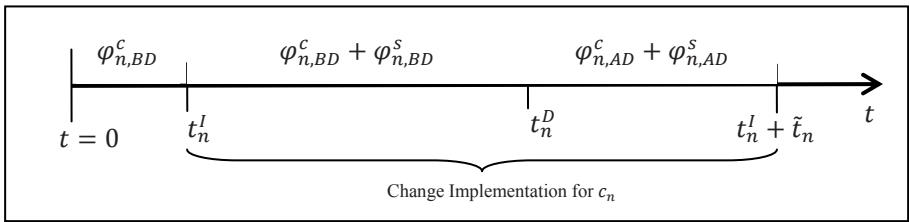

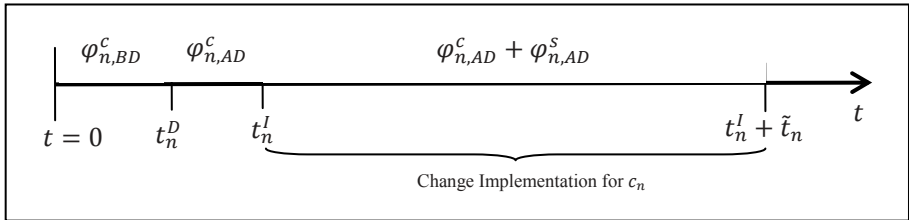**Fig. 1.** – Change implementation starts before deadline

Fig. 2. – Change implementation starts after deadline

3   Assigning Priorities to Changes

The risk exposure measure given in Equation (1) can be used in several manners. In this section, we show how to calculate a change's priority level ("Immediate", etc.) from it, indicating how quickly the change should be dealt with. Priority thus influences the order of implementation; it also influences whether the CAB gets involved or even whether a CAB meeting is called to consider a very high-risk change.

How can the priority level be chosen? ITIL recommends that priority be chosen after evaluating service impact and business urgency (through the deadline and financial loss, for example). Since both these dimensions are captured in the above risk exposure measure, it should be useful in establishing priority. From Equation (1), risk depends on the time t_n^I at which the change implementation starts. How is this

time to be chosen and how can the risk exposure measure be translated to a priority level?

A naïve algorithm can calculate risk exposure at $t_n^I = 0$ (now) and use risk thresholds for each priority level. This algorithm is not adequate. Consider, for example, the case of a change with very high penalty starting tomorrow, or next week. Merely evaluating risk at $t = 0$ does not reveal just how close the precipice really is!

Informally, the algorithm we propose is as follows. Company policy sets a *tolerance limit to risk exposure* (or a pain threshold), say \$10,000.00. Risk is evaluated assuming the change implementation will start at several instants in time, one for each priority level. Intuitively, this time instant captures "how long implementing changes with this priority can be delayed". For example, *now* is the instant associated with "Immediate", "this weekend" is associated with level "High", "two weeks from now" is associated with level "Medium", and so on. Now, the priority level chosen for a change is that which initiates change implementation furthest in the future but which does *not* cause risk exposure to cross the tolerance limit. Thus, if a change has \$1,000.00 risk exposure at $t_n^I = 0$ and \$50,000.00 at $t_n^I =$ "this weekend", then the change would be "Immediate" since delaying till the weekend makes risk exposure cross the pain threshold.

Let us formalize the concepts. Let $P = \{p_1, \dots, p_{|P|}\}$ be the set of priority levels, where level p_i is more urgent than level p_{i+1}. Time instant t_i is associated with priority level p_i. Further, let the tolerance level to risk be η, as set by corporate policy. Let e_n be the priority level associated with change c_n. Here is the result we were seeking:

$$e_n = \begin{cases} p_1 & \text{if } R_n(t_1) \geq \eta \\ p_{|P|} & \text{if } R_n(t_{|P|}) < \eta \\ p_i & \text{if } R_n(t_i) < \eta \text{ and } R_n(t_{i+1}) \geq \eta, 1 \leq i < |P| \end{cases}$$

Prioritizing a single change can be done in time $O(|P|)$, which is constant in the number of changes, making the algorithm adequate to prioritize very many changes with no performance restrictions.

4   Case Study and Validation

A case study was undertaken in conjunction with a large multinational IT service provider in Brazil. A scenario (also used in [5], but in the context of change scheduling) was set up in conjunction with the service provider and all parameters used here were furnished by the provider. We first describe the scenario and then discuss how validation of the models and methods presented in this paper was performed.

The services and infrastructure. Please refer to Figure 3. The IT infrastructure supports a credit card payment service and must be extended to support a new e-commerce service. Each service is subject to an SLA that specifies service level objectives, penalties, deadlines, etc. Configuration items supporting the services are as follows: two servers support the credit card payment service, a database server and an application server. Firewall B controls traffic to both servers. The e-commerce

service will be supported by the e-commerce server, the router and Firewall A. Furthermore, other service (whose exact nature was not specified by the provider) also and may be affected by maintenance activity to IT infrastructure components.

The SLAs. The three services are subject to SLA clauses as follows.

- General clauses for all SLAs: Any security-related RFC costs $1000/hour until the change is implemented. Any service affected by security issues must be brought down.
- Credit-card service SLA: Downtime costs $9000/hour before the deadline and $12000/hour from the deadline onward; deadlines are negotiated per incident. By default, RFCs raised due to incidents cost $1500/hour before the deadline and $2500/hour from the deadline onward; deadlines are negotiated per incident. Performance-related problems are penalized at the rate of $300/hour before the deadline and $400/hour from the deadline onward; deadlines depend on the severity of the performance problem.
- New e-commerce service. The service must be up by a certain date, 18 days in the future. From that point on, a penalty of $18000/hour is exacted from the provider.
- Other services. Performance-related penalties amount to $4000/hour before the deadline and $5000/hour from the deadline onward; deadlines depend on the severity of the performance problem.
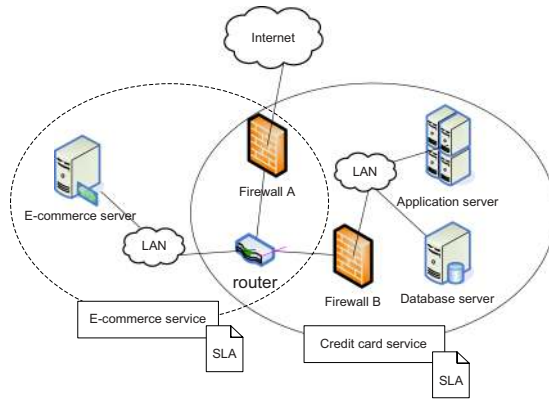
Fig. 3. Case study scenario infrastructure

The changes. Four changes were included in the scenario:

- Change c_1: Bring the new e-commerce service online. Since the Internet Connection used for the service is already being paid for even before the service is up, a link cost of $25/hour applies.
- Change c_2: Because of change c_1, new firewall rules must be installed in firewall A. Since this is tied to bringing up the e-commerce service, it must be done by the same deadline (18 days). The e-commerce service cannot come up until this change is performed.
- Change c_3: An incident occurring in the credit card service forces maintenance on the database server which will bring down the service. This change was negotiated to be performed within 20 days.

- Change c_4: Due to traffic overload, communication technology used for "other services" needs to be tweaked. This will have heavy performance impact and degrade service quality while the change is being done.

Data from the service SLAs and the changes described above are summarized in Table 1. Observe that $\varphi^s_{1,AD}$ (associated with the service) has value zero since, in this case, the service is down right from the start (RFC submission) and the impact is already included in $\varphi^c_{1,AD}$ associated with the change.

Table 1. Changes and their parameters

#	Change	t^D_n (days)	Affected service	$\varphi^c_{n,BD}$	$\varphi^c_{n,AD}$	$\varphi^s_{n,BD}$	$\varphi^s_{n,AD}$
c_1	Provision e-commerce service	18	e-comm	$25/h	$18000/h	$0/h	$0/h
c_2	Firewall configuration	18	e-comm	$1000/h	$1000/h	$0/h	$18000/h
c_3	Maintenance to database server	20	Credit Card	$1500/h	$2500/h	$9000/h	$12000/h
c_4	Maintenance to infrastructure	13	Others	$300/h	$400/h	$4000/h	$5000/h

Cumulative probability functions. In order to complete the risk model, the service provider supplied a historical log of 977 changes performed over 1 month from which probabilistic parameters were obtained. This log provided the RFC registration time, change implementation start time and implementation duration. Still, there were not enough changes of all types to extract a meaningful distribution from the data. We therefore extracted the mean μ_n and standard deviation σ_n from changes of the same type as the ones considered in the scenario and assumed a normal (Gaussian) distribution for the implementation time $\tilde{t}_n$. The appropriateness of this assumption can be argued as follows: a change implementation consists of several activities and the duration of each activity is a random variable with its own probability distribution. Since $\tilde{t}_n$ is simply the sum of several such random variables, we can use the Central Limit Theorem (see, e.g., [4]) which states that the distribution of a sum of independent random variable will tend to the normal distribution. We therefore use the normal distribution: $M_n(t) = 1 - \frac{1}{2}\left(1 + \text{erf}\left(\frac{t-\mu_n}{\sigma_n\sqrt{2}}\right)\right)$. Finite values for μ_n and σ_n guarantee that the infinite integrals of Equation (1) will converge.

The parameters shown in Table 2 were obtained from the historical change log.

Table 2. Mean and standard deviation for change duration

#	μ_n (hours)	σ_n (hours)
c_1	5.6840	6.2520
c_2	8.1421	7.4317
c_3	6.1018	6.1852
c_4	9.0152	7.3802

The case study scenario. These 4 changes must be prioritized by the change manager now (at $t = 0$). Let us use the following common priority levels: P = {"Immediate", "High", "Medium", "Low"}. The time instants used to evaluate risk exposure are *now* and the end of next three change windows, which occur weekly. Thus, $t_1{=}0$, $t_2{=}7$

Table 3. Priority levels when evaluating at t=0

#	$R_n(0)$	$R_n(7)$	$R_n(14)$	$R_n(21)$
c_1	$0	$4,000	$9,000	$1,520,000
c_2	$8,700	$176,000	$344,000	$1,100,000
c_3	$76,000	$496,000	$916,000	$2,021,000
c_4	$29,000	$72,000	$257,000	$325,000

(days), t_3=14, t_4=21. Company policy set the pain threshold at η =$200,000. Table 3 shows the results of the risk exposure calculation.

Recall that the algorithm sets the priority according to the largest time before the pain threshold is crossed. This would set change priorities as: {"Medium", "High", "Immediate", "High"}. For example, change c_1 crosses threshold between t=14 (medium) and t=21 (low), which makes it a medium-priority change. The time t=14 does not mean that this change should be implemented 14 days from now; it means that leaving it till the next change window (t=21) is too painful. Ideally, it can be done much ahead of that time limit. This will depend on the set changes to be performed, resources available, etc. The change scheduling problem is studied in [5].

Priority levels change with time as can be seen in Figure 4. Let us assume that time passes and we are now one week later. Deadlines are now closer by 7 days and, if evaluating priorities again, yields {"High", "Immediate", "Immediate", "Immediate"}. If the change manager waits a further 7 days, all changes are tagged as "Immediate".

Validation. The case study scenario described above was configured with the help of an experienced change manager working at a large multinational IT service provider in Brazil. In order to validate our method, we asked this manager to prioritize the changes using his current approach. The manager typically uses proximity to deadlines and loss rate before the deadline to assign priorities. His final priorities were: {"Medium", "**Medium**", "Immediate", "High"}. We then ran our algorithm and obtained the results shown previously: {"Medium", "**High**", "Immediate", "High"}. There is only one difference in priority. We showed our results, the model used and
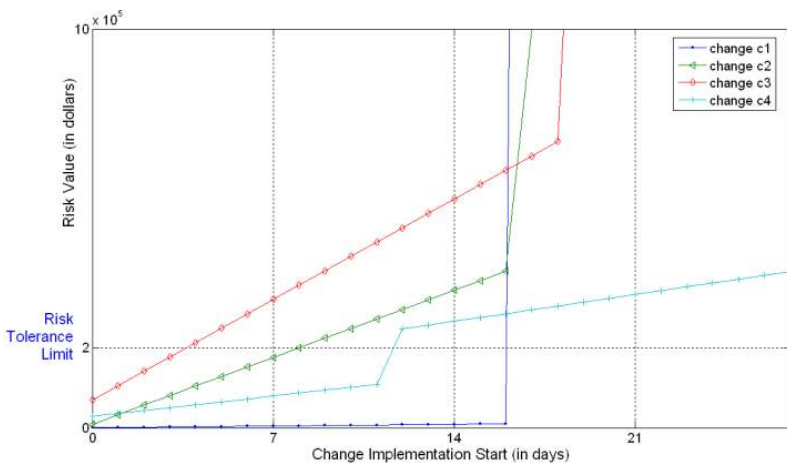

Fig. 4. Priority levels changing with time

the graph in Figure 4 and the manager agreed that "High priority" is more adequate for change c_2, since risk exposure would be too high if the change were left till later.

Discussion. Several points concerning the proposed method for evaluating risk exposure and assigning priorities need clarification. A model is no better than the parameters fed into it. The model used here has several inputs such as loss rates and the probability distribution functions for change duration. Where do these come from?

Loss rates can either come from SLAs (as we did here) or can come from a business impact model such as used in BDIM models. As an example, for e-commerce applications, business process throughput (monetary transactions per second) can be used to estimate loss due to downtime (see, e.g., [17]).

To obtain probability distributions for change duration, we suggest two alternatives: historical information can be used as we did here; furthermore, if historical information is available for individual change implementation activities, then distributions for each activity can be used to estimate distribution for change duration using the central limit theorem. A second alternative is to ask change implementers to specify minimum and maximum values for expected change implementation times and use a Weibull distribution; a distribution with negative skew is more likely to match reality since experience shows that actual change duration is more likely to edge closer to the maximum value.

5   Related Work

This section reviews some past work concerning risk that bears some relationship to our own work. The discussion progresses from the general to the more specific.

Work concerning risk in general. Risk management is a well-studied subject in various areas of human endeavor. The definition of risk itself varies according to the area of application. For example, in *Statistics*, risk is taken as a probability of some event that is seen as undesirable. More frequently used is the definition from *Probabilistic Risk Analysis* (PRA) whereby risk is characterized by two quantities: the severity of the possible adverse consequence(s), and the probability of occurrence of each consequence. In *Finance*, risk is the probability that an investment's actual return will be different than expected. In *Information Security*, risk is a similar to the PRA definition but considers two separate probabilities: that there are threats and that there are vulnerabilities to be exploited by threats. Our work is based on the PRA definition of risk (see, for example, [3]).

Risk in IT. In the world of IT, risk has been used in project management [6], software development, security [7, 8], and other areas [12]. Most risk assessment methodologies use probabilistic analysis. All of these approaches were instrumental in helping reach the model we propose here. However, whereas the approaches listed here calculate numerical values of risk, they use ad hoc weights and impact measures that are not direct business metrics; by contrast, our method carefully defines risk parameters and calculates values for risk exposure in terms of metrics that are directly understandable by business people. Also, our approach uses historical information to estimate model parameters.

Risk in change management. Some tools are commercially available that claim to help in managing changes, although no details are given that can be used to evaluate and compare methods [9, 10]. Several papers have presented approaches to qualitatively evaluate risk (e.g., [11, 13]). These studies do not provide quantitative risk analysis. On the other hand, most of these methods evaluate more dimensions than our analysis which limits itself to the risks associated with the uncertainty in change implementation duration.

Keller and others, in [14, 15], present the CHAMPS system to automate some steps in the execution of changes. Even though the authors try to solve a different problem in Change Management, this work was one of the first to model changes formally and influenced our own work.

Finally, some of our own past work in change scheduling led to the model developed and presented here [16, 5]. Our past work in change management dealt with scheduling and business impact. The similarity with our past work is that *business loss* is a basic metric used to solve the scheduling (past work) and prioritization (this work) problems. However, our present work deals with another change management process activity (prioritization) and includes risk (due to uncertainties) in the model formulation; both of these things are completely new.

Assigning priority to changes using risk. To our knowledge, this is the first formal, quantitative, business-driven method to automatically quantify risk and to assign priorities to a set of changes in ITSM.

6   Conclusions and Future Work

Summary. This paper has dealt with the Change Management process within IT Service Management. Change Management includes several activities, some of which need to evaluate the risk associated with changes to be made to the infrastructure and services. We presented a method by which risk exposure associated with a change can be evaluated and the risk metric was applied to the problem of automatically assigning priorities to changes. A formal model was developed to this end; the model captures the business perspective by using financial metrics in the evaluation of risk exposure. A case study was performed in conjunction with a large IT service provider and provides good results when compared to decisions made by human managers. To the best of our knowledge, this is the first such automatic solution published in the literature. The method is scalable and can be applied to evaluate risk exposure and prioritize hundreds or thousands of changes.

Conclusions. The validation exercise has shown the method to be useful. Risks associated with changes can be calculated and changes prioritized in an automatic fashion. We understand that we have not concluded full validation and that more extended use is required to reach final conclusions regarding the worth of the approach. Still, preliminary results are very encouraging and the change manager participating in the study wholeheartedly supports our continued efforts. Observe also that our method need not altogether substitute human managers or claim to "do better" than human managers: it claims to help managers handle a larger scale of changes by automating some of the risk and priority calculations and provides better visibility into the possible impact of changes from a business (financial) perspective.

Future work. We would like to improve the following deficiencies: model parameters may be difficult to obtain if SLAs are deficient, the business impact models used may not be applicable in all type of business processes affected by IT services, the model only considers risk due to uncertainties in time and lateness in implementing changes; other risk dimensions such as change complexity and the presence of back-out plans can be investigated.

Finally, the risk metric proposed here can be applied to other change management activities such as change scheduling and also to other ITSM processes.

References

1. IT Governance Institute, Cobit 3rd edn. (2000), www.isaca.org/cobit.htm
2. IT Infrastructure Library, ITIL Service Delivery and Support, OGC, UK (2003)
3. Kaplan, S., Garrick, B.J.: On the Quantitative Definition of Risk. Risk Analysis 1, 11–27 (1981)
4. Rohatgi, V.K.: An Introduction to Probability Theory and Math. Statistics. Wiley, Chichester (1976)
5. Rebouças, R., Sauvé, J., Moura, A., Bartolini, C., Trastour, D.: A Decision Support Tool for Optimizing Scheduling of IT Changes. In: 10th IFIP/IEEE Symp. on Integrated Mgmt., IEEE Computer Society Press, Los Alamitos (2007)
6. PMBOK - Project Management Book of Knowledge, http://www.pmi.org
7. http://www.cert.org/octave/
8. http://www.cramm.com/
9. Cisco IT Balances Innovation and Risk With Change Management Process, http://www.cisco.com/web/about/ciscoitatwork/case_studies/business_management_dl2.html
10. IT Service Management - Change and Configuration Management: Reducing Risk by understanding your infrastructure, http://www-03.ibm.com/solutions/itsolutions/doc/content/bin/itsol_it_service_management_change_and_configuration_management.pdf
11. Goolsbey, J.: Risk-Based IT Change Management, http://web.reed.edu/nwacc/programs/awards/excellence_award/pnnl_submissions_07/pnnl_ risk-based_it_change_management.pdf
12. Benoit A., Rivard S., Patry M.: Managing IT Outsourcing Risk, Cirano, Montreal (2001)
13. Mosier, S., Gutenberg, S., Raphael, R.: The Relationship of Technology Change Management to Risk Management, Engineering Management Society (2000)
14. Keller, A., Hellerstein, J.L., Wolf, J.L., Wu, K.-L., Krishnan, V.: The CHAMPS system: change management with planning and scheduling. In: Network Operations and Management Symposium, pp. 395–408 (2004)
15. Brown, A.B., Keller, A., Hellerstein, J.L.: A model of configuration complexity and its application to a change management system. In: IM 2005, pp. 631–644 (2005)
16. Sauvé, J., Rebouças, R., Moura, A., Bartolini, C., Boulmakoul, A., Trastour, D.: Business driven support for change management: planning and scheduling of changes. In: DSOM 2006. 17th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management, Dublin, Ireland (October 23–25, 2006)
17. Menascé, D., Almeida, V.A.F., Fonseca, R., Mendes, M.A.: Business-Oriented Resource Management Policies for e-Commerce Servers, Performance Evaluation 42, pp. 223–239. Elsevier Science, Amsterdam (2000)