Snowball Effects on Risk Mitigation Scheduling

Process and Tool

Abstract—Risk management is widely accepted as a routine activity in software project management. Many project risks are often complex and intertwined. Once some risks are materialized, they spawn active issues which can cause some other risks to occur. Moreover, these risks will have stronger impact than they could happen alone. Such snowball effect has been well known in risk management; however, few studies have addressed their impacts to risk mitigation plan. This paper will apply the autoregressive model to estimate the snowball effect on six risk mitigation strategies.

Keywords—Scheduling strategy, Risk mitigation, Time element, Risk management, Software project management

I. INTRODUCTION

The positive correlation between effective risk management and project success was emphasized in [1, 2, 3]. The adoption of risk management practices can help to increase the success rate of project and then enhance the competitiveness of organizations. Risk factors are often intertwined [4] making any assumption of their independence underestimate the complexity of risk problems in the end. When we consider that the impact of a risk is brought forward from some other previously materialized risks, which is named as “snowball effects.” In order to manage this kind of accumulated effect, one may add a number of risk review sessions from time to time. For example, agile software development, which is a highly iterative process, intends to reduce risk interconnectedness or snowball effect in different development phases [5].

Risk mitigation is essential for risk management because it aims to reduce or eliminate risks. To make the best use of project resources and to prioritize which risk factors should be dealt with, a scheduling strategy for risk mitigation is needed to determine the risks to be mitigated and when to mitigate them. The three basic strategies for scheduling risk mitigation are risk value first strategy, emergency first strategy and lowest-effort first strategy [6]. For easy reference, these strategies are denoted as “V”, “E” and “L” in this paper, respectively. As in V strategy, risks are scheduled for mitigation according to their risk values (i.e. Risk Exposure) [7] so that risks with higher risk values will be treated earlier. As in E strategy, risks are scheduled for mitigation according to possible time they may occur. As in L strategy, the mitigation is based on how much effort we will need to control each risk factor so that higher productivity throughput can be achieved earlier. Although a risk mitigation plan in reality is a highly customized combination of V, L and E based on not only resources but also project politics and project experience, understanding the three basic strategies is still helpful.

How the three mitigation strategies perform in different situations has been studied through a computer simulation [6]. However, one unrealistic assumption has been made: risk factors are independent and a risk can only be considered mitigated as far as the mitigation process for that risk is done. Such limitation is huge and it hinders the development of a risk software management tool in practice, in which users work out a list of risk factors with probability and impacts, and a customized mitigation plan, the tool will perform the simulation and address the overall impact as feedback.

This paper aims to investigate different scheduling strategies for risk mitigation by a computer simulation. The impact of risks is accumulated as an autoregressive model AR(1) [8] and a partial mitigation is allowed.

The paper is organized as follows. We briefly review the risk process and simulation model of risk management process (SMRMP) in section 2. In section 3, we formally present different scheduling strategies, the autoregressive model of risk management process and partial risk mitigation. The next section compares the performance of identified strategies with or without snowball effects. In the final section, we conclude our study.

II. BACKGROUND

A. Risk Management Process and Simulation Tool

Risk management aims to identify risks and take actions to reduce or eliminate their probability and/or impact so that the project is kept from being damaged by risks. There are many models and standards to guide the risk management practice, such as risk management paradigm developed by Software Engineering Institute, PMI framework, IEEE Std 1540, AS NZS 4360 and ISO 31000. Although these models and standards address the risk management processes in different manners, they can be mapped to each other to a large extent. Generally, these paradigms, models and standards follow the cyclic process shown in Fig. 1.

Risk Management Planning defines how to conduct risk management practices throughout the project. Risk
identification aims to identify risks that would affect the project objectives and document their characteristics. It is expected that a list of risk factors can be identified at this stage. The risk analysis aims to understand the identified risks and provide data to assist in managing them. Generally, risk analysis includes: (1) estimate the probability, impact, and the expected timing of the risk; (2) establish a risk mitigation plan. Risk monitoring and control aims to tracking the change of all identified risks and identifying new risks, monitoring residual risks, and evaluating risk response effectiveness and performance of risk management.

When a risk simulation tool is in place, it enhances the cyclic process of risk management shown in the right part of Fig 1. A risk manager can input several risk mitigation plans together with a list of detailed risk factors and the overall impact for each draft on the project can be estimated. He/she may then revise and finalize the mitigation plan. As mentioned, a risk mitigation plan often takes project politics and organization culture into account. Thus it may not be practical to have a fully optimized plan generated by a system. This paper will propose an autoregressive model for risk simulation.

B. Simulation Model of Risk Management Process (SMRMP)

Few studies have explicitly modeled the time elements of risk. [6] proposed variants of risk, presented a model of risk lifecycle, and gave the relationship between the risk variants by explicit consideration of the occurrence time of risk.

A stochastic simulation model of risk management process called SMRMP with due consideration of time elements of risks is defined in [6]. The simulation model can be used for many risk management issues, such as understanding of risk management process, predicting risk management outcome, and making informed risk management decision.

There is one major limitation. Each risk is independent to each other. The simulation process is a strictly static and the expected variance is constant.

III. METHODOLOGY

A. Modeling of Risk Mitigation

A project may have a number of phases and each phase can have its cycle of risk management. It is possible that one cycle of risk management crosses the whole project. For easy explanation, we will consider one project phase. Although each project phase has different length of time, it can be easily normalized so that each phase has 100 time units. A risk (r) occurrence period will be a closed time interval \([t_{eo}, t_{lo}]\) within the phase. The risk can only be materialized within its occurrence period. Surely some risk will cover the whole length; however, the more precise information can be defined, the better estimate can be obtained. The probability \((p)\) to occurrence of a risk is 0 to 1. Each risk impact scale \((i)\) is 0 to 1. The mitigation effort \((eff)\) to input is person-day and it can be converted into person-time-unit as a project phase always has 100 time units.

Once a detailed list of risks has been identified, a risk manager based on the manpower resources can work out a risk mitigation plan. Fig 3 illustrates the relationships between a risk mitigation plan and a risk. In Fig 3, a risk \(r\) with \([t_{eo} = 40, t_{lo} = 90]\) and \(eff = 20\) person-unit-time has been identified at the stage of analysis of a cyclic process of risk management given in Fig 1. Assume that the risk would be materialized and identified at time \(k\). There are four cases of a risk mitigation plan for \(r\). Case 1 is preferred. The duration of case 2 has passed \(t_{eo}\) which stands for a chance that the risk \(r\) might be materialized before its full mitigation. Case 3 illustrates that the mitigation for \(r\) is not yet complete. In this situation, we consider that the impact has been proportionally lessened. Case 4 is unfavorable since the \(r\) occurs at time \(k\).
Figure 3. Cases for a mitigation plan

From Fig 3, we know that how the mitigation plan is defined could affect the outcome. Although mitigation plans involve a number of factors including organization culture and project politics to dominate which risks should be mitigated, it will be still practical and beneficial that a risk manager defines several draft plans and a simulation engine based on an autoregressive model estimates the overall impacts for each draft.

In this paper, we will adopt three different risk mitigation strategies to develop six mitigation plans. The objective is to understand whether some of six mitigation plans may substantially affect the overall impact as outliers. These plans should be taken out of consideration. It should be noted that one should not look for an optimized mitigation plan. It is impractical because many risk factors and values in a project are subjectively defined. Their values are not objectively obtained by measurement.

The three mitigation strategies are discussed below.

1) Risk value first strategy (V strategy): It is straightforward. Risks are prioritized by their impact scales. The V strategy does not consider the time elements or efforts taken for mitigation.

2) Emergency first strategy (E strategy): We order all risks according to their teo, and then risks with an earlier teo will be treated earlier. For example, a risk with $teo=30$ will be mitigated before one with $teo=40$. This strategy attempts to mitigate the risk before it would occur.

3) Lowest effort first strategy (L strategy): We schedule all risks according to the efforts needed for mitigating the risk, then risks requiring a lower effort will be treated earlier. For example, a risk with $eff=40$ person-unit-time will be handled before one with $eff=80$. This strategy can mitigate more risks within the same time period because mitigating a risk with lower effort will use less time.

The combination of V, E and L generates three more strategies as VE, VL and EL. For example, EL is defined as from $i \cdot teo$. In this paper, we adopt a naïve approach as our goal is to discover any significant difference between strategies after the snowball effect by AR(1) rather than optimization for mitigation strategy.

B. Snowball Effects Simulation

Snowball effect is a figurative term for a process that accumulates from earlier effects or impacts and builds upon them. In this regard, the autoregressive model AR(q) [8] is adopted. The model is a stochastic process in which the variable at time $k$ depends linearly on its own previous values at $k-1$. It is a special case of the more general ARIMA(0,0,q) model of time series. It has been used to describe certain time-varying processes in nature and economics. The AR(q) model is generally defined as:

$$X_t = c + \sum_{i=1}^{p} \varphi_i X_{t-i} + \eta_t$$

where $\varphi_1, \ldots, \varphi_p$ are the parameters of the model, $c$ is a constant, and $\epsilon_t$ is white noise.

In this paper, the snowball effect to accumulate the overall impact is an AR(1) model defined as:

$$S_k = \alpha S_{k-1} + i_k - m_k,$$

where $S_k$ is the accumulated project impact at time $k$, $\alpha$ is a constant, $i_k$ is the planned impact scale of an identified risk at time $k$ and $m_k$ is an mitigation effect for $i_k$. Fig. 4 illustrates a snowball simulation with $k$ consecutive integers from 1 to 100, $\alpha=1$, $i_k \equiv \mathcal{N}(0,1)$ and $m=0$.

Figure 4. Snowball Simulation Using AR(1)

IV. PERFORMANCE OF RISK MITIGATION STRATEGIES

A. Results of Simulation

We generated 1000 projects with different risk parameters. According to the risk sets and six scheduling strategies, we establish a set of six mitigation plans for each project. The implementation was written in R which is a tool for statistical computing and provides libraries for the AR process and quantitative risk management. The implementation platform was R Cloud Workbench. The service is provided by European Bioinformatics Institute [9].
The result is given in Fig 5. We include a normalized curve, labeled “No Mitigation”, which accumulated all the planned impacts along time without mitigation or snowball effect as a baseline.

Figure 5. Normalized Impacts by Autoregressive Model AR(1)

B. Discussion

Without snowball effect, the impact range of those strategies is 0.24, calculated by (Highest – Lowest)/Lowest. This is consistent with the results of average overall impact (Zhou and Leung, 2012), in which (Highest – Lowest)/Lowest is 0.26. Although we can do the ranking in order to select better strategies, it may not be very practical as (1) many (if not all) risks are subjectively defined by a risk team and they are not objectively measured variables (2) the overall impacts vary in a range smaller than three standard deviations provides little meaning that there is a statistical significant difference between them.

Once the snowball effect is considered, two groups “E, L, VE & V” and “VL & EL” are formed. Their difference is substantial and the VL and EL can be considered as outliers. It is recommended that a risk manager should not select VL or EL.

V. Conclusion

Modeling and simulation for the purpose of software development risk management has been considered as quite limited [10]; however, this paper proposes a practical way to enhance software development risk management. Given a cyclic process of risk management in Fig 1, a risk team defines a set of detailed risks and several mitigation draft plans. The team can adopt our proposed simulation engine as a tool to estimate which mitigation plans should be taken out. The risk manager can select one of the remaining plans and then finalize the plan before execution. The approach maximizes the usage of risk information. Although risk items and mitigation plan are often subjectively defined, it is valuable and effective to reduce unpleasant surprises [11, 12]. In this paper, we are even able to quantitatively compare the overall impacts of different mitigation plans on the project so that worse plans can be eliminated. In addition, the paper illustrates that the simulation engine should include the snowball effect; otherwise, many mitigation plans will have no statistical significant difference. As the nature of risk is uncertain, such simulation engine provides little practical values to risk management process in reality. With snowball effects, mitigation strategies can vary significantly. We can be better informed by the tool and take out those mitigation plans when their overall impacts are huge.

REFERENCES