A Proactive Approach to Support Risk Management in Software Projects using Multi-agent Systems

Thayse Alencar[1], Mariela Cortés[1], Nécio Veras[2] and Lui Magno[1]

[1]*State University of Ceará, Fortaleza, Brazil*

[2]*Federal Institute of Education, Science and Technology of Ceará, Tianguá, Brazil*

Keywords: Software Project Management, Risk Management, Multi-agent Systems, MAS, Metrics.

Abstract: Software project management is a complex and demanding task full of threats or negative risks that lead to the delay or the failure of the project. Risks stem from many different internal sources as well as external ones in the company and the project. In addition, these events can originate in any phase of the project life cycle, and thereby increase the complexity of the decisions for the project manager. Aiming to reduce the negative consequences caused by these events, we propose an approach that extends a multi-agent system to provide support for risk management in software projects by using metrics and contingency reserves. The approach is evaluated with a feasibility study demonstrating that agent-oriented approaches are promising solutions that support risk management processes.

1 INTRODUCTION

One of the challenges in software development projects is the high level of uncertainty that emerges from inaccurate estimates of schedule and resources, inadequately defined scope or requirements, frequent changes in the requirements, among other reasons. In 2015, about 71% of projects have failed or had problems to meet deadlines, budget targets or quality in general (Hastie and Wojewoda, 2015). These data confirm that these uncertainties cause risks that compromise the performance and success of the achievement goals in the project.

A *risk* is an event of an uncertain condition that, if it occurs, will either have a positive or negative effect on the project objectives by: (1) threatening the *project* itself through the schedule, cost, and resources; (2) impacting the quality of the *product* that is being developed; (3) or affecting the *organization* at the business-level (Sommerville, 2011) (Pressman, 2011) (PMI, 2013). In the organization or project, risks can emerge from both internal and external sources. As a result, both of these sources and triggers, which contribute to the progress of these events, require continuous monitoring and management.

According to Sommerville (2011), risk management is recognized as one of the most important tasks of project management. Knowing how to handle risks is one of the decisive factors for project success as it can compromise goals of time, cost, quality and customer satisfaction. However, only 32% of worldwide organizations employ a formal methodology (structured by policies, procedures and forms) for risk management (PMI, 2014).

To handle the negative events threatening projects, organizations must have a proactive and consistent approach to support risk management through the entire project life cycle (PMI, 2013). In this context, intelligent agents have been held as a promising solution for supporting project management activities, due to their abilities: to detect and monitor changes in complex and highly dynamic environments; to reason about these changes; and to act proactively (Veras et al., 2015). In this paper, we propose a proactive and automated approach based on agent technology to assist the software project manager in the execution of the Risk Management processes (SEI, 2010) (PMI, 2013), regardless of the application domain.

This paper is organized as follows: the theoretical background is presented in Section 2. Section 3 contains the related works. The proposed approach for risk management is described in Section 4. The design and implementation of the multi-agent platform is detailed in Section 5. Section 6 shows some findings obtained from the implementation of our proposal. Finally, conclusions and future work are presented in Section 7.

Alencar, T., Cortés, M., Veras, N. and Magno, L.
A Proactive Approach to Support Risk Management in Software Projects using Multi-agent Systems.
DOI: 10.5220/0006703804150424
In *Proceedings of the 20th International Conference on Enterprise Information Systems (ICEIS 2018)*, pages 415-424
ISBN: 978-989-758-298-1

2 THEORETICAL BACKGROUND

2.1 Risk Management

Risk Management (RM) aims to increase the chance and impact of positive events (*opportunities*), while decreasing those of negative events (*threats*) (PMI, 2013). RM is one of the critical activities of a project manager and is important for software projects because of the inherent uncertainties faced by most projects (Sommerville, 2011). Risk management is a continuous process and should address issues that may compromise critical project objectives (SEI, 2010). A risk management strategy should be developed early in the project to detect negative events, due to the fact that introducing changes in the initial phases is more efficient than in the final phases of the project. Project Risk Management has several processes, such as: risk management planning, identification, qualitative and quantitative analysis, response planning, and control (SEI, 2010) (PMI, 2013).

The main focus of *Risk Identification* is to identify potential problems, threats and vulnerabilities that can affect the work effort or project plans (PMI, 2013). It can be performed with tools and techniques such as checklists, questionnaires and stakeholder interviews, documentation reviews, Strengths-Weaknesses-Opportunities-Threats analysis (SWOT), Failure Mode and Effect Analysis (FMEA), expert opinions, brainstorming, etc. *Qualitative Analysis* evaluates the *priority* or *exposure* to the individual risk for each identified risk based on the comparison parameters (PMI, 2013). The most common risk comparison parameters are: *probability occurrence* and *impact* (or seriousness) of the occurrence. The *Quantitative Analysis* involves analyzing the aggregate effect of all risks in the project in order to provide a correct understanding of the flaws, outcomes and events that are difficult to explain in a qualitative approach. The *Planning of Responses* handles the selection strategies or merging of risk-taking strategies by considering their specific probability and impact. There are strategies normally used to treat both negative (prevent, transfer, mitigate and accept) and positive risks (exploit, improve, share, and accept) in projects. Finally, *Risk Control* involves continuous and periodic support of the risk factors, by performing all processes previously reported. They include several activities: implementation of risk response plans, tracking identified risks, monitoring residual risks, identifying new risks, and evaluating the effectiveness of the risk management process throughout the project.

These processes are iterative and require many steps of planning. Even with comprehensive consideration for planning, hardly any of the risks will be eliminated from the project. In this context there are several factors that contribute to the complexity of these processes, among them being: the occurrence of negative events at any step of the project; multiple sources of risk in projects; and the need for specialized knowledge to conduct the processes. Perhaps, for this reason, many organizations neglect some of these processes, hence jeopardizing the success of the projects. In the current state of the area, we consider finding solutions that support RM in specific contexts or apply restrictions. Among the solutions found in the literature, many offer partial support by excluding several RM processes (Fontoura and Price, 2008), while some are theoretical (Rafele et al., 2005) or semi-automated (Knob et al., 2006), and others require greater efforts from the manager or were developed for a specific context (Rad, 2013).

2.2 Agents and Multi-agent Systems

Agent-based technology has been applied increasingly by the scientific community to develop complex computational systems (Jennings, 2001). An agent is an entity able to *perceive* its environment through *sensors*, and acts on it through its *actuators* (Russell and Norvig, 2013). Agents can be biological (e.g., people or animals), robotic or computational, and can perform several tasks, usually to support some human individual (Coppin, 2004). Agent perception refers to perceptual inputs acquired from the environment at an instant, while the actuation refers to the agent's response to the stimulus received. This idea is presented in Figure 1.

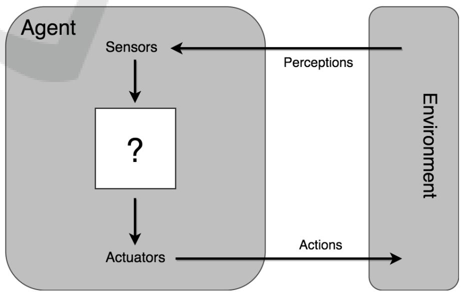

Figure 1: A standard agent illustration adapted from (Russell and Norvig, 2013).

Artificial Intelligence (AI) is the field of science and engineering that attempts to understand and construct intelligent entities (Russell and Norvig, 2013). In this context, agents are different software programs in comparison to other computational programs, due to features such as: *autonomy*, the ability to perceive the environment in which they are inserted; *persistence* during a specific length of time; *adaptation* to change; and *proactivity* to create and reach goals (Russell and Norvig, 2013).

Another concept assigned to agent technology is *rationality*. A rational agent is able to act in order to achieve the best result or, when there is uncertainty, the best expected result (Russell and Norvig, 2013). Intelligent agents should not only collect information, but should also learn as much as possible from environment. This learning allows agents to improve their performance in executing a specific task over the time (Coppin, 2004). Moreover, agents have the ability to learn from other agents — as the case of multi-agent systems (MAS). When there is more than one agent collaborating or disputing with each other in the same environment, this is called *multi-agent system*.

Achieving a perfect rationality (doing the right thing) is not simple in complex environments (Russell and Norvig, 2013). Agents are problem solvers, capable of balancing flexible behavior to pursue their project objectives, acting both *reactively* (able to respond to changes occurring in the environment) and *proactively* (able to adopt targets and take initiatives) (Jennings, 2001). For this reason, agent-based solutions are suitable for dynamic and complex environments, such as software project management and development.

In complex problems of Software Engineering, the adoption of an agent-oriented approach utilizes the decomposition of the problem into multiple sub-problems, as well as autonomous components (agents) that act and interact in a flexible way to achieve established objectives (problem resolution) (Jennings, 2001). For instance, in the field of Software Engineering, agents can be used to perform tasks relevant to processes or people. Similar to subjects who have tasks allocated in a process, agents can perform delegated actions for them.

3 RELATED WORK

In order to provide an overview of the methodologies and solutions that support project management, while focusing on Risk Management, a literature review was conducted. Our current research covers topics of how organizations conduct management, which processes of risk management are more applicable, which tools and techniques are utilized, and what limitations are encountered the most.

The usage of multi-agent systems supporting the risk management can be observed in some works. Nienaber (2008) proposes a *framework* to support all areas of project management proposed by (PMI, 2013), by using specialized software agent technology, where each agent is responsible for a general task. The architecture of the *framework* integrates various multi-agent systems, and each of these systems is responsible for the processes of a management area, while fulfilling a specific objective. The proposed approach was implemented in the prototype containing only the multi-agent module for risk management and including only the processes of Qualitative Analysis and Risk Monitoring. The technique of *Probability and Impact Matrix* in its most simplistic version is applied in the qualitative analysis. The tool is not available to the scientific community.

Another multi-agent platform for project management was developed by Veras et al. (2015) , which provides the monitoring and control of the project work and the integrated changes management. The approach applies the Aggregate Asset Management (AAM) and the Critical Path Method (CPM) (PMI, 2013) to provide a consistent view of project progress, and assist the project manager in decision making. The agents are able to detect deviations during the execution of project's activities and suggest corrective actions to reduce the negative impact of deviations. The approach does not provide support for all the project management processes (PMI, 2013) (SEI, 2010) and is not available for the scientific community.

Focusing specifically on risk management, Shah (2004) suggests the association of qualitative and quantitative approaches to identify risks in complex systems. Based on the combination of different theoretical *frameworks* for risk management and the (*Probabilistic Risk Assessment Model - PRA*) evaluation model, the approach is able to identify and quantify risk factors in complex systems. In addition, the approach supports only three risk management processes (Identification, Qualitative and Quantitative Analysis) and bases on prioritizing risks to allocate resources, aiming at reducing costs and mitigating risks. The approach lacks the implementation of an automated tool as a proof of concept.

Rafael et al. (2005) also present another method for risk management in projects utilizing a *Risk Breakdown Matrix* - RBM, comprised of the *Work Breakdown Structure* - WBS and the *Risk Breakdown Structure* - RBS. This strategy is useful for associating risks with the activities of a project. While the WBS defines the activities and packages of work in the project, RBS identifies possible sources of risk, enabling the approach to perform a more robust analysis than the simple *Probability and Impact Matrix*. The approach includes three risk management processes (Identification, Qualitative and Quantitative Analysis), but it does not offer any automated support tools.

Intended as a support to all risk management processes (PMI, 2013)(SEI, 2010), the semi-automated tool proposed by Knob et al. (2006), *RiskFree*, aims to help software development project teams to collaboratively manage risks in their projects. Due to the fact that processes can be executed by various techniques, the tool *RiskFree* is designed to allow organizations to develop components that meet their own needs. In such approach, the only process that is truly semi-automated is the Qualitative Analysis performed through the *Probability and Impact Matrix* technique, denoting that the manager or person in charge is responsible for the manual execution of the other processes.

In a technical report, Rad (2013) describes the *GOES-R Series RM*, a decision-making tool used to ensure safety and functionality of the *Geostationary Operational Environmental Satellite* - GOES system. The *GOES-R Series RM* has the risks in a *Risk Distribution Matrix* — a more robust version of the *Probability and Impact Matrix* — and positions them accordingly with the value of their risk exposure (*RE*). When some significant change occurs in the project, the affected risks are updated and repositioned in the matrix. The tool provides reports, suggestions for mitigation actions and supports all risk management processes (PMI, 2013) (SEI, 2010). Although this tool is developed privately for a specific domain, the theoretical approach of this work can be adapted to other application domains.

The use of project metrics can also be observed as a technique for supporting risk management processes. Fontoura et al. (2004) proposed an approach for risk prevention based on the customization of the organization's software process. The approach is oriented as defined metrics from the Goal/Question/Metric paradigm, and supports the Identification, Qualitative Analysis, Response Planning, and Risk Control processes. Considering the previously cited works, the approach also uses the *Probability and Impact Matrix* technique in its most simplistic version to calculate the effect of a risk. In an extended version of this work, Fontoura and Price (2008) presented a tool that implements such an approach, but the tool is not available online.

The use of agents in the context of software management projects, in particular, is a relatively new field of research, and as such the literature is not widely available. Based on the analysis of the works, RBS and the *Impact and Probability Matrix* are structures commonly used in the detection and evaluation of risks. This is justified by the fact that RBS provides the visualization of complex projects and systems in smaller segments; and the *Probability and Impact Matrix* is a simple, quick, and inexpensive way to obtain the critical level of each risk. However, both techniques are static strategies, applied at specific times of the project. Another observation is the simplicity and the limitation of the mathematical formulations that calculate the risk exposure (ER), which exclude project or organization factors contributing to the criticality of the risks. Finally, parameters that confirm the obtained results are lacking in the majority of works, since many of them do not contain an automated support tool. In the next section, a proactive approach to risk management in software projects will be discussed in detail.

4 PROPOSED APPROACH

Aiming to assist software project managers with the Risk Management Process (RM) as suggested by the PMI (2013) and SEI (2010), we propose the development of an intelligent agent to treat risks (ARis) in an integrated way with diverse aspects of the project such as scope, schedule, cost, and changes management. To perform a robust analysis of the project's risks, the mathematical formulation developed in our approach takes into account these parameters: (i) the impact of each risk for the various project aspects (cost, schedule, scope and others); (ii) requested changes in the project; and (iii) the amount of available contingency reserve.

The proposed approach is divided into four macro-process — *Risk Analysis*, *Simulation Environment*, *Updating Environment* and *Monitoring Project Metrics* — which are performed by the risk agent according to the current state of the project environment. Figure 2 shows the execution flow diagram of these processes implemented by the risk agent ARis in the approach.

At the onset of the project, the properly identified and documented risks as explained in Section 2 are incorporated into the internal state of ARis. Once the existence of risk factors jeopardizing the project is detected, the agent executes the process *Risk Analysis*, which consists of calculating the priority of each risk factor and updating its internal state. The occurrence of changes must be predicted during a project, but only formally approved change requests can be incorporated into the project's baseline (PMI, 2013). To be approved, a change request needs to be evaluated because it might result in one or more modifications in the project attributes. In this scenario, according to the approach proposed here, whenever the agent ARis is notified of any change request, it performs the *Simulation Environment* process to simulate

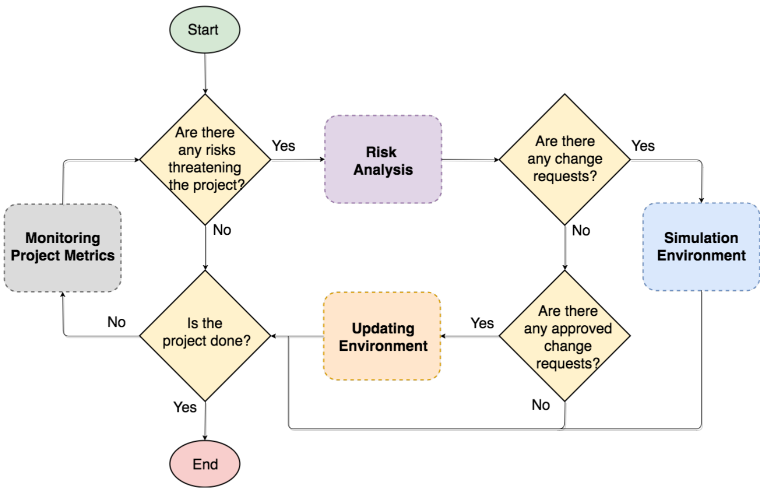

Figure 2: The flow diagram of the macro-processes performed by the ARis agent.

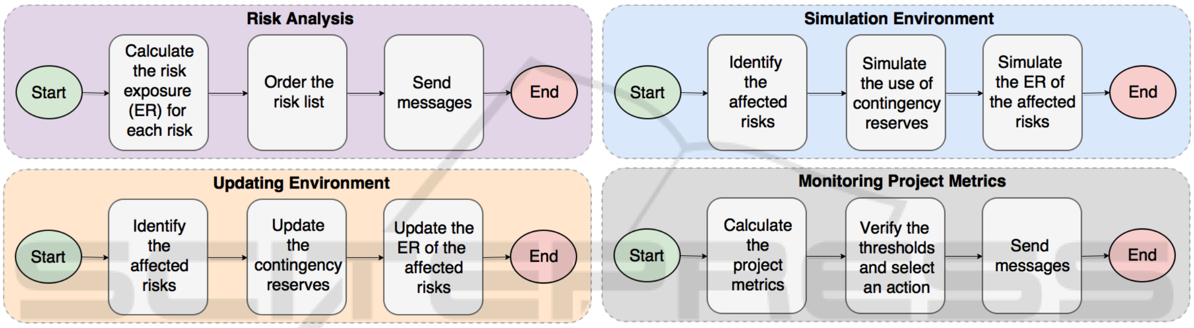

Figure 3: Macro-processes in-depth.

the new state of the environment from the application of the change. If the change is approved by the manager, then ARis executes the *Updating Environment* process to update the information about the project's risks in its internal state.

Due to the interactive nature of project management, the macro-process proposed in this approach overlap and interact in different ways. The union of all these processes reflect the Risk Management process proposed in the literature that can be seen in Section 2. More details about each macro-process and their mathematical modeling are discussed in the following sections.

4.1 Approach Processes

4.1.1 Risk Analysis

The qualitative risk analysis process explained in Section 2 is executed in this approach through the *Risk Analysis* illustrated in Figure 3. Let $LR = \{r_1, r_2, r_3, ..., r_N\}$ be the set of N risks that endanger a project P and let $CI = \{c_1, c_2, c_3, ..., c_M\}$ be the set of M attributes of the project affected by risks. Considering that each $r_i \in LR$ is able to simultaneously affect more than one attribute $c_j \in CI$, each risk r_i therefore has probability and impact values associated with each attribute c_j, respectively $P_{i,j}$ and $I_{i,j}$. Thus the estimated risk exposure RE for each risk r_i is given by Equation 1:

$$RE_{r_i} = \sum_{j=1}^{M} P_{i,j} * I_{i,j}, \qquad (1)$$

where RE_{r_i} = total risk effect i; $P_{i,j}$ = risk probability i in an attribute $j \in CI$ and $I_{i,j}$ = risk impact i in an attribute $j \in CI$.

The impact of each risk is measured from a scale of 1-5, where 1 is very low and 5 is very high. The probability is defined in percentage values from 0% to 100%, representing the occurrence chances of the project events. An example of the application of Equation 1 for a set of risks LR is given by Table 1. In this example, the set CI is composed of the attributes Cost, Time, and Scope. The stored values in these columns of the table represent the risk impact in the attributes of the project. The columns labeled PC, PT and PE store the probability values of the risk in the same attributes of cost, time, and scope — in other words, the probability of a risk affects the attributes of the project. The row labeled *TOTAL RE BY AREA* store the total value of risk exposures by area of the project. The arrangement and analysis of the risks in this approach uses a *Risk Breakdown Structure* combined with a *Probability and Impact Matrix* throughout the project life cycle.

Considering the number base values in Table 1, some of the conclusions derived are: (i) identify what project areas/attributes are most vulnerable to risks (in this case it is *Time* with 5.2 total risk exposure); (ii) identify the most important risk, i.e. the highest *RE* value (in this case it is $R3$); (iii) identify the most significant relationships (in the example, the relation between *Cost* and Incorporation of a new technology).

4.1.2 Simulation and Updating the Environment

According to PMI (2013) , a contingency reserve should be designated for both known and unknown risks, which can not be managed proactively. In the RM, project contingency reserves are used to respond to risks or to apply mitigation actions, possibly leading to changes in the project. The current approach uses the contingency reserve of time and cost both for the treatment of risks and possible changes in the project baseline requested by stakeholders. The amount of contingency reserve should be defined by the manager considering the size of the project, contract terms and the profile of the organization of the clients or investors. In this approach, the time and cost reserves of a project P are determined by:

$$RC_C = x\% * CP, \tag{2a}$$

$$RC_T = y\% * TP, \tag{2b}$$

where RC_C = Reserve of the Cost Contingency; CP = Total Cost of the Project; RC_T = Reserve of the Time Contingency; TP = Total Time of the Project; and x/y = Percentage established for reserves.

In Figure 3, the *Simulation Environment* process measures the evolution of the risks based on the usage of the contingency reserve. Since a decrease in this reserve implies a reduction in the resources needed to apply changes or handle threats, the use of contingency reserves for change requests has a direct impact on the progress of the project's risks. In other words, the smaller the amount of available reserves, the more critical the treatment of risks is. For this reason, the strategy of simulating change impacts in the project environment, before they are approved, assists the manager in his/her decision making and provides clear and concise visualization of risk progress; hence saving time and money, and simplifying the project execution of risk management processes in the organization.

The *Simulation Environment* process is executed every time that a change is solicited in an activity of the project. Let $A = \{a_1, a_2, a_3, ..., a_n\}$ be the set of N activities of the project P, an activity $a_i \in A$ is represented by a tuple (*id, title, estimatedTime, actualTime, estimatedCost, actualCost, estimatedScope, actualScope*). Thus a change in any activity $a_i \in A$ can affect its *estimatedTime*, *estimatedCost*, *estimatedScope*. In this approach, the variation in cost (VC_i) and time (VT_i) are defined by Equations 3a and 3b. Regarding variation in scope or any other attributes of the project, they are all mapped to cost and time variations.

$$VC_i = AC_i - EC_i, \tag{3a}$$

$$VT_i = AT_i - ET_i, \tag{3b}$$

where VC_i = Variation of cost in activity i; AC_i = Actual cost of activity i; EC_i = Estimated cost of activity i; VT_i = Variation of time in activity i; AT_i = Actual time of activity i; and ET_i = Estimated time of activity i.

As stated earlier, the act of changing the project baseline may require the use of contingency reserves of time or cost. It is important to emphasize that the time reserves are only used for time changes in activities of the *critical path* in the project, hence these are able to impact the total duration of the project. The calculation of the amount of contingency reserves used for variations in the project P, at the same instant t, is given by:

$$URC_C = \frac{\sum_{i=1}^{k} VC_i}{RC_C}, \tag{4a}$$

$$URC_T = \frac{\sum_{i=1}^{k} VT_i}{RC_T}, \tag{4b}$$

where k = number of change requests at an instant t; URC_C = Use of cost contingency reserve; URC_T = Use of time contingency reserve.

Due to the direct impact on the progress of the project's risks, the amount of available contingency reserve increases or decreases the progress of the risks proportionally. Consequently, updating the progress of the affected risks requires updating their probability values. In this approach, the updating of the risk probability, i.e. the chance of affecting the attributes (time or cost) of the project is given by:

$$PC'_i = \begin{cases} URC_C * (1 - PC_i) + PC_i, & \text{if } URC_C > 0 \\ URC_C * PC_i + PC_i, & \text{otherwise.} \end{cases} \tag{5a}$$

Table 1: Risk Exposure Matrix of Equation 1.

	Cost	Time	Scope	PC	PT	PE	RE	Classification
R1. Definition of Scope			IE = 4	-	-	30%	1.2	5
R2. Misunderstanding of the requisites		IT = 5		-	50%	-	2.5	3
R3. Incorporation of a new technology	IC = 5			70%	-	-	3.5	1
R4. Unrealistic schedule		IT = 3	IE = 1	-	90%	40%	3.1	2
R5. Unrealistic budget	IC = 4		IE = 1	30%	-	50%	1.7	4
TOTAL RE BY AREA	4.7	5.2	2.1					

$$PT_i' = \begin{cases} URC_T * (1 - PT_i) + PT_i, & \text{if } URC_C > 0 \\ URC_T * PT_i + PT_i, & \text{otherwise.} \end{cases} \tag{5b}$$

PC_i' and PT_i' are new probabilities of risk i affecting the cost and time of the projects, respectively. PC_i and PT_i are the initial probabilities of risk i affecting the the cost and time, respectively.

After updating the probabilities of the affected risks in the *Simulation Environment*, the agent recalculates the *RE* value of the same risks to close the simulation scenario and send a set of messages to the change requestor. The requestor, in turn, can analyze the scenario information and decide whether or not to approve the change in the project. After the *Simulation Environment*, the ARis agent verifies for approved changes to initiate the *Updating Environment* process shown in Figure 3. The environment update consists of changing the project environment variables by applying what was shown in the simulation scenario, as well as updating the internal state of the ARis agent. The process of controlling risks (see Section 2) is executed in this approach by the combination of the *Simulation Environment*, *Updating Environment* and *Monitoring Project Metrics* processes.

4.1.3 Monitoring Project Metrics

Soon after the processes of simulation or updating the environment, a set of defined metrics using the Goal/Question/Metric paradigm (extracted and adapted from the work of (Fontoura et al., 2004)) is calculated by the ARis agent in order to identify new risks or sources of risks. ARis' decision-making subsystem computes the metrics and information for its internal state, while considering a set of condition-action rules and metric thresholds. Such actions include alert messages to the manager, prediction of new risks, suggestions or preventive/corrective actions that should be applied to the project, etc. In this approach, metrics are used for both triggering condition-action rules and assisting the manager to project the probability of risks in future projects. The flow is illustrated in Figure 3 and one example of the metric used by the ARis agent is shown in Table 2.

Table 2: GQM metric example.

Goal 1	To evaluate the ability of workers on a team based on the perspective of the project manager
Question 1	What percentage of qualified workers perform their role on a team?
Metric 1.1	Percentage of qualified workers PQW = (number of qualified workers / number of workers on team) * 100

5 THE ARis AGENT AND THE MULTI-AGENT PLATFORM

The implementation of the approach comes from the development and integration of the ARis agent into the MAS proposed by (Veras et al., 2015). Originally, the platform includes three agents (AMon, ACon and AMud) that provides the monitoring and control of the project work and the integrated changes management through environment simulations of the project. Figure 4 illustrates the interactions between the agents and the components of the platform. During the execution, ARis perceives the evolution of the project environment and exchanges messages with the AMud agent, supervising the progress of the risks in the environment. In the referred illustration, the arrows indicate what actions the agents perform and what information the components send and receive.

AMon is a model-based reflex agent (Russell and Norvig, 2013) responsible for monitoring the “Project Environment” with the goal of verifying differences between the current and planned project performance. This agent incorporates a set of condition-action rules based on the theory of AAM to verify, in real time, the progress of the project in relation to the cost and schedule. According to the obtained indicators, this agent is able to detect deviations and send alerts to the manager.

ACon is a model-based reflex agent (Russell and Norvig, 2013) in charge of the integrated control process proposed by (PMI, 2013) and obtains the information from AMon. ACon consists of a set of condition-action rules that allows it to suggest preventive/corrective actions to the manager in order to reduce detected deviations. When the suggested acti-

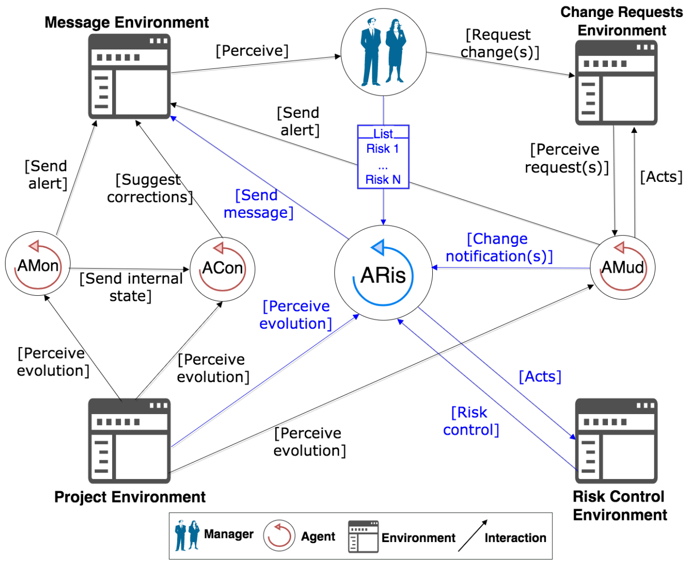

Figure 4: ARis and the multi-agent platform.

ons are taken, this agent generates new estimations based on the compensations of cost or time, and begins to monitor and control the project activities with the new incoming information from the new consolidated plan.

AMud is a simple-reflex agent (Russell and Norvig, 2013) responsible for monitoring and controlling change requests registered in the "Change Request Environment". Through the SUT (Severity, Urgency and Trend) Matrix, AMud calculates the priority of the change requests. The severity of a change represents the impact in cost (IC) and time (IT) of the project. The urgency (U) represents the available time to insert the change in the project. Thus, the priority (P) of a change is given by the equation P = ICxITxU.

ARis, the new agent, is a model-based reflex agent (Russell and Norvig, 2013) responsible for the risk management of the project and acts based on the information obtained by its perceptions of the "Project Environment" and a set of condition-action rules. Moreover, ARis exchanges messages with AMud, calculates projects metrics, and performs the macro-processes of the proposed approach described in Section 4.1. This agent aims at contributing to the comprehensiveness of the multi-agent platform, by providing a robust analysis of the change requests managed by AMud and assisting the project manager in decision-making.

The multi-agent platform proposed by (Veras et al., 2015) uses the *JaCa Programming Model* [1] where *Jason* is adopted as the programming language to implement and execute agents that work and cooperate inside common environments; and *CArtAgo*, as the framework to program and execute these environments. As a result, in the JaCa model, the *agents* are programmed on one side, encapsulating the tasks logic control that must be executed; and on the other side, the *environment*, providing a first-class abstraction to the actions and functionalities exploited by the agents. As seen in Figure 4, each agent has access to at least one environment to use or observe its resources.

[1] http://jacamo.sourceforge.net/?page_id=40

6 FINDINGS

Aiming at performing a feasibility study of the proposed approach, we developed the risk agent ARis fully implemented with all the macro-processes except the *Updating Environment*. To run the experiments, six fictitious project scenarios were created. The scenario explored in this section is comprised of nine activities (A-I), totaling 160 time units until completion and a total budget of 7000 money units. Due to space restrictions, the details about the project's activities (pre-requisite, cost, duration and others) were omitted. The list of risks threatening this project can be seen in Table 1. The project manager had established 48% and 30% for cost and time contingency reserves respectively, translating to 3360 extra cost units and 48 extra time units. The manager is represented by another agent *[manager]* and through the execution of this scenario various events happen, such as time and cost change requests and the emergence of new risks in the project.

Initially, when the scenario is loaded in the MAS, all the agents observe the "Project Environment" to obtain partial information about their world — in this case, the running project (see Figure 5) — and afterwards all agents start their respective tasks at the same time. ARis perceives the existence of five risks (originally registered by the manager) threatening the project, and it initiates the process *Risk Analysis* continuously until it receives a notification of a change request or observes a metric reaching a certain threshold. At the instant 40, ARis receives a notification of a change request made by the manager for activity I, requiring an increase of 11.9% in cost and 11.8% in time. Upon being notified, ARis begins executing the macro-process *Simulation Environment* and sends messages to the manager. The set of information shown in Figure 6 represents the project's future state based on the application of the change such as: (i) variation in the cost and/or time of activity I, (ii) the new cost and/or new time of activity I, (iii) the available amount of contingency reserves, and (iv) the list of affected risks including the new probabilities and *RE* values.

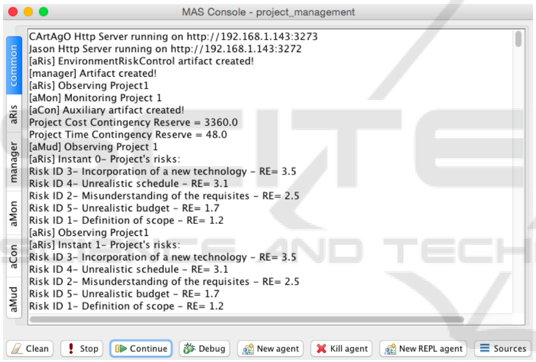

Figure 5: ARis executes the macro-process Risk Analysis.

Following the execution of the macro-process *Simulation Environment* before the completion of the project, ARis initiates the macro-process *Monitoring Project Metrics* as shown in Figure 7. The metrics calculated by ARis in the current version of the MAS are show in Table 3. Each one of these metrics has its acceptance interval. Let $B = \{m_1, m_2, m_3, \ldots, m_n\}$ be the set of N metrics of the project P. The acceptance interval of each metric is given by $[a, b] = \{m_i \in \mathbb{R} \,|\, a \leq m_i \leq b\}$, in which a and b are values between 0 and 1, previously defined by the project manager. The relation between metrics and risks is given by 6a:

$$m_i = \begin{cases} \text{no risk detected,} & \text{if } m_i > b \\ \text{a new risk detected,} & \text{if } a \leq m_i \leq b \\ \text{risk occurred,} & \text{if } m_i < a \end{cases} \tag{6a}$$

Our fictitious scenario simulates a lack of qualified workers, representing a new risk in the project. By qualified workers we refer to team members who have the required skills for the project. Meanwhile at instant 40, ARis detects a $\leq$ m6 $\leq$ b leading to the identification of this risk shown in Figure 7. ARis sends warnings to the manager and suggests corrective actions to reduce the probability of the negative event. Afterwards at instant 41, the routine *Risk Analysis* is re-executed and the new risk is included in the process; now the project's risk list contains 6 instead of 5 elements. During the remainder of the scenario execution, more changes will be requested, the agents will continue to trade information and execute their roles, as well as the macro-processes of this approach.

Table 3: Metrics for the approach.

Percentage of Cost Changes: m1 = 1 - (NumberOfCostChanges / TotalNumberOfChanges)
Percentage of Available Cost Reserve: m2 = CurrentCostReserve / TotalCostReserve
Percentage of Time Changes: m3 = 1 - (NumberOfTimeChanges / TotalNumberOfChanges)
Percentage of Available Time Reserve: m4 = CurrentTimeReserve / TotalTimeReserve
Percentage of Scope Changes: m5 = 1 - (NumberOfScopeChanges / TotalNumberOfChanges)
Percentage of Qualified Workers: m6 = NumberOfQualifiedWorkers / TotalNumberOfWorkers

Based on our findings by executing fictitious scenarios, we conclude that ARis provides an anticipated and efficient view of the project's future state that aids the project manager into his/her decision making process; moreover, the metrics are effective support for predicting new risks at no extra cost. Based on these results, we have confirmed our hypothesis that agent technology contributes to automated project management, mainly in proactive risk and change management. In conclusion, we claim that agent-oriented approaches are promising solutions that support the risk management processes regardless of the application context. To improve our findings, the macro-process *Updating Environment* will be included in the release of the next version of this tool, which will be accessible online for the scientific community.

7 CONCLUSIONS AND FUTURE WORK

In this paper, we present a proactive and automated approach based on agent technology to assist the software project manager in the execution of the Risk Management processes, as well as the decision-making throughout the project life cycle (SEI, 2010) (PMI,

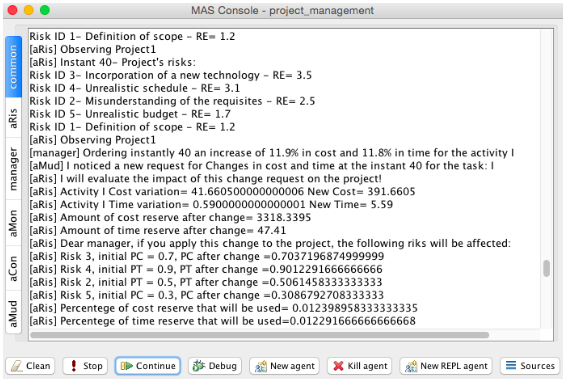

Figure 6: ARis executes the macro-process Simulation Environment.

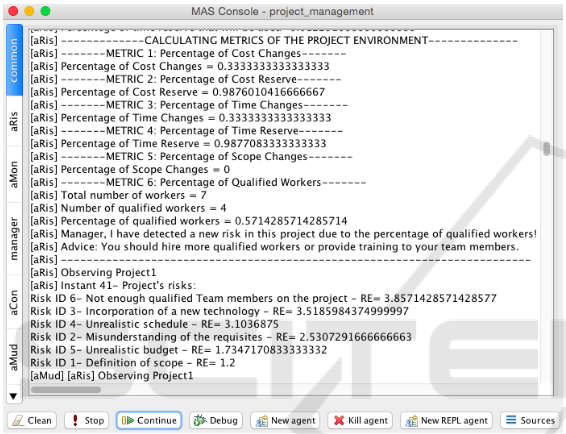

Figure 7: ARis executes the macro-process Monitoring Project Metrics.

2013). The approach provides support in identification, analysis, planning of responses and controlling of risks in an integrated and sensitive manner that responds to changes in the project environment. The proposed approach is based on simulations of the project environment, and its processes are triggered by change requests and metrics, revealing the progress of the project's risks. The approach incorporates a rich mathematical formulation that considers the usage of contingency reserve of the project to calculate probability and risk exposure.

As a result, the agent-oriented approach has proven to be a promising solution for supporting risk management regardless of the application context. Moreover, the approach provides evidence to monitor the risks throughout the project life cycle at no cost for the organization, and intends to improve and control the project risk indices. The MAS proposed by Veras et al. (2015) has been extended with the integration of the ARis agent, but it is currently still being developed for the completeness of the macro-process *Updating Environment*. Aiming at predicting a wider risk spectrum, a more in-depth group of metrics will be included in the release of the next version.

REFERENCES

Coppin, B. (2004). *Artificial intelligence illuminated*. Jones & Bartlett Learning.

Fontoura, L. and Price, R. (2008). Systematic approach to risk management in software projects through process tailoring. In *SEKE*, pages 179–184.

Fontoura, L., Price, R., and Phil, D. (2004). Usando gqm para gerenciar riscos em projetos de software. In *Anais do XVIII Brazilian Symposium on Software Engineering*, pages 39–54. Brazilian Symposium on Software Engineering (SBES).

Hastie, S. and Wojewoda, S. (2015). The 2015 chaos report. Technical report.

Jennings, N. (2001). An agent-based approach for building complex software systems. *Communications of the ACM*, 44(4):35–41.

Knob, F., Silveira, F., Orth, A., and Prikladnicki, R. (2006). Riskfree: Uma ferramenta de gerenciamento de riscos baseada no pmbok e aderente ao cmmi. *V Brazilian Symposium on Software Quality (SBQS)*, pages 203–217.

Nienaber, R. (2008). *A Model for Enhancing Software Project Management Using Software Agent Technology*. PhD thesis, University of South Africa (South Africa). AAI0821180.

PMI (2013). *A Guide to the Project Management Body of Knowledge: PMBOK Guide*. Project Management Institute, Inc., 14 Campus Boulevard Newtown Square, Pennsylvania 19073-3299, Estados Unidos.

PMI (2014). Pmsurvey.org 2014 edition. Technical report, Project Management Institute.

Pressman, R. (2011). *Software Engineering: A Practitioner's Approach, 7th Edition*. The McGraw Hill.

Rad, A. (2013). Goes-r series risk management plan. Technical report, DOC, NOAA, NESDIS, NASA.

Rafele, C., Hillson, D., and Grimaldi, S. (2005). Understanding project risk exposure using the two-dimensional risk breakdown matrix. Technical report, Project Management Institute.

Russell, S. and Norvig, P. (2013). *Artificial Intelligence: A Modern Approach, 3rd Edition*. Pearson.

SEI (2010). *CMMI for Development, Version 1.3*. Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA.

Shah, J. (2004). Probabilistic risk assessment method for prioritization of risk factors. Master's thesis, Master of Science in Industrial Engineering. Louisiana State University and Agricultural and Mechanical College.

Sommerville, I. (2011). *Software Engineering, 9th Edition*. Pearson.

Veras, N., Cortés, M., Queiroz, A., and Souza, L. (2015). Abordagem proativa para a gestão integrada dos trabalhos de projeto. *Theoretical and Applied Computing Journal (RITA)*, 22(2):166–180.