

Arquitetura para Monitoramento de Riscos usando Agentes de *Software*

Giliane Bernardi¹, Lisandra Manzoni Fontoura^{1,2}

¹Área de Ciências Naturais e Tecnológicas - UNIFRA - Santa Maria – RS - Brasil

² Universidades Regionais Integradas – URI – Santiago – RS - Brasil

{giliane, lisandramf}@unifra.br

Resumo. *Esse artigo propõe uma arquitetura para controle de projetos de software utilizando a tecnologia de agentes, mais especificamente o monitoramento de cenários de riscos. Os agentes são usados para acompanhar o progresso dos fatores de risco, possibilitando ao gerente de projeto tomar ações corretivas, quando necessário e no momento adequado.*

Abstract. *This paper proposes an architecture to software projects control using the agents technology to monitor risks scenarios. The agents are used to keep up with the risk factors progress, making possible to the project's manager take corrective actions, if necessary and on the right moment.*

1. Introdução

Considera-se um projeto bem-sucedido aquele que satisfaz as necessidades do cliente e é entregue dentro do prazo, com custo e qualidade estimados (PMI, 2000). Segundo estudos do *Standish Group* (2004), em 2004, 29% dos projetos foram bem-sucedidos, 18% foram cancelados e 53% foram entregues fora do prazo, com custo acima do estimado, ou com menos funcionalidades que as inicialmente especificadas. O gerenciamento sistemático de riscos pode evitar esse caos. Por meio do acompanhamento dos fatores de riscos, o gerente de projeto pode tomar ações preventivas e corretivas durante o projeto, evitando comprometer prazo, qualidade, custo e funcionalidades.

A Gerência de Riscos de *Software* (GRS) é uma abordagem que busca organizar o enfoque dos riscos correlatos ao desenvolvimento bem-sucedido de sistemas de *software* em um conjunto de princípios e técnicas para analisar, preparar ações preventivas e controlar os riscos de projetos de software (Addison, 2002). Entre as atividades para gerenciamento de riscos, encontra-se o monitoramento dos mesmos, alvo deste trabalho. O monitoramento dos riscos é fundamental para que os mesmos sejam mantidos sob controle, sendo que possibilita ao gerente manter o controle sobre os projetos e identificar o momento em que ações de

resolução de riscos devem ser tomadas. Este artigo propõe o desenvolvimento de uma arquitetura para monitoramento de cenários de riscos, usando agentes de *software*. O monitoramento de cenários de riscos provê mecanismos para antecipar situações de perigo de forma que ações preventivas sejam tomadas antes da materialização dos riscos. Para a modelagem e especificação desta arquitetura será considerada a abordagem de Engenharia de *Software* Orientada a Agentes.

2. Gerência de Riscos de *Software*

Os principais riscos envolvidos podem ser classificados em riscos de *Cliente*, de *Requisitos*, de *Planejamento* e de *Execução*. Neste artigo serão usados como exemplo os *Riscos de Requisitos*, que descrevem preocupações quanto ao entendimento das necessidades do cliente, tais como: definição inadequada de escopo, definições ambíguas ou inexistentes dos requisitos, entre outras. Após a identificação dos riscos que podem afetar o projeto é necessário definir as métricas que serão utilizadas para seu monitoramento. A tabela 1 apresenta exemplos de métricas que são usadas para monitoramento de riscos de requisitos.

Tabela 1. Exemplo de Métricas para Monitoramento de Riscos

Risco: Não-entendimento dos requisitos e instabilidade de requisitos	
<i>Métrica 1a:</i> Percentual de validação de requisitos pelo Cliente	
PVRC = (número de documentos de requisitos validados/ número total de documentos de requisitos) * 100	
<i>Métrica 1b:</i> Percentual de requisitos elaborados pelo Cliente	
PRE = (número de requisitos elaborados pelos usuários/requisitos elaborados pelo pessoal de sistema) * 100	
Risco 2: Escopo/objetivos não-claros e diferentes expectativas sobre o sistema entre distintos usuários	
<i>Métrica 2a:</i> Número de solicitações de mudança (comparar com dados históricos)	
NSM = Número de solicitações de mudança de escopo/objetivos	
<i>Métrica 2b:</i> Percentual de mudanças realizadas no prazo	
PMRP = (número de mudanças realizadas no prazo / número de mudanças solicitadas) *100	

Tabela 2. Associação de Atividades a Métricas

Riscos de Requisitos	Atividades Monitoradas				
	Review Requirements	Elicit Stakeholders Request	Submit Change Request	Manage Acceptance Tests	Update Change
<i>Não-entendimento dos requisitos e requisitos instáveis</i>					
Métrica 1a: Percentual de validação de requisitos pelo Cliente	X				
<i>Métrica 1b:</i> Percentual de requisitos elaborados pelo Cliente		X			
<i>Escopo/objetivos não claros e conflito entre os departamentos do usuário</i>					

Métrica 2a: Número de solicitações de mudança			X		
Métrica 2b: Percentual de mudanças realizadas no prazo					X

As métricas precisam estar associadas às atividades de processo que estão sendo utilizadas no projeto para que sejam coletadas durante a execução dessas atividades. No caso deste trabalho, o processo de *software* considerado é o Processo Unificado *Rational* (RUP). A tabela 2 mostra a associação das métricas às atividades propostas pelo RUP que precisam ser monitoradas para acompanhamento dos fatores de risco.

3. Proposta de Arquitetura para Monitoramento de Riscos

Para Collier (2002), o cenário da Engenharia de *Software* mudou nos últimos anos devido ao aumento da complexidade dos *softwares* que vêm sendo desenvolvidos. Para o autor, dois aspectos podem ser citados como fatores primordiais que influenciaram tal mudança: a complexidade das tarefas e a integração de numerosos componentes ou sistemas para a realização de tarefas. Neste contexto, emerge a Engenharia de Software Orientada a Agentes, uma abordagem oriunda da área de Inteligência Artificial, que utiliza o conceito de agentes de *software*, ditos “inteligentes”, e os conceitos da orientação a objetos, onde agentes “encapsulam” uma estrutura e um comportamento associados a si, porém, um comportamento dinâmico e adaptável, de acordo com as experiências e decisões a serem tomadas. Um conjunto de benefícios pode ser obtido no desenvolvimento de sistemas dentro da perspectiva de orientação a agentes, sendo que (Collier, 2002) destaca alguns para a Engenharia de *Software*: disponibilidade de diversos instrumentos de impacto para tomada de decisão; simplificação de sistemas distribuídos complexos através da abstração; suporte para padrões flexíveis de interação; suporte à criação de aplicações abertas; e robustez das aplicações.

Para o efetivo monitoramento dos riscos em projetos de *software* identificou-se que é necessário que os agentes tenham as seguintes capacidades: obter informação sobre o projeto; monitorar atividades do processo de *software*; e fornecer alertas quando riscos estão próximos a limites pré-estabelecidos, indicando que o gerente deve executar os planos de resolução de riscos. Os agentes foram divididos em duas classes, que são: Agentes *Coletores* de informação e agentes *Monitores*. Agentes *Coletores* são responsáveis por coletar informações sobre a execução das atividades associadas aos riscos, para que as métricas sejam calculadas. Agentes *Monitores* são responsáveis pelo monitoramento dos fatores de risco, isto é, as métricas obtidas pelos agentes *Coletores* são comparadas com limites pré-estabelecidos pelo gerente e esses agentes são responsáveis por emitir alertas quando as métricas coletadas ultrapassam os limites permitidos.

Por exemplo, considerando o risco “*Não-entendimento dos requisitos e requisitos instáveis*”, as métricas: “*Percentual de validação de requisitos pelo Cliente*”, “*Percentual de requisitos elaborados pelo Cliente*”, “*Percentual de alteração dos requisitos*”, “*Requisitos aceitos*” e “*Requisitos rejeitados*” são usadas no monitoramento desses riscos. Essas métricas são calculadas a partir de medidas coletadas durante a execução de atividades que compõem o processo de *software*, sendo assim os agentes *Coletores* de informação são responsáveis por coletar essas métricas em tempo de execução. Por exemplo, a métrica “*Percentual de alteração dos requisitos*”, é calculada a partir do número de solicitações de mudança submetidas pelo cliente. Considerando um formulário *Web* onde o cliente faz essa solicitação, o agente deve monitorar e coletar dados durante a atividade de submissão da modificação. Os agentes *Monitores*, por outro lado, utilizando os dados coletados pelos agentes *Coletores* são responsáveis por calcular as métricas e atualizar o painel de controle do projeto. Caso o valor da métrica esteja acima de intervalos permitidos, é feita uma marcação para indicar o problema ao gerente.

4. Considerações Finais e Trabalhos Futuros

As discussões sobre riscos e o processo de gerência de riscos apresentadas servem como base para a identificação dos requisitos e para o projeto da arquitetura de monitoramento de riscos baseada agentes. Além disso, a arquitetura proposta apresenta-se como uma solução flexível para a implantação de uma infra-estrutura de *software* que dê suporte ao monitoramento de riscos.

No estágio atual de andamento deste projeto, está sendo realizado um estudo sobre as diferentes metodologias existentes para modelagem e especificação orientada a agentes, de forma que possa ser adotada uma metodologia para o desenvolvimento do mesmo. Entre as metodologias e *frameworks* que estão sendo analisados, encontram-se: AUML, GAIA, JADE e *Prometheus*.

5. Referências Bibliográficas

- Addison, T.; Vallabh, S. (2002) “Controlling Software Project Risks – An Empirical Study of Methods used by Experienced Project Managers”. In: South African Institute of Computer Scientists & Information Technologists, Saicsit, Port Elizabeth, South Africa.
- Collier, R.W. (2002) “Agent Factory: A Framework for the Engineering of Agent-Oriented Applications”. Tese (Doutorado), Universidade Nacional da Irlanda.
- Project Management Institute (PMI) (2000). “A Guide to the Project Management Body of Knowledge”. Pennsylvania.
- Standish Group. (2004) “Third Quarter Research Report”. Disponível em: <http://www.standishgroup.com/sample_research/PDFpages/q3-spotlight.pdf>.