

Uma Ontologia para Apoiar Discussões de Riscos em Projetos de Software

Camila H. Brondani, Lisandra M. Fontoura, Luís Alvaro L. Silva

Programa de Pós Graduação em Informática – Universidade Federal de Santa Maria
(UFSM) – Santa Maria, RS – Brasil

{chbrondani, lisandra, luisalvaro}@inf.ufsm.br

Abstract. *Having effective risk management in software projects is important to guarantee the control of factors that can affect both the success and quality of a developed system. In addition, improve communication between participants of a project, as well as to integrate their different experiences and perspectives regarding these risk management problems is essential. The purpose of this work is to demonstrate how to explore the representation of a knowledge domain through ontologies in order to support the collaborative identification and analysis of risks in software projects. In this context, the proposed ontology was developed to support the development of risk management activities typically involved in these risks collaborative discussion system.*

Resumo. *Um gerenciamento de riscos efetivo em projetos de software é importante para garantir que os fatores que podem impactar no sucesso e na qualidade de um sistema desenvolvido estão sob controle. Além disso, melhorar a comunicação entre participantes de um projeto, bem como integrar suas diferentes experiências e perspectivas sobre este domínio de problema é imprescindível. O objetivo deste trabalho é demonstrar como explorar a representação de um domínio de conhecimento por meio de ontologias para suportar a identificação e análise colaborativa de riscos em projetos de software. Para isso, a ontologia proposta foi criada para apoiar o desenvolvimento de atividades de gerenciamento de riscos tipicamente envolvidas nestas discussões colaborativas de riscos.*

1. Introdução

O desenvolvimento de software envolve inúmeros fatores como inovações tecnológicas e mudanças constantes nos requisitos dos clientes. Esta complexidade faz com que grande parte dos projetos de software exceda o prazo e o orçamento previstos, além de não atender às expectativas dos clientes em termos de funcionalidades e qualidade [Standish Group 2013]. Neste contexto, a ampla utilização de processos de gerenciamento de riscos integrando os diferentes *stakeholders* de um projeto é fundamental para o sucesso de projetos de software. Para nortear essa tarefa, o gerenciamento de riscos de projetos é tratado por modelos de avaliação da qualidade de processos de software, tais como: *Capability Maturity Model Integration* (CMMI) [SEI 2010], ISO/IEC15504 [ISO/IEC 15504 2003] e MPS.BR [SOFTEX 2011]. Além disso, o gerenciamento de riscos é tratado como uma das dez áreas de conhecimento do

Project Management Body of Knowledge (PMBOK) [Project Management Institute (PMI) 2013], em que seis processos relacionados ao gerenciamento de riscos são descritos. Além do PMBOK, o *framework* de processo denominado *Rational Unified Process (RUP)* [Rational Software Corporation 2003] propõe atividades que podem ser usadas para um efetivo gerenciamento dos riscos em um projeto de software. A avaliação desses modelos e guias permite claramente observar que uma grande quantidade de conceitos relacionados a esta área de conhecimento estão disponíveis na literatura. Contudo, ainda não existem padrões visando a formalização dos muitos conceitos de gerenciamento de riscos existentes (por exemplo, tipos de riscos, planos, etc) em modelos de conhecimento reusáveis, muitas vezes dificultando a construção e integração de sistemas desenvolvidos para apoiar o desenvolvimento de tarefas de gerenciamento de riscos.

Para desenvolver atividades de gerenciamento de riscos, é fundamental promover a colaboração entre as partes interessadas no projeto e o reuso das informações coletadas, assim como pode ser observado em [Project Management Institute (PMI) 2013], por exemplo. Conforme discutido em [Severo et al. 2013][Pozzebon et al. 2014], o processo de colaboração entre os participantes de uma discussão de riscos pode ser organizado por meio de um processo de argumentação [Moulin et al. 2002]. Nestes processos colaborativos de diálogo, o objetivo é permitir que participantes de um projeto cheguem a um consenso sobre os riscos que um projeto possui e as melhores formas de tratá-los. Este processo colaborativo também auxilia no compartilhamento de experiências e perspectivas oriundas destes diferentes participantes. Para apoiar tais processos colaborativos, bem como permitir a construção de uma memória reusável contendo experiências concretas de gerenciamento de riscos em projetos particulares, assim como proposto em [Machado et al. 2014], este trabalho propõe a construção e emprego de uma ontologia para gerenciamento de riscos.

No contexto do projeto de pesquisa onde este trabalho está inserido [Severo et al. 2013][Pozzebon et al. 2014][Machado et al. 2014], um dos objetivos desta ontologia é formalizar e tornar explícito o vocabulário que comumente é explorado por *stakeholders* de um projeto de software em discussões colaborativas de riscos. Além disso, a ontologia visa possibilitar que participantes de um projeto explorem um padrão para o compartilhamento de informações sobre riscos, fornecendo um modelo comum para a troca e reuso de informações sobre riscos, causas de riscos, planos de tratamento de riscos, entre outras. Ao mesmo tempo, o modelo de uma ontologia para gerenciamento de riscos pode ser adotado no apoio de consultas que podem ser realizadas em uma memória de discussões de riscos. Entre outras vantagens, a ontologia permite a análise semântica dos termos de gerenciamento de riscos especificados por *stakeholders* de projetos tanto em discussões de riscos quanto em consultas de riscos de projetos de software.

Este artigo está estruturado da seguinte forma: a Seção 2, trabalhos relacionados são analisados. Na Seção 3, a ontologia criada é discutida. Na Seção 4, um estudo de caso visando avaliar a ontologia proposta é descrito. Por fim, conclusões e trabalhos futuros são apresentados na Seção 5.

2. Trabalhos Relacionados

Propostas descrevendo ontologias sobre riscos de software foram apresentadas na literatura. Nesse contexto, destacam-se os trabalhos de Robin e Uma [Robin and Uma 2011], Falbo [Falbo 2010] e Gualberto *et al.* [Gualberto et al. 2012]. Os dois primeiros propõem ontologias para o domínio em riscos de projetos de software e o terceiro propõe uma ontologia relacionada a gestão de riscos de segurança da informação.

Robin e Uma [Robin and Uma 2011] apresentam uma proposta de uma ontologia chamada SRMONTO (*Software Risk Management Ontology*). A SRMONTO consiste em 5 ontologias: de identificação, análise, planejamento, monitoramento e rastreamento de riscos de software. A ontologia foi construída com base nas informações disponíveis na literatura e na experiência de desenvolvedores de software. Esta ontologia foi construída com um objetivo educacional. Neste caso, a ontologia visa possibilitar a construção e a utilização de sistemas *e-learning* na área de gerenciamento de riscos em projetos de software.

Falbo [Falbo 2010], por sua vez, propõe uma ontologia de riscos de software que visa formalizar o conhecimento envolvido neste domínio. A ontologia é um modelo conceitual e não um artefato de implementação, escrito em alguma linguagem de representação específica, tal como OWL. O propósito é capturar conceitos básicos do domínio, no qual compromissos técnicos de formalização não são descritos.

Por fim, Gualberto *et al.* [Gualberto et al. 2012] apresentam uma ontologia para formalizar, compartilhar, manipular e processar conceitos e informações relacionadas ao domínio de gestão de riscos de segurança da informação. A ontologia desenvolvida chama-se InfoSecRM. Ela pode ser caracterizada como uma ontologia de domínio, visto que dispõe de conceitos básicos relacionados ao domínio de Gestão de Riscos de Segurança da Informação (GRSI) e de Gestão da Segurança da Informação (GSI). Por meio da utilização desta ontologia, pode-se documentar e operar o processo de GRSI e subsidiar decisões gerenciais relacionadas à GSI. Para a elaboração da InfoSecRM, foram utilizadas as perspectivas de três abordagens: a) a metodologia *methontology*, como o processo que define o arcabouço das atividades a serem realizadas, b) o método 101, para definir o “como fazer” de algumas destas atividades e c) a metodologia proposta por Fox e Gruninger [Gruninger and Fox 1995] no projeto TOVE.

Deste modo, em termos de propósito, há muito em comum entre o objetivo deste trabalho com o artigo do Falbo [Falbo 2010]. Porém, diferencia-se ao fato de que o autor usa a ontologia como um modelo conceitual e não um artefato de implementação. O autor apenas definiu um modelo conceitual consensual descrevendo os principais conceitos envolvidos no domínio da Gerência de Riscos que pode ser adotada por organizações de software como um vocabulário básico para se falar sobre riscos. E este trabalho tem o objetivo de integrar a ontologia desenvolvida com a ferramenta *RD System*, a fim de sugerir riscos a serem empregados como argumentos em um processo de discussão colaborativo de riscos.

3. O Desenvolvimento da Ontologia

O desenvolvimento de uma ontologia é um processo iterativo. Para apoiar o processo de construção da ontologia, duas abordagens metodológicas foram exploradas: a do projeto

Enterprise Ontology, proposto por Uschold e King [Uschold and King 1995], e a metodologia proposta por Gruninger e Fox [Gruninger and Fox 1995] no projeto TOVE (*Toronto Virtual Enterprise*). Baseado nestas metodologias, os passos empregados para a construção desta ontologia foram:

1) Identificação do propósito da ontologia: a ontologia proposta aborda os conceitos relacionados a riscos de software. Tal ontologia é integrada a uma ferramenta de discussões colaborativas de riscos [Severo et al. 2013][Pozzebon et al. 2014][Machado et al. 2014], tendo como objetivo sugerir riscos a serem utilizados como argumentos em um processo de discussão colaborativo de riscos.

2) Aquisição de conhecimento: uma ontologia busca ser um modelo de consenso dentro de uma certa comunidade. Neste contexto, para desenvolver a ontologia apresentada, foram utilizados como base os modelos: PMBOK, RUP e CMMI. Dessa análise, os termos que compõem a ontologia foram selecionados.

3) Definição de questões informais de competência: as questões que a ontologia deve ser capaz de responder são as seguintes:

QC1. O que é um risco?

QC2. Qual é a categoria de um risco?

QC3. Qual é a fonte de um risco?

QC4. Quais são os riscos de um projeto?

QC5. Quais são as consequências de um risco?

QC6. Qual é o impacto de um determinado risco?

QC7. Qual é a probabilidade de um determinado risco?

QC8. Que ações podem ser tomadas para tratar um determinado risco?

4) Integração com ontologias já existentes: repositórios de ontologias foram consultados. Em particular, os repositórios foram:

OntoLP <<http://www.inf.pucrs.br/~ontolp/downloads.php>>. Neste repositório, foram encontradas ontologias sobre agrissolos, nanotecnologia, nanociência, currículo *lattes*, gerência de projetos (PMBOK), organizações, privacidade, entre outras.

DAMLOntologyLibrary <<http://www.daml.org/ontologies/>>. Neste repositório, a palavra-chave: *risk* foi usada como consulta. Como resultado, foram encontradas ontologias sobre riscos de doenças, risco de câncer, fatores de riscos ambientais. Neste caso, nenhuma ontologia disponível estava relacionada a riscos de software.

A partir dessa consulta, é possível perceber que os repositórios citados não disponibilizam uma ontologia sobre riscos a ser amplamente reutilizada. Contudo, [Falbo 2010] apresenta a estrutura básica de uma ontologia de riscos. Esta ontologia foi adaptada com a adição de novos conceitos para a definição da ontologia deste trabalho.

5) Manutenção: a manutenção deve ocorrer à medida que surgirem novas necessidades ou situações que demandem ajustes na ontologia.

6) Documentação: realizada como parte do processo de construção da ontologia.

7) Avaliação: esta etapa consiste em verificar se a ontologia está de acordo com o projetado durante a etapa de definição de escopo. Neste caso, é necessário confirmar se a ontologia pode responder as questões de competências estipuladas e se representa as principais características do domínio. Esta etapa foi realizada pela exploração de mecanismos de inferências e consultas feitas nas linguagens SWRL e SQWRL.

Em geral, a ferramenta para edição Protégé foi amplamente utilizada no apoio a formalização da ontologia proposta. Para a visualização gráfica desta ontologia foi utilizado o *plugin Jambalaya*. Na representação desta ontologia, a linguagem OWL [W3C 2004] foi utilizada.

3.1. Classes e Propriedades

A ontologia desenvolvida neste trabalho é formada por 19 classes. Baseado nestas classes, a ontologia representa o conceito de risco associado a um cenário de incidente de discussão de riscos. Assim, um *risco* está relacionado a um ou vários *projetos*. Em particular, a ocorrência de um risco em um projeto é representada pelo *perfil de risco individual*. Estes perfis são avaliados com o objetivo de priorizar e definir quais e como os riscos identificados serão tratados no contexto do projeto. Riscos possuem *fontes de riscos*. A identificação destas fontes são relevantes para determinar quais são as áreas de eventos que causam riscos para um projeto ou uma organização. Neste caso, essas fontes identificam áreas comuns das quais os riscos podem se originar.

Os riscos podem ser identificados e organizados dentro de *categorias de riscos*. Estes devem refletir os princípios comuns de um risco para determinada área de aplicação. As categorias de riscos criadas para esta ontologia são as seguintes: *Riscos de Cliente, Riscos de Execução, Riscos de Manutenção, Riscos de Negócio, Riscos de Planejamento e Riscos de Requisitos*. O perfil destes riscos é definido pelas diferentes avaliações de um mesmo risco. Este perfil registra cronologicamente a informação do estado corrente e a informação histórica associada a um dado risco em um projeto. Quando todos esses perfis de riscos são tomados em conjunto, os perfis dos riscos definem o *perfil de risco do projeto*.

Cada perfil de risco individual é sujeito a uma *avaliação de risco*. Tal avaliação registra o estado de um risco em um dado instante do projeto. A cada avaliação de risco é preciso identificar a *probabilidade* de sua ocorrência, bem como o seu *impacto*, caso o risco venha a ocorrer. A partir dessas informações é possível definir o grau de exposição de um projeto a um risco. Com este grau de exposição, é possível priorizar os riscos e identificar a *opção de tratamento de risco* a ser aplicada para o risco em questão. Quando um risco ocorre, isto é, transforma-se em um problema, então há *consequências* para o projeto. As *ações* que serão tomadas no projeto em questão deverão ser planejadas após definir quais riscos serão gerenciados, para então, tratá-los. Neste caso, a ontologia define os *tipos de ações* que podem ser empregados para tratar certos riscos.

A Figura 1 apresenta as 19 classes vinculadas à classe padrão *Thing*.

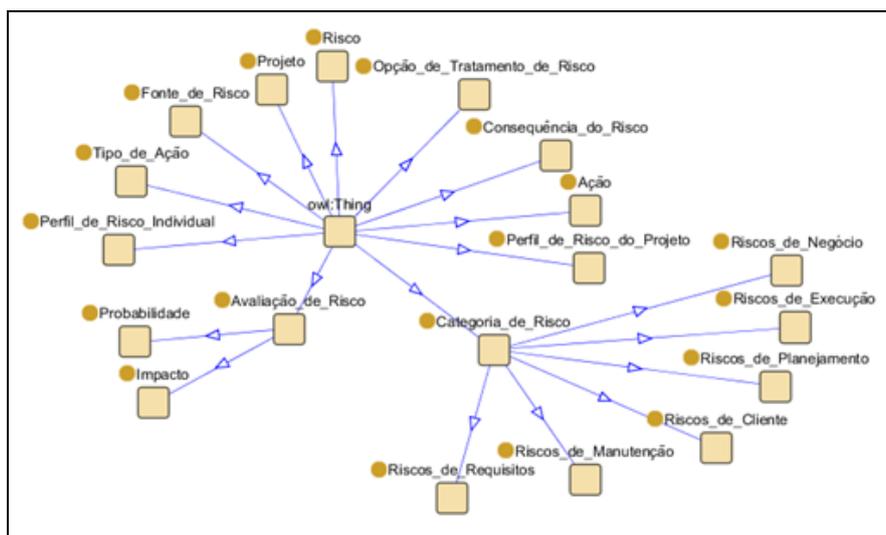


Figura 1. Classes da ontologia

A representação da ontologia também envolve a representação de propriedades de dados (atributos), as quais são responsáveis por caracterizar as instâncias de cada classe, e propriedades de objeto (relacionamentos), as quais definem os relacionamentos existentes entre os indivíduos de cada classe. Para exemplificar o uso de relações na ontologia desenvolvida, uma relação criada foi entre a classe *Risco* e as classes *Fonte de Risco*, *Categoria de Risco*, *Tipo de Ação*, através da propriedade *has*, indicando que cada risco possui fonte de risco e podem ser identificados e organizados dentro das categorias de riscos. E também será definido um ou mais tipos de ações que serão empregados para tratar este risco. A propriedade *belongs* representa a relação entre a classe *Risco* e *Projeto*, mostrando que um risco pertence a um ou vários projetos.

3.2. Consultas e Inferências

Somente a descrição de classes, propriedades e suas respectivas restrições e relacionamentos não são suficientes para representar um domínio de conhecimento numa ontologia. Em geral, tais recursos de representação de conhecimento modelam a informação estática de um domínio. Contudo, para obter novos conhecimentos a partir de uma ontologia, é preciso realizar inferências sobre ela. Uma das formas de realizar inferências é a partir da existência das regras da ontologia. Neste caso, usuários podem utilizar estas definições de regras para produzir novas conclusões a serem acrescentadas na ontologia.

Consultas em ontologias são utilizadas para obter dados relevantes sobre um domínio que está sendo representado. Neste projeto, tais consultas foram criadas usando a linguagem SQWRL, na qual utiliza uma biblioteca com os métodos do SWRL, em que regras para fazer buscas em OWL estão disponíveis. Diferentemente das regras SWRL, que aplicam o resultado diretamente na ontologia, as consultas SQWRL apenas retornam as informações que satisfazem a restrição descrita e, então, esses resultados podem ser tratados utilizando uma linguagem de programação qualquer.

Uma consulta pode utilizar uma ou mais regras. Uma regra também pode ser utilizada por diversas consultas. Para possibilitar o processamento de regras e consultas sobre a ontologia criada, foi necessário à utilização do *plugin Jess* no ambiente Protégé.

Como exemplo, algumas regras criadas para a manipulação da ontologia são apresentadas a seguir:

Regra 01: Todo risco de cliente que possui impacto maior ou igual a 80% deve ter uma opção de tratamento, pois este risco é considerado como tendo um impacto alto:

$$\text{Riscos_de_Cliente} (?rc) \wedge \text{hasImpacto} (?rc, ?i) \wedge \text{swrlb:greaterThanOrEqual} (?i, 80) \rightarrow \text{Opção_de_Tratamento_de_Risco} (?rc)$$

Regra 02: Toda avaliação de um risco acarreta em identificar a opção de tratamento de risco:

$$\text{Avaliação_de_Risco} (?a) \rightarrow \text{Opção_de_Tratamento_de_Risco} (?a)$$

Regra 03: Todo risco de requisitos que possui probabilidade superior a 60% acarreta em um tipo de ação:

$$\text{Riscos_de_Requisitos} (?rr) \wedge \text{hasProbability} (?rr, ?p) \wedge \text{swrlb:greaterThan} (?p, 60) \rightarrow \text{Tipo_de_Ação} (?rr)$$

Regra 04: Se existe um risco de requisitos mal entendidos e uma equipe inexperiente, isto irá acarretar no risco de cliente insatisfeito:

$$\text{Riscos_de_Negócio} (\text{Requisitos_mal_entendidos}) \wedge \text{Riscos_de_Execução} (\text{Equipe_inexperiente}) \rightarrow \text{Riscos_de_Cliente} (\text{Cliente_insatisfeito})$$

Regra 05: Se um projeto possui cliente e equipe com visões diferentes, baixa qualidade do projeto e avanço do projeto sem aprovação do cliente, acarreta que o produto final não corresponde às expectativas do cliente:

$$\text{Riscos_de_Requisitos} (\text{Cliente_e_equipe_têm_visões_diferentes_de_um_mesmo_requisito}) \wedge \text{Riscos_de_Planejamento} (\text{Baixa_qualidade_do_projeto}) \wedge \text{Riscos_de_Execução} (\text{Avanço_do_projeto_sem_aprovação_do_cliente}) \rightarrow \text{Riscos_de_Cliente} (\text{Produto_final_não_corresponde_as_expectativas_do_cliente})$$

Neste contexto, regras de inferência como estas aumentam o poder de expressividade da ontologia. Neste caso, ao executar uma regra, seu resultado afeta diretamente a estrutura da ontologia. Isso se deve ao fato de que toda vez que as restrições presentes em uma regra forem aceitas, um novo conhecimento sobre o domínio é acrescentado à ontologia.

4. Estudo de Caso

Devido ao foco principal do trabalho ser a utilização de ontologias para representação de conhecimento no contexto de discussões de riscos de software, foi desenvolvido um estudo de caso para realizar uma avaliação da proposta deste trabalho. Este estudo de caso é baseado em um projeto para desenvolvimento de um sistema de administração de condomínios. O objetivo deste sistema é o gerenciamento das contas do condomínio e divisão dos valores por condômino de forma a gerar a fatura para quitação mensal das despesas. As informações deste projeto foram utilizadas como entrada na ferramenta *Web Risk Discussion System (RD System)* [Severo et al. 2013]. Esta ferramenta é um sistema colaborativo de discussão de riscos, que faz a integração de jogos de diálogo

[Machado et al. 2014] e esquemas de argumentação [Pozzebon et al. 2014] no apoio ao gerenciamento colaborativo de riscos em projetos de software.

Discussões de riscos são gerenciadas no sistema como atividades de um projeto. Cada projeto é definido por um conjunto de características, descritas como informações factuais sobre o projeto. No sistema, os usuários são capazes de definir e ajustar essas características às necessidades de uma organização. É importante ressaltar que o *RD System* mantém registros de todas as etapas da discussão colaborativa de riscos (ex.: identificação de riscos, análise de riscos e planejamento de riscos), permitindo que usuários formem uma memória reusável de gerenciamento de riscos, assim como proposto em [Machado et al. 2014]. A linguagem OWL foi explorada na representação da ontologia, SWRL na representação de regras e SQWRL na execução de consultas semânticas sobre os termos representados.

Assim como exemplificado na Figura 2, os participantes de uma discussão colaborativa de riscos chegaram ao consenso que para o risco “Desenvolvimento de interface do usuário inadequada”, o impacto e probabilidade deste risco seriam baixos.

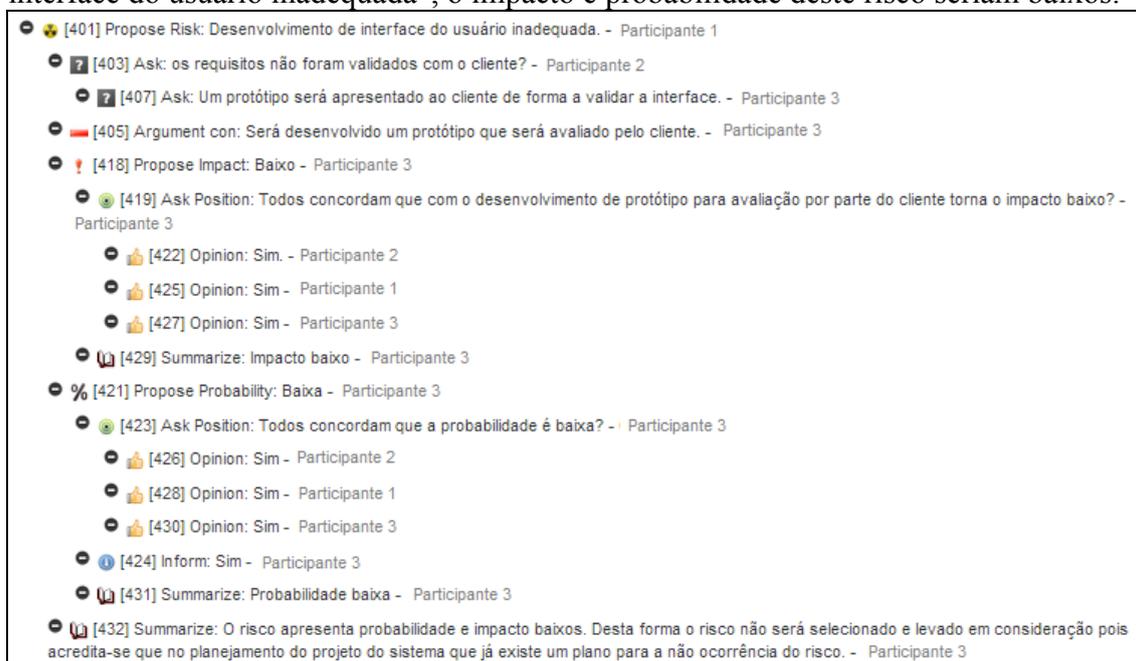


Figura 2. Discussão de Riscos na ferramenta *RD System*

A partir disso, as seguintes classes foram instanciadas na ontologia:

- Para a classe Projeto, a instância Projeto_Condomínio foi criada.
- Para a classe Impacto, foram criadas cinco instâncias: Impacto_5_por_cento, Impacto_10_por_cento, Impacto_20_por_cento, Impacto_40_por_cento e Impacto_80_por_cento. Tais instâncias foram relacionadas a valores inteiros de impacto de 5, 10, 20, 40, 80, respectivamente, atribuídos pela propriedade (valor_impacto).
- Para a classe Probabilidade, foram criadas cinco instâncias: Probabilidade_10_por_cento, Probabilidade_30_por_cento, Probabilidade_50_por_cento, Probabilidade_70_por_cento e Probabilidade_90_por_cento. Tais

instâncias foram relacionadas a valores inteiros de probabilidade de 10, 30, 50, 70, 90, respectivamente, atribuídos pela propriedade (valor_probabilidade).

Os valores de probabilidade e impacto sugeridos estão de acordo com a matriz de probabilidade e impacto do PMBOK.

- Também foram criadas instâncias de alguns riscos para as subclasses da classe Categoria de Risco, visualizadas na Tabela 1: Riscos de Cliente, Riscos de Execução, Riscos de Manutenção, Riscos de Negócio, Riscos de Planejamento e Riscos de Requisitos.

Efetuada a instanciação das classes definidas na ontologia, algumas consultas foram elaboradas e executadas para verificar resultados que poderiam ser obtidos a partir das instanciações dessas classes. Entre outras, as consultas executadas foram as seguintes:

Consulta 01: Como o risco da discussão foi “Desenvolvimento de interface do usuário inadequada”, e este é considerado um risco de execução, a ontologia permite fazer uma consulta que retorna todos os riscos de execução representados, por exemplo:

Riscos_de_Execução (?re) → sqwrl:select (?re)

Em seguida, cada risco de execução foi associado com determinado valor de impacto e probabilidade. Desta forma, esta consulta foi realizada da seguinte maneira:

Consulta 02: Consulta que retorna o(s) risco(s) de execução que possui (em) impacto com valor de 5%:

Riscos_de_Execução (?re) ∧ hasImpact (?re, Impacto_5_por_cento) → sqwrl:select (?re)

Neste trabalho, a estrutura da ontologia deu-se com a definição de um conjunto de termos/conceitos e dos relacionamentos semânticos entre eles, estes então, foram utilizados para formular as consultas. A ontologia serviu para definir as regras de combinação entre os termos e seus relacionamentos, estes relacionamentos foram criados na ferramenta Protégé.

Tabela 1. Instâncias da classe Categoria de Riscos

Riscos de Manutenção (6)	Alto custo para testar uma mudança em termos de orçamento e tempo	Riscos de Cliente (6)	Aprovação inadequada de documentos por parte do cliente
	Alto nível de complexibilidade técnica das mudanças solicitadas		Cliente insatisfeito
	Código do programa a ser mantido complexo e desestruturado		Disponibilidade de fundos do cliente
	Documentação excessiva referente ao software a ser mantido		Produto final não corresponde às expectativas do cliente
	Grande número de linhas de código afetadas com a mudança		Resistência a mudanças por parte dos empregados do cliente
	Mudanças contínuas no objetivo e escopo ao manter software		Treinamento inadequado dos usuários
Riscos de Negócio (5)	Condições de quebra e manutenção de contrato	Riscos de Execução (14)	Alta rotatividade dentro da organização usuária
	Diferenças legais entre países do cliente e da companhia		Ambiente de desenvolvimento diferente do ambiente de implantação
	Impostos locais e nacionais		Ambiente improdutivo
	Instabilidade organizacional		Ambientes de integração e testes inadequados
	Mudanças nas leis		Avanço do projeto sem aprovação do cliente
Riscos de Planejamento (7)	Ausência de um líder		Deficiência nos componentes fornecidos externamente
	Baixa qualidade do projeto		Dificuldade de uso por causa de um design da interface pobre
	Cronograma irreal		Dificuldade em obter cenários e dados de testes realísticos
	Falta de consciência dos riscos		Equipe com carência de pessoas
	Orçamento irreal		Equipe inexperiente
	Sem fundos para treinamento e tutoria		Falha na comunicação entre o time
Riscos de Requisitos (5)	Tempo inadequado para realização de testes		Falta de padrão de programação durante o ciclo de vida do software
	Análise inadequada quando um novo requisito é adicionado		Gerentes inexperientes
	Cliente e equipe têm visões diferentes de um mesmo requisito		Métricas inadequadas
	Objetivos pouco claros ou equivocados		
	Requisitos de baixa qualidade		
Requisitos mal entendidos			

Em geral, este estudo de caso demonstrou que é possível tornar a ferramenta *RD System* mais objetiva, pois atualmente o processo tradicional de discussão de riscos é realizado com argumentos informais. A partir do desenvolvimento dessa ontologia foi possível ter uma representação formal de conceitos de riscos comumente apresentados nestes argumentos. Com as instâncias criadas de conceitos importantes para a discussão de riscos, participantes durante a discussão podem usar os mecanismos de consulta da ontologia para avaliar os diferentes riscos e suas características sendo discutidos.

5. Considerações Finais e Trabalhos Futuros

A utilização de ontologias é imprescindível no apoio às necessidades de comunicação e no melhor reuso de informações e conhecimento em atividades colaborativas de gerenciamento de riscos em projetos de software. Neste trabalho, o processo de desenvolvimento de uma ontologia no domínio de conhecimentos de discussões de riscos de software foi descrito, definindo conceitos, propriedades, consultas e inferências. Para construir a ontologia apresentada neste trabalho, além de realizar uma pesquisa sobre as principais tecnologias envolvidas na criação de ontologias e mecanismos de inferência, diferentes abordagens para gerência de riscos foram revisadas, a fim de buscar conceitos para a construção da mesma.

Uma das contribuições apresentadas neste trabalho foi a formalização de conceitos de riscos, causas de riscos, planos de tratamento de riscos, entre outros, podendo ser compartilhada e reutilizada por pessoas que desenvolvam aplicações dentro desse mesmo domínio. Sendo assim, uma boa prática é integrar essa ontologia em questão, visando aproveitar conceituações previamente estabelecidas. Além disso, a ontologia permite a análise semântica dos termos de gerenciamento de riscos especificados por *stakeholders* de projetos tanto em discussões de riscos quanto em consultas de riscos de projetos de software.

A partir de regras desenvolvidas em SWLR, este trabalho demonstrou como inferir novos conhecimentos na ontologia, bem como realizar consultas usando a linguagem SQWRL, a qual é utilizada para buscar informações na ontologia. Um estudo de caso desenvolvido possibilitou a experimentação prática da ontologia construída, por meio da instanciação de classes e execução de consultas.

Como trabalho futuro, pretende-se ampliar a integração desta ontologia ao sistema de discussões de riscos *RD System*. Neste caso, espera-se que o mecanismo de busca de riscos possa trazer resultados mais adequados à semântica dos termos procurados pelo usuário. Com isso, irá aumentar e otimizar a relevância dos mesmos, os quais são amplamente usados por usuários durante atividades de debate de riscos. Desta forma a ontologia poderá ser testada em ambiente real, além de aprimorar a ontologia com a criação de mais indivíduos caracterizando uma base de conhecimento maior e a realização de inferências mais complexas.

Agradecimentos

Agradece-se à Fundação de Amparo à Pesquisa do Rio Grande do Sul (Fapergs) pelo suporte financeiro a este trabalho.

Referências

- Falbo, R. D. A. (2010). Uma Ontologia de Riscos de Software. *IX Simpósio Brasileiro de Qualidade de Software (SBQS 2010)*.
- Gruninger, M. and Fox, M. S. (1995). Methodology for the Design and Evaluation of Ontologies.
- Gualberto, É. S., Souza, R. T. De, Duque, C. G. and Deus, F. E. G. De (2012). InfoSecRM: Uma Abordagem Ontológica para a Gestão de Riscos de Segurança da Informação. *VIII Simpósio Brasileiro de Sistemas de Informação (SBSI 2012)*.
- ISO/IEC 15504 (2003). Information Technology. Process Assessment. Part 1: Concepts and Vocabulary.
- Machado, N. L. R., Silva, L. a L., Lisandra, M. and Campbell, J. a (2014). Case-based Reasoning for Experience-based Collaborative Risk Management. *XXVI International Conference on Software Engineering and Knowledge Engineering (SEKE 2014)*, p. 262–267.
- Moulin, B., Irandoust, H., Bélanger, M. and Desbordes, G. (2002). Explanation and Argumentation Capabilities: Towards the Creation of More Persuasive Agents. *Artificial Intelligence Review*, v. 17, p. 169–222.
- Pozzebon, R. C. B., Silva, L. A. L., Fontoura, L. M. and Campbell, J. A. (2014). Argumentation schemes for the reuse of argumentation information in collaborative risk management. *15th International Conference on Information Reuse and Integration*, p. 179–186.
- Project Management Institute (PMI) (2013). A Guide to the Project Management Body of Knowledge.
- Rational Software Corporation (2003). Rational Unified Process: Version 2003.06.12.
- Robin, C. R. R. and Uma, G. V (2011). Design and Development of Ontology Suite for Software Risk Planning , Software Risk Tracking and Software Risk Control. *Journal of Computer Science*, v. 7, n. 3, p. 320–327.
- SEI (2010). CMMI® for Development, Version 1.3.
- Severo, F., Fontoura, L. M. and Silva, L. A. L. (2013). A Dialogue Game Approach to Collaborative Risk Management. *XXV International Conference on Software Engineering and Knowledge Engineering (SEKE 2013)*, p. 548–551.
- SOFTEX (2011). MPS.BR - Melhoria de Processo do Software Brasileiro, Guia Geral.
- Standish Group (2013). The Chaos Manifesto.
- Uschold, M. and King, M. (1995). Towards a Methodology for Building Ontologies. *Workshop on Basic Ontological Issues in Knowledge Sharing*.
- W3C (2004). OWL Web Ontology Language Overview.