Uma Ontologia de Riscos de Software

Ricardo de Almeida Falbo

Mestrado em Informática – Universidade Federal do Espírito Santo (UFES)
Av. Fernando Ferrari s/n, Campus de Goiabeiras – 29.060-900 – Vitória – ES – Brasil
falbo@inf.ufes.br

Resumo. *Apesar de reconhecidamente importante para o sucesso de projetos de software, muitas organizações têm negligenciado o processo de Gerência de Riscos (GRI). Isso decorre, dentre outros, do fato de haver diversas visões parciais sobre esse domínio, cada qual adotando um vocabulário próprio, o que dificulta a integração e a comunicação. Para superar essa barreira, é útil definir um modelo conceitual consensual descrevendo os principais conceitos envolvidos no domínio da GRI, com a finalidade de apoiar o entendimento, aprendizado e a comunicação. Uma vez que ontologias podem ser usadas para esse fim, este artigo apresenta uma ontologia de riscos de software, estabelecendo uma conceituação comum acerca do domínio de riscos de software, a qual pode ser adotada por organizações de software como um vocabulário básico para se falar sobre riscos.*

Abstract. *In spite of being an important process for projects to succeed, Risk Management (RSKM) is being disregarded by many software organizations. This happens because of several factors. One of them is the fact that there are several partial views about this domain, each one of them using its own vocabulary. This makes integration and communication difficult. Thus, it is worthwhile defining a consensual conceptual model describing the main concepts involved in the RSKM domain, for purposes of understanding, learning and communication. Since these are purposes of ontologies, this paper presents an ontology of software risk that establishes a common conceptualization about the RSKM domain to be used by organizations as a basic vocabulary for talking about risks.*

1. Introdução

A importância de se gerenciar riscos em projetos de software é amplamente reconhecida. A maioria dos modelos de qualidade de processo (p. ex., CMMI [SEI, 2006] e MPS.BR [Softex, 2009]) e padrões de gerência de projetos (p.ex., PMBOK [PMI, 2008]) aponta essa prática como essencial para o sucesso de um projeto. O MPS.BR, por exemplo, possui um processo de Gerência de Riscos (GRI) no nível C de maturidade, além de possuir um resultado esperado no processo Gerência de Projetos (GPR6) que diz respeito à gerência de riscos. Situação análoga ocorre no CMMI.

Contudo, apesar de ter sua importância reconhecida, muitas organizações têm negligenciado a realização desse processo e grande parte dos projetos de software sequer experimentou alguma forma de gerência de riscos [Lister, 1997]. Segundo resultados da pesquisa em Qualidade e Produtividade no Setor de Software Brasileiro realizada em

2001 [PBQP, 2002], apenas 11,8% das organizações de software brasileiras gerenciavam riscos. Diversos fatores podem ser apontados como razões para esse quadro, dentre eles [Carr, 1997]: cultura organizacional avessa a riscos, infraestrutura inadequada e falta de uma abordagem sistemática para a gerência de riscos. De fato, a Gerência de Riscos (GRI) é uma ferramenta de aplicação relativamente recente em projetos de software e os conceitos envolvidos são, muitas vezes, abstratos e difíceis de compreender.

Para superar essa dificuldade de aplicação, algumas ações são importantes, dentre elas: (i) estabelecer um processo padrão organizacional para a GRI; (ii) prover ferramentas de apoio a esse processo; (iii) gerenciar o conhecimento organizacional relacionado à GRI; e (iv) prover um modelo conceitual consensual descrevendo os principais conceitos envolvidos nesse domínio, com o objetivo de apoiar a comunicação, entendimento, aprendizado e todas as ações anteriormente citadas (i a iii). Esta última ação é o foco deste trabalho.

Uma vez que ontologias podem ser usadas para promover um entendimento comum entre pessoas atuando em uma área de conhecimento, desenvolveu-se uma Ontologia de Riscos de Software, estabelecendo uma conceituação comum acerca desse domínio. A ontologia proposta tem muitos potenciais usos, dentre eles: (i) para a comunicação humana, sendo os seus conceitos usados como um vocabulário comum para se falar sobre riscos em organizações de software; (ii) como uma especificação base para o desenvolvimento de ferramentas de apoio à GRI; (iii) para apoiar a definição de uma infraestrutura de gerência de conhecimento sobre riscos de software, propiciando, desse modo, aprendizagem organizacional sobre riscos.

O objetivo deste artigo é apresentar a ontologia de riscos de software desenvolvida visando formalizar parcialmente o conhecimento envolvido nesse domínio. Para tal, a seção 2 trata brevemente do universo de discurso da ontologia proposta, a gerência de riscos, e da abordagem de desenvolvimento adotada. A seção 3 apresenta a ontologia de riscos desenvolvida, enquanto a seção 4 é dedicada a uma avaliação preliminar da mesma. Finalmente, as seções 5 e 6 apresentam, respectivamente, trabalhos correlatos e as conclusões e perspectivas futuras deste trabalho.

2. Gerência de Riscos e Ontologias

De acordo com o PMBOK [PMI, 2008], um risco é uma condição ou evento incerto que, caso ocorra, tem efeito em pelo menos um objetivo de um projeto. Um risco combina a probabilidade de um evento acontecer e suas consequências e geralmente o termo risco é usado quando há pelo menos uma possibilidade de consequências negativas [ISO/IEC, 2006].

O propósito da Gerência de Riscos (GRI) é identificar potenciais problemas de cunho técnico ou gerencial antes que eles ocorram, de forma que ações possam ser tomadas a fim de reduzir ou eliminar a probabilidade e o impacto desses problemas [ISO/IEC, 2006]. Uma implementação bem sucedida de um processo de GRI leva a importantes resultados, dentre eles: (i) riscos são identificados; (ii) a probabilidade e as consequências desses riscos são entendidas; (iii) a prioridade na qual riscos devem ser tratados é estabelecida; (iv) alternativas adequadas de tratamento de riscos são

estabelecidas; e (v) as ações adequadas são selecionadas para riscos que estiverem em um nível acima do limiar aceitável.

Existem na literatura diversas propostas de processos de GRI. Gusmão e Moura (2004), após um exame de várias delas, apontam que as seguintes atividades tipicamente compõem um processo de GRI: Planejamento da Gerência de Riscos, Identificação de Riscos, Análise de Riscos, Planejamento de Respostas aos Riscos, Monitoramento de Riscos, Controle de Riscos e Comunicação de Riscos. Mesmo sem discutir aqui detalhes acerca de cada uma dessas atividades, pode-se perceber que este é um processo complexo, o que explica em parte o fato da indústria de software como um todo ainda não aplicar ativa e sistematicamente as práticas de gerenciamento de riscos.

Gusmão (2007) aponta que uma grande dificuldade atual para a gerência de riscos em projetos de software é a existência de diversas visões parciais sobre esse domínio. Cada visão parcial carrega consigo um vocabulário e determinados valores próprios, dificultando a integração entre os diversos profissionais pela ausência de padronização. Esse problema pode ser minimizado se a conceituação envolvida no domínio da gerência de riscos for, pelo menos parcialmente, explicitada, o que pode ser feito por meio de uma ontologia de domínio.

Uma ontologia de domínio é um artefato de engenharia que busca explicitar a conceituação de um domínio particular. Uma conceituação, por sua vez, corresponde ao conjunto de conceitos usados para interligar abstrações de entidades de um dado domínio. Assim, uma ontologia de domínio tem como objetivo explicitar e formalmente definir os conceitos, relações, propriedades e restrições em um domínio particular [Guizzardi, 2005]. Ontologias de domínio têm diversos usos, dentre eles [Jasper e Uschold, 1999]: (i) apoio à comunicação entre pessoas envolvidas no domínio; (ii) integração de dados e interoperabilidade de sistemas desenvolvidos para o domínio e (iii) especificação reutilizável para a construção de sistemas no domínio. Antoniou e van Harmelen (2004) citam, ainda, que o uso de ontologias favorece a construção de sistemas de gerência de conhecimento nos quais, dentre outros, o conhecimento pode ser organizado em espaços conceituais de acordo com seu significado e a busca por palavras-chave pode ser substituída por consultas mais elaboradas.

Uma vez que uma ontologia é um artefato de engenharia, ela deve ser construída usando métodos apropriados. Para desenvolver a ontologia de gerência de riscos, foi adotado o método SABiO (***S**ystematic **A**pproach for **Bui**lding **O**ntologies*) [Falbo, 2004], que sugere a realização das seguintes atividades: (i) identificação do propósito e especificação de requisitos, cujo produto principal são questões de competência que a ontologia deve ser capaz de responder, (ii) captura da ontologia, na qual são capturados conceitos, relações, propriedades e restrições relevantes sobre o domínio em questão; e (iii) formalização, na qual os axiomas da ontologia são escritos em uma linguagem formal (neste trabalho, optou-se pela lógica de primeira ordem). Paralelamente a essas atividades, ocorrem as atividades de: (iv) integração com ontologias existentes, cujo objetivo é reutilizar conceituações existentes e integrá-las à ontologia em desenvolvimento; (v) avaliação da ontologia, que, dentre outros, trata de avaliar se a ontologia é capaz de responder às questões de competência; e (vi) documentação da ontologia, que tem por objetivo registrar o desenvolvimento da ontologia.

SABiO advoga, ainda, o uso de uma linguagem de modelagem para facilitar a comunicação dos modelos da ontologia. Neste trabalho é utilizado o perfil UML ilustrado na Figura 1 como linguagem gráfica para representação de ontologias, o qual faz distinção entre três tipos de entidades em uma ontologia: conceitos, relações e propriedades. Conceitos da ontologia são representados como uma classe estereotipada (<<concept>>). Propriedades de um único indivíduo são representadas como atributos da UML. Relações podem ser de dois tipos: formais ou materiais [Guizzardi 2005]. Relações formais acontecem entre duas ou mais entidades diretamente sem que haja necessidade que outro indivíduo intervenha e são representadas como uma associação nomeada entre os conceitos relacionados. Relações materiais, por outro lado, possuem estrutura material própria, sendo necessário que exista uma outra entidade para mediar os conceitos envolvidos. Essa entidade, dita um modo relacional, conecta os outros indivíduos e é representada por meio de uma classe associativa estereotipada (<<relator>>). A linha pontilhada conectando a relação material e o modo relacional representa que a primeira deriva do último, i.e., indica como as instâncias da relação material podem ser derivadas a partir das instâncias das relações de mediação. A derivação de uma relação material a partir de um modo relacional é representada como uma classe associativa com um pequeno círculo preto anexado, como sugerido em [Guizzardi 2005]. Além de representar a relação formal, são representadas as relações de mediação entre o modo relacional e os conceitos relacionados, usando o estereótipo (<<mediation>>). Por fim, relações todo-parte e de subtipo são representadas, respectivamente, com a notação de agregação e de generalização da UML. Neste trabalho, são consideradas apenas relações todo-parte do tipo "subcoleção de" [Guizzardi 2005], que são irreflexivas, antissimétricas e transitivas.

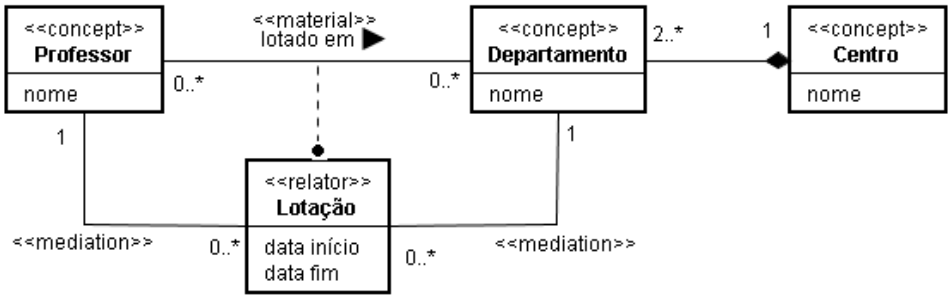

Figura 1 – Exemplo do Perfil UML usado para representar ontologias.

Vale destacar que neste artigo está-se considerando ontologia como um modelo conceitual e não um artefato de implementação, escrito em alguma linguagem de implementação específica, tal como OWL (*Ontology Web Language*) [Smith et al. 2004]. O propósito é capturar a conceituação básica do domínio, sem introduzir compromissos de codificação que poderiam restringir a representação de aspectos importantes do domínio. A partir dos modelos conceituais escritos no perfil UML mostrado na Figura 1, diversas implementações podem ser derivadas, considerando as características das linguagens adotadas.

3. Uma Ontologia para o Domínio de Gerência de Riscos de Software

Segundo o método SABiO, primeiro deve-se identificar o propósito da ontologia. A ontologia de riscos de software apresentada neste trabalho é uma evolução da ontologia

parcialmente apresentada em [Falbo et al. 2004]. A primeira versão dessa ontologia foi desenvolvida para servir de base para o desenvolvimento de ferramentas de apoio à GRI incluindo ferramentas de gerência de conhecimento. Assim, a ontologia de GRI apresentada neste trabalho ainda mantém o propósito de servir como uma especificação conceitual de referência para o desenvolvimento e integração de ferramentas de apoio ao processo de GRI, como a versão original. Contudo, espera-se que a nova versão tenha usos mais amplos. A nova versão da ontologia visa capturar a conceituação básica envolvida no domínio da GRI, servindo de referência para pessoas, sistemas e organizações se comunicarem usando um vocabulário comum.

Uma vez que uma ontologia busca ser um modelo de consenso dentro de uma certa comunidade, para desenvolver a ontologia aqui apresentada, foram utilizadas como base referências bastante importantes na área, muitas delas padrões internacionais, tais como ISO/IEC 16085 [ISO/IEC 2006], ISO 31000 [ISO 2009], PMBOK [PMI 2008] e SWEBOK [IEEE 2004]. Além disso, foram consultados especialistas na área, os quais opinaram sobre as questões a serem respondidas pela ontologia, os modelos propostos e as restrições a serem consideradas.

Dada a complexidade do domínio, optou-se por considerar inicialmente questões de competência relacionadas a um processo de GRI considerando os aspectos tidos como mais relevantes em níveis mais altos de maturidade, tal como o nível C do MPS.BR. Assim, para derivar as questões de competência, os modelos MPS.BR e CMMI foram analisados, procurando identificar que questões deveriam ser respondidas pela ontologia para capturar parcialmente a conceituação subjacente à GRI nesses modelos. Tomando por base esse referencial, chegou-se às seguintes questões de competência para a nova versão ontologia de riscos de software:

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 os riscos gerenciados em um projeto em um determinado momento?
QC6. Qual o grau de exposição de um risco em um projeto em um dado momento?
QC7. Qual o perfil de um risco individual em um projeto?
QC8. Qual o perfil de riscos de um projeto?
QC9. Que ações podem ser tomadas para tratar um determinado risco?
QC10. Quais as ações planejadas para tratar um risco no contexto de um projeto?
QC11. Quais as ações efetivamente tomadas para tratar um risco em um projeto?
QC12. Quais as atividades de um processo de gerência de riscos?
QC13. Quais os artefatos produzidos e consumidos por atividades do processo de GRI?
QC14. Quais os recursos necessários para se realizar a gerência de riscos?

Em função da forte relação entre algumas das questões de competência anteriormente relacionadas, a ontologia foi desenvolvida de forma modular, considerando quatro aspectos:

- Riscos, Fontes e Categorias de Riscos (questões 1 a 3);
- Riscos de um Projeto de Software e suas Avaliações (questões 4 a 8);
- Gerência de Riscos de um Projeto de Software (questões 9 a 11);
- Processo da Gerência de Riscos (questões 12 a 14).

Riscos e Categorias de Riscos

A Figura 2 mostra o modelo conceitual da parte da ontologia que trata de riscos e sua categorização, apresentando seus conceitos e relações.

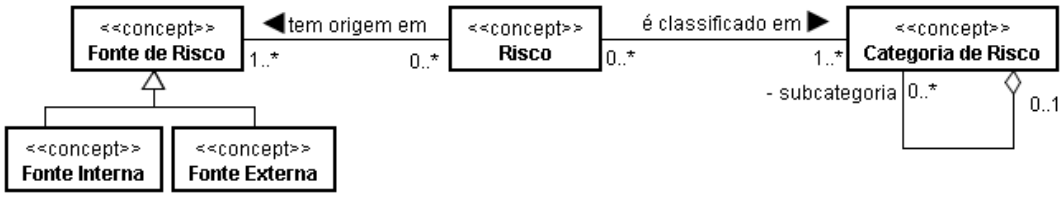

Figura 2 – Riscos e Categorias de Riscos

O conceito de risco indica "exposição a consequências da incerteza", podendo ser aplicado tanto no gerenciamento de perdas como no de ganhos potenciais [Gusmão 2007]. Contudo, geralmente o termo risco é usado quando há pelo menos uma possibilidade de consequências negativas [ISO/IEC 2006]. Exemplos de riscos típicos de projetos de software incluem rotatividade de pessoal, volatilidade de requisitos, troca do patrocinador do projeto, determinação de prazos inviáveis etc.

Trabalhar com uma série de riscos aleatórios pode ser um fator complicador, principalmente em grandes projetos, em que o número de riscos é relativamente grande. Assim, a classificação de riscos em categorias de risco é importante. Uma categoria de risco é uma classe ou tipo de um risco. Muitas classificações para riscos são encontradas na literatura, como, por exemplo, a classificação proposta na taxonomia de riscos do SEI (*Software Engineering Institute*) [Carr et al. 1993], que identifica três grandes categorias de riscos: Engenharia de Produto, Ambiente de Desenvolvimento e Restrições de Programa. Categorias de riscos podem ser refinadas em subcategorias. Por exemplo, a categoria Engenharia de Produto apresenta como subcategorias Riscos Relacionados a Requisitos e Riscos Relacionados à Segurança [Carr et al. 1993].

A identificação das *fontes de risco* é importante para saber quais são as áreas de eventos fundamentais que causam riscos para um projeto ou uma organização, podendo as fontes serem internas ou externas. Fontes de riscos identificam áreas comuns das quais os riscos podem se originar. Com a antecipação das fontes de riscos internas e externas, é possível que riscos sejam identificados também de forma antecipada [SEI 2006].

O modelo apresentado na Figura 2, contudo, não é capaz de capturar uma importante restrição, descrita, então, pelo axioma (A1):

- Se um risco *r* é classificado em uma categoria *c2* que é subcategoria de outra categoria *c1*, dita sua supercategoria, então *r* também é classificado em *c1*.

$$(\forall r \in Risco, c1, c2 \in CategoriaRisco)\ (classificadoEm\ (r, c2) \wedge subcategoria\ (c2,c1) \rightarrow classificadoEm\ (r, c1))$$

Vale a pena destacar que, além do axioma acima descrito, o uso do perfil UML apresentado na Figura 1 indica outros axiomas, tal como a transitividade na relação de subcategorias (A2), abaixo formalizado. Esses axiomas, contudo, não serão mais aqui apresentados.

$$(\forall c1, c2, c3 \in CategoriaRisco)\ (subcategoria(c3, c2) \wedge (subcategoria(c2, c1) \rightarrow subcategoria(c3, c1))$$

Riscos de um Projeto de Software e suas Avaliações

A Figura 3 mostra o modelo conceitual da parte da ontologia que trata de avaliações de risco. Durante o processo de GRI, os *riscos* relacionados ao *projeto* são identificados na forma de *perfis de risco individuais*, os quais são avaliados com o objetivo de priorizar e definir quais riscos serão tratados e como serão tratados no contexto do projeto em questão. Posteriormente, em pontos definidos no processo de GRI, os riscos do projeto devem ser reavaliados.

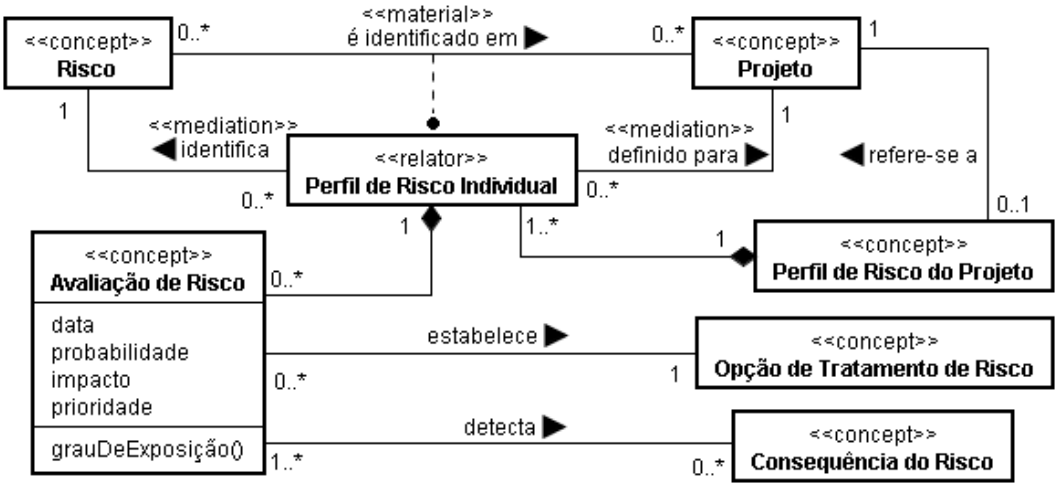

Figura 3 – Riscos de um Projeto de Software.

Tomadas em conjunto, as avaliações de um mesmo risco definem o *perfil do risco*, que registra cronologicamente a informação do estado corrente e a informação histórica associada a um dado risco em um projeto. Juntos, os perfis dos riscos definem o *perfil de risco do projeto*, que, sendo um compêndio ou agregado de todos os perfis de risco individuais do projeto, registra a informação relacionada a riscos de um projeto, corrente e histórica [ISO/IEC 2006].

Uma *avaliação de risco* 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 e qual o seu *impacto*, caso o risco venha a ocorrer. A probabilidade é a extensão até a qual o evento associado ao risco é provável de acontecer [ISO/IEC 2006]. O impacto é uma expressão quantitativa ou qualitativa da magnitude das consequências de sua ocorrência. Essas informações são muito importantes, porque, a partir delas, é possível definir o *grau de exposição de um risco*, i.e., a perda potencial que um risco pode ocasionar, que é uma função da probabilidade do risco ocorrer e de seu impacto [ISO/IEC 2006]. Com a informação do grau de exposição, é possível priorizar os riscos e estabelecer a opção de tratamento de risco a ser aplicada para o risco em questão. Opções de tratamento tipicamente citadas na literatura incluem: aceitar, monitorar, reagir, eliminar, transferir e mitigar. Por fim, quando um risco torna-se realidade, então há *consequências* concretas para o projeto. Uma consequência do risco é, portanto, o resultado da ocorrência do risco.

Assim como no caso do diagrama da Figura 2, o diagrama da Figura 3 não é capaz de capturar algumas restrições importantes, descritas a seguir na forma de axiomas:

- Se um perfil de risco de projeto *prp* refere-se a um projeto *prj*, então os perfis de risco individuais que o compõem têm de ser definidos para o mesmo projeto *prj*.

$$(\forall prp \in PerfilRiscoProjeto, prj \in Projeto, pri \in PerfilRiscoIndividual)\ (refereSeA\ (prp, prj) \wedge parteDe(pri,prp) \rightarrow ehDefinidoPara\ (pri,prj))$$

- Em um projeto *prj*, só pode haver um único perfil de risco individual *pri* identificando um risco específico *r* no contexto desse projeto.

$$(\forall pri1, pri2 \in PerfilRiscoIndividual, prj \in Projeto, r \in Risco)\ ((ehDefinidoPara\ (pr1, prj) \wedge identifica(pr1,r)) \wedge (ehDefinidoPara\ (pr2, prj) \wedge identifica(pr2,r)) \rightarrow (pr1 = pr2))$$

Gerência de Riscos de um Projeto de Software

Uma vez que tenham sido definidos quais riscos serão gerenciados, devem-se planejar as *ações* para tratá-los. A Figura 4 mostra o modelo conceitual da parte da ontologia que trata do planejamento de resposta aos riscos.

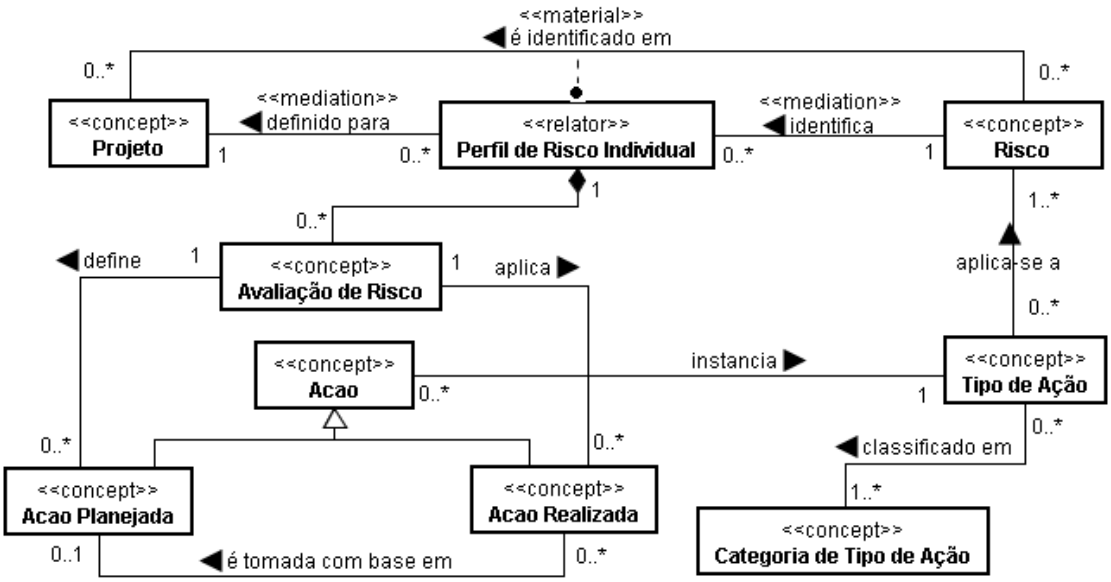

Figura 4 – Planejamento de Respostas aos Riscos.

Geralmente, organizações definem *tipos de ações* que potencialmente se aplicam para tratar certos riscos e o planejamento consiste em instanciar ações desse tipo para fazerem parte do plano de riscos do projeto. Duas categorias principais de tipos de ações são comumente encontradas na literatura: *mitigação* (tipos de ações a serem tomadas para evitar que um risco ocorra) e *contingência* (tipos de ações que serão tomadas caso um risco ocorra). O PMBOK [PMI 2008] sugere além de ações de mitigação e contingência, ações de transferência de responsabilidade.

Tipicamente, ações são planejadas levando-se em consideração o estado atual do perfil do risco, registrado na avaliação mais recente do mesmo. As ações efetivamente tomadas para tratar o risco também devem ser registradas. Uma ação realizada pode ou não ter por base uma ação planejada, ainda que idealmente deva sempre ser baseada em uma ação previamente planejada.

Além da conceituação capturada no diagrama da Figura 4, os seguintes axiomas devem ser observados:

- Se uma ação planejada *ap*, que instancia o tipo de ação *t*, é definida em uma avaliação de risco *av* que é parte do perfil de risco individual *pri* que identifica o risco *r* em um projeto *prj*, então o tipo de ação *t* deve se aplicar ao risco *r*.

$$(\forall ap \in AçãoPlanejada, av \in AvaliaçãoRisco, t \in TipoAção, r \in Risco, pri \in PerfilRiscoIndividual) ((instancia(ap,t) \wedge define(av,ap) \wedge parteDe(av,pri) \wedge identifica(pri, r)) \rightarrow aplicaSeA(t,r))$$

- Se uma ação realizada *ar*, que instancia o tipo de ação *t*, é aplicada em uma avaliação de risco *av* que é parte do perfil do risco individual *pri* que identifica o risco *r* em um projeto *prj*, então o tipo de ação *t* deve se aplicar ao risco *r*.

$$(\forall ar \in AçãoRealizada, av \in AvaliaçãoRisco, t \in TipoAção, r \in Risco, pri \in PerfilRiscoIndividual) ((instancia(ar,t) \wedge aplica(av,ar) \wedge parteDe(av,pri) \wedge identifica(pri, r)) \rightarrow aplicaSeA(t,r)$$

- Seja uma ação realizada *ar* tomada com base em uma ação planejada *ap*. Se *ap* foi definida no contexto da avaliação de risco *av1* e *ar* foi aplicada no contexto da avaliação de risco *av2*, então *av1* e *av2* devem pertencer ao mesmo perfil de risco individual *pri*.

$$(\forall ap \in AçãoPlanejada, ar \in AçãoRealizada, av1, av2 \in AvaliaçãoRisco) (tomadaComBaseEm(ar, ap) \wedge define(av1,ap) \wedge aplica(av2,ar) \rightarrow (\exists pri \in PerfilRiscoIndividual) (parteDe(av1, pri)) \wedge parteDe(av2,pri))$$

Processo de Gerência de Riscos

Assim como qualquer processo, um processo de GRI é definido para um projeto, possivelmente instanciando um processo padrão organizacional. Durante a definição de um processo de GRI para um projeto, definem-se as atividades do processo (incluindo decomposição e precedência), os tipos de recursos e papéis de pessoas requeridos por cada atividade, os procedimentos (métodos, técnicas etc) a serem utilizados na realização das atividades e os tipos de artefatos requeridos e produzidos.

Assim, a definição de um processo de GRI é, na verdade, um caso específico de definição de processos e para responder às questões de competência relacionadas a essa porção da ontologia, foi reutilizada a conceituação da ontologia de processos de software definida em [Guizzardi et al. 2008] e parcialmente apresentada na Figura 5.

Por essa figura, pode-se observar que os elementos do domínio da GRI podem ser vistos como instâncias dos elementos da ontologia de processo de software e, portanto, essa última responde às questões de competência QC12 a QC14, colocadas para a ontologia de riscos. Ilustrando, tem-se que um processo padrão de GRI é uma instância do conceito Processo Padrão. Os tipos de atividades da GRI (Identificação de Riscos, Análise de Riscos etc) são instâncias do conceito Tipo de Atividade. A ocorrência de uma atividade *a* de identificação de riscos realizada no contexto de um projeto *prj* é uma instância do conceito Atividade, que está relacionada ao tipo de atividade Identificação de Riscos. E assim por diante.

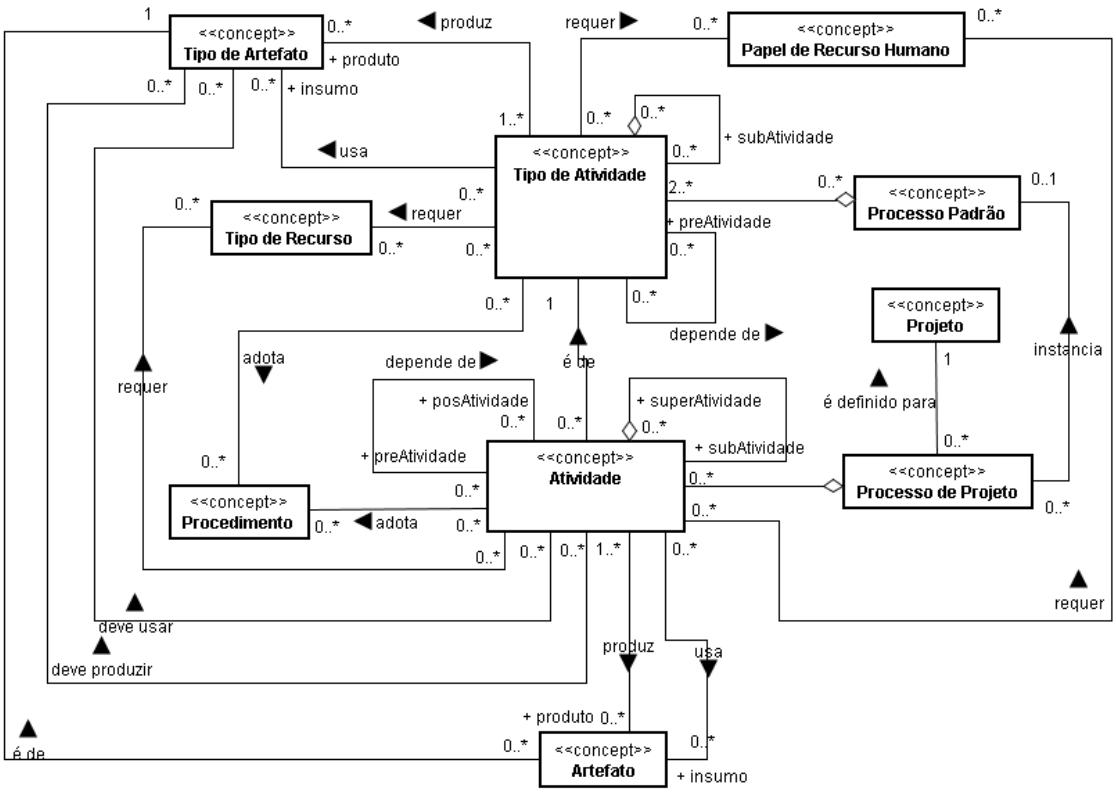

Figura 5 – A Ontologia de Processo de Software Reutilizada – Modelo Parcial.

4. Avaliação Preliminar da Ontologia Proposta

Uma importante atividade no processo de desenvolvimento de uma ontologia é a sua avaliação, o que envolve verificação e validação. Para realizar a verificação, deve-se avaliar a competência da ontologia, i.e., se a mesma é capaz de responder às questões de competência colocadas. Para realizar a validação, incluindo a validação das questões de competência, deve-se avaliar a utilidade da ontologia, tomando por base a opinião de especialistas e cenários reais.

Para avaliar a utilidade da ontologia, inicialmente a ontologia de riscos foi avaliada por especialistas que apontaram diversas necessidades de alteração. Algumas delas, após exame minucioso para definir termos e restrições, foram incorporadas à ontologia. Uma vez efetuada esta primeira etapa, passou-se à instanciação da ontologia. Instanciando a ontologia de riscos com dados de planos de risco, é possível avaliar se a ontologia é capaz de representar as situações que ela pretende descrever. Assim, a ontologia foi instanciada com dados de planos de riscos de projetos reais, sendo essa instanciação ilustrada na Tabela 1. Devido a limitações de espaço, não é possível apresentar um exemplo completo da instanciação. Entretanto, vale destacar que as várias instanciações realizadas apontaram para a utilidade da ontologia aqui proposta, conforme previamente indicado pela avaliação de especialistas.

Tabela 1 – Exemplo de Instanciação da Ontologia de Riscos de Software.

Conceito / *Relação*	**Instância**
Categoria de Risco	C1 - Riscos relativos ao produto
	C1.1 - Riscos relativos a requisitos do produto
sub-categoria	C1.1 *é sub-categoria de* C1.
Risco	R1 - Requisitos instáveis
é classificado em	R1 *é classificado em* C1.1
Tipo de Ação	TA1 – Envolver os usuários durante todo o ciclo de vida
Categoria de Tipo de Ação	Mitigação
aplica-se a	TA1 *aplica-se a* R1
Projeto	P1
Perfil de Risco Individual	R1 é identificado em P1 (Perfil R1-P1). Perfil R1-P1 *identifica* R1. Perfil R1-P1 *é definido para* P1.
Avaliação de Risco	AV1: Data: 10.07.2008; Probabilidade: 70%; Impacto: Alto (7/10); Grau de Exposição: 4,9; Prioridade: Alta.
parte de	AV1 *é parte de Perfil* R1-P1
Opção de Tratamento de Risco	OTR1: Mitigar e Reagir.
estabelece	AV1 estabelece OTR1.
Ação Planejada	AP1 - Designar o usuário José para compor a equipe e acompanhar o desenvolvimento dos requisitos.
instancia	AP1 *instancia* TA1.
define	AV1 *define* AP1.

Para se avaliar se as questões de competência foram respondidas, podem-se utilizar as instanciações feitas e ver se os conceitos, relações, propriedades e axiomas da ontologia são capazes de prover as respostas corretas. Por exemplo, a questão de competência QC2 (Qual é a categoria de um risco?) é respondida pela relação *é classificado em* entre os conceitos *Risco* e *Categoria de Riscos*, pela relação de agregação entre categorias de risco e pelos axiomas (A1) e (A2). Tomando por base o exemplo de instanciação da Tabela 1, ao se fazer essa questão para o risco R1, obtém-se como resposta as categorias C1 e C1.1. A categoria C1.1 é uma resposta direta à questão QC2, pois R1 é classificado em C1.1, como mostra a Tabela 1. Entretanto, o axioma (A1) diz que se um risco *r* é classificado em uma categoria *c2* que é subcategoria de outra categoria *c1*, então *r* também é classificado em *c1*. Logo, R1 é também classificado em C1, tendo em vista que C1.1 é uma subcategoria de C1.

Neste trabalho, esse processo de verificação da competência da ontologia foi conduzido manualmente, verificando-se cada uma das questões de competência. Contudo, para torná-lo mais ágil e menos suscetível a erros, uma boa opção é implementar a ontologia em alguma linguagem processável por máquina, tal como a OWL [Smith et al., 2004], contendo, inclusive, a instanciação da ontologia. As questões de competência poderiam ser colocadas, então, na forma de consultas SPARQL, uma linguagem de consulta para RDF [Prud'hommeaux e Seaborne, 2008], e submetidas para uma máquina de inferência capaz de processar tais linguagens, tal como *Racer* (http://www.sts.tu-harburg.de/~r.f.moeller/racer/) ou Jena (http://jena.sourceforge.net/).

Ainda que muito útil para esse e outros propósitos, vale reforçar que se entende ontologia como um modelo conceitual e não como um artefato de código processável apenas. Uma ontologia pode ser implementada em diversas linguagens e a escolha das linguagens mais apropriadas deve ser feita considerando o uso dessa implementação

para um propósito específico. Por exemplo, para a realização de anotações baseadas em ontologias feitas em planos de riscos desenvolvidos na forma de *Wikis*, uma implementação em OWL pode ser útil. Já para o desenvolvimento de ferramentas de apoio à GRI, uma implementação em Java, seguindo a abordagem proposta em [Falbo et al. 2002], é mais indicada.

5. Trabalhos Correlatos

Há diversos trabalhos relacionados ao desenvolvimento de ontologias para a Engenharia de Software, com propósitos bastante variados. Entretanto, há poucos trabalhos de ontologias sobre riscos de software. Além do trabalho usado com ponto de partida para a ontologia aqui apresentada ([Falbo et al, 2004]), foram encontrados apenas dois outros trabalhos com foco em riscos de software: [Gusmão 2007] e [Aiello et al. 2008].

Gusmão (2007) desenvolveu uma ontologia no domínio de riscos de software, denominada mPRIME *Ontology*. Segundo a autora, a mPRIME *Ontology* permite a identificação de riscos por meio do uso de um vocabulário comum que pode ser utilizado para representar conhecimento útil dentro de um ambiente de desenvolvimento de software. Assim, em termos de propósito, há muito em comum entre a ontologia apresentada neste trabalho e mPRIME *Ontology*. Contudo, as conceituações apresentadas focam aspectos diferentes. mPRIME *Ontology* teve por base o estudo e adaptação da taxonomia de riscos do SEI [Carr et al. 1993] e sua estrutura espelha bem sua origem, sendo essa ontologia decomposta em três subontologias voltadas, respectivamente, para riscos de Engenharia do Produto, Ambiente de Desenvolvimento e Restrições de Programa. A subontologia de Engenharia de Produto considera cinco classes de riscos (requisitos, design, código e teste unidade, integração e teste, e engenharia de especialidades) e sua conceituação se restringe a apontar que, se um projeto possui um risco dentro de uma dessas categorias de risco, então ele possui um risco de Engenharia de Produto. De maneira análoga, as outras duas subontologias, definem classes de riscos e indicam que se um projeto possui um risco dentro de uma dessas categorias de risco, então ele possui um risco de Ambiente de Desenvolvimento ou Restrições de Programa.

Pode-se notar que a conceituação proposta neste trabalho diverge bastante da proposta em mPRIME *Ontology*. mPRIME *Ontology* basicamente trata o conceito de risco, ainda que sem defini-lo explicitamente, e incorpora como parte da ontologia determinadas categorias de risco, oriundas, basicamente, da taxonomia de riscos do SEI. Assim, comparando com a ontologia aqui proposta, em essência, a conceituação de mPRIME *Ontology* corresponde à porção que trata de riscos e categorias de risco (parte da Figura 2). Mas, mesmo neste aspecto, ainda há outras diferenças. A ontologia de gerência de riscos proposta neste trabalho procura ser mais genérica, não se comprometendo com categorias de risco específicas. Considera-se aqui que a definição de categorias é uma decisão organizacional e que não há um consenso sobre quais devem ser as categorias a serem consideradas. No mais, todo o restante da conceituação explicitada pela ontologia apresentada neste artigo parece não encontrar correspondência em mPRIME *Ontology*.

Aiello, Nota e Gregorio [Aiello et al. 2008] propuseram uma ontologia de riscos de software para apoiar uma abordagem de gerência de riscos baseada em ontologias em

projetos distribuídos. Há conceitos de risco, classe de risco e outros relacionados a processo de software, a saber: organização, pessoa, responsável por risco, projeto, fase, atividade e produto de trabalho. Há certa correspondência entre os conceitos relativos a processos de software, mas não há, por exemplo, distinção entre tipos de atividades e atividades. De fato, há muitas diferenças e, uma vez que apenas um modelo simples é apresentado, fica a impressão que falta uma formalização ao trabalho.

Em relação à ontologia de riscos proposta em [Falbo et al. 2004], vale a pena destacar as algumas das muitas diferenças para a versão apresentada neste artigo. Primeiro, fez-se uma revisão das diversas partes da ontologia, quando foram introduzidos novos conceitos, tais como Fonte de Risco, Perfil de Risco de Projeto e Opção de Tratamento de Risco. No que se refere ao conceito de Categoria de Risco, passou-se a tratar a sua decomposição em subcategorias, adicionando a axiomatização correspondente. Algumas relações também foram revisadas, tal como a relação entre Risco e Categoria de Risco que agora indica que todo risco deve ser classificado em pelo menos uma categoria. Outro ponto a ser considerado, foi a preocupação com uma axiomatização bem mais completa, visto que o trabalho descrito em [Falbo et al. 2004] era muito pobre nesse sentido. Por fim, passou-se a tratar de aspectos relacionados ao processo da GRI por meio da integração da ontologia de riscos com a ontologia de processo de software descrita em [Guizzardi et al. 2008]. A partir do exame da distinção ontológica feita em [Guizzardi et al. 2008] entre atividade e tipo de atividade, toda a conceituação relativa às ações para tratamento de riscos foi reformulada, de modo a ficar em linha com o proposto na ontologia de processo de software utilizada.

6. Conclusões e Trabalhos Futuros

Atualmente, reconhece-se a Gerência de Riscos (GRI) como elemento de grande importância para o sucesso do desenvolvimento profissional de software. Entretanto, uma barreira para a sua efetiva disseminação em organizações de software é a existência de diversas visões parciais sobre esse domínio, cada qual adotando um vocabulário próprio, dificultando a integração entre os diversos profissionais pela ausência de padronização [Gusmão, 2007]. Este artigo apresentou uma ontologia para o domínio de riscos de software. Essa ontologia pode ser utilizada, dentre outros, como uma referência para uso de um vocabulário comum para se falar sobre esse domínio e como uma especificação reutilizável para a construção de ferramentas de apoio ao processo de GRI. Esse mesmo vocabulário pode ser usado por organizações para definir metodologias de gerência de riscos.

A ontologia apresentada neste artigo é uma evolução da ontologia proposta em [Falbo et al., 2004], a qual foi utilizada como base para a construção de uma ferramenta de apoio à GRI no ambiente ODE, denominada GeRis [Carvalho et al, 2007]. Tendo em vista que diversos aspectos foram melhorados na ontologia, está-se evoluindo também a correspondente ferramenta, de modo a adequá-la à nova conceituação. Este trabalho está em andamento e uma análise preliminar aponta para a necessidade de evoluir a ferramenta, dentre outros, nos seguintes aspectos: (i) permitir a representação de uma hierarquia de categorias de riscos; (ii) permitir o registro de fontes de risco; (iii) permitir instanciar tipos de ações em ações para tratar mais adequadamente os riscos no contexto de projetos específicos, tanto no que se refere ao planejamento das ações, quanto na efetiva realização das mesmas; e (iv) permitir definir opções de tratamento de riscos.

Está-se desenvolvendo também um sistema de gerência de conhecimento voltado para apoiar a GRI em ODE, o qual está fortemente estruturado na ontologia proposta.

É importante observar que a ontologia em si pode ser aperfeiçoada. Estudos preliminares focando a gerência de riscos em níveis mais altos de maturidade, tais como o nível C do MPS.BR e o nível 3 do CMMI, apontam para a necessidade de considerar outros aspectos da Gerência de Riscos não contemplados nesta versão da ontologia, tais como: escopo da gerência de riscos (p.ex., considerando riscos para a organização e não apenas riscos para projetos), acompanhamento das ações de tratamento de riscos até a sua efetiva conclusão, limiares para disparar ações de gerenciamento, estabelecimento de critérios para avaliar a significância de um risco, estabelecimento de medidas para monitorar riscos e relações com áreas de melhoria de processo. Neste sentido, está-se iniciando um estudo voltado para a definição de uma ontologia de riscos de software em organizações de alta maturidade. Essa ontologia deve ser uma extensão da ontologia apresentada neste artigo.

Ainda em relação à evolução da ontologia, está em curso a realização de uma análise ontológica tomando por base a Ontologia de Fundamentação Unificada (*Unified Foundational Ontology* – UFO) [Guizzardi et al. 2008]. Uma ontologia de fundamentação descreve conceitos gerais que são independentes de um problema ou domínio particular e pode ser usada para melhorar a qualidade de modelos conceituais, incluindo ontologias de domínio [Guizzardi et al. 2008]. Algumas distinções definidas em UFO, como a distinção entre relações formais e materiais, já foram consideradas nesta versão da ontologia de riscos, mas outras mais ainda serão realizadas, tal como feito em [Guizzardi et al. 2008] para a ontologia de processos de software.

Referências

Aiello, R., Nota, G., Gregorio, M.P. (2008), "Ontology Based Risk Management in Distributed Software Engineering Projects", in NEW 2008 - Decision Theory and Choice: a Complexity Approach, Salermo, Italy.

Antoniou, G., van Harmelen, F. (2004) *A Semantic Web Primer*, The MIT Press.

Carr, M. J., Konda, S.L., Monarch, I., Ulrich, F. C., Walker, C. F. (1993) Taxonomy Based Risk Identification. Tecnical Report CMU/SEI-93-TR-6. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University. USA.

Carr, M.J. (1997) "Risk Management May Not Be For Everyone", IEEE Software, May/June 1997, p. 21 – 24.

Carvalho, V.A., Coelho, A.G. N., Falbo, R.A. (2007) "Apoio Automatizado à Gerência de Riscos Cooperativa", X Workshop Iberoamericano de Ingeniería de Requisitos y Ambientes de Software - IDEAS'07, Isla de Margarita, Venezuela, p. 297-310.

Falbo, R. A. (2004) "Experiences in Using a Method for Building Domain Ontologies" Proc. of the 16th International Conference on Software Engineering and Knowledge Engineering, International Workshop on Ontology In Action, Banff, Canada.

Falbo, R.A., Guizzardi, G., Duarte, K.C. (2002) "An Ontological Approach to Domain Engineering", in Proc. of the 14th Int. Conference on Software Engineering and Knowledge Engineering, 351- 358, Ischia, Italy.

Falbo, R.A., Ruy, F.B., Bertollo, G., Togneri, D.F. (2004) "Learning How to Manage Risks Using Organizational Knowledge". 6th International Workshop on Advances in Learning Software Organizations, LSO'2004, pp. 7-18, Banff, Canada.

Guizzardi, G. (2005) *Ontological Foundations for Structural Conceptual Models*, Universal Press, The Netherlands.

Guizzardi, G. Falbo, R.A. Guizzardi, R.S.S. (2008) "Grounding Software Domain Ontologies in the Unified Foundational Ontology (UFO): The case of the ODE Software Process Ontology", Proceedings of the XI Iberoamerican Workshop on Requirements Engineering and Software Environments, Recife, Brazil.

Gusmão, C.M.G., Moura, H.P. (2004) "Gerência de Risco em Processos de Qualidade de Software: uma Análise Comparativa", III Simpósio de Brasileiro de Qualidade de Software – SBQS´2004, Brasília – DF.

Gusmão, C.M.G. (2007) "Um Modelo de Processo de Gestão de Riscos para Ambientes de Múltiplos Projetos de Desenvolvimento de Software", Tese de Doutorado, Centro de Informática, Universidade Federal de Pernambuco.

IEEE (2004) SWEBOK - Guide to the Software Engineering Body of Knowledge, 2004 Version, IEEE Computer Society.

ISO (2009) *ISO 31000 Risk Management – Principles and Guidelines*.

ISO/IEC (2006) *ISO/IEC 16085 Systems and Software Engineering – Life Cycle Processes – Risk Management*, Second edition (IEEE Std 16085-2006).

Jasper, R., Uschold, M. (1999) "A Framework for Understanding and Classifying Ontology Applications", Proceedings of the IJCAI99 Workshop on Ontologies and Problem-Solving Methods, Stockholm, Sweden.

Lister, T. (1997) "Risk Management is Project Management for Adults", IEEE Software, May/June 1997, p. 20 – 22.

PBQP (Programa Brasileiro da Qualidade e Produtividade, Subcomitê Setorial da Qualidade e Produtividade em Software) (2002) *Qualidade e Produtividade no Setor de Software Brasileiro – Pesquisa 2001.*

PMI (Project Management Institute) (2008) *A Guide to the Project Management Body of Knowledge* (PMBOK Guide), 4th edition.

Prud'hommeaux, E., Seaborne, A. (2008) *SPARQL Query Language for RDF*, W3C [Online]. Disponível em: http://www.w3.org/TR/rdf-sparql-query/.

SEI (Software Engineering Institute) (2006), *CMMI for Development, Version 1.2*, Technical Report CMU/SEI-2006-TR-008, ESC-TR-2006-008, 2006.

Smith, M.K., Welty, C., and McGuinness, D.L. (2004) *OWL Web Ontology Language Guide*, W3C [Online]. Disponível em: http://www.w3.org/TR/owl-guide/.

Softex (2009) *MPS.BR - Melhoria de Processo do Software Brasileiro, Guia Geral Versão 1.3*, Maio 2009.