

**UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA**

**ESQUEMAS DE ARGUMENTAÇÃO PARA
DISCUSSÃO DE RISCOS EM PROJETOS DE
SOFTWARE**

DISSERTAÇÃO DE MESTRADO

Ruan Carlo Bonilha Pozzebon

Santa Maria, RS, Brasil

2014

ESQUEMAS DE ARGUMENTAÇÃO PARA DISCUSSÃO DE RISCOS EM PROJETOS DE SOFTWARE

Ruan Carlo Bonilha Pozzebon

Dissertação apresentada ao Curso de Mestrado do Programa de Pós-Graduação em Informática, Área de Concentração em Computação Aplicada, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Mestre em Ciência da Computação.**

Orientador: Prof. Dr. Luís Alvaro de Lima Silva
Co-orientadora: Prof^a. Dr^a. Lisandra Manzoni Fontoura

Santa Maria, RS, Brasil

2014

Ficha catalográfica elaborada através do Programa de Geração Automática da Biblioteca Central da UFSM, com os dados fornecidos pelo(a) autor(a).

Pozzebon, Ruan Carlo Bonilha

Esquemas de argumentação para discussão de riscos em projetos de software. / Ruan Carlo Bonilha Pozzebon.- 2014.

94 p.; 30cm

Orientador: Luís Alvaro de Lima Silva

Coorientadora: Lisandra Manzoni Fontoura

Dissertação (mestrado) - Universidade Federal de Santa Maria, Centro de Tecnologia, Programa de Pós-Graduação em Informática, RS, 2014

1. Esquemas de argumentação 2. Gerenciamento de riscos de software 3. Argumentação I. Silva, Luís Alvaro de Lima II. Fontoura, Lisandra Manzoni III. Título.

© 2014

Todos os direitos autorais reservados a Ruan Carlo Bonilha Pozzebon. A reprodução de partes ou do todo deste trabalho só poderá ser feita mediante a citação da fonte.

E-mail: rb.pozzebon@gmail.com

**UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA**

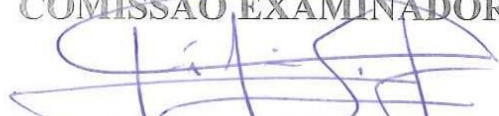
**A Comissão Examinadora, abaixo assinada,
aprova a Dissertação de Mestrado**

**ESQUEMAS DE ARGUMENTAÇÃO PARA DISCUSSÃO DE RISCOS
EM PROJETOS DE SOFTWARE**

Elaborada por
Ruan Carlo Bonilha Pozzebon

Como requisito parcial para obtenção do grau de
Mestre em Ciência da Computação


COMISSÃO EXAMINADORA:



Luis Alvaro de Lima Silva, Dr.
(Presidente/Orientador)



Leandro Krug Wives, Dr. (UFRGS)



Raul Ceretta Nunes, Dr. (UFSM)

Santa Maria, 29 de agosto de 2014.

“A persistência é o melhor caminho para o êxito”.

Charles Chaplin.

AGRADECIMENTOS

Agradeço primeiramente a Deus por me dar força e me oportunizar finalizar mais esta etapa da minha vida.

A minha mãe Cristina, por me mostrar o caminho, me ensinar a seguir em frente e ser meu grande exemplo e porto seguro.

A minha namorada Taiana por todo apoio nas horas mais difíceis, por ter permanecido ao meu lado e ter me incentivado a sempre buscar mais.

As minhas irmãs Ariane e Bruna, por servirem de grande inspiração por todas as minhas buscas e conquistas.

Aos meus orientadores, Luís Alvaro e Lisandra, que tiveram papel fundamental para que este trabalho fosse concluído. Pela oportunidade fornecida de podermos trabalhar juntos. Pelos ensinamentos, dedicação e amizade.

Aos colegas do Laboratório de Computação Aplicada (LaCA) e Grupo de Pesquisa em Engenharia de Software (PEEnSo), pela troca de experiências, convivência, amizade e apoio durante o desenvolvimento deste trabalho.

A todos os familiares e amigos, por estarem sempre na torcida e apoiarem a minha caminhada.

Ao CNPq que proveu minha bolsa de mestrado, dando o imprescindível apoio financeiro.

I would like to thank teacher Campbell for his kind comments on the improvement of this work.

Muito obrigado!

RESUMO

Dissertação de Mestrado
Programa de Pós-Graduação em Ciência da Computação
Universidade Federal de Santa Maria

ESQUEMAS DE ARGUMENTAÇÃO PARA DISCUSSÃO DE RISCOS EM PROJETOS DE SOFTWARE

Autor: Ruan Carlo Bonilha Pozzebon

Orientador: Luís Alvaro de Lima Silva

Co-orientadora: Lisandra Manzoni Fontoura

Data e Local da Defesa: Santa Maria, 29 de agosto de 2014.

Gerenciamento de riscos é um componente fundamental do gerenciamento de projetos, que tem como objetivo identificar, analisar e tratar possíveis eventos que podem comprometer o sucesso do projeto. Neste contexto, a proposição e a melhoria de técnicas que apoiem processos de Gerenciamento de Riscos é uma necessidade constante, em particular buscando padronizar e formalizar discussões colaborativas de riscos em projetos de software, por exemplo. Neste contexto, o emprego de técnicas alternativas de Argumentação na área de Inteligência Artificial aplicadas a área de gerenciamento de riscos é discutido neste trabalho. Este trabalho apresenta esquemas de argumentação para proposição e discussão de riscos, em conjunto com técnicas de jogos de diálogo, as quais visam mediar discussões colaborativas de riscos em projetos de software. Estes esquemas visam auxiliar usuários pouco experientes a propor riscos, os quais podem ser descritos por argumentos bem formados e estruturados. Além disso, o trabalho discute formas de promover a continuidade e a profundidade de uma discussão de riscos por meio do uso de um conjunto de questões críticas, as quais compõem a estrutura dos esquemas propostos. O trabalho também apresenta um processo para criação/padronização de esquemas de argumentação voltados para este domínio de aplicação. Em particular, este processo de construção apresenta passos necessários para criar esquemas de argumentos na área de gerenciamento de riscos. A partir do emprego deste processo, o desenvolvimento de esquemas pode ser melhor compreendido, permitindo a ampliação desta biblioteca de esquemas e a consequente utilização destes esquemas em discussões de riscos. Baseado nos esquemas propostos, este trabalho apresenta um sistema *web* - *Risk Discussion System (RD System v2.0)* - construído para fornecer um ambiente colaborativo para discussões de risco, onde esquemas podem ser utilizados segundo diferentes formas implementadas e testadas no nosso projeto. A fim de validar a abordagem proposta neste trabalho, dois experimentos foram realizados. Os resultados desses experimentos apresentam evidências positivas para a aceitação e a aplicabilidade da abordagem em discussões colaborativas de riscos, assim como um incremento de qualidade das informações normalmente gerenciadas por meio de outras estratégias de gerenciamento de riscos.

Palavras-chave: Esquemas de argumentação. Gerenciamento de riscos de software. Argumentação.

ABSTRACT

Master's Dissertation
Post-Graduate Program in Computer Science
Federal University of Santa Maria

ARGUMENTATION SCHEMES FOR THE DISCUSSION OF RISKS IN SOFTWARE PROJECTS

Author: Ruan Carlo Bonilha Pozzebon

Advisor: Luís Alvaro de Lima Silva

Coadvisor: Lisandra Manzoni Fontoura

Defense Place and Date: Santa Maria, August 29th, 2014.

Risk management is a key component of project management, which aims to identify, analyze and treat possible events that can compromise project success. In this context, the proposition and the improvement of technical processes that support risk management is a constant need, in particular seeking to standardize and formalize collaborative discussions of risks in software projects, for example. In this context, the use of alternative techniques of argumentation in the area of Artificial Intelligence applied to the area of risk management is discussed in this paper. This work presents schemes for argument proposition and discussion of risks, together with techniques of dialogue games, which aim to mediate collaborative discussions of risks in software projects. These schemes are designed to help novice users to propose risks, which can be described by well-trained and structured arguments. Furthermore, the paper discusses how to promote continuity and depth discussion of the risks with a set of critical issues, which make up the structure of the proposed schemes. The paper also presents a process for creation / standardization of argumentation schemes meant for this application domain. In particular, this construction process presents steps required to create arguments schemes in the area of risk management. From the use of this process, the development of regimens may be better understood, allowing the extension of this schema library and the consequent use of these schemes discussions risks. Based on the proposed schemes, this work presents a web system - Risk Discussion System (*RD System v2.0*) - built to provide a collaborative environment for discussions of risk, where different forms implemented and tested in our project can use schemes. In order to validate the approach proposed in this paper, two experiments were conducted. The results of these experiments show positive evidence for the acceptance and applicability of the approach in collaborative discussions of risks, as well as an increase in quality of the information typically managed through other risk management strategies.

Keywords: Argumentation schemes. Risk management of software. Argumentation.

LISTA DE FIGURAS

Figura 1 – Exemplo de Diagrama de causa e efeito	19
Figura 2 – Modelo padrão de argumentação de Toulmin.....	26
Figura 3 – Fragmento de uma discussão de riscos do RD System	36
Figura 4 – Regras de combinação.....	38
Figura 5 – Processo de desenvolvimento de esquemas de argumentação.	42
Figura 6 – Aplicação de um risco ao diagrama de espinha de peixe.....	44
Figura 7 – Arquitetura do RD System v2.0.	62
Figura 8 – Módulo do RD System v1.0 para manter Projetos.....	63
Figura 9 – Módulo do RD System v1.0 para manter e acessar as Discussões	63
Figura 10 – Módulo do RD System v1.0 para manter Protocolo.	64
Figura 11 – Módulo do RD System v2.0 para manter Esquemas de Argumentação.	64
Figura 12 – Submódulo do RD System v2.0 para manter Tipo de Argumento.....	65
Figura 13 – Tela do RDSsystem v2.0 para adicionar um Esquema de Argumentação a Discussão de Riscos – Forma I.	67
Figura 14 – Tela do RDSsystem v2.0 para adicionar um esquema de argumentação a discussão de riscos – Forma II.	68
Figura 15 – Tela do RD System v2.0 para uso das questões críticas	70
Figura 16 – Resultados do primeiro experimento.....	80
Figura 17 – Resultados do segundo experimento.	84

LISTA DE TABELAS

Tabela 1 –	Atos de locuções do protocolo de comunicação que determinam o início e término de uma discussão de riscos	32
Tabela 2 –	Atos de locuções específicos para mediar o desenvolvimento de tarefas de gerenciamento de riscos	33
Tabela 3 –	Atos de locuções de propósito geral que visam permitir uma discussão crítica de riscos e seus planos.....	33
Tabela 4 –	Lista de verificação (checklist) de riscos adaptados de (EMAM; KORU, 2008).....	41
Tabela 5 –	Argumento de causa e efeito. Adaptado de (WALTON; REED; MACAGNO, 2008).....	45
Tabela 6 –	Esquema de argumento sobre mudanças de requisitos do projeto.	50
Tabela 7 –	Instanciação do esquema de argumento sobre muitas mudanças de requisitos.	51
Tabela 8 –	Esquema de argumento sobre requisitos incertos.....	53
Tabela 9 –	Instanciação do esquema de argumento sobre requisitos incertos.	53
Tabela 10 –	Esquema de argumento sobre risco de software utilizado.	55
Tabela 11 –	Instanciação do Esquema de argumento sobre risco de software utilizado.	55
Tabela 12 –	Esquema de argumento sobre risco de usabilidade.	57
Tabela 13 –	Instanciação do esquema de argumento sobre risco de usabilidade.....	58
Tabela 14 –	Esquema de argumento sobre risco de tempo de capacitação.....	59
Tabela 15 –	Instanciação do esquema de argumento sobre risco de tempo.....	59
Tabela 16 –	Comparativo entre trabalhos relacionados.	76

LISTA DE ABREVIATURAS E SIGLAS

IA	–	Inteligência Artificial
TA	–	Tipos de Argumentos
<i>RD System</i>	–	<i>Risk Discussion System</i>
TAP	–	<i>Toulmin's Argument Pattern</i>
QC	–	Questão Crítica
EA	–	Esquemas de Argumentação
JD	–	Jogos de Diálogos
CRM	–	<i>Customer Relationship Management</i>
SEI	–	<i>Software Engineering Institute</i>
TBQ	–	<i>Taxonomy Based Questionnaire</i>

SUMÁRIO

1.	INTRODUÇÃO	13
1.1.	Objetivos e contribuições	15
1.2.	Organização dos capítulos	16
2.	GERENCIAMENTO DE RISCOS EM PROJETOS DE DESENVOLVIMENTO DE SOFTWARE	17
2.1.	Métodos para identificação de riscos	18
2.2.	Tipos de riscos em projetos de software	20
2.3.	Gerenciamento colaborativo de riscos em projetos de software	21
3.	ARGUMENTAÇÃO	23
3.1.	Modelos de argumentação	25
3.1.1.	Toulmin.....	25
3.1.2.	Jogos de diálogos	27
3.1.3.	Esquemas de argumentação	28
4.	CONTEXTUALIZAÇÃO DO PROJETO DE PESQUISA	31
4.1.	Jogos de diálogo para gerenciamento de riscos	32
5.	ESQUEMAS DE ARGUMENTAÇÃO PARA GERENCIAMENTO COLABORATIVO DE RISCOS	39
5.1.	Tipos de argumentos	39
5.2.	Processo de desenvolvimento de esquemas de argumentos para gerenciamento de riscos	40
5.2.1.	Selecionar o risco.....	41
5.2.2.	Elaborar o diagrama de Ishikawa	43
5.2.3.	Fazer uma análise em biblioteca de esquemas de argumentação existentes	44
5.2.4.	Definir premissas e conclusão do esquema	45
5.2.5.	Definir questões críticas para o esquema.....	46
5.2.6.	Associar o esquema de argumentos a tipos de argumentos	47
5.2.7.	Validar esquema	48
5.2.8.	Adicionar esquema de argumentos a biblioteca	49
5.3.	Esquemas de argumentação para proposição e discussão de riscos	49
5.3.1.	Esquema de argumento sobre mudanças de requisitos do projeto	50
5.3.2.	Esquema de argumento sobre requisitos incertos	52
5.3.3.	Esquema de argumento sobre risco de software utilizado.....	54
5.3.4.	Esquema de argumento sobre risco de usabilidade	56
5.3.5.	Esquema de argumento sobre risco de tempo para capacitação	58

6.	SISTEMA DE DISCUSSÃO DE RISCOS: <i>Risk Discussion System</i>	61
6.1.	O uso de esquemas de argumentação em discussões colaborativas de riscos	65
6.1.1.	Propor um novo risco na discussão usando um esquema de argumentação para gerenciamento de riscos.....	66
6.1.2.	Conectar/anotar um esquema de argumentação para gerenciamento de riscos ao texto de um risco proposto anteriormente na discussão	67
6.1.3.	Questionar a veracidade de uma proposta de risco que é construída segundo um esquema de argumentação para gerenciamento de riscos.....	68
7.	TRABALHOS RELACIONADOS	71
7.1.	Esquemas de argumentação em sistemas de segurança	71
7.2.	Esquemas de argumentação auxiliando em conflitos de banco de dados biológicos	72
7.3.	Esquemas de argumentação no auxílio a tomada de decisões de transplante de órgãos	73
7.4.	Esquemas de argumentação para gerenciamento de riscos	74
8.	EXPERIMENTOS DE VALIDAÇÃO	77
8.1.	Primeiro experimento: Cooperativa Médica – Unimed Ijuí/RS	78
8.1.1.	Configuração do experimento na Cooperativa Médica – Unimed Ijuí/RS.....	78
8.1.2.	Discussão do experimento na Cooperativa Médica – Unimed Ijuí/RS	79
8.2.	Segundo experimento: alunos da graduação em computação da UFSM	81
8.2.1.	Configuração do experimento envolvendo alunos de graduação em computação da UFSM	81
8.2.2.	Resultados do experimento envolvendo alunos de graduação em computação da UFSM	84
8.2.3.	Discussão do experimento envolvendo alunos de graduação em computação da UFSM	85
9.	CONCLUSÃO E TRABALHOS FUTUROS	87
9.1.	Considerações finais	87
9.2.	Sugestões para pesquisas futuras	89
	REFERÊNCIAS	90

1. INTRODUÇÃO

Gerenciamento de riscos em projetos de software consiste em identificar e analisar problemas potenciais e elaborar planos para preveni-los, evitando que o sucesso do projeto seja comprometido ou mesmo venha a ser cancelado (BOEHM; DEMARCO, 1997; PMI, 2013). Assim como descrito em (HALL, 1998), os fatores relacionados ao risco são: o evento ou fato que caracteriza o risco, a probabilidade de que ele irá ocorrer, o impacto que ele pode causar e a perda resultante de sua ocorrência. O risco caracteriza-se pela incerteza (visto que o evento que caracteriza o risco pode ou não ocorrer) e perda (se o evento ocorrer, consequências inesperadas ou perdas ocorrerão). Segundo Souza, Vasconcelos, Judice e Jamil (2010), uma das ferramentas mais poderosas no gerenciamento de riscos em projeto de desenvolvimento de software é o conhecimento. Assim como descrito por (NEVES et al., 2013), por exemplo, entre as estratégias que a literatura registra para gerar as condições de que as empresas de software necessitam para se manterem competitivas no mercado está a gestão e, em particular, o compartilhamento de conhecimento. Sob esses aspectos, esta dissertação discute que tarefas de gerenciamento de risco devem ser desenvolvidas de forma colaborativa, bem como com a reutilização de experiências obtidas em projetos passados. A colaboração entre os membros de um projeto de software visa garantir que diferentes pontos de vista e experiências sejam identificados e explorados, assim considerando diferentes perspectivas sobre riscos e seus planos de tratamento. Além disso, a reutilização de informações sobre riscos capturadas nessas tarefas colaborativas pode auxiliar nas etapas da gerência de riscos em projetos atuais. Por exemplo, esta reutilização pode auxiliar participantes e partes interessadas em projetos de softwares atuais a selecionar planos que tenham evitado/amenizado riscos em projetos passados, como também evitar a utilização de algum plano que tenha falhado no passado.

Este trabalho está fundamentado em um processo colaborativo de discussão de riscos. Diferente de (GREER; BUSTARD, 2002; KAYIS; ZHOU; SAVCI, 2007; MILER; GÓRSKI, 2002; PAPADAKI; POLEMI; DAMILOS, 2008), nos quais abordagens colaborativas baseadas em bate-papo e fóruns para a gestão de riscos são discutidas, é apresentada uma abordagem baseada em Argumentação para a gestão de riscos. Argumentação (BENCH-CAPON; DUNNE, 2007) pode ser sumarizada como um processo de construir e usar argumentos para suportar ou justificar uma afirmação. A finalidade básica de empregar um argumento é apresentar uma razão (ou mais que uma) para apoiar (ou derrubar) uma

afirmação que está sujeita a dúvida e, assim, eliminar essa dúvida. Recentemente, sistemas de argumentação têm sido propostos e explorados em diferentes domínios de aplicação, tais como: (MCLEOD; FERGUSON; BURGER, 2009; TOLCHINSKY; MODGIL; ATKINSON, 2012; YUAN; KELLY, 2012). Esses sistemas são fundamentados em modelos de argumentação variados - esquemas de argumentação (WALTON; REED; MACAGNO, 2008), jogos de diálogos (HADJINIKOLIS et al., 2012), Toulmin (TOULMIN, 1948), entre outros - os quais podem promover a captura e representação informal/semiformal de coleções de argumentos, bem como a representação e computação de argumentos baseados em diferentes tipos de lógica (TOLCHINSKY; MODGIL; ATKINSON, 2012). Neste contexto, sistemas de argumentação estão gradualmente promovendo o projeto e construção de memórias de argumentos (ou bases de argumentos, em contraste com bases de conhecimento representadas por meio de regras de produção, por exemplo) em diferentes domínios de aplicação.

Assim como observado na pesquisa desenvolvida nesta dissertação, diferentes argumentos apresentados por participantes de projetos de software em debates de gestão de riscos muitas vezes são apresentados de forma incompleta, onde partes importantes da estrutura dos argumentos estão implícitas. Em muitos casos, tais argumentos são apresentados de forma bastante informal, onde participantes de uma discussão de gestão de riscos não conseguem compreender a sua natureza/ideia, além de não conseguirem identificar quais etapas da gestão de riscos estão sendo executadas nestes debates. Visando propor solução para parte deste problema, a abordagem de argumentação para a gestão colaborativa de riscos discutida neste trabalho está fundamentada na noção de “jogos de diálogo” (SEVERO; FONTOURA; SILVA, 2013; SEVERO *et al.*, 2013). Estes modelos dialéticos de argumentação são estruturas de representação do conhecimento que visam identificar e representar passos significativos de interação, os quais são típicos de debates envolvendo dois ou mais agentes, sejam estes humanos ou computacionais. Além de explorar a noção de jogos de diálogo, o processo de apresentação e análise de argumentos em discussões de riscos é investigado segundo a noção de “esquemas de argumentos” (REED; WALTON, 2007). Na literatura de Argumentação em Inteligência Artificial (IA), estes esquemas visam capturar e descrever claramente (e de forma explícita) a estrutura de argumentos típicos encontrados em trocas de diálogos (WALTON, 2012). Buscando investigar este modelo de argumentação em um novo domínio de aplicação e pesquisa, este trabalho demonstra que esquemas voltados para o gerenciamento de riscos podem ser usados para auxiliar o processo colaborativo entre os membros de uma equipe de desenvolvimento de software. Neste sentido, o trabalho propõe a construção e utilização de um conjunto de esquemas de argumentação para suporte a

identificação de riscos em atividades colaborativas de análise de riscos em projetos de software. O trabalho demonstra como estruturar e formular esquemas neste domínio de aplicação, assim permitindo que usuários possam inserir ou alterar esquema de argumentos armazenados em uma memória de argumentos reusável. A principal ideia dessa memória é armazenar esquemas de argumentos padronizados, ou *templates* para argumentos típicos empregados em discussões de riscos. Assim como investigado neste trabalho, tais esquemas podem ser consultados e reutilizados por diferentes participantes de um projeto porventura envolvidos em tarefas de análise de riscos em projetos futuros.

Esta dissertação também apresenta uma nova versão de um sistema de discussão de risco baseado na *web*: o *Risk Discussion System*. Este sistema oferece um ambiente colaborativo onde os diferentes membros de um projeto de software podem interagir e, sistematicamente, registrar discussões de risco em uma memória, permitindo futura reutilização dessa informação, assim como descrito em (SEVERO; FONTOURA; SILVA, 2013). Na versão *RD System v2.0*, assim como produzida nesta dissertação, o sistema traz módulos para o gerenciamento de tipos de argumentos e esquemas de argumentação variados. Baseado nesta base de conhecimento envolvendo esquemas de argumentação, a dissertação discute como empregar tais esquemas no auxílio ao processo de proposição e discussão de riscos em meio a um debate colaborativo de riscos de projetos de software.

1.1. Objetivos e contribuições

O objetivo deste trabalho é descrever formas de construir esquemas de argumentação voltados para a determinação de riscos em atividades de gerenciamento de riscos. Baseado nestes esquemas, os quais são organizados em uma biblioteca, o trabalho busca identificar e descrever como usuários podem manter tais esquemas, bem como eles podem ser empregados em discussões colaborativas de riscos de projetos de software. Em geral, a ideia do trabalho é investigar esquemas de argumentação, vista como uma técnica da Argumentação em Inteligência Artificial, no contexto do melhor desenvolvimento de tarefas de gerenciamento de risco de projetos de software.

Uma contribuição desta dissertação é demonstrar como construir e explorar esquemas de argumentação para gerenciamento de riscos. A dissertação descreve modos de utilizar estes esquemas em meio a discussões de riscos, assim como modeladas no ambiente do *RD System*

v2.0. Outra contribuição desta dissertação é discutir a integração de duas técnicas de argumentação distintas, as quais são esquemas de argumentação e jogos de diálogos. A partir dessa integração, a utilização destas técnicas visa permitir a aquisição e representação de diferentes tipos de argumentos naturalmente apresentados por participantes de projetos de software em discussões colaborativas de riscos. Além disso, a dissertação apresenta um processo de criação de esquemas de argumentação, buscando descrever como engenheiros de software/engenheiros de conhecimento podem elaborar esquemas de argumentação a serem explorados em sistemas colaborativos no domínio de gerenciamento de riscos em projetos de software.

1.2. Organização dos capítulos

O texto desta dissertação está organizado da seguinte forma.

No Capítulo 2 são introduzidos conceitos relacionados à disciplina de Gerenciamento de Riscos, também é abordado o funcionamento de um processo colaborativo de gerenciamento de riscos.

No Capítulo 3 é apresentado conceitos de Argumentação em Inteligência Artificial. O capítulo também discute modelos de argumentação, dando o enfoque ao modelo de esquemas de argumentação, assim como explorado nesta dissertação.

No Capítulo 4 é apresentado uma contextualização do projeto de pesquisa onde esta dissertação está inserida.

No Capítulo 5 é apresentada e discutida uma biblioteca de esquemas de argumentação, o seu processo de criação e uso em atividades colaborativas de discussão de riscos, como também os tipos de argumentos usados na indexação dos esquemas desenvolvidos.

No Capítulo 6 é descrito o sistema para discussão de riscos de software implementado, *Risk Discussion System*.

No Capítulo 7 é discutido trabalhos relacionados a esta dissertação, comparando abordagens, demonstrando os aspectos comuns e distintos da abordagem proposta.

No Capítulo 8 é apresentado e analisado os resultados obtidos com o *RD System* em cenários de testes propostos.

No Capítulo 9, por fim, é apresentada conclusões desta dissertação e perspectivas para trabalhos futuros.

2. GERENCIAMENTO DE RISCOS EM PROJETOS DE DESENVOLVIMENTO DE SOFTWARE

Para (STERN; ARIAS, 2011), um projeto visa proporcionar algum tipo de benefício ou vantagem a uma organização. Apesar dos benefícios potenciais que o gerenciamento de projetos pode proporcionar, estes podem não ser alcançados sem que haja a transposição de barreiras tais como: a complexidade do projeto, as mudanças de escopo do projeto, as mudanças de tecnologias e os riscos (PMI, 2012).

Segundo (BOEHM, 2008; SUN, 2009), um risco pode ser qualquer evento indesejável associado a equipe, ao ambiente externo e com o trabalho em um projeto. Geralmente, riscos são produtos de dois fatores: os impactos esperados de um evento e a probabilidade de que o evento possa ocorrer. Inicialmente, um processo de gestão de riscos pode apresentar informações confusas e duvidosas, uma vez que, tudo no início do projeto é incerto (BOEHM, 2010; KWAN; LEUNG, 2011).

O processo de gerenciamento de riscos é considerado uma das principais áreas do gerenciamento de projetos, embora ainda não seja executada pela maioria das equipes/empresas desenvolvedoras de software. Gerenciamento de riscos (ALBERTS; DOROFEE, 2010) tem como objetivo ajudar uma equipe de desenvolvimento de software a entender e gerenciar incertezas, as quais são comuns em um processo de desenvolvimento de sistemas. Este processo de gerenciamento de riscos engloba ações como identificação dos eventos de risco, determinação de níveis para probabilidades e impactos de eventos de riscos em um determinado projeto, bem como o desenvolvimento de respostas para eliminar, reduzir ou minimizar estes eventos, caso eles ocorram, permitindo um controle do processo de gerenciamento de riscos como um todo.

Em geral, riscos podem capturar eventos que tendem a ocorrer no futuro. Caso estes riscos sejam concretizados, eles se transformam em problemas que afetam o projeto de software. Para evitar a concretização destes riscos, é necessário a identificação destas incertezas no projeto, o planejamento de como tais incertezas podem ser monitoradas e as contramedidas a serem tomadas caso elas ocorram. Assim como descrito em (ALBERTS; DOROFEE, 2010; BOEHM, 1991), o gerenciamento de riscos prevê um conjunto de atividades que tem como objetivo o planejamento e controle destas incertezas. Em geral, tarefas de gerenciamento de riscos focalizam i) a identificação de riscos, ii) a análise dos

riscos mais relevantes, iii) o planejamento de respostas para riscos selecionados, iv) o monitoramento e controle dos riscos de um projeto.

Existem na literatura diversas propostas de processos de gerenciamento de riscos. (BOEHM; DEMARCO, 1997; PMI, 2013) apontam atividades típicas que compõem um processo de gerenciamento de riscos: planejamento do gerenciamento de riscos, identificação de riscos, análise de riscos (envolvendo a determinação de probabilidade e impacto de riscos identificados), planejamento de respostas aos riscos priorizados (isto é, riscos com probabilidade e impacto altos), monitoramento de riscos, controle de riscos e comunicação de riscos. É importante ressaltar que diferentes autores, tais como (ALBERTS; DOROFEE, 2010; PMI, 2012), discutem que a grande dificuldade atual para a gerenciamento de riscos em projetos de software é a existência de diversas visões/pontos de vista sobre esse domínio. Cada visão tem por si só um modo de discussão e exposição de riscos, dificultando a integração entre os diversos profissionais pela ausência de uma padronização.

2.1. Métodos para identificação de riscos

Existe uma grande variedade de métodos para a identificação de riscos. Dentre outros, podemos citar: listas de verificação (*checklist*), entrevistas, comparação análoga, análise causal e *brainstorm*. Contudo, existe um consenso que a melhor abordagem para a identificação precisa de riscos é dada usando uma combinação de vários métodos.

- Listas de Verificação (*checklists*): nesse método, os *stakeholders* (usuários envolvidos no projeto) utilizam listas de riscos pré-definidas para ajudar a identificar potenciais riscos de um projeto. Estas listas podem ser desenvolvidas com base em informações históricas e no conhecimento acumulado a partir de projetos anteriores, bem como a partir da análise de fontes de informações variadas (CARR et al., 1993). Ao final de cada projeto, a lista de verificação de riscos pode ser revisada, para que seu uso seja aperfeiçoado em projetos futuros (PMI, 2013).
- Entrevistas com especialista: podem ser realizadas com especialistas no assunto, os quais podem ser participantes experientes no projeto e *stakeholders*. Uma das vantagens desse método é a capacidade de obter diversas visões sobre os riscos, pelo fato dos entrevistados, normalmente, apresentarem perfis e experiências

diferentes. Um questionário pode ser preparado na condução de tais entrevistas, mas deve-se ter o cuidado ao definir as perguntas de modo que estas perguntas não venham a limitar uma possível troca de idéias.

- Comparação análoga: esse método está fundamentado na idéia de que nenhum projeto representa um sistema totalmente novo, sendo possível encontrar entre sistemas já desenvolvidos características em comum com o novo sistema, tais como: tecnologias, funcionalidades, processo de desenvolvimento, clientes (CARR et al., 1993). Sendo assim, os riscos de um projeto também podem ser identificados com base em projetos similares.
- Análise causal: por meio dessa análise, são investigadas as causas principais dos riscos de um projeto. Esta análise está fundamentada na ideia de que se um erro ocorreu, ele irá acontecer novamente, a menos que algo seja feito para evitá-lo (HALL, 1998). A ferramenta mais utilizada para fazer este tipo de análise causal é o Diagrama de Ishikawa, também conhecido como Diagrama de Causa e Efeito ou Diagrama Espinha-de-peixe, cuja estrutura é representada na Figura 1. A partir da identificação das causas de um risco, é possível planejar respostas eficazes (planos) para tratá-lo.

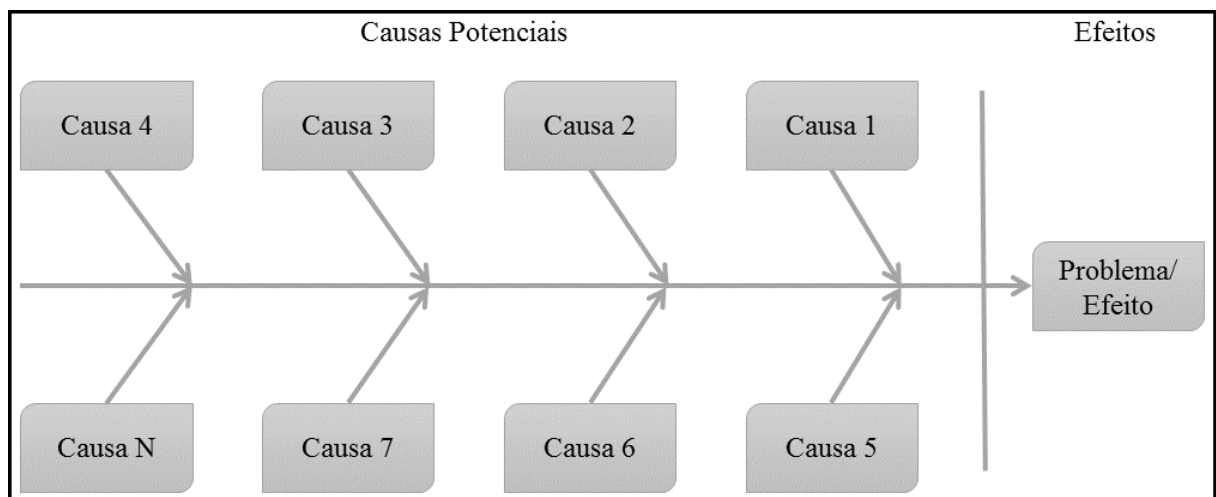


Figura 1 – Exemplo de Diagrama de causa e efeito

Fonte: Adaptado de Hall (1998).

- *Brainstorm*: método convencional, em que um facilitador pode reunir a equipe do projeto, bem como um conjunto multidisciplinar de especialistas que não fazem parte da equipe, para discutir ideias sobre os riscos de um projeto (PMI, 2013).

2.2. Tipos de riscos em projetos de software

Diversos tipos de riscos que podem afetar projetos de desenvolvimento de software foram propostos na literatura de Engenharia de Software. Uma breve análise de um conjunto de tipos de riscos torna evidente a diversidade de fontes de incerteza que fazem parte de um projeto. Esta diversidade dificulta a análise de riscos, exigindo a criação de processos e sistemas para facilitar e auxiliar esta tarefa.

Pressman (2007) classifica os riscos de um projeto de desenvolvimento de software em três grupos: riscos de projeto, riscos de negócio e riscos técnicos. Os riscos de projeto se referem à equipe de desenvolvimento, custos, cronograma, recursos, relações com clientes e problemas com os requisitos do software. Os riscos de negócio se referem a mudanças nas estratégias de mercado da empresa, a falta de apoio da gerência e a falta de apoio financeiro ao projeto. Por fim, os riscos técnicos se referem a problemas que possam vir a ocorrer durante o processo de desenvolvimento de software, como desempenho inadequado, falta de experiência e conhecimento técnico da equipe de desenvolvimento, entre outros.

Para facilitar a identificação de riscos em projetos de desenvolvimento de software, o SEI – *Software Engineering Institute* – definiu uma taxonomia para identificação de riscos. A taxonomia provê uma estrutura para identificação de riscos técnicos no desenvolvimento de software (CARR et al., 1993). A taxonomia é dividida em três tipos: produto, ambiente e organização. Os riscos de produto envolvem aspectos técnicos do trabalho a ser realizado. Os riscos de ambiente envolvem os métodos, ferramentas e técnicas utilizadas no projeto. Os riscos de organização envolvem restrições externas a que o projeto está sujeito, sejam elas contratuais, organizacionais ou operacionais.

Cada tipo de risco contém um conjunto de elementos que, por sua vez, são divididos em atributos. Os atributos servem como guia para formulação de um questionário para identificação de riscos, o TBQ – *Taxonomy Based Questionnaire*. Este questionário é especializado para cada projeto, removendo-se as perguntas referentes a atributos que não são

relevantes no contexto do projeto (CARR et al., 1993). O TBQ está organizado em dois níveis. O primeiro nível contém perguntas abrangentes, cujas respostas selecionam perguntas mais específicas do segundo nível. Este filtro permite manter o foco nos aspectos relevantes de um projeto, sem perder a visão em suas outras características. As perguntas do questionário contêm listas de exemplos que definem seu escopo. Estes exemplos facilitam a comunicação com o usuário, reduzindo a chance de interpretação incorreta das perguntas.

HALL (1998) também separa os riscos de desenvolvimento de software em três tipos: riscos de projeto, riscos de processo e riscos de produto. Os riscos de projeto compreendem problemas contratuais, relações com fornecedores, restrições de recursos e de interfaces externas à organização de desenvolvimento. Os riscos de processo compreendem problemas de planejamento, equipe, acompanhamento de projeto e controle de qualidade. Os riscos de produto estão relacionados com a estabilidade dos requisitos, desempenho, complexidade e confiabilidade. Hall utiliza estes tipos para dividir as responsabilidades por cada risco dentro de um projeto. Assim, os riscos de projeto são delegados para a equipe de gerenciamento, enquanto os riscos de produto são delegados para a equipe de desenvolvimento. Já os riscos de processo são compartilhados pelas duas equipes.

Os tipos de riscos discutidos servem como referência para os riscos que podem ser encontrados em um projeto de software. Eles também servem como um alerta para a diversidade destas incertezas. HALL (1998) mostra a dependência existente entre os riscos considerados relevantes por gerentes e os contextos em que se realizam seus projetos. Tal contexto é definido pelos elementos envolvidos no projeto, tais como seu domínio de aplicação, as tecnologias utilizadas, as características da equipe, entre outros. Riscos que podem afetar um projeto também devem ser associados a estes elementos contextuais. Esta associação, além de servir como classificação, pode auxiliar na reutilização de riscos, determinando os riscos relevantes para um projeto a partir dos elementos que o compõem.

2.3. Gerenciamento colaborativo de riscos em projetos de software

O uso de colaboração durante a realização de atividades relacionadas ao gerenciamento de riscos é abordado em trabalhos como: IRMAS (KAYIS; ZHOU; SAVCI, 2007), RA!SE (GREER; BUSTARD, 2002) e RiskGuide (MILER; GÓRSKI, 2002). Estes trabalhos propõem ambientes colaborativos para o registro e consulta de informações

relacionadas ao gerenciamento de riscos. Em geral, tais trabalhos buscam melhorar a visibilidade, por parte da equipe, em relação aos riscos de projetos. Porém, estes sistemas estão limitados a oferecer ambientes colaborativos somente voltados para o registro de informações (catalogação, arquivamento, armazenamento em banco de dados) sobre os riscos de projetos.

Embora relevantes para a compreensão de diálogos em diferentes áreas de aplicação, assim como descrito em (MCBURNEY; HITCHCOCK; PARSONS, 2007; REED; WELLS, 2007), as diferentes interações que ocorrem entre os participantes de projetos enquanto essas informações são elicitadas em processos de discussão de riscos não são exploradas, sendo perdidas, visto que elas não são consideradas.

O trabalho proposto por (PAPADAKI; POLEMI; DAMILOS, 2008) apresenta um sistema de gerenciamento de riscos voltado para a análise de segurança de informação. Neste sistema, informações sobre o processo de análise de riscos são apresentadas em um fórum de discussão, onde projetos e riscos de segurança são organizados em tópicos. Apesar deste trabalho considerar alguns tipos de interações entre os usuários, a abordagem colaborativa descrita não é fundamentada em nenhuma teoria da argumentação como forma de adquirir e representar essas discussões. Em particular, um fórum pode registrar informações muitas vezes desestruturadas, onde o foco de discussões pode ser perdido neste registro, uma vez que diferentes assuntos podem ser intercalados na discussão.

3. ARGUMENTAÇÃO

Argumentação é um assunto amplo, pertencente em parte a Inteligência Artificial. Em geral, a argumentação pode ser usada como um método para localizar, destacar e organizar a informação relevante a favor e contra as afirmações propostas em debates (BENCH-CAPON; DUNNE, 2007; GRASSO et al., 2010). Argumentação, entre outros significados, indica que algo deve estar claramente descrito, ou simplesmente deixar claro. Entre outras áreas da Argumentação, o seu estudo investiga problemas enfrentados por um grupo de pessoas dentro de um debate. Este estudo tenta definir esquemas e ferramentas que auxiliem o processo de entendimento e participação de discussões.

De acordo com (EEMEREN; GROOTENDORST, 2004), argumentação é a forma verbal e a atividade social da razão que visa aumentar (ou diminuir) a aceitabilidade de um ponto de vista. Tal ponto de vista pode ser muitas vezes contraditório entre as partes de um diálogo. Neste contexto, é importante ressaltar que técnicas de argumentação buscam capturar e modelar o processo de proposição e análise de argumentos, os quais podem justificar (ou refutar/derrubar) pontos de vista apresentados por participantes de debates.

Entre outros objetivos, técnicas de argumentação em ciência da computação buscam permitir a construção de programas capazes de capturar e computar dados e conhecimento oriundos de processos de diálogo, tal como se tais discussões fossem realizadas por agentes computacionais ou seres humanos. O desenvolvimento de sistemas de argumentação tornou-se um tema de pesquisa bastante ativo na Inteligência Artificial, que inclui atividades de pesquisa como o desenvolvimento de modelos teóricos e implementação de sistemas inteligentes em diferentes áreas de aplicação. Conforme (REED; WALTON; MACAGNO, 2007; TONIOLO; NORMAN; SYCARA, 2012), a argumentação é útil em cenários onde informações são incertas ou incompletas, por exemplo, no contexto de gerência de riscos em projetos de software conforme proposto nesta dissertação.

De acordo com diferentes modelos de argumentação, participantes de uma discussão podem coordenar os seus atos de comunicação pelo estabelecimento de rotinas argumentativas. Tais rotinas determinam quem pode falar, quem deve escutar, que tipos de argumentos são permitidos, como resolver conflitos, entre outros. A tecnologia de argumentação usada por um grupo de pessoas introduz funcionalidades que induz os mesmos a realizarem diferentes tipos de interações. Neste caso, tais interações de argumentação podem ser restritivas ou preventivas (BENCH-CAPON; DUNNE, 2005). As interações

restritivas limitam o escopo de ação de uma determinada interação, reduzindo a quantidade de opções de argumentação disponíveis a um número pré-definido. As interações preventivas tentam evitar que determinado padrão de comportamento aconteça, permitindo que a discussão se desenvolva sem que ocorra um comportamento indesejável muitas vezes considerado incorreto.

É essencial selecionar a tecnologia argumentativa que auxilie um grupo de pessoas a alcançar seus objetivos, e que ao mesmo tempo diminua as indesejáveis limitações que a tecnologia pode colocar no processo de argumentação. No entanto, esta tarefa de seleção não é trivial (RAHWAN, 2008; TOLCHINSKY; CORTES, 2006). Entre outros motivos, não é simples a elaboração de uma tecnologia/ferramenta que absorva grande parte da complexidade que os modelos de argumentação possui e os exponha de maneira fácil para que um usuário consiga utilizá-los nas suas discussões. Um aspecto importante geralmente tratado na área de argumentação é o processo de tomada de decisões em grupos. Este processo envolve múltiplos agentes buscando tomar decisões, onde estes agentes possivelmente possuem diferentes níveis de conhecimento, experiências, preferências e perspectivas do problema em análise. No decorrer de um processo de decisão é frequente ocorrerem as mais diversas formas de desentendimento e de conflito, sendo necessário ultrapassá-los para se conseguir obter consenso. Neste caso, a argumentação pode ser compreendida como um mecanismo que investiga e auxilia a tomada de decisão em situações de conflito, como também possibilita a troca de informações em situações em que agentes deliberam e cooperam na tomada de uma decisão. Em particular, um argumento é uma peça de informação que permite a um agente (PARSONS, 1998):

- Justificar as suas opções – um agente, ao tomar determinadas decisões, certamente tem razões que as suportam. Neste caso, o envio de argumentos de suporte permite que os demais participantes de uma discussão disponham de mais informação para julgarem decisões selecionadas.
- Influenciar terceiros – um participante em processos de tomada de decisão em grupo tenta convencer os demais participantes acerca da qualidade de sua proposta, construindo argumentos que levem os mesmos a concordar com o seu ponto de vista.

O processo de tomada de decisão por meio da troca de argumentos, segundo (PARSONS, 1998), tem o potencial de aumentar a probabilidade de se atingir um acordo, o qual pode ser alcançado de uma forma mais eficaz. Como argumentação é utilizada no nosso

dia a dia para justificar uma escolha, ela também auxilia na tomada de decisão em grupo. A argumentação, portanto, pode ser apresentada como uma forma natural de tornar claras e justificar as escolhas realizadas e de convencer outros em uma discussão.

3.1. Modelos de argumentação

Esta seção traz uma breve revisão de argumentação com base no modelo argumentativo de Toulmin, visto que este é o modelo de argumentação mais conhecido e usado em diferentes áreas de aplicação e pesquisa. Este capítulo também apresenta uma breve revisão do modelo de Jogos de Diálogos, que por sua vez busca explorar processos de diálogo de modo a tornar tais processos estruturados e organizados. Por fim, uma revisão do modelo de Esquemas de Argumentação é realizada, visto que este modelo visa facilitar o processo de exposição de raciocínio/argumentos sobre um determinado assunto.

3.1.1. Toulmin

Stephen Toulmin (1948), estudando a estrutura e validade de argumentos em diferentes áreas, concluiu que há alguns elementos presentes na argumentação, os quais são comuns a variados campos do conhecimento. Assim como descrito por Toulmin, esses campos foram descritos no TAP (*Toulmin's Argument Pattern*) (Figura 2).

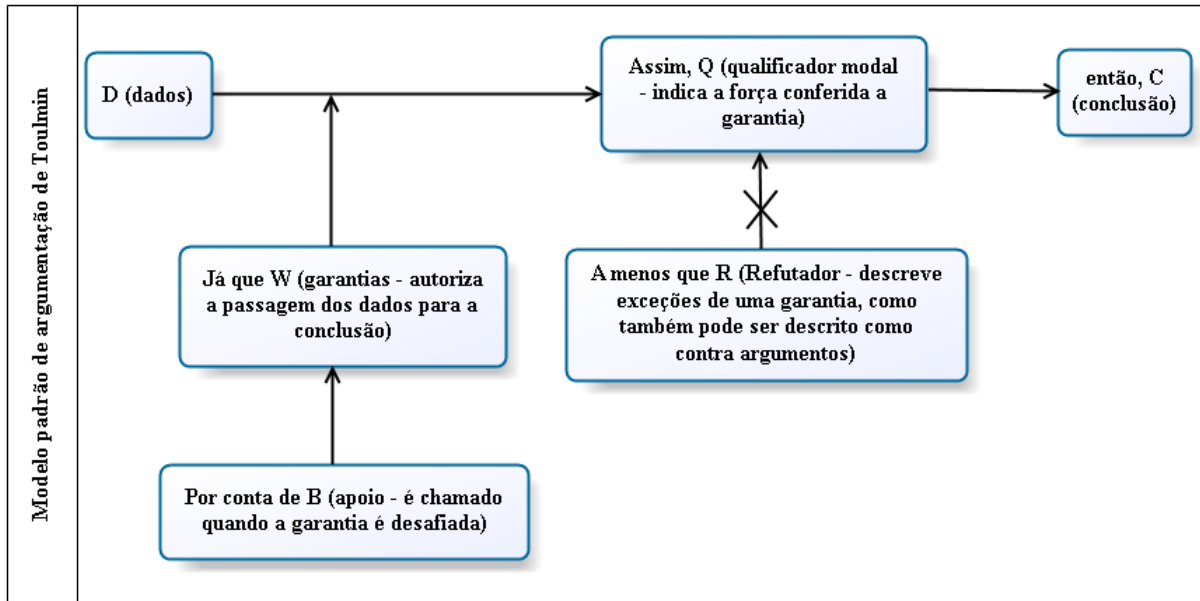


Figura 2 – Modelo padrão de argumentação de Toulmin

Baseado no padrão de argumentação apresentado por Toulmin (TOULMIN, 2003), questões diversas podem ser expressas. Entre elas, podemos citar: “O que está envolvido no processo de estabelecer conclusões mediante a produção de argumentos?” Para refletir sobre essa questão, o autor busca tornar explícito os “elementos” constituintes de seu modelo padrão de argumentação. Segundo Toulmin, é preciso de fatos (dados D) para apoiar uma afirmação. Neste caso, cabe a nós recorrer àqueles fatos e apresentá-los como fundamento no qual se baseia nossa alegação a conclusão (C), a qual os méritos nós procuramos estabelecer. Podemos inclusive ressaltar que o processo (se D então C) pode ser expandido, com vantagem, a favor da imparcialidade. A esta expansão, foi introduzido de um novo “elemento”, ao qual Toulmin chamou garantias – proposições (W). Segundo Toulmin, talvez seja necessário acrescentar alguma referência explícita ao grau de forças que os dados conferem à alegação, vistas a garantia. Neste sentido, Toulmin apresenta a necessidade de inserir um qualificador modal (Q). Encontramos ainda no modelo de Toulmin condições de exceção ou refutação (R). Estas são circunstâncias nas quais alguém tem que deixar de lado a autoridade geral das garantias. Em suas assertivas, Toulmin enfatiza que por trás das garantias normalmente “haverá outros avais, sem os quais nem as próprias garantias teriam autoridade ou vigência”, assim chama estes avais de apoio (B) – campo variante. Em resumo, é importante observar que um argumento pode ser descrito e apresentado de forma completa a partir da apresentação/identificação das suas várias partes constituintes. Muitas vezes, tais

elementos de um argumento estão implícitos, visto que não são apresentados ou são apresentados de forma pouco clara.

3.1.2. Jogos de diálogos

Em argumentação, a estruturação e representação de diálogos podem ser realizadas por meio de modelos de jogos de diálogo (TOLCHINSKY; MODGIL; ATKINSON, 2012). Um jogo de diálogo é uma técnica de argumentação que permite representar conhecimento normalmente oferecido em um diálogo e/ou debate. Entre outros elementos, um jogo de diálogo descreve o início e o fim de um debate (por exemplo, Iniciar Discussão e Terminar Discussão, assim como proposto por (SEVERO; FONTOURA; SILVA, 2013)), um conjunto de atos de locuções ou regras voltadas para mediar a comunicação entre os participantes (por exemplo, Propor Risco, Propor Plano, Propor Probabilidade, entre outros (SEVERO; FONTOURA; SILVA, 2013)) e regras de combinações entre estes atos de locução. Por exemplo, Propor Plano, Impacto, Probabilidade só podem ser utilizados em uma discussão de riscos de software após a proposição de um risco, a qual é realizada por meio da locução Propor Riscos, conforme definido por (SEVERO; FONTOURA; SILVA, 2013). (MCBURNEY; PARSONS, 2003). Em geral, um conjunto de atos de locuções e suas combinações possibilitam capturar e organizar uma discussão em um formato de árvore (ver Figura 3). A hierarquia desta árvore permite visualizar o encadeamento dos argumentos apresentados pelos participantes envolvidos na discussão, bem como a dependência entre os diferentes atos de fala usados pelos participantes envolvidos no debate.

Em um jogo de diálogo, as locuções escolhidas como válidas (permitidas em um diálogo) e suas regras de combinações permitem mediar discussões na medida em que os participantes apresentam suas contribuições no desenvolvimento de um diálogo. As regras de interação definidas no jogo de diálogo são relevantes para controlar as possíveis interações entre os agentes. Elas podem ser usadas para validar estas interações e auxiliar na formulação de respostas, chamando a atenção de participantes para a utilização de um conjunto de movimentos de fala permitidos no diálogo. Em geral, jogos de diálogo permitem estruturar e guiar o processo de diálogo, tal como, por exemplo, guiar um processo de interação baseado em perguntas e respostas, assim como descrito em (KARUNATILLAKE; JENNINGS, 2009).

Para a Inteligência Artificial, jogos de diálogo, segundo (KARUNATILLAKE; JENNINGS, 2009), oferecem uma forma de combinação entre algum conteúdo linguístico, o qual deve ser capturado na construção de um sistema inteligente, e uma restrição do papel que este conteúdo desempenha em processos de discussão, a ser modelado e implementado nestes sistemas. Apesar da pesquisa em jogos de diálogo ter um foco predominante em jogos envolvendo uma dupla de participantes, há um crescente interesse na modelagem e interação de tarefas colaborativas de argumentação envolvendo múltiplos participantes. Para melhor entender o funcionamento e estrutura de um jogo de diálogo, ver a Figura 3 como também as tabelas (Tabela 1; Tabela 2; Tabela 3), na seção 4.1. Devido a relevância desta técnica de argumentação no nosso projeto de pesquisa, esta seção visa contextualizar jogos de diálogo na representação de atividades colaborativas de discussão de riscos.

3.1.3. Esquemas de argumentação

Segundo Walton (WALTON; REED; MACAGNO, 2008; WALTON, 2005), esquemas de argumentos são formas de inferência a partir de premissas para uma conclusão. Esquemas buscam capturar argumentos do cotidiano encontrados em trocas de diálogos. A ideia de muitos esquemas é descrever formas de argumento em que uma pessoa em uma discussão tenta convencer outra a aceitar uma conclusão que está em questão, ou mesmo explicar uma conclusão sendo discutida.

Em geral, a estrutura em que esquemas de argumentação são descritos visa facilitar e padronizar a exposição do raciocínio ou “fala” de um indivíduo em meio a uma discussão. Neste caso, esquemas são descritos em uma estrutura definida por meio de i) uma ou várias premissas, seguidas de ii) uma conclusão e de iii) um conjunto de questões críticas. Uma vez que um processo de raciocínio seja descrito neste formato, ele tende a levar os participantes de um diálogo a aceitar uma conclusão, a qual parte da contextualização das premissas de um argumento.

É importante ressaltar que esquemas descrevem argumentos que podem ser considerados como verdadeiros ou falsos em um determinado contexto. Dessa forma, podem ser levantadas questões sobre a veracidade de argumentos apresentados. Estas questões são denominadas Questões Críticas – QC. Entre outros usos, QC podem ser utilizadas para fazer a validação de argumentos que são propostos de acordo com um esquema selecionado. Neste

caso, se as questões críticas não puderem ser respondidas em uma discussão, um argumento em questão pode ser “derrubado”. Como resultado, o argumento não é mais verdadeiro no contexto da discussão. Assim, o emprego de QCs durante as discussões pode garantir ou melhorar a qualidade de uma discussão, pois a utilização de questões críticas permite testar argumentos propostos. Desta forma, os argumentos que irão seguir na discussão podem ser considerados confiáveis e verdadeiros no contexto de um debate.

Um exemplo de um esquema que podemos citar é um “Argumento da Posição do Saber” (WALTON; REED; MACAGNO, 2008):

- Premissa 1: α está em condições que permitem saber se A é falsa ou verdadeira.
- Premissa 2: A afirma que A é verdadeira (falsa).
- Conclusão: É plausível que A seja verdadeiro (falsa).

O esquema de argumento “Argumento da Posição do Saber” pode ser compreendido da seguinte forma. Em princípio, podemos assumir que alguma pessoa estava em um local privilegiado para ver algo. Logo, essa pessoa diz claramente que sabe algo porque ela estava na hora e local do acontecimento. Este esquema pode auxiliar em uma situação tal como, por exemplo, quando alguém está perdido em uma cidade e pede uma informação a um estranho. Neste caso, este estranho pode ser uma pessoa natural da cidade; logo, este estranho deve possuir a informação necessária. Em situações como esta, a pessoa perdida necessita buscar uma informação que não tem. Se o estranho der uma resposta citando um local, pode fazer a pessoa perdida pensar que a informação dada pelo estranho pode ser falsa, ou tomar essa informação como verdadeira.

Para a pessoa perdida poder seguir a informação dada, ela pode recorrer a algumas questões. Respostas para essas questões podem tornar os argumentos/informações do estranho falaciosos, assim “derrubando-os”. Da mesma forma, se tais questões forem respondidas satisfatoriamente, os argumentos apresentados pelo estranho podem se tornar mais confiáveis e verdadeiros. Neste caso, tais questões são representadas em esquemas de argumento como Questões Críticas (QC). Assim como descrito em (WALTON; REED; MACAGNO, 2008), o esquema de argumento “Argumento da Posição do Saber” possui três QCs que são as seguintes:

- Questão crítica 1: A realmente tem condições de saber se A é verdadeiro (falso)?
- Questão crítica 2: A fonte α é honesta e confiável?
- Questão crítica 3: A realmente afirmou que A é verdadeiro (falso)?

Em geral, para argumentar sobre um determinado assunto, é necessário que se tenha conhecimento sobre o mesmo e justificativas plausíveis para gerar argumentos, sejam eles contra ou a favor de questões levantada em uma discussão (BEX; VERHEIJ, 2011). Um esquema de argumentação visa estruturar argumentos apresentados em uma discussão de forma a identificar como cada participante está contribuindo, encadeando as informações de forma coerente e facilitando o acesso ao conhecimento gerado, particularmente relacionando este conhecimento com processos de decisão (LINDGREN, 2009).

4. CONTEXTUALIZAÇÃO DO PROJETO DE PESQUISA

A proposta de pesquisar esquemas de argumentação para auxiliar a proposição de riscos em meio a uma discussão colaborativa de gerenciamento de riscos é parte de um projeto de pesquisa maior que está sendo desenvolvido no Programa de Pós-Graduação em Informática (PPGI) da Universidade Federal de Santa Maria (UFSM). Entre outros objetivos, este projeto visa investigar técnicas colaborativas baseadas em conhecimento para apoiar o processo de gerenciamento de riscos em projetos de desenvolvimento de software.

Neste projeto de pesquisa, a técnica de jogos de diálogo está sendo amplamente explorada visando estruturar e guiar uma discussão de riscos, de forma que sejam criados e armazenados dados e conhecimento obtidos nestes processos colaborativos de gerenciamento de projetos. Porém, a técnica de jogos de diálogo, assim como proposta em trabalhos desenvolvidos por outros pesquisadores de nosso grupo de pesquisa (SEVERO et al., 2013), não é voltada para organizar o conteúdo das mensagens trocadas durante discussões de riscos. Para expandir e aprimorar o emprego de técnicas de argumentação na área de gerenciamento de riscos, esta dissertação associa o uso de jogos de diálogo com esquemas de argumentação. Desta forma, participantes de uma discussão podem adicionar argumentos mais completos e estruturados em uma discussão.

É importante ressaltar que a utilização de atos de locução em conjunto com esquemas de argumentação pode permitir a captura de informações relevantes para a contextualização e explicação do porquê de um risco ter sido priorizado ou o porquê de um plano ter sido aceito ou rejeitado para o tratamento deste risco, por exemplo. Esta contextualização e explicação são fatores importantes em processos de tomada de decisão, de forma que essas experiências possam ser consultadas e reutilizadas em discussões de riscos em projetos futuros.

A abordagem proposta neste trabalho visa expandir a pesquisa realizada por (SEVERO; FONTOURA; SILVA, 2013). Em geral, a pesquisa no projeto visa permitir o desenvolvimento sistemático de discussões de gerenciamento de riscos. Além disso, ela permite que tais discussões possam ser registradas de modo organizado em uma memória. Atualmente, o projeto de pesquisa em que esta dissertação está inserida também aborda, por meio de trabalhos de outros participantes do grupo de pesquisa, o emprego de técnicas de raciocínio baseado em casos (MANTARAS, 2005; SILVA et al., 2010) como forma de consultar e reusar argumentos apresentados em discussões de riscos passadas (MACHADO et al., 2014).

4.1. Jogos de diálogo para gerenciamento de riscos

Em (SEVERO; FONTOURA; SILVA, 2013), uma abordagem baseada em jogos de diálogo para guiar, organizar e estruturar discussões colaborativas de riscos é proposta e implementada em um ambiente de discussão *web* – *RD System v1.0*. A principal ideia dessa abordagem é explorar o emprego de atos de locução no desenvolvimento de tarefas de gerenciamento de riscos, em particular, na identificação, análise e planejamento de riscos. Essas são as tarefas que mais exigem o uso de um protocolo de discussão (embora tarefas de monitoramento e controle de riscos ainda não sejam tratadas neste protocolo). Para um melhor entendimento, o conjunto de atos de locuções do protocolo é apresentado nas Tabela 1; Tabela 2; Tabela 3. Estas tabelas exemplificam com mais clareza quais são os atos de locuções disponíveis no nosso protocolo de comunicação, bem como seus objetivos e exemplos típicos de uso destas locuções.

Tabela 1 – Atos de locuções do protocolo de comunicação que determinam o início e término de uma discussão de riscos

Nome: Iniciar Discussão
Descrição: Iniciar a discussão com uma afirmação à respeito do assunto a ser abordado no processo de gerenciamento de riscos.
Exemplo:
<u><i>Iniciar Discussão</i></u> : Discussão do módulo de consulta.

Nome: Terminar Discussão
Descrição: Terminar a discussão, não permitindo mais a inserção de novos argumentos no processo de gerenciamento de riscos.
Exemplo:
<u><i>Terminar Discussão</i></u> : Discussão finalizada.

Tabela 2 – Atos de locuções específicos para mediar o desenvolvimento de tarefas de gerenciamento de riscos

Nome: **Propor Risco**
 Descrição: Propor um novo risco a ser analisado na discussão. (Foco principal em discussões de riscos).
 Exemplo:
Propor Risco: A equipe de desenvolvimento tem pouca experiência em ruby.

Nome: **Propor Probabilidade**
 Descrição: Propor a probabilidade de um risco acontecer no projeto em questão. Alta, média, baixa ou usando valores numéricos (por exemplo: 70%).
 Exemplo:
Propor Risco: A equipe de desenvolvimento tem pouca experiência em ruby.
Propor Probabilidade: Média.

Nome: **Propor Impacto**
 Descrição: Propor o impacto de um risco acontecer no projeto em questão. Alto, médio, baixo ou usando valores numéricos (por exemplo: 70%).
 Exemplo:
Propor Risco: A equipe de desenvolvimento tem pouca experiência em ruby.
Propor Impacto: Alto.

Nome: **Propor Plano**
 Descrição: Propor um plano (ações) para lidar com um risco levantado na discussão.
 Exemplo:
Propor Risco: A equipe de desenvolvimento tem pouca experiência em ruby.
Propor Plano: A equipe poderia receber um treinamento da linguagem ruby.

Tabela 3 – Atos de locuções de propósito geral que visam permitir uma discussão crítica de riscos e seus planos

Nome: **Propor Consequência**
 Descrição: Propor uma consequência de um risco sendo analisado no projeto.
 Exemplo:
Propor Risco: A equipe de desenvolvimento tem pouca experiência em ruby.
Propor Consequência: Este risco pode causar atraso na entrega do módulo de consulta ou até mesmo o fracasso do projeto.

Nome: **Perguntar**
 Descrição: Perguntar aos outros participantes sobre qualquer assunto de gerenciamento de riscos pertinente à discussão.
 Exemplo:
Perguntar: Qual linguagem de consulta será utilizada?

Nome: **Informar**
 Descrição: Informar uma informação de gerenciamento de riscos relevante aos outros participantes ou responder alguma pergunta realizada na discussão.

Exemplo:

Informar: O módulo de consulta será implementado em ruby.

Nome: **Argumento a favor**

Descrição: Expressar uma opinião de apoio a alguma informação de gerenciamento de riscos apresentada durante a discussão.

Exemplo:

Propor Plano: Monitorar os prazos de conclusão (marcos) propostos no projeto.

Argumento a favor: Concordo, e não podemos passar nenhum marco sem apresentar todas as tarefas concluídas para aquele determinado período.

Nome: **Argumento contra**

Descrição: Expressar uma opinião contrária a alguma informação de gerenciamento de riscos apresentada durante a discussão.

Exemplo:

Propor Risco: A utilização de esquemas de argumentação em uma discussão de gerenciamento pode não ser compreendida pelos participantes.

Argumento contra: O uso dos esquemas de argumentação pode ser algo complexo que pode ser resolvido com um treinamento neste assunto.

Nome: **Retirar**

Descrição: Expressar a anulação ou abandono de alguma informação de gerenciamento de riscos apresentada previamente na discussão.

Exemplo:

Argumento contra: Esse projeto é acadêmico e não disponibilizamos de verba para treinamentos.

Argumento contra: O treinamento poderia ser feito pelas pessoas capacitadas que possuímos no grupo.

Retirar: Ok.

Nome: **Resumir**

Descrição: Resumir um grupo de informações de gerenciamento de riscos em uma única mensagem.

Exemplo:

Propor Risco: A equipe de desenvolvimento tem pouca experiência em ruby.

Resumir: O risco tem probabilidade média e impacto alto; logo, este risco será selecionado para a etapa de planejamento.

Nome: **Pedir Posicionamento**

Descrição: Pedir que cada participante da discussão expresse sua opinião sobre um determinado assunto de gerenciamento de riscos.

Exemplo:

Pedir posicionamento: Todos concordam que o impacto do risco é alto?

Nome: **Opinar**

Descrição: Expressar a opinião sobre algum pedido de posicionamento de gerenciamento de riscos.

Exemplo:

Pedir posicionamento: Todos concordam que o impacto do risco é alto?

Opinar: Sim.

O emprego deste jogo de diálogo em uma discussão de gerenciamento de riscos pode ser visualizado na Figura 3. Neste exemplo, a discussão é organizada a partir do ato de locução *Propose Risk*, onde um debate é efetivamente iniciado. O objetivo desta locução é definir se o risco será identificado (e posteriormente priorizado) ou não no processo de gerenciamento de riscos. A etapa de análise do risco é desenvolvida por meio das locuções *Propose Probability* e *Propose Impact*. Estes atos de locução permitem que os participantes entrem em um consenso a fim de priorizar os riscos com probabilidade e impactos definidos como altos. O planejamento de riscos priorizados é capturado a partir da utilização do ato de locução *Propose Plan*. Esta locução permite que usuários descrevam diferentes tipos de planos voltados para o tratamento de riscos, como, por exemplo: “A equipe poderia receber um treinamento da linguagem ruby” (ver Tabela 2). Em conjunto com essas locuções-chaves para o desenvolvimento de tarefas colaborativas de gerenciamento de riscos, o protocolo contém locuções para argumentar contra ou a favor, questionar e informar, entre outras. Essas locuções viabilizam o desenvolvimento organizado de um debate de riscos com o objetivo de capturar a opinião de participantes, permitir uma análise crítica de riscos e planos, além de buscar consenso entre os vários integrantes do diálogo envolvidos no projeto.

É importante notar que o passo inicial de uma discussão colaborativa de gerenciamento de riscos é a proposição de um risco. Após um risco ter sido proposto de forma clara, a discussão deste risco pode ser continuada a partir de sub-argumentos. De forma similar, outros argumentos e proposições também podem ser apresentados nas demais etapas de um processo de gerenciamento de riscos. Na Figura 3, por exemplo, pode ser observado um fragmento de um debate no qual são discutidas tarefas relacionadas a etapas do processo de gerenciamento de riscos. Esse debate, normalmente, contém questionamentos feitos pelos participantes. Nestes questionamentos, os participantes do diálogo visam capturar informações adicionais para aprofundar o conhecimento sobre determinados aspectos do problema sendo discutido, assim permitindo continuar a discussão. Uma vez que tais questionamentos sejam feitos, os participantes tendem a respondê-los. Essas respostas permitem apresentar informações que supostamente estavam implícitas no processo de gerenciamento de riscos do projeto em questão, por exemplo. Em meio ao processo de apresentação de proposições, questionamentos e informações, participantes também podem argumentar contra ou a favor sobre diferentes tópicos da discussão. Tais locuções são fundamentais para viabilizar discussões de propostas, por exemplo, bem como a tomada de decisão baseada na análise de pros e cons, as quais podem ser apresentados em debates de riscos pelos participantes de um projeto.

Na discussão apresentada na Figura 3, pode ser observado que são propostos probabilidade alta e impacto médio para determinado risco, indicando que o risco deve ser priorizado no processo de gerenciamento de riscos. A partir dessas definições, outras locuções permitem que os participantes façam a proposta de planos para tratar tal risco. Apesar deste exemplo não apresentar, a discussão também acontece nas etapas de proposição de probabilidade e impacto, bem como na proposição de planos. Em particular, os papéis dos ícones “S” (argument Scheme) e “Q” (critical Question), os quais aparecem no fragmento de discussão da Figura 3, serão discutidos posteriormente nesta dissertação.

The screenshot shows a discussion tree in the RD System. The root node is a proposal of risk: "[244] Propose Risk: Módulo de consulta é muito complexo e possui problemas de usabilidade. - Ruan Pozzebon" with an 'S' icon. It branches into several nodes: a question "[249] Ask: A pessoa responsável não pode melhorar a usabilidade do módulo? - Nielsen Rechia" with a 'Q' icon; an information node "[250] Inform: Reuniões para discussão sobre usabilidade e melhoria podem resolver o problema, acredito. - Catherine Barchet"; an argument against node "[254] Argument con: Acredito, que testes e demonstração com usuários sem conhecimento do módulo seria mais apropriado - Nielsen Rechia"; another information node "[251] Inform: Acredito que o gerente deveria fazer experimentos, para ter um feedback dos usuários e assim poder melhorar a usabilidade do módulo. - Maicon Balke"; an argument pro node "[255] Argument pro: Concordo, pois feedback de usuários é sempre bom para qualquer projeto - Nielsen Rechia"; another question "[268] Ask: O módulo tem condições de ir para experimentos? - Catherine Barchet"; and an information node "[274] Inform: O módulo terá experimentos, mas somente depois que for implementado. - André Lúcio Hahn". Below these are three more nodes: "[252] Propose Probability: Alta - Ruan Pozzebon", "[253] Propose Impact: Médio - Ruan Pozzebon", and "[273] Propose Plan: Realizar testes com usuários e ir melhorando o modulo durante o tempo determinado do projeto para que quando o deadline final chegar a usabilidade esteja em condições ideais de utilização - Nielsen Rechia". At the bottom, a legend lists icons for: Iniciar Discussão, Propor Risco, Propor Impacto, Propor Probabilidade, Propor Plano, Propor Consequência, Argumentar a favor, Argumentar contra, Perguntar, Informar, Resumir, Pedir Posicionamento, Opinar, Retirar, and Terminar Discussão.

Figura 3 – Fragmento de uma discussão de riscos do RD System

A Figura 3 permite observar como uma discussão colaborativa de riscos é organizada por meio de uma estrutura de árvore. A partir desta árvore de discussão, os recursos de diálogo do *RD System* funcionam da seguinte forma: após a discussão ter sido iniciada, um participante seleciona um nodo na árvore. Em seguida, este participante propõe um argumento (descreve o texto do argumento, em uma linguagem informal) que vai ser indexado

internamente como um ramo/subárvore do nodo selecionado. Além disso, a seleção de um nodo da árvore, bem como a proposição de um argumento, pode acontecer em paralelo com a apresentação de argumentos em vários pontos da árvore (ou vários pontos de um processo de discussão no qual vários tópicos estão sendo debatidos em paralelo). Neste caso, usuários podem apresentar argumentos relacionados à identificação, análise e planejamento de riscos a qualquer momento do processo de discussão. Tais subdebates podem estar ocorrendo em diferentes fases do processo de gerenciamento de riscos, bem como essas fases podem estar ocorrendo em paralelo, nas quais vários riscos podem estar sendo discutidos. Outra funcionalidade importante existente no ambiente de discussão do *RD System* é a possibilidade que os usuários têm de expandir e filtrar a árvore (ramos ou subárvores) durante a discussão. Isso permite a estes participantes focalizar e analisar os diferentes aspectos do debate sendo realizado.

Após entender a estrutura da árvore de discussões e suas funcionalidades, este trabalho ilustra na Figura 4, as regras de combinação do protocolo de comunicação. Essas regras funcionam da seguinte forma, o processo de discussão é iniciado por meio da locução “Iniciar Discussão” (*Start Discussion*). Após iniciada a discussão, os participantes têm a possibilidade de realizar questionamentos “Perguntar” (*Ask*) referentes ao projeto em questão e/ou apresentar informações “Informar” (*Inform*) complementares sobre o projeto. Após os participantes sanarem suas dúvidas sobre o projeto, inicia-se o processo de gerenciamento de riscos no qual os usuários irão identificar os riscos do projeto usando a locução “Propor risco” (*Propose Risk*). Para cada risco proposto, consequências que os riscos podem trazer ao projeto, caso os mesmos ocorram, podem ser discutidas usando a locução “Propor consequência” (*Propose Consequence*). Em paralelo a etapa de identificação, inicia-se a etapa de análise dos riscos, realizada por meio da proposição do impacto e da probabilidade dos riscos, utilizando as locuções “Propor Impacto” (*Propose Impact*) e “Propor probabilidade” (*Propose Probability*), respectivamente. Por fim, o planejamento dos riscos priorizados no projeto é proposto por meio da proposição de planos, com a locução “Propor plano” (*Propose Plan*). Para terminar uma discussão de riscos, o protocolo de comunicação possui a locução “Terminar discussão” (*End Discussion*). Além das regras definidas pelas etapas da discussão, é possível utilizar locuções de uso geral em qualquer ponto da discussão. Tais locuções visam remover possíveis dúvidas, informar algo relevante para o desenvolvimento do debate, argumentar contra ou a favor “Argumentar a favor; Argumentar contra” (*Argument pro; Argument con*) em relação a alguma afirmação proposta por meio de um ato da discussão, entre outras ações.

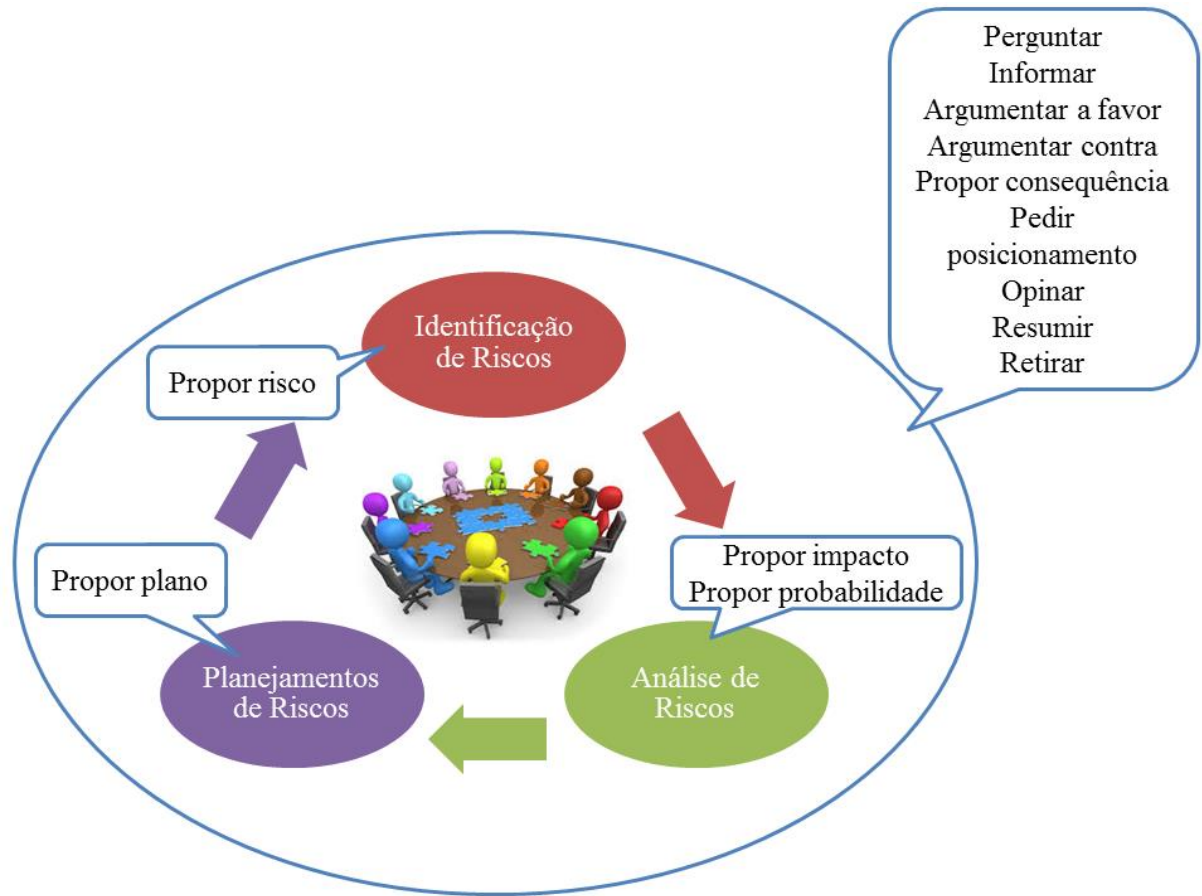


Figura 4 – Regras de combinação

5. ESQUEMAS DE ARGUMENTAÇÃO PARA GERENCIAMENTO COLABORATIVO DE RISCOS

Este capítulo trata do modelo de argumentação que norteia a pesquisa desta dissertação. Dentre outros fatores para a escolha desse modelo, tal escolha está relacionada ao fato de que há necessidade de melhorar o processo de proposição de riscos em meio a uma discussão. Neste contexto, foi observada a necessidade de disponibilizar formas para que participantes de discussões de riscos possam padronizar e formalizar seus argumentos. Uma das propriedades principais que define o risco é a incerteza, isto é, riscos são eventos incertos. Portanto, argumentos envolvendo a proposição e discussão de riscos podem ser compreendidos como falácias. O modelo de argumentação que é melhor ajustado para a captura e representação destes fatores é o modelo de esquemas de argumentação. Este modelo permite que usuários contextualizem a natureza dos riscos que estão sendo propostos, ou que tais argumentos sejam derrubados por não serem baseados em suposições ou evidências válidas.

Este capítulo apresenta e discute esquemas de argumentação para gerenciamento de riscos. O capítulo também aborda como é realizada a classificação destes esquemas de argumentação. Tal classificação é realizada pelo emprego da noção de tipos de argumento, a qual é devidamente adaptada para a área de gerenciamento de riscos. Em seguida, é descrito um processo de criação destes esquemas, além de descrever e exemplificar como instâncias de esquemas (ou *templates*) podem ser realizadas por usuários.

5.1. Tipos de argumentos

No intuito de construir uma biblioteca de esquemas de argumentos, decidiu-se explorar a noção de tipos de argumentos - TA (SILVA, 2010). Embora esta noção tenha sido proposta pelo estudo de modelos de argumentação em outros domínios de aplicação, tal noção é relevante para auxiliar na classificação/indexação e uso de esquemas voltados para o apoio a tarefas de proposição e discussão de riscos em projetos de software. Em essência, tipos de argumentos visam capturar a natureza de um argumento no domínio da aplicação (ou seja, sobre o que é o argumento). Nesta dissertação, estes tipos são utilizados para anotar o

conteúdo textual dos argumentos propostos em discussões de risco. Isso significa que os usuários podem vincular os seus argumentos com um ou vários tipos, visto que argumentos propostos podem ser construídos por usuários a partir da combinação de diferentes assuntos no domínio de aplicação.

Nesta dissertação, uma lista de tipos de argumentos para o domínio de gerenciamento de riscos foi construída. Esta lista é compatível com outras classificações de tipos (por exemplo, tipos de risco) que estão disponíveis na literatura de gestão de projetos de software (CARR et al., 1993; PRESSMAN, 2007; SOMMERVILLE, 2011), por exemplo. Baseado em diferentes listas de riscos, foram definidos os seguintes Tipos de Argumentos:

- TA01 – Cliente;
- TA02 – Requisitos;
- TA03 – Planejamento;
- TA04 – Execução;
- TA05 – Manutenção;
- TA06 – Negócio;
- TA07 – Tecnologia;
- TA08 – Organizacional;
- TA09 – Ferramentas.

Esta lista de tipos de argumentos pode ser expandida e subdivisões podem ser criadas, assim permitindo um maior detalhamento destes tipos e uma indexação de argumentos possivelmente mais detalhada. Além disso, usuários podem requerer o ajuste desta lista de acordo com a necessidade de indexação dos esquemas de argumentação desenvolvidos para análises de riscos em aplicações ou empresas particulares.

5.2. Processo de desenvolvimento de esquemas de argumentos para gerenciamento de riscos

Um processo de desenvolvimento de esquemas foi elaborado com o intuito de avançar as pesquisas sobre esquemas de argumentação, pois os autores da área (MCLEOD; FERGUSON; BURGER, 2009; PARSONS et al., 2012; TOLCHINSKY; MODGIL; CORTÉS, 2006; WALTON; REED; MACAGNO, 2008; YUAN; KELLY, 2011), não descrevem nenhum processo de como desenvolver esquemas de argumentação. Desta forma,

o desenvolvimento de esquemas de argumentação para gerenciamento de riscos pode ser descrito por meio de um conjunto de atividades sequenciadas em um fluxo de execução. Para ilustrar o processo de criação dos esquemas de argumentação, um diagrama de processos de negócios em *Business Process Management Notation* (BPMN) foi elaborado e pode ser visualizado na Figura 5 e cada uma dessas tarefas é detalhada. No geral, todas as tarefas são executadas por um especialista em Gerência de Riscos, ele que tem a responsabilidade de criar os esquemas de argumentação. O processo proposto visa a criação de uma biblioteca de esquemas.

5.2.1. Selecionar o risco

O processo de desenvolvimento de esquemas de argumento para proposição de riscos é iniciado pela escolha de riscos relevantes, os quais podem ser propostos, priorizados e tratados em discussões colaborativas de riscos em projetos de software. Na área de gerenciamento de riscos, estes “fatores de riscos” são normalmente descritos em catálogos, tais como: (EMAM; KORU, 2008; PERSSON; MATHIASSEN, 2010). Estes catálogos são amplamente consultados na construção de listas de riscos iniciais a serem avaliadas em processos de gerenciamento de riscos. Assim como tais catálogos são explorados em atividades tradicionais de identificação de riscos, o processo de desenvolvimento de esquemas para este domínio de aplicação também deve explorar tais catálogos no auxílio à identificação de esquemas candidatos a serem definidos. Além disso, o processo de construção de esquemas aqui proposto também visa explorar os tipos de riscos descritos nesses catálogos na indexação de esquemas de argumentos desenvolvidos para esta área de aplicação. Tais índices são relevantes na busca e utilização de esquemas na proposição de riscos em discussões colaborativas de riscos, assim como implementado no *RD System v2.0* (ver capítulo 6).

Tabela 4 – Lista de verificação (checklist) de riscos adaptados de (EMAM; KORU, 2008)

Gerente do projeto não está suficientemente envolvido;
Muitas mudanças de requisitos;
Muitas mudanças de escopo;
Falta de experiência necessária;
Requisitos incertos.

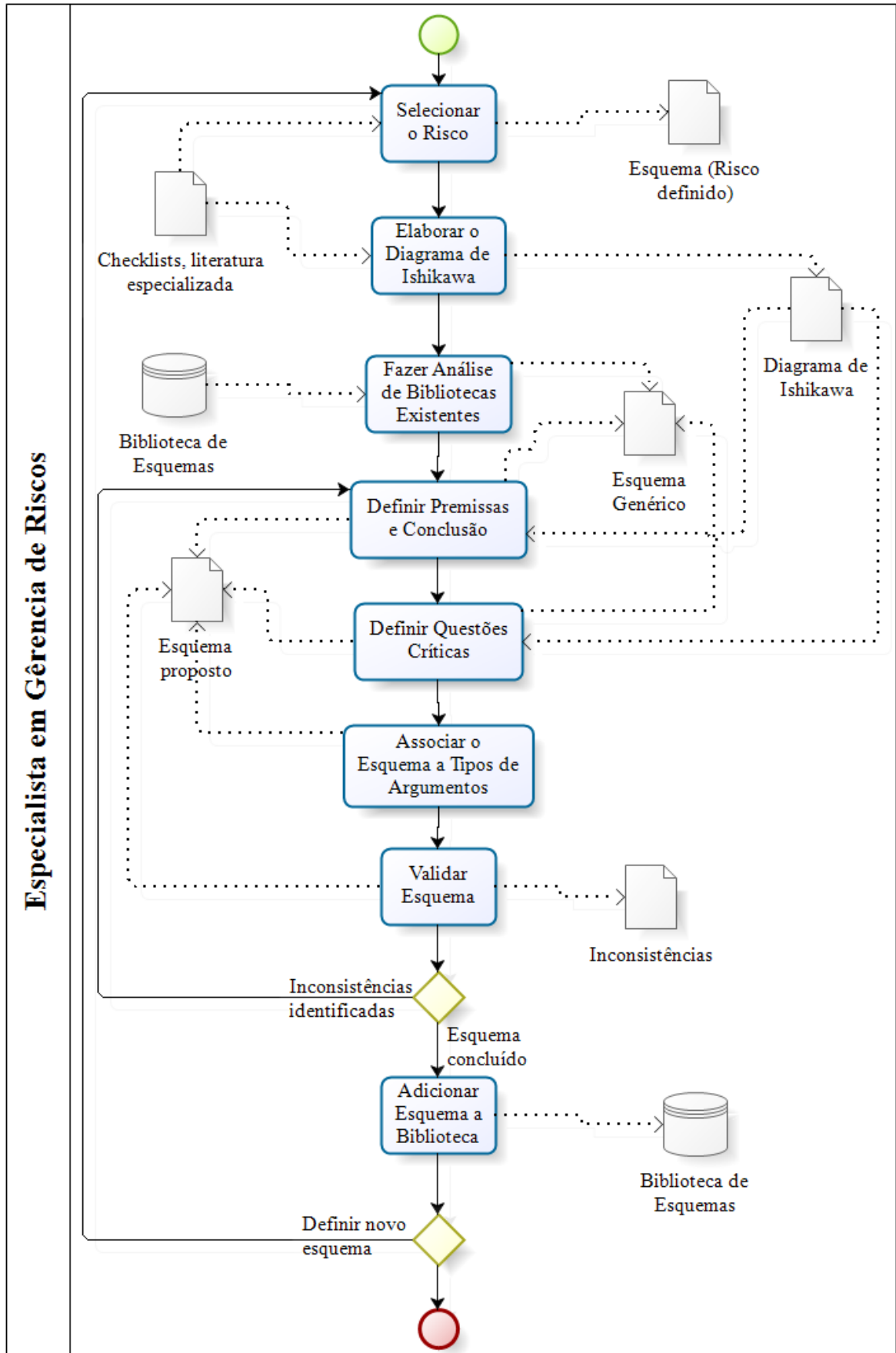


Figura 5 – Processo de desenvolvimento de esquemas de argumentação.

Tendo um checklist dos principais riscos tratados em projetos de software, tal como exemplificado na Tabela 4, o próximo passo no processo de construção de esquemas é utilizar riscos selecionados na definição de tipos de argumentos (2).

Portanto, a finalidade da atividade “Selecionar Risco” é definir para qual risco o novo esquema será criado. O responsável pela execução da atividade é um especialista da área de Gerência de Riscos. Para executar essa atividade é necessário ter listas de verificação (*checklists*) e literatura especializada como artefatos de entrada e o resultado da execução desta atividade (artefato de saída) é a definição do risco para qual o esquema será criado.

5.2.2. Elaborar o diagrama de Ishikawa

A segunda atividade propõe a adição do risco ao diagrama de Ishikawa, também conhecido como diagrama espinha de peixe (ROCHA; MALDONADO; WEBER, 2001).

Esquemas de argumentação podem focalizar a representação de causas e efeito de riscos os quais são típicos de projetos de software, assim como proposto em um esquema de argumento genérico de causa e efeito descrito em (WALTON; REED; MACAGNO, 2008). Em geral, a principal ideia deste diagrama é auxiliar na identificação e listagem de causas que tem um risco como efeito. As causas são os fatores de risco, isto é, fatores que levam a ocorrência de determinado risco. A partir da utilização deste diagrama, o processo de construção de um esquema é fundamentado nesta análise de causas de riscos. Em consequência, tais causas e efeitos identificados são usadas para motivar a representação de premissas, conclusão e questões críticas de esquemas. Uma vez que tais causas tenham sido empregadas na descrição de um esquema de argumento, elas podem também motivar a proposição de planos de tratamento de riscos em uma discussão. Em particular, estes planos podem descrever formas de tratar estes fatores de riscos (isto é, suas causas), visto que causas típicas de riscos são identificadas explicitamente no corpo de um esquema. Em resumo, a adição de uma proposição de risco no diagrama de Ishikawa torna possível uma melhor visualização e compreensão de causas e efeito de riscos, tal como ilustrado na Figura 6. Tal compreensão é relevante na identificação e formalização de esquemas de argumentos nesta área de aplicação.

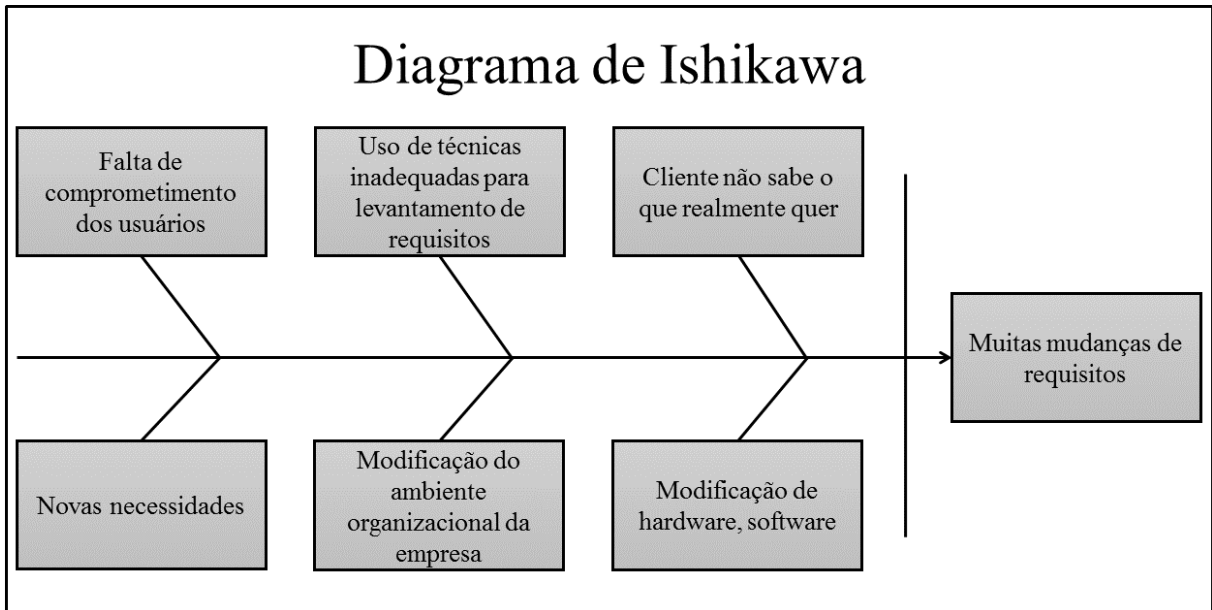


Figura 6 – Aplicação de um risco ao diagrama de espinha de peixe

Portanto, a finalidade da tarefa “Elaborar o diagrama de Ishikawa” é criar um diagrama de Ishikawa visando compreender os fatores de riscos (causas) que podem ocasionar o risco. Essas causas são usadas para a elaboração das premissas, conclusões e questões críticas do esquema. Para realização desta atividade é necessário consultar a literatura especializada e o resultado da execução da atividade é o diagrama de Ishikawa.

5.2.3. Fazer uma análise em biblioteca de esquemas de argumentação existentes

Esta atividade tem o objetivo de identificar padrões/*templates* genéricos de esquemas de argumentos que possam nortear a criação de bibliotecas de esquemas em um domínio de aplicação particular, tal como o domínio de gerenciamento de riscos em projetos de software. Além disso, esta análise também visa identificar a natureza e contexto de esquemas existentes, os quais podem também sustentar a criação de biblioteca de esquemas. Por exemplo, o “Argumento de causa e efeito” proposto por Walton (Tabela 5) foi utilizado como base para a criação de esquemas de argumentação para proposição de riscos de software, assim como descrito posteriormente nesta dissertação. Assim como este esquema genérico de argumentação, o qual é voltado para a captura de argumentos de causa e efeito, motivou a

construção de esquemas para a proposição de riscos nesta dissertação, outros esquemas genéricos descritos em catálogos na área de argumentação (por exemplo: (WALTON; REED; MACAGNO, 2008)) poderiam ser usados para motivar a construção de esquemas de argumentação particularmente ajustados para domínios de aplicação particulares. Em geral, a ideia é explorar bibliotecas de esquemas disponíveis na literatura e possivelmente adaptar esquemas disponíveis aos requisitos e suposições de propostas e discussões colaborativas de riscos em projetos de software, assim permitindo construir esquemas que sejam compreensíveis e simples de utilizar por usuários típicos destes de domínios de aplicação particulares.

Tabela 5 – Argumento de causa e efeito. Adaptado de (WALTON; REED; MACAGNO, 2008)

Premissa maior: Geralmente, se A ocorre, então B vai (pode) ocorrer.
Premissa menor: Neste caso, A ocorre (pode ocorrer).
Conclusão: Portanto, neste caso, B vai (pode) ocorrer.
Questões críticas:
QC1: Quão forte é a generalização causal?
QC2: É forte a evidência citada (se houver) o suficiente para justificar a generalização casual?
QC3: Existem outros fatores causais que possam interferir com a produção do efeito, no caso dado?

5.2.4. Definir premissas e conclusão do esquema

Essa atividade tem como objetivo a criação de premissas e conclusão do esquema, utilizando causas e efeito descritos no diagrama de Ishikawa.

Em uma situação típica de desenvolvimento de esquemas de argumento, a criação das premissas e conclusão de um esquema é iniciada na tarefa (4). Neste passo, a conclusão do esquema é representada por meio da definição dos riscos em si. As premissas do esquema, por sua vez, partem das causas dos riscos descritas em um diagrama de Ishikawa. Neste caso, as possíveis premissas de um esquema podem ser claramente observadas no diagrama, as quais

podem capturar causas para um risco, o qual seria descrito na conclusão do esquema. Em resumo, devido a importância da definição de causas e efeito de riscos na área de gerenciamento de riscos, acreditamos que este passo de desenvolvimento de esquemas seja relevante para a construção de esquemas de causa e efeito voltados para a proposição e discussão de riscos em projetos de software, assim como exemplificado por esquemas descritos posteriormente nesta dissertação.

O resultado desta atividade é a definição das premissas e conclusão do esquema a partir do esquema genérico selecionado na tarefa anterior e das causas e efeito descritos no diagrama Ishikawa.

5.2.5. Definir questões críticas para o esquema

A finalidade desta atividade é a criação do conjunto de questões críticas, tendo em vista a natureza, exceções, requisitos, suposições e relações entre premissas e conclusão, bem como contra argumentos típicos da área.

Assim como são criadas as premissas de um esquema de argumento, é também criado o conjunto de questões críticas do esquema. Neste caso, a partir da identificação das causas e efeito de um risco, questões críticas podem ser também identificadas. Em particular, a construção do diagrama de Ishikawa pode favorecer a obtenção de uma melhor compreensão de elementos de riscos, os quais podem ser explorados na elaboração de questões críticas para um esquema. Na definição de questões críticas, por exemplo, causas (premissas) e efeito (conclusão) de riscos podem ser usados para testar (questionar) se um risco está realmente ocorrendo (ou tem grande chance de ocorrer) em um projeto. Desta forma, a natureza das causas e efeito de argumentos estaria sendo explorada na identificação de possíveis questões críticas a serem adicionadas na definição de um esquema em desenvolvimento. Da mesma forma, a análise da natureza das informações capturada por argumentos descritos como esquemas pode também favorecer a identificação de contra-argumentos típicos usados para derrubar argumentos originalmente propostos. Na mesma linha de raciocínio, a definição de questões críticas para esquema de argumentos pode ser fundamentada em exceções conhecidas para as regras que sustentam estes argumentos. Em geral, argumentos tendem a ser baseados em regras que podem ser válidas (ou não) em determinados contextos, visto que tais regras estão sujeitas a exceções.

Questões críticas de esquemas também podem ser motivadas na análise dos principais requisitos identificados na proposição de argumentos. Tais requisitos capturam informação que é tomada como verdadeira no contexto onde o argumento é proposto. Logo, o ataque a tais requisitos é normalmente usado como uma forma de derrubar argumentos originalmente fundamentados nestes requisitos. Da mesma forma, argumentos capturados como esquemas podem ser baseados em suposições, as quais descrevem informação que é compreendida/tomada como verdadeira em determinadas situações. Neste caso, o simples ataque por meio de questões críticas a tais suposições usadas na proposição de argumentos é uma forma de questionar a veracidade de argumentos, os quais são generalizados na estrutura de esquemas. Em geral, não existe um número ideal de questões críticas a ser definido em um esquema de argumento. A ideia principal, portanto, é explorar a natureza de argumentos, exceções conhecidas, requisitos, suposições, entre outros, na definição de questões que sejam relevantes para o desenvolvimento de análises críticas de riscos em projetos de software.

Portanto, questões críticas podem ser definidas a partir de uma análise de fatores de risco (causas) descritas no diagrama de Ishikawa e do *template* selecionado na segunda tarefa, que propõe algumas formas para a criação de questões críticas. O resultado é a inclusão das premissas no esquema que está sendo elaborado.

5.2.6. Associar o esquema de argumentos a tipos de argumentos

Riscos em projetos de software são tradicionalmente agrupados em categorias ou tipos, assim como demonstrado em (EMAM; KORU, 2008; PERSSON; MATHIASSEN, 2010). No contexto de desenvolvimento de esquemas para a proposição e discussão de riscos, também existe a necessidade de explorar essas classificações tradicionais de riscos por tipos (ex.: tecnologia, requisitos). Assim como descrito anteriormente, essa classificação é relevante para auxiliar na indexação de esquemas de argumento. Contudo, estes esquemas devem ser indexados por meio de “tipos de argumentos” (SILVA, 2010). Similar a tipos de riscos, tipos de argumentos visam capturar a natureza de argumentos capturados/descritos no formato de esquemas. Em particular, um tipo de argumento busca identificar o conteúdo de informação de um argumento na área de aplicação onde esse argumento está inserido. Por exemplo, se um esquema de argumento fosse voltado para a proposta de riscos de requisitos em um projeto, o esquema por si só seria identificado como um argumento do tipo “riscos de requisitos”. Um

exemplo deste processo de indexação pode ser descrito para o esquema relacionado a “Muitas mudanças de requisitos” (descrito, posteriormente, neste capítulo), onde sua classificação/indexação por tipo é a seguinte: Requisitos. Como um argumento pode ser construído a partir de diferentes conteúdos de informação, uma combinação de tipos de argumentos definidos na área de aplicação pode ser usada para indexar esquemas desenvolvidos.

Portanto, o objetivo desta atividade é classificar (indexar) o esquema que está sendo elaborado de acordo com alguns tipos de argumentos pré-definidos.

5.2.7. Validar esquema

A finalidade desta tarefa é a revisão e validação do esquema de argumento, com especialistas e outros usuários da área de aplicação.

Por fim, uma validação geral do esquema resultante deve ser realizada, onde as ideias e a linguagem dos esquemas construídos são revisados e ajustados segundo diferentes necessidades. Em geral, esta validação pode ser realizada a partir da análise de especialistas em gerenciamento de riscos, ou mesmo por usuários com experiência no desenvolvimento destas tarefas, os quais são convidados a apresentar opiniões a respeito da coerência e natureza do esquema. Mesmo diante desta análise, é importante salientar que esquemas de argumento para proposição e discussão de riscos devem ser descritos em uma forma que usuários sem experiência em gerenciamento de riscos sejam capazes de compreender. Na prática, a descrição dos esquemas deve ser apresentada em uma linguagem típica do domínio de aplicação, visto que esquemas descritos de forma muito genérica podem ser de difícil compreensão e conseqüente uso por participantes de projetos de software (clientes, desenvolvedores, por exemplo).

Após a validação do esquema duas situações podem ocorrer: o esquema pode ser considerado pronto e, portanto, pode ser adicionado a biblioteca de esquemas ou os especialistas podem considerar que o esquema necessita de ajustes e, neste caso, as atividades “Definir Premissas e Conclusão” e “Definir Questões Críticas”, bem como a “Validar Esquema” precisam ser novamente executadas.

5.2.8. Adicionar esquema de argumentos a biblioteca

A atividade adicionar esquema a biblioteca visa a inserção do esquema, indexado usando tipos de argumentos pré-definidos, a biblioteca de esquemas da organização. A partir dessa inclusão, os projetos da organização podem utilizar o esquema definido.

Pode ser que a partir do uso do esquema melhorias sejam identificadas e, para isso, o esquema precisa ser alterado.

5.3. Esquemas de argumentação para proposição e discussão de riscos

Equipes gerenciam riscos em seus projetos de diferentes maneiras. Para usar argumentação para raciocinar sobre gerenciamento de riscos, é preciso de ser capazes de capturar estas diferentes maneiras dentro de um argumento, assim permitindo identificar premissas que levam a uma conclusão ou risco.

Nesta dissertação, vamos focalizar a apresentação e discussão de cinco esquemas de argumentação, os quais fazem parte da biblioteca de esquemas de riscos de software construída. A escolha desses esquemas deve-se ao fato de que eles foram os mais usados e testados por usuários em discussões de riscos realizadas no nosso projeto e experimentos realizados, bem como validados por um especialista na área de gerenciamento de riscos, durante o desenvolvimento desta dissertação. Em particular, escolhemos apresentar dois esquemas relacionados a riscos de requisitos, visto que a análise de requisitos é uma atividade fundamental em diferentes tipos de projetos de software. Também escolhemos apresentar um esquema sobre riscos de ferramentas, um sobre riscos de treinamento e um esquema sobre risco de tempo devido ao fato que estes tipos de riscos são parte dos 10 riscos mais comuns (*top10*) em projetos de software (EMAM; KORU, 2008). Contudo, na versão corrente, a biblioteca construída contém 20 esquemas de argumentação, os quais estão devidamente catalogados e disponíveis para uso no *RD System v2.0*.

Os cinco esquemas que vamos descrever e exemplificar são: 1. Esquema de argumento sobre mudanças de requisitos do projeto; 2. Esquema de argumento sobre requisitos incertos; 3. Esquema de argumento sobre risco de software utilizado; 4. Esquema de argumento sobre risco de usabilidade; 5. Esquema de argumento sobre risco de tempo para capacitação.

5.3.1. Esquema de argumento sobre mudanças de requisitos do projeto

O risco de “muitas mudanças de requisitos” é um risco bastante comum no ambiente de desenvolvimento de software. Este risco afeta diretamente o desenvolvimento de um projeto, de maneira que se ele não for descrito de forma correta e completa por meio da utilização de um esquema de argumento, esta proposição pode levar a discussão de planos pouco efetivos para tratar tal ameaça. Considerando sua relevância, um esquema de argumento para este tipo de risco foi criado, visando formalizá-lo para que sua importância possa ser observada logo no início de uma discussão colaborativa de riscos.

A classificação por tipos de argumentos deste esquema é a seguinte: TA02 – Requisitos. A Tabela 6 apresenta a estrutura do esquema de argumento sobre mudanças de requisitos do projeto:

Tabela 6 – Esquema de argumento sobre mudanças de requisitos do projeto.

1. Esquema de argumento sobre mudanças de requisitos do projeto.
<ul style="list-style-type: none"> • Risco: Muitas mudanças de requisitos
Premissa 1: Geralmente, se os requisitos do projeto X são alterados com frequência, então os objetivos Y do projeto X (pode) não ser atingido.
Premissa 2: Neste caso, os requisitos do projeto X estão mudando com frequência.
Conclusão: Portanto, devemos gerenciar as mudanças nos requisitos do projeto X para que os objetivos Y do projeto X sejam alcançados.
Questões Críticas:
QC1: As mudanças nos requisitos do projeto X e os problemas no alcance dos objetivos Y do projeto têm uma relação causal significativa?
QC2: As informações de que mudanças constantes de requisitos do projeto X vai (pode) causar problemas no alcance dos objetivos Y do projeto são fundamentadas em evidências observáveis e mensuráveis?
QC3: Em vez de mudanças nos requisitos do projeto X, existem outras características do projeto que estão interferindo no alcance dos objetivos Y do projeto?
QC4: Quanto trabalho adicional as mudanças nos requisitos de projeto X geram de modo que a realização dos objetivos Y do projeto sejam comprometidas?

A Tabela 6 mostra que o esquema de argumento possui variáveis como X e Y. Tais variáveis podem ser instanciadas por termos diferentes em uma discussão de gerenciamento de risco. Por exemplo, assumindo X um projeto que está sendo desenvolvido e Y seus objetivos que devem ser alcançados para ser obtido o sucesso do projeto em questão. Neste caso, as mudanças frequentes ao decorrer do projeto X devem acarretar num fracasso do mesmo, tendo em vista que os objetivos Y não serão alcançados devido a essa frequente mudança de requisitos. A Tabela 7 apresenta uma possível instancição deste esquema de argumento, assim como realizada por um participante de uma discussão colaborativa de riscos.

Tabela 7 – Instancição do esquema de argumento sobre muitas mudanças de requisitos.

Premissa 1: Geralmente, se os requisitos do módulo de consulta do RD System são alterados com frequência, então os resultados das consultas poderão ficar inconsistentes, assim como também, a lista de projetos passados recuperados pela consulta e a possibilidades de reusa-los em uma discussão atual são objetivos que poderão não ser atingidos.

Premissa 2: Neste caso, os requisitos do módulo de consulta do RD System estão mudando com frequência.

Conclusão: Portanto, devemos gerenciar as mudanças nos requisitos do módulo de consulta do RD System para que as metas do módulo de consulta do RD System sejam alcançadas.

Em geral, as questões críticas do esquema expõem formas de validar os argumentos construídos a partir de um esquema. Questões críticas também apresentam formas de dar seguimento à discussão de riscos, assim contextualizando essa discussão. A partir do emprego destes recursos de argumentação, participantes de discussões podem ter meios de selecionar riscos que sejam realmente relevantes para um projeto. Um exemplo relacionado à utilização de uma questão crítica para dar continuidade a uma discussão de risco é o seguinte. Primeiro, um participante poderia utilizar a QC1 do esquema: as mudanças nos requisitos do módulo de consultas do *RD System* e os problemas no alcance dos objetivos do projeto têm uma relação causal significativa? O participante da discussão, sabendo que os objetivos do módulo de consultas do *RD System* dependem totalmente dos seus requisitos, pois se as mudanças nos

requisitos podem comprometer funcionalidade e ações fundamentais do módulo de consultas como: os resultados das consultas poderão ficar inconsistentes, assim como também, a lista de projetos passados recuperados pela consulta e a possibilidades de reusa-los em uma discussão atual. Neste contexto, usuários podem argumentar a favor do risco com base na QC1, afirmando que as mudanças frequentes de requisitos devem ser gerenciadas a fim de que não comprometa os objetivos do projeto, de modo que o mesmo seja concluído com sucesso. Dessa forma, este risco deve ser priorizado no projeto, assim indicando que um plano de contingência seja construído em etapas subsequentes de um processo de discussão de riscos.

5.3.2. Esquema de argumento sobre requisitos incertos

A identificação da importância do risco “cliente incerto sobre os requisitos” foi o ponto de partida para a criação deste esquema de argumento. Neste caso, este risco é bastante comum no ambiente de desenvolvimento de software, afetando um projeto em vários aspectos. Caso este risco ocorra, pode haver retrabalho nas funcionalidades do sistema a ser desenvolvido, pode haver atraso significativo na conclusão do projeto, entre outras consequências. As ocorrências citadas, as quais este risco pode trazer a um projeto, incentivou então a sua representação usando um esquema de argumentação, de forma que o risco seja proposto e compreendido pelos participantes de uma discussão.

A classificação por tipos de argumentos para este esquema é a seguinte: TA01 – Cliente; TA02 – Requisitos; TA09 – Ferramentas. A escolha desses tipos de argumentos foi feita visando todas as áreas/natureza do esquema, que são: Cliente (pessoa que define os requisitos), Requisitos que é o foco principal, já é o nome por si só já trata de requisitos e por último Ferramenta (resultado final de um projeto e os requisitos levantados são para desenvolver uma ferramenta/sistema). Pode-se visualizar na Tabela 8 a estrutura do esquema de argumento sobre requisitos incertos:

Tabela 8 – Esquema de argumento sobre requisitos incertos.

2. Esquema de argumento sobre requisitos incertos.
<ul style="list-style-type: none"> • Risco: Cliente incerto sobre os requisitos
Premissa 1: Geralmente, o cliente X define os requisitos necessários para o desenvolvimento do sistema Y.
Premissa 2: Neste caso, não está claro para o cliente X quais funcionalidades devem ser implementadas no sistema Y.
Premissa 3: Se o cliente X define os requisitos necessários ao sistema e não está claro para o cliente X quais funcionalidades devem ser implementadas no sistema Y, logo, o cliente X está incerto sobre os requisitos.
Conclusão: Portanto, os requisitos definidos pelo cliente X são incertos.
Questões Críticas:
QC1: O cliente X realmente está comprometido com o desenvolvimento do sistema Y?
QC2: O quanto o cliente X não ter certeza dos requisitos necessários para o sistema Y pode prejudicar o desenvolvimento deste sistema Y?
QC3: O cliente X não ter certeza dos requisitos necessários para o desenvolvimento do sistema Y implica em os requisitos estão realmente incertos?

Um exemplo de instanciação das variáveis do esquema de argumento sobre requisitos incertos pode ser visto na Tabela 9.

Tabela 9 – Instanciação do esquema de argumento sobre requisitos incertos.

Premissa 1: Geralmente, o gerente bancário define os requisitos necessários para o desenvolvimento do sistema para caixas eletrônicos.
Premissa 2: Neste caso, o gerente bancário não tem certeza das funcionalidades que o sistema para caixas eletrônicos necessita.
Premissa 3: Se o gerente bancário define os requisitos e não tem certeza das funcionalidades que o sistema para caixas eletrônicos necessita, logo, o gerente bancário está incerto sobre os requisitos.
Conclusão: Portanto, os requisitos definidos pelo gerente bancário são incertos.

A questão crítica QC3 deste esquema: “O cliente X não ter certeza dos requisitos necessários para o desenvolvimento do sistema Y implica em os requisitos serem incertos?”, por exemplo, pode ser usada por um participante de um projeto para validar ou derrubar um argumento construído a partir do esquema. Desse modo, os envolvidos no processo colaborativo de discussão de riscos podem chegar a um consenso se este risco deve ser priorizado ou não.

Em projetos de desenvolvimento de software, é comum haver muitas mudanças de requisitos, assim como tratado no esquema 1. Neste caso, não ter certeza sobre os requisitos necessários para o desenvolvimento do projeto é um dos principais fatores que levam a ocorrência de frequentes mudanças de requisitos. Entre outras coisas, a QC3 pode ser explorada por participantes de uma discussão para tornar a proposta do risco clara e completa (onde informação relevante não esteja implícita), assim indicando que o risco proposto deve ser priorizado em um projeto particular. Uma vez priorizado, o risco pode ser tratado por meio de um plano, tal como, “um gerente bancário deve visitar a outras agências bancárias que já possuem sistemas para seus caixas eletrônicos a fim de observar o funcionamento dos sistemas para caixas eletrônicos e adquirir conhecimento para poder definir os requisitos do sistema”.

5.3.3. Esquema de argumento sobre risco de software utilizado

O risco “problemas com softwares usados durante o desenvolvimento do projeto” é um risco frequente em empresas que mudam as tecnologias que apoiam seus serviços, mesmo considerando que tais mudanças são motivadas pela constante necessidade de atualização que estas empresas estão sujeitas. Contudo, a questão de manter os sistemas de software sempre atualizados muitas vezes não é favorável para um projeto, pois podem acontecer conflitos entre as versões dos sistemas utilizados no desenvolvimento de um produto, como também outros problemas decorrentes da utilização de sistemas em versões ainda em fases de teste (versões Beta, por exemplo). Com este cenário, decidiu-se expor este risco por meio de um esquema de argumentação de forma que fique claro a proposição do mesmo em meio a uma discussão. Neste caso, participantes de uma discussão de riscos podem usar este esquema para realizar uma análise mais profunda da questão de mudança de software a ser utilizado em um projeto.

Este esquema pode ser classificado pelos seguintes tipos: AT02 – Requisitos; AT04 – Execução; AT07 – Tecnologia; e AT09 – Ferramentas. A estrutura do “esquema de argumento sobre risco de software utilizado” é segue na Tabela 10:

Tabela 10 – Esquema de argumento sobre risco de software utilizado.

3. Esquema de argumento sobre risco de software utilizado
<ul style="list-style-type: none"> • Risco: Problemas com softwares usados durante o desenvolvimento do projeto.
Premissa 1: Geralmente, se a equipe de desenvolvimento usa o software X no desenvolvimento do projeto Y, então pode ter problemas com o software Y usado no desenvolvimento.
Premissa 2: Neste caso, a equipe de desenvolvimento está usando o software X.
Conclusão: Portanto, a equipe de desenvolvimento teve problemas com o software Y durante o desenvolvimento do projeto Y.
Questões Críticas:
QC1: É realmente necessário usar o software X para auxiliar no desenvolvimento do projeto Y?
QC2: O uso do software X e os problemas no desenvolvimento do projeto Y tem uma relação causal?
QC3: O desenvolvimento do projeto Y realmente pode ser comprometido com o uso do software X?

Um exemplo de instanciação das variáveis do “esquema de argumento sobre risco de software utilizado” pode ser observado na Tabela 11.

Tabela 11 – Instanciação do Esquema de argumento sobre risco de software utilizado.

Premissa 1: Geralmente, se a equipe de desenvolvimento usa o software Tortoise, para controle de versão, no desenvolvimento do projeto RD System, então pode ter problemas com o software NetBeans IDE usada no desenvolvimento.
Premissa 2: Neste caso, a equipe de desenvolvimento está usando o software Tortoise.
Conclusão: Portanto, a equipe de desenvolvimento teve problemas com o software NetBeans durante o desenvolvimento do projeto RD System.

Uma questão crítica que pode auxiliar os participantes de uma discussão a repensar a proposição do risco “Problemas com softwares usados durante o desenvolvimento do projeto” é a QC1: “É realmente necessário usar o software X para auxiliar no desenvolvimento do projeto Y?”. Em geral, esta QC pode induzir os participantes de uma discussão a analisarem se realmente é necessário usar um determinado recurso no desenvolvimento de um sistema. Neste caso, muitas vezes estes participantes podem escolher quais recursos eles querem utilizar uma vez que podem existir diversos recursos com uma mesma funcionalidade almejada em um projeto. Assim como descrito no exemplo apresentado na Tabela 11, tal questão crítica pode ser respondida de forma negativa, indicando que não é necessário usar o software Tortoise para controle de versões. Em particular, existem outros softwares para controle de versões que não causam conflitos ou problemas com a IDE de desenvolvimento NetBeans, por exemplo. Considerando este fato, os participantes da discussão estariam derrubando o argumento construído a partir do uso deste esquema de argumentação, assim refutando o risco proposto através do esquema.

5.3.4. Esquema de argumento sobre risco de usabilidade

O risco “problemas de usabilidades” é um risco que possui relevância em diferentes cenários de gerenciamento de riscos. Por exemplo, se um sistema for desenvolvido sem pensar na sua usabilidade, os usuários finais terão grandes dificuldades ao utilizar o sistema em questão. Pensando neste cenário, os usuários finais solicitariam uma atualização, melhoria, enfim a dificuldade de uso do sistema irá gerar um retrabalho. Este retrabalho fará com que a equipe de desenvolvimento tenha que reimplementar alguns módulos ou funcionalidades do sistema de modo que os usuários finais consigam utilizar e realizar suas atividades. Atualmente ocorre uma preocupação maior por parte dos desenvolvedores em relação aos usuários finais, diferente do que ocorria a alguns anos atrás, pois atualmente com o desenvolvimento de software em alta, ou seja, a alta concorrência do mercado, o sucesso do software depende da aceitação dos usuários finais, dentre outros fatores. Devido a sua importância perante o processo de desenvolvimento, foi percebida a necessidade de propor este risco de modo que o mesmo seja melhor compreendido por meio da formalidade e padronização dos esquemas de argumentação.

Utilizando tipos de argumentos, podemos classificar este esquema da seguinte forma: AT01 – Cliente; AT03 – Planejamento; AT05 – Manutenção; AT06 – Negócio; AT09 – Ferramentas. A estrutura do esquema de argumento sobre risco de usabilidade é descrita na Tabela 12:

Tabela 12 – Esquema de argumento sobre risco de usabilidade.

4. Esquema de argumento sobre risco de usabilidade
<ul style="list-style-type: none"> • Risco: Treinamento inadequado dos usuários.
<p>Premissa 1: Geralmente, se um sistema X não é intuitivo para os usuários, então, o sistema tem problemas de usabilidade Y que poderão (talvez) pôr em perigo a realização dos objetivos do projeto Z.</p>
<p>Premissa 2: Neste caso, os usuários estão reclamando sobre a falta de clareza nas propriedades e funcionalidades do sistema X.</p>
<p>Conclusão: Portanto, devemos gerenciar os problemas de usabilidade Y no sistema para que sejam alcançados os objetivos do projeto Z.</p>
<p>Questões Críticas:</p>
<p>QC1: Os problemas de usabilidade do projeto X e os problemas no alcance dos objetivos do projeto Z têm uma relação causal significativa?</p>
<p>QC2: As informações de que os problemas de usabilidade do projeto X vai (pode) causar problemas no alcance dos objetivos do projeto Z são fundamentadas em evidências observáveis e mensuráveis ?</p>
<p>QC3: Além de problemas de usabilidade do projeto X, existem outras características do projeto Z que estão interferindo no alcance dos objetivos do projeto Y?</p>
<p>QC1: As reclamações dos usuários são significativas?</p>
<p>QC2: Apenas um treinamento expositivo não proporciona aos usuários finais um conhecimento prático sobre o uso do sistema X?</p>
<p>QC3: O tipo de treinamento influencia na aprendizagem dos usuários finais?</p>

Uma possível instanciação das variáveis do Esquema de argumento sobre risco de usabilidade é descrita na Tabela 12.

Tabela 13 – Instanciação do esquema de argumento sobre risco de usabilidade.

Premissa 1: Geralmente, se o sistema RD System não é intuitivo para os usuários, então, o sistema tem problemas de usabilidade que poderão (talvez) pôr em perigo a realização dos objetivos do projeto de pesquisa sobre gerenciamento de riscos.

Premissa 2: Neste caso, os usuários estão reclamando sobre a falta de clareza nas propriedades e funcionalidades do sistema RD System.

Conclusão: Portanto, devemos gerenciar os problemas de usabilidade no sistema para que sejam alcançados os objetivos do projeto de pesquisa sobre gerenciamento de riscos.

Para refutar o risco proposto a partir da utilização deste esquema, um participante poderia utilizar a QC3: “O tipo de treinamento influencia na aprendizagem dos usuários finais? ”. Neste caso, este participante utilizaria uma locução *Argument con* (esta forma de utilização de esquemas é tratada na seção 6.1). Na terceira forma de uso dos esquemas podemos deixar as QCs explícitas, usando a locução *Ask*, ou implícitas, utilizando as demais locuções, na discussão de riscos. Neste caso em específico, a QC3 ficará implícita na discussão, pois estará sendo usada em conjunto com a locução *Argument con*. A utilização da QC3 neste contexto seria da seguinte forma, sabendo que o tipo de treinamento influencia na aprendizagem dos usuários finais, pois cada pessoa tem facilidade de gravar ou aprender por meio de uma determinada metodologia. Desta forma, se o treinamento for apenas expositivo e a maioria dos usuários aprendem melhor com um treinamento prático, a QC3 estará refutando/derrubando o risco em questão.

5.3.5. Esquema de argumento sobre risco de tempo para capacitação

O risco “o tempo disponível para desenvolver o projeto não é suficiente para capacitar a equipe envolvida” envolve uma questão recorrente em diferentes tipos de projetos de software. Este risco quase sempre causa transtornos durante o processo de desenvolvimento de um produto, como também pode levar o projeto a falhar se o prazo de entrega estimado não for cumprido. Neste contexto, a criação e representação do risco, por meio deste esquema

de argumentação, são fundamentais para que tal risco possa ser devidamente compreendido e porventura priorizado.

A classificação por tipos de argumentos para este esquema é a seguinte: AT03 – Planejamento; AT04 – Execução; AT06 – Negócio; AT08 – Organizacional. A estrutura do esquema de argumento sobre Risco de tempo de capacitação é descrita na Tabela 14:

Tabela 14 – Esquema de argumento sobre risco de tempo de capacitação.

5. Esquema de argumento sobre risco de tempo para capacitação.
<ul style="list-style-type: none"> • Risco: O tempo disponível para desenvolver o projeto não é suficiente para capacitar a equipe envolvida.
Premissa 1: O projeto tem tempo X para ser desenvolvido.
Premissa 2: O projeto tem participantes Y que precisam ser treinados.
Premissa 3: Visto que participantes Y precisam ser treinados, o tempo X não é suficiente para desenvolver o projeto.
Conclusão: Portanto, o tempo X disponível não é suficiente para desenvolver o projeto.
Questões Críticas:
QC1: O tempo X para treinar os participantes Y é maior que 30% da duração do projeto?
QC2: Realmente é necessário treinar participantes Y do projeto?
QC3: O treinamento tem que fazer parte do tempo X que o projeto tem para ser desenvolvido?

Uma possível instanciação das variáveis do “esquema de argumento sobre risco de tempo de capacitação” é apresentada na Tabela 14.

Tabela 15 – Instanciação do esquema de argumento sobre risco de tempo.

Premissa 1: O projeto tem 7 meses para ser desenvolvido.
Premissa 2: O projeto tem 3 programadores que precisam ser treinados.
Premissa 3: Com 3 programadores que precisam ser treinados, 7 meses não é suficiente para desenvolver o projeto.
Conclusão: Portanto, 7 meses disponível não é suficiente para desenvolver o projeto.

A QC3 (O treinamento tem que fazer parte do X que o projeto tem para ser desenvolvido?) é uma questão do esquema que pode ser usada com o intuito de derrubar um argumento proposto de acordo com esse esquema. Neste caso, se a resposta a esta questão for que o tempo de treinamento não está incluso no tempo para desenvolvimento do projeto, um argumento proposto tende a ser refutado. Essa questão também pode ser usada para apoiar o risco proposto com o intuito de priorizá-lo. Caso a resposta da questão for que o tempo de treinamento está dentro do cronograma para o desenvolvimento do projeto, o argumento referente a proposta deste risco seria reforçado em uma discussão.

6. SISTEMA DE DISCUSSÃO DE RISCOS: *Risk Discussion System*

Para que um projeto seja bem-sucedido, é necessário que o produto desenvolvido seja entregue no prazo, dentro do orçamento e que cumpra os requisitos definidos pelo cliente. Para cumprir todos estes fatores e evitar que algum evento/risco se concretize e ameace o sucesso do projeto, esta dissertação possibilitou o planejamento e desenvolvimento de um sistema colaborativo que tem por objetivo auxiliar, coordenar e facilitar o processo de gerenciamento de riscos. Este sistema, na sua primeira versão – *RD System v1.0*, foi desenvolvido para apoiar e suportar discussões colaborativas de riscos em tarefas de gerenciamento de riscos (SEVERO; FONTOURA; SILVA, 2013). O sistema visa promover a participação dos diferentes membros de uma equipe de projeto nas atividades de gerenciamento de riscos, mesmo que estes se encontrem geograficamente distribuídos. As atividades colaborativas de discussão apoiadas e controladas pelo *RD System* seguem um protocolo de interação bem definido, o qual é fundamentado na noção de jogos de diálogo (SEVERO; FONTOURA; SILVA, 2013), discutido na seção 4.1.

Experimentos práticos com a versão 1.0 do *RD System* permitiu observar que existe a necessidade de formalizar os argumentos apresentados como proposições de riscos em um projeto. Assim como proposto nesta dissertação, isso é realizado a partir do emprego de esquemas de argumentação. Uma das principais funções destes esquemas, tal como descrito em (BEX; REED, 2011; MCLEOD; FERGUSON; BURGER, 2009), é apoiar a solução de conflitos de opiniões, caso estes existam em um debate. Em gerenciamento de riscos não é diferente, visto que em uma discussão de riscos os participantes podem ter dificuldades de selecionar e identificar os riscos mais relevantes para um projeto. Neste contexto, uma expansão do sistema de discussão de riscos, o *RD System v2.0*, foi desenvolvida no nosso projeto. Esta nova versão do sistema possui módulos para manter esquemas de argumentação (ver Figura 11), como também possui recursos voltados para o uso sistemático destes esquemas em meio a discussões de riscos. Para melhor entender como este sistema está organizado, a Figura 7 apresenta uma visão da arquitetura do *RD System v2.0*.

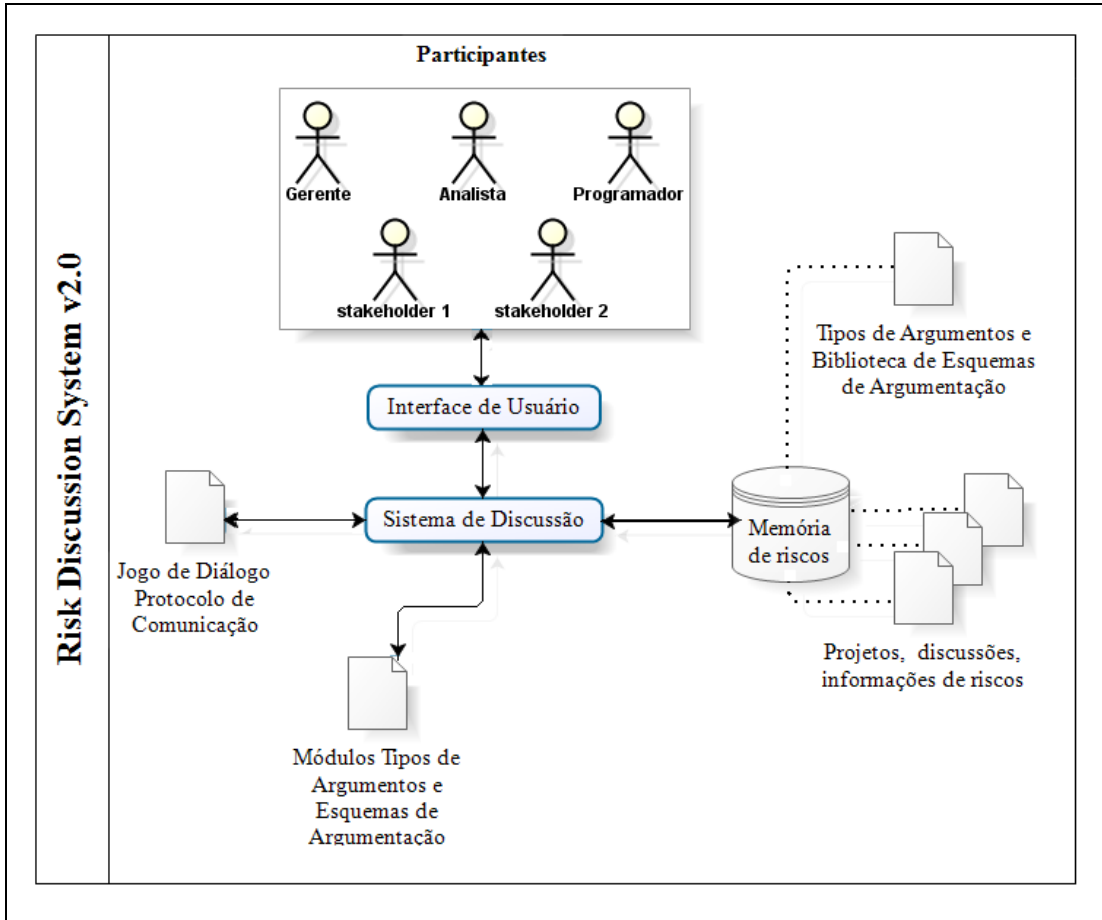


Figura 7 – Arquitetura do RD System v2.0.

Conforme a arquitetura apresentada na Figura 7, os participantes de um projeto podem acessar o *RD System* por meio de uma Interface de Usuário, onde estes podem desenvolver atividades colaborativas de riscos. O ambiente do *RD System* é organizado e controlado por um Sistema de Discussão, o qual é responsável por carregar e interpretar uma representação externa do Protocolo de Comunicação desenvolvido por (SEVERO; FONTOURA; SILVA, 2013) na v1.0 do *RD System*. Esta representação externa do jogo de diálogo é relevante para simplificar possíveis passos de adaptação do protocolo, os quais podem ser necessários quando uma organização desejar alterar/personalizar os elementos do protocolo para suas necessidades específicas. Para melhor entender o ambiente do *RD System* que é dividido em três módulos principais, e cada módulo possui submódulos particulares, são eles: a) Módulo de Projetos, onde inicialmente são cadastrados os projetos a serem discutidos no ambiente do *RD System* (Figura 8), dentro deste módulo de Projetos, temos acesso ao submódulo “Discussão”, onde são cadastradas as discussões de cada projeto (Figura 9), para que desta forma as discussões possam se acessadas e desenvolvidas/discutidas (Figura 13);

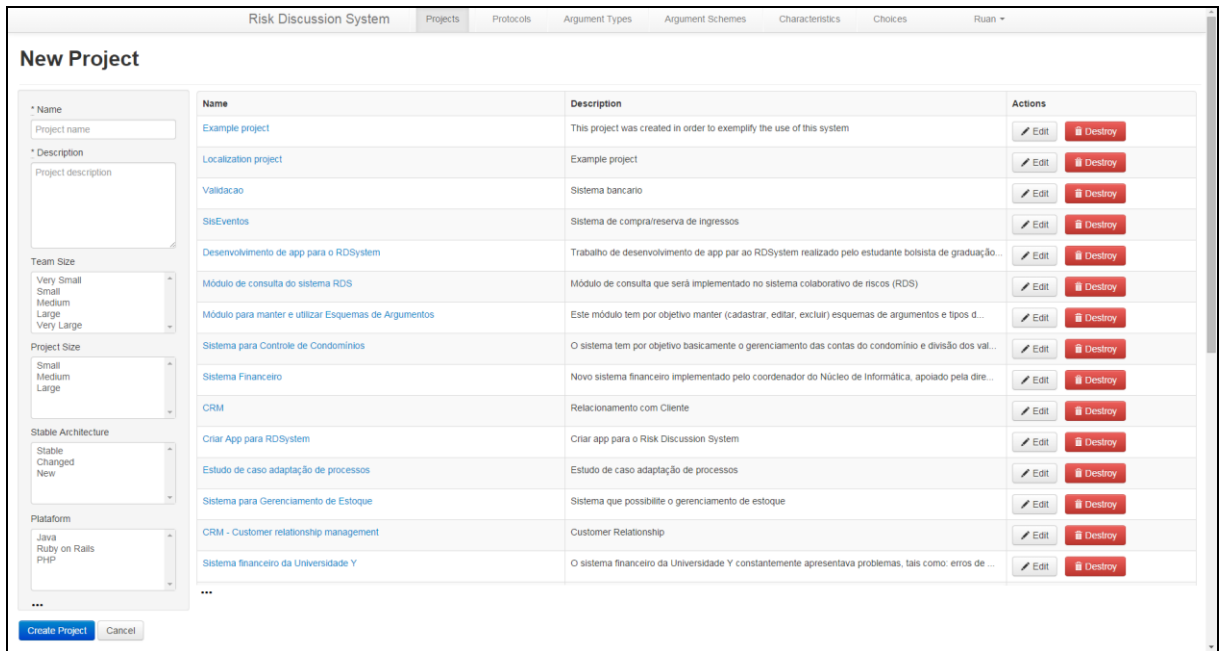


Figura 8 – Módulo do RD System v1.0 para manter Projetos



Figura 9 – Módulo do RD System v1.0 para manter e acessar as Discussões

b) Módulo de Protocolos, este módulo é reponsavel por manter os jogos de diálogo que são cadastrados no RD System (Figura 10);

The screenshot shows the 'New Protocol' interface in the Risk Discussion System. At the top, there are tabs for 'Projects' and 'Protocols'. The main heading is 'New Protocol'. On the left, there is a form with a '* Name' label and an input field. Below the input field are 'Create Protocol' and 'Cancel' buttons. On the right, there is a table with two columns: 'Name' and 'Actions'. The table lists four protocols: RMDG, PGQM, Dependence Risk Protocol, and security protocol. Each protocol has an 'Edit' button (with a pencil icon) and a 'Destroy' button (with a trash icon). At the bottom center, there is a green button with a plus icon and the text 'New Protocol'.

Figura 10 – Módulo do RD System v1.0 para manter Protocolo.

c) Módulo Esquemas de Argumentação, este módulo foi inserido no RD System na sua versão v2.0, este módulo é onde são cadastrados os esquemas de argumentação criados para proposição de riscos (Figura 11), como também os tipos de argumentos no seu submódulo “Tipos de Argumentos” (Figura 12).

The screenshot shows the 'New Argument Scheme' interface. At the top, there is a heading 'New Argument Scheme'. Below it, there is a section for '* Type(s)' with checkboxes for Client, Requirements, Planning, Execution, Maintenance, Bussiness, Technology, Organizational, and Tools. Below this, there are four input fields: '* Name' (with the text 'Name's Argument Scheme'), '* Premise(s)' (with the text 'Premise's Argument Scheme'), '* Conclusion' (with the text 'Conclusion's Argument Scheme'), and '* Critical Question(s)' (with the text 'Critical Question's Argument Sch'). Below these fields are 'Create Argument scheme' and 'Cancel' buttons. On the right side, there is a preview window showing the generated argument structure. The preview includes:

- Argument Type(s):** Client, Planning, Bussiness, Technology, Organizational, Tools
- Name:** Esquema de argumento sobre falta de treinamento
- Premise(s):**
 - X (Ruby) e uma tecnologia nova.
 - Y (O gerente) quer usar X (Ruby) em seus projetos.
 - Y (O gerente) não tem conhecimento sobre X (Ruby).
 - X (Ruby) não possui treinamento disponível.
- Conclusion:** Portanto, Y (o gerente) não poderá usar X (Ruby).
- Critical Question(s):**
 - Y não pode usar outra tecnologia?
 - É necessário usar X?
 - O uso de X ira melhor a qualidade do projetos?

 At the bottom of the preview window, there is a 'Back' link.

Figura 11 – Módulo do RD System v2.0 para manter Esquemas de Argumentação.

The screenshot shows the 'Argument Types' management interface. At the top, there are navigation tabs: Risk Discussion System, Projects, Protocols, Argument Types (selected), Argument Schemes, and Characteristics. Below the tabs, there's a search bar for 'New Argument Type' and a 'Create Argument type' button. The main content is divided into two sections:

- Left Sidebar:** A list of argument types with 'Edit' and 'Destroy' buttons for each.

Name	Actions
Client	Edit, Destroy
Requirements	Edit, Destroy
Planning	Edit, Destroy
Execution	Edit, Destroy
Maintenance	Edit, Destroy
Bussiness	Edit, Destroy
Technology	Edit, Destroy
Organizational	Edit, Destroy
Tools	Edit, Destroy
Others	Edit, Destroy
- Main Content Area:**
 - Argument Type: Client** header with 'Edit' and 'Back' buttons.
 - Related Argument Schemes** table:

Name	Actions
Esquema de argumento sobre Risco de falta de usabilidade	Show, Destroy
Esquema de argumento sobre Risco de falta de envolvimento	Show, Destroy
Esquema de argumento sobre falta de treinamento	Show, Destroy
Esquema de argumento sobre Risco de Cliente	Show, Destroy
Esquema de argumento sobre Risco de Viabilidade do Projeto	Show, Destroy
Esquema de argumento sobre Risco de Pessoas	Show, Destroy
Esquema de argumento sobre riscos de mudança de requisitos	Show, Destroy
Esquema de argumento sobre Risco de Falta de Integrante	Show, Destroy
 - A 'New Argument scheme' button at the bottom.

Figura 12 – Submódulo do RD System v2.0 para manter Tipo de Argumento

Em particular, o Sistema de Discussão organiza e controla as interações entre os participantes, as quais ocorrem de acordo com regras de combinação entre locuções e utilização dos esquemas de argumentação. Como resultado, tais discussões são estruturadas e registradas em uma Memória de Riscos.

6.1. O uso de esquemas de argumentação em discussões colaborativas de riscos

Durante a proposição de argumentos a serem usados em uma discussão de riscos, um participante pode amplamente utilizar esquemas de argumentação para proposição e discussão de riscos armazenados na memória do *RD System v2.0*. Assim como desenvolvido nesta dissertação, o processo de utilização proposto para estes esquemas pode ser resumido em três formas.

6.1.1. Propor um novo risco na discussão usando um esquema de argumentação para gerenciamento de riscos

Nesta forma de utilização de esquemas em discussões colaborativas de riscos, o primeiro passo é identificar a área do risco a ser proposto, neste caso, o tipo do risco (1) (ex.: execução, negócio, pessoa, etc.). Após a escolha do tipo do risco no *RD System v2.0*, todos os esquemas de argumentação indexados (2) ao mesmo estarão disponíveis para serem selecionados por um participante. Uma vez que tal seleção seja realizada, todas as partes do esquema selecionado são apresentadas para o usuário. Com isso, a ideia e o contexto do risco que o esquema de argumento está representando podem ser examinados. Desta forma, é possível selecionar partes reutilizáveis da estrutura do esquema (premissas, conclusão, questões críticas) na construção de uma proposta de risco para a discussão corrente. Feita a escolha de quais partes do esquema devem ser re-usadas, o próximo passo é “copiar” (3) o esquema para um campo de edição de argumentos a serem apresentados em uma discussão, o qual é apresentado na forma de texto livre (*Free Text*). Isso permite que o usuário realize modificações no texto de propostas de riscos que porventura sejam consideradas necessárias, bem como o preenchimento/instanciação das variáveis do esquema. Com as variáveis preenchidas com termos que sejam apropriados para a discussão de riscos do projeto corrente, o próximo passo é selecionar uma locução (4) adequada para indexar o argumento recém construído. Neste caso, a locução “propor risco” do protocolo de comunicação utilizado no nosso trabalho deve ser utilizada. Ao final, o argumento construído deve ser submetido (5) junto a árvore de discussão para identificarmos este argumento, além da locução *Propose risk* é sinalizado com o ícone “S”, assim como pode ser observado na Figura 13.

The screenshot displays the RDSYSTEM v2.0 interface. On the left, a list of risk discussions is visible, including:

- [244] Propose Risk: Módulo de consulta é muito complexo e possui problemas de usabilidade. - Ruan Pozzebon
- [249] Ask: A pessoa responsável não pode melhorar a usabilidade do módulo? - Nielsen Rechia
- [250] Inform: Reuniões para discussão sobre usabilidade e melhoria podem resolver o problema, acredito. - Catherine Barchet
- [254] Argument con: Acredito, que testes e demonstração com usuários sem conhecimento do módulo seria mais apropriado - Nielsen Rechia
- [251] Inform: Acredito que o gerente deveria fazer experimentos, para ter um feedback dos usuários e assim poder melhorar a usabilidade do módulo. - Maicon Balke
- [255] Argument pro: Concordo, pois feedback de usuários é sempre bom para qualquer projeto - Nielsen Rechia
- [268] Ask: O módulo tem condições de ir para experimentos? - Catherine Barchet
- [252] Propose Probability: Alta - Ruan Pozzebon
- [256] Ask Position: Todos acreditam que a probabilidade é alta? - Nielsen Rechia
- [266] Argument con: Acredito que o modulo tera melhorias com o decorrer do tempo sendo assim a probabilidade seria media - Nielsen Rechia
- [277] Summarize: A probabilidade do risco é alta. - André Lúcio Hahn
- [253] Propose Impact: Médio - Ruan Pozzebon
- [260] Ask Position:
- [271] Summarize:
- [273] Propose Plan: R determinado do projeto para utilização - Nielsen Rechia

The main area shows a 'Free Text' editor with the text: 'Portanto, X (RD System) possui problemas de usabilidade.' Below the editor are 'Submit' and 'Unbind Scheme' buttons.

On the right, the 'Argument Scheme' editor is active, showing:

- 'Client' dropdown set to '1'.
- 'Esquema de argumento sobre' dropdown set to '2'.
- 'Premises' section with a checked box and text: 'X (RD System) é um sistema complexo. Y (a equipe de desenvolvimento) desenvolveu X (RD System) sem pensar na usabilidade. Se X (RD System) é um sistema complexo e Y (a equipe de desenvolvimento) desenvolveu X (RD System) sem pensar na usabilidade, logo não é um sistema funcional.'
- 'Conclusion' section with text: 'Portanto, X (RD System) possui problemas de usabilidade.'
- 'Link to AS' and 'Copy to Free Text' buttons.
- 'Critical Question(s)' section with text: 'Não pode ser feito treinamento para o uso de X? Y não pode melhorar a usabilidade de X?'
- 'Query' section at the bottom.

Figura 13 – Tela do RDSYSTEM v2.0 para adicionar um Esquema de Argumentação a Discussão de Riscos – Forma I.

6.1.2. Conectar/anotar um esquema de argumentação para gerenciamento de riscos ao texto de um risco proposto anteriormente na discussão

Esta forma de usar esquemas de argumentação em discussões colaborativas de riscos é relevante na contextualização de um risco, buscando complementar a ideia apresentada em propostas de riscos e/ou fornecer informação adicional para apoiar um risco já proposto anteriormente através de um argumento descrito em texto livre (argumento construído originalmente sem qualquer referência a esquemas disponíveis em uma biblioteca). Esta forma de utilizar esquemas (II) em discussões colaborativas de riscos funciona da seguinte maneira. Primeiro, o participante deve selecionar um risco (1) existente na árvore de discussão. Esta seleção pode ser motivada pela dificuldade de compreensão destes riscos, bem como por eles estarem inconsistentes com outras informações levantadas na discussão do projeto, ou mesmo porque estes riscos estão descritos de forma incompleta (assim como observado em experimentos desenvolvidos nestes projetos, muitos usuários com pouca experiência em gerenciamento de riscos tendem a apresentar argumentos pouco compreensíveis por diversos motivos). Após selecionar este risco, o usuário deve identificar a sua área, neste caso, o tipo do risco (2) (ex.: execução, negócio, pessoa, etc.). Uma vez que

um tipo seja selecionado, esquemas indexados por aqueles tipos serão apresentados para o usuário. A partir desta lista de esquemas, o usuário então deve selecionar um esquema de argumentação (3) a ser usado na indexação do argumento proposto na discussão corrente (por exemplo, uma proposição de riscos submetida a partir do uso de uma locução “propor risco”). Após a conclusão de tais etapas, o argumento é conectado/anotado (4) explicitamente pelo esquema, assim conectando um esquema a um risco, tal como ilustrado na Figura 14.

The screenshot displays the RDSysSystem v2.0 interface. On the left, a list of discussion items is visible, including a highlighted item [244] 'Propose Risk: Módulo de consulta é muito complexo e possui problemas de usabilidade.' and several other items with icons for Ask, Inform, Argument con, Argument pro, and Summarize. On the right, the 'Argument Scheme' configuration panel is shown. It includes a 'Client' dropdown menu (labeled 2) and an 'Esquema de argumento sobre' dropdown menu (labeled 3). The 'Premise(s):' section contains a text area with the following content: 'X (RD System) é um sistema complexo. Y (a equipe de desenvolvimento) desenvolveu X (RD System) sem pensar na usabilidade. Se X (RD System) é um sistema complexo e Y (a equipe de desenvolvimento) desenvolveu X (RD System) sem pensar na usabilidade, logo não é um sistema funcional.' The 'Conclusion' section contains a text area with the text: 'Portanto, X (RD System) possui problemas de usabilidade.' Below this are buttons for 'Link to AS' (labeled 4) and 'Copy to Free Text'. The 'Critical Question(s):' section contains a text area with the text: 'Não pode ser feito treinamento para o uso de X? Y não pode melhorar a usabilidade de X?'

Figura 14 – Tela do RDSysSystem v2.0 para adicionar um esquema de argumentação a discussão de riscos – Forma II.

6.1.3. Questionar a veracidade de uma proposta de risco que é construída segundo um esquema de argumentação para gerenciamento de riscos

O questionamento de riscos propostos tende a impactar na fluência e profundidade de uma discussão de riscos, bem como na consequente seleção de riscos que devem ser priorizados em um projeto. Assim como proposto nesta dissertação, este questionamento pode ser realizado a partir da análise e utilização de questões críticas descritas em esquemas de

argumentação. Desta forma, este questionamento baseado em questões críticas de um esquema é feito da seguinte forma: o participante de uma discussão pode clicar no ícone “S” de um esquema usado na proposição de riscos em um debate. Com isso, este usuário pode visualizar a estrutura completa do esquema utilizado. Visualizando este esquema e suas questões críticas associadas, o usuário pode então selecionar uma questão crítica que atenda as suas necessidades de argumentação na discussão corrente. Com a questão crítica selecionada, o próximo passo é editar o conteúdo textual desta questão, bem como preencher as variáveis descritas na questão selecionada. Para submeter tal questão na discussão corrente, o participante da discussão deve empregar um ato de locução adequado para o uso de uma questão crítica. No protocolo de discussão de riscos, este ato de locução é “perguntar”, visto que este movimento de argumentação tornar o questionamento explícito no debate. Neste caso, a utilização da questão crítica pode auxiliar no questionamento da veracidade do argumento proposto conforme um esquema de argumentação selecionado. Contudo, tais questões críticas podem também ser usadas em conjunto com outros atos de locuções além de perguntar. Se questões críticas forem usadas com atos de locução argumentar a favor, argumentar contra, informar, propor plano, impacto ou probabilidade, por exemplo, este uso também poderia ser considerado adequado (neste caso, o conteúdo da discussão seria melhorado, sem haver a adição de uma inconsistência no processo de diálogo) assim como observado em experimentos práticos desenvolvidos no nosso projeto. Neste caso, embora o ato de questionar possa ficar implícito no diálogo, questões críticas também podem motivar a construção e submissão de outros movimentos de argumentação em uma discussão colaborativa de riscos. Por exemplo, pode ser utilizada a questão crítica – QC3 do “esquema de argumento sobre risco de tempo de capacitação”, conforme Tabela 14 – “O treinamento tem que fazer parte do tempo X que o projeto tem para ser desenvolvido?” juntamente com o ato de locução “argumentar contra”, visto que é sabido que o tempo de treinamento não faz parte do cronograma de desenvolvimento do projeto. A questão crítica, neste caso, estará implícita no diálogo gravado, porém ela estará contextualizando a ideia de que o risco proposto não é relevante para o projeto, pois a falta de experiência ou conhecimento da equipe sobre uma determinada tecnologia pode ser sanada com um treinamento e este tempo de treinamento não será computado no tempo de desenvolvimento do projeto.

Após ser feita a escolha de qual ato de locução vai ser utilizado em uma discussão, deixando a questão crítica explícita ou implícita no debate, o argumento construído a partir do reuso do conteúdo de informação descrito em um esquema é submetido na árvore representando a discussão de gerenciamento de risco, conforme Figura 15.

Esquema de argumento sobre Risco de falta de usabilidade

Premise(s):

X (RD System) é um sistema complexo.

Y (a equipe de desenvolvimento) desenvolveu X (RD System) sem pensar na usabilidade.

Se X (RD System) é um sistema complexo e Y (a equipe de desenvolvimento) desenvolveu X (RD System) sem pensar na usabilidade, logo não é um sistema funcional.

Conclusion

Portanto, X (RD System) possui problemas de usabilidade.

Critical Question(s):

Não pode ser feito treinamento para o uso de X? Use

Y não pode melhorar a usabilidade de X? Use

Figura 15 – Tela do RD System v2.0 para uso das questões críticas

7. TRABALHOS RELACIONADOS

A partir da pesquisa realizada nesta dissertação, podemos afirmar que não há estudos anteriores que discutam a construção e utilização de esquemas de argumentação voltados para auxiliar o processo de gerenciamento de riscos em projetos de desenvolvimento de software. No entanto, o emprego de esquemas de argumentação tem sido estudado em outras áreas de aplicação, tais como: transplante de órgãos (TOLCHINSKY; MODGIL; ATKINSON, 2012), biologia (genética) (MCLEOD; FERGUSON; BURGER, 2009) e sistemas de segurança (YUAN; KELLY, 2012). Este capítulo brevemente revisa trabalhos nestas diferentes áreas de aplicação, onde tais trabalhos discutem o papel de esquemas no apoio a problemas de tomada de decisão.

7.1. Esquemas de argumentação em sistemas de segurança

O trabalho de Yuan (YUAN; KELLY, 2012) discute a importância do investimento em segurança em sistemas de computador, visto que a demanda destes tem aumentado cada vez mais.

Yuan desenvolveu uma biblioteca de esquemas com o intuito de controlar e listar os principais riscos de segurança em sistemas de computadores de alta complexidade. Os esquemas de argumentação propostos possuem um conjunto de questões críticas que têm a tarefa de sustentar ou derrubar um argumento proposto de acordo com o modelo de um esquema selecionado.

No nosso trabalho, em particular, as questões críticas além de testarem a veracidade dos esquemas de argumento, servem para aprofundar o nível da discussão. Além disso, essas questões podem ser associadas a atos de locuções (*inform*, *argument-con*, *argument-pro*, *propose probability*, por exemplo) com o papel de contextualizar os argumentos apresentados a partir do emprego dessas locuções.

Um esquema de argumentação proposto que pode ser citado é o esquema de argumentação de “prevenção de riscos” (adaptado de (YUAN; KELLY, 2012)):

- Premissa: Todos os riscos identificados para o sistema X são abordados.
- Conclusão: Sistema X é aceitavelmente seguro.

- QC1. O quão completa é a lista de riscos identificados?
- QC2. Qual é a precisão de cada um dos riscos identificados?
- QC3. Os riscos são controlados de forma adequada?

7.2. Esquemas de argumentação auxiliando em conflitos de banco de dados biológicos

O trabalho de McLeod (MCLEOD; FERGUSON; BURGER, 2009) discute o uso de esquemas de argumentação para resolver conflitos de banco de dados biológicos. Nestes sistemas, resultados de experimentos biológicos são armazenadas em banco de dados. Porém, devido à complexidade destes experimentos, as decisões tomadas e os resultados obtidos podem ser contraditórios. Desta forma, resolver estes conflitos é fundamental visto que este banco de dados pode se tornar instável. Devido a este problema, esquemas de argumentação são empregados no auxílio à construção de conclusões as quais podem ser alcançadas a partir de resultados objetivos obtidos nestes experimentos.

A biblioteca de esquemas de argumentação proposta por Mc Leod é classificada por ordem de importância por um especialista da área. Questões críticas são utilizadas para aprovar o uso ou não de um esquema na solução destes conflitos no sistema, pois McLeod afirma que os esquemas só podem ser empregados por usuários caso eles responderem todas as QC positivamente. Caso exista alguma resposta negativa, um esquema não poderá ser usado no sistema, visto que ele pode comprometer o resultado de alguma análise biológica a ser gravada no banco de dados.

Um exemplo de esquema que auxilia na especificação de um resultado de experimento é o esquema de “anotações textuais e anotações espaciais”, adaptado de (MCLEOD; FERGUSON; BURGER, 2009):

- Premissa 1: Resultado R1 veio de uma anotação textual.
- Premissa 2: Resultado R2 veio de uma anotação espacial.
- Premissa 3: Anotações textuais podem ser mais precisas do que anotações espaciais.
- Conclusão: Portanto, R1 pode ser mais preciso do que R2.

Questões críticas:

- Quem criou a anotação espacial?
- Quem aprovou a anotação espacial?
- Existe um conflito genuíno entre as anotações espaciais e textuais?
- A anotação textual é confiável?
- Que anotações podem ser obtidos a partir de outros experimentos?
- Que conclusões podem ser obtidas a partir de outras análises?

7.3. Esquemas de argumentação no auxílio a tomada de decisões de transplante de órgãos

Outro trabalho relevante para esta dissertação, o qual utiliza esquemas de argumentação na solução de problemas em domínios de aplicação particulares, é o trabalho de Tolchinsky et al. (TOLCHINSKY; MODGIL; ATKINSON, 2012). Os esquemas propostos neste trabalho foram criados a partir da biblioteca de esquemas apresentada por Walton (WALTON; REED; MACAGNO, 2008). Este trabalho propõe uma discussão, a qual é sustentada pelo conjunto de duas técnicas de argumentação: jogos de diálogo e esquemas de argumentação. O processo de discussão desenvolvido com o emprego destas técnicas de argumentação tem o objetivo de decidir a viabilidade de doação de um órgão a ser transplantado em um determinado paciente. Em particular, esse processo de tomada de decisão é apoiado por esquemas de argumentação. Por exemplo, os esquemas auxiliam na decisão de qual paciente é mais indicado a receber um determinado transplante. Em particular, esse trabalho discute a utilização de um jogo de diálogo que é usado apenas internamente em um sistema desenvolvido (composto por três camadas: *i) Open Stage*, *ii) Deliberation Stage (a-Context Layer, b-Argumentation Layer e c-Endorsement Layer)*, *iii) Resolution Stage; contendo os seguintes atos de locução: i(open_dialogue, enter_dialogue), ii((a-assert, retract, propose, retract), (b-argue, challenge), (c-endorse, retract)) e iii(no_more_moves, time_out, solution, accept, close_deliberation)).*

O sistema CARREL implementa o modelo proposto. Assim como implementado neste sistema, um exemplo de esquema pode ser apresentado: um agente doador - DA1 - apresenta

o argumento A1 para a inviabilidade do pulmão de um doador. Neste caso, Mark é doador de pulmão. Devido a este fato, o órgão está relacionado a um histórico de tabagismo, e histórico de tabagismo é uma contraindicação para a doação de um pulmão. Sendo assim, o pulmão não é viável para transplante. Em resumo, o argumento aqui apresentado é uma instanciação do “esquema de não-viabilidade”:

- Premissa 1: D doador de órgão O tinha condição C
- Premissa 2: C é uma contraindicação para a doação de O
- Conclusão: Portanto, o órgão O é não viável.

As seguintes questões críticas estão associadas a este esquema:

- QC1: O doador D tinha um histórico de C?
- QC2: Uma história de C é uma contraindicação para a doação de O?

7.4. Esquemas de argumentação para gerenciamento de riscos

O emprego de esquemas em processos de debate está relacionado com a importância de apresentar informações mais completas (pouco ambíguas, por exemplo) em uma discussão colaborativa de riscos, bem como para facilitar processos de tomada de decisão neste domínio de aplicação. Neste caso, os esquemas propostos são *templates* de argumentos pré-definidos, os quais podem capturar argumentos típicos no domínio de aplicação estudado, bem como argumentos baseados em melhores práticas descritas na literatura de gerenciamento de riscos. Tais esquemas também podem capturar exemplos de argumentos considerados convincentes em discussões passadas, bem como argumentos representando *checklists* de riscos, entre outras fontes que podem ser exploradas para a ampliação da nossa biblioteca de esquemas na área de gerenciamento de riscos.

Esta dissertação, além de discutir a construção e emprego de uma biblioteca de esquemas para gerenciamento de risco, propõe a união de duas técnicas de argumentação, assim descrito por Tolchinsky et al. (TOLCHINSKY; MODGIL; ATKINSON, 2012). Em particular, essa dissertação explora a noção de jogos de diálogos para gerenciamento colaborativos de riscos, tal como apresentada por Severo et al. (SEVERO; FONTOURA; SILVA, 2013). Contudo, a nossa dissertação amplia essa proposta pelo emprego de esquemas de argumentação para gerenciamento de riscos. Porém, nós abordamos a união destas duas

técnicas de argumentação de maneira diferente do trabalho do Tolchinsky, pois os atos de locuções utilizados no nosso trabalho têm a tarefa de guiar uma discussão de riscos, expondo em qual etapa da discussão o participante está. No nosso trabalho, os atos de locuções além de guiarem as etapas da discussão podem ser usados em conjunto com esquemas de argumentação selecionados pelos participantes envolvidos no debate. Por exemplo, esquemas podem ser usados em determinadas etapas de uma discussão, tal como na proposição de um risco juntamente com o ato de locução *propose risk*. Da mesma maneira, outras locuções descritas no nosso protocolo de comunicação podem ser usadas em conjunto com esquemas de argumentação cadastrados no *RD System v2.0*. A seguir, a Tabela 16 apresenta uma comparação entre trabalhos que utilizam esquemas de argumentação em diferentes áreas de aplicação e podem ser relacionados a proposta descrita nesta dissertação.

Tabela 16 – Comparativo entre trabalhos relacionados.

	Yuan et al. (2012)	McLeod et al. (2009)	Tolchinsky et al. (2012)	Severo (2013)	Esquemas de Argumentação para Discussão de Riscos em Projetos de Software
Qual a área de aplicação destes trabalhos?	Segurança de sistemas.	Biológica, genética.	Transplante de órgãos.	Gerenciamento de riscos.	Gerenciamento de riscos.
Qual o formalismo de representação utilizado?	Não está claro o padrão de representação utilizado.	Utiliza uma linguagem semi-formal para a representação de premissas, conclusão e questões críticas, assim como proposto em Walton.	Utiliza uma linguagem baseada na proposta de Walton, embora esquemas e jogos de diálogo devam ser descritos em um formalismo lógico.	Jogos de diálogo foram representados em uma linguagem semi formal, composta por atos de locução e regras de combinação entre estes.	Utiliza uma linguagem semi-formal para a representação de jogos de diálogo, bem como para a representação de esquemas em termos de premissas, conclusão e questões críticas, assim como proposto em Walton.
Como os modelos de argumentação foram desenvolvidos?	Esquemas foram desenvolvidos em um processo tradicional de engenharia de conhecimento (baseado em um estudo da literatura de engenharia da segurança de sistemas).	Esquemas foram construídos a partir de consultas típicas e resultados destas, as quais estavam sendo realizadas em bancos de dados biológicos.	Usando um processo tradicional de engenharia de conhecimento, esquemas foram baseados em normas (guidelines) de transplante de órgãos, bem como a partir de revisões de especialistas no domínio de aplicação.	Um jogo de diálogo foi criado a partir do estudo de casos práticos de discussões de riscos, bem como a partir da análise de jogos de diálogo propostos na literatura.	Esquemas de argumentação são desenvolvidos seguindo uma sequência de passos. (<i>Checklists</i> , discussões passadas, literatura de GR, diagrama de espinha de peixe...).
Quais os problemas que os modelos de argumentação estão buscando atacar?	Esquemas são empregados no controle e listagem de riscos de segurança.	Esquemas são usados para apoiar o processo de solução de inconsistências entre resultados obtidos a partir de diferentes métodos usados na obtenção de dados biológicos.	Jogos de diálogo são utilizados juntamente com esquemas para guiar o processo de decisão de viabilidade de transplantes de órgãos.	Jogos de diálogo são utilizados na padronização de um processo de discussão de riscos (mediar tais discussões pelo emprego de um protocolo de comunicação bem definido), bem como construir uma memória reusável de discussões de riscos.	Jogos de diálogo e esquemas são usados na padronização e apoio ao processo de proposição e discussão de riscos em discussões colaborativas.
Qual o papel das questões críticas?	Questões críticas são usadas para indicar pontos de fraqueza (tais como suposições e hipóteses, por exemplo) usadas na construção de argumentos onde dúvidas podem ser apresentadas por meio de contra-argumentos. Neste caso, desafios e ataques podem ser feitos a partir dessas fraquezas.	Questões críticas são usadas para sustentar ou derrubar um esquema proposto.	Questões críticas são usadas para determinar quais possíveis movimentos dialéticos (atacar ou fortalecer) podem ser usados em uma discussão. Elas também fazem a conexão entre os diferentes argumentos apresentados na discussão, em um grafo onde esquemas são usados como nodos e questões críticas são usadas como arcos.	Questões críticas não foram exploradas neste trabalho.	Questões críticas têm o papel de testar a veracidade de riscos propostos através de esquemas de argumentação, como também estimular o aprofundamento de uma discussão pela utilização de em conjunto com atos de locução apropriados.
Existem sistemas desenvolvidos?	Não existe informação disponível sobre um sistema que esteja disponível.	Há um sistema no qual os esquemas são carregados e visualizados, permitindo o uso destes esquemas.	Sistema CARREL (contendo uma interface web e um mecanismo de raciocínio lógico voltado para a representação e computação de argumentos).	Risk Discussion System versão 1.0. Os argumentos de uma discussão são visualizados na forma de uma árvore, possibilitando o desenvolvimento organizado de um debate colaborativo de gestão de riscos.	Risk Discussion System versão 2.0 (contendo um módulo para a representação e utilização de esquemas de argumentação, utiliza esquemas para apoiar o desenvolvimento de discussões colaborativas de riscos, com recursos para consultas de argumentos apresentados em discussões passadas).

8. EXPERIMENTOS DE VALIDAÇÃO

Para validar o uso de esquemas de argumentação em um processo de gerenciamento de riscos de projetos de software desenvolvido no ambiente de discussão de riscos do *RD System 2.0*, foram realizados dois experimentos. O primeiro experimento foi realizado com a equipe de desenvolvimento de software da cooperativa de médicos, Unimed de Ijuí/RS. O segundo experimento foi realizado envolvendo alunos da graduação da UFSM.

Os experimentos foram realizados após um treinamento breve sobre gerenciamento de riscos. Após o treinamento ter sido concluído, foi escolhido um projeto para realizar o gerenciamento de riscos, sendo que a equipe da Unimed realizou o gerenciamento de um sistema real e os alunos da UFSM realizaram o gerenciamento de riscos de projetos propostos. Ao final dos experimentos, foram aplicados questionários aos participantes, contendo questões de múltipla escolha, organizadas na escala: concordo plenamente; concordo parcialmente; não concordo nem discordo; discordo parcialmente; e discordo plenamente. Tais experimentos foram organizados com configuração e objetivos específicos, conforme apresentado nas seções seguintes, onde estes experimentos são descritos em detalhes. A partir deles, os resultados dos questionários respondidos pelos participantes e as conclusões alcançadas a partir da análise destes resultados são apresentadas.

O objetivo dos experimentos foi investigar as possíveis melhorias que esquemas de argumentação poderiam apresentar em um ambiente colaborativo de discussão de riscos. Em particular, os subobjetivos dos experimentos de validação foram: a) avaliar se esquemas de argumentação para gerenciamento de riscos podem auxiliar o processo de discussão de riscos (fluência e profundidade da discussão); b) avaliar se esquemas de argumentação para gerenciamento de riscos podem facilitar a proposição de riscos (completude e clareza de riscos propostos); c) avaliar a usabilidade do sistema colaborativo para gerenciamento de riscos (*RD System v2.0*);

Com base em resultados de experimentos passados (SEVERO; FONTOURA; SILVA, 2013), os quais tinham como objetivo avaliar o uso de jogos de diálogos em discussões colaborativas de riscos, decidiu-se avaliar possíveis melhorias que porventura poderiam ser obtidas com a integração de esquemas de argumentação e jogos de diálogos no suporte ao desenvolvimento de discussões voltados para o gerenciamento de riscos em projetos de software.

8.1. Primeiro experimento: Cooperativa Médica – Unimed Ijuí/RS

Um estudo de caso em uma empresa de desenvolvimento de sistemas médicos foi realizado nesta dissertação. O principal objetivo deste estudo de caso foi buscar novas formas de validar a abordagem de uso de esquemas de argumentação no auxílio da proposta de riscos em discussões colaborativas de gerenciamento de riscos.

8.1.1. Configuração do experimento na Cooperativa Médica – Unimed Ijuí/RS

A equipe de desenvolvimento do projeto escolhido para a realização do experimento é formada por três programadores, três analistas de suporte, um analista de sistemas e um gerente de projetos. Esta equipe costuma executar atividades de gerenciamento de riscos sem seguir um método particular. Em geral, este gerenciamento de riscos é feito manualmente, onde é produzida uma lista de possíveis riscos complementando a documentação e análise dos sistemas propostos.

O projeto discutido neste estudo de caso foi um projeto que estava sendo iniciado na empresa. Este projeto envolvia a construção de um sistema de CRM (*Customer Relationship Management* – Gestão de Relacionamento com o Cliente). Neste projeto, o levantamento de riscos focalizou necessidades de atendimento ao cliente, as quais deveriam ser atendidas por um dos módulos do sistema de CRM. Este módulo deveria permitir o controle e gerenciamento das chamadas de atendimento, as quais seriam realizadas pelo setor de relacionamento ao cliente. Em geral, o objetivo deste novo módulo seria promover um melhor gerenciamento de chamadas de atendimento através de um controle de chamadas pendentes e *feedback* contínuo para clientes. Por fim, o módulo proposto deveria ser integrado a outros sistemas já existentes na empresa.

No estudo de caso, conceitos de gerenciamento de riscos e argumentação foram inicialmente apresentados para a equipe de desenvolvimento da empresa. Neste caso, o protocolo de comunicação usado para suportar discussões colaborativas de riscos no *RD System* foi apresentado brevemente. Esquemas de argumentação também foram explicados, onde tal explicação descreveu como e quando empregar estes esquemas em discussões. Uma vez que a apresentação do sistema *RD System v2.0* foi concluída, a equipe de

desenvolvimento envolvida no estudo de caso foi convidada a realizar uma discussão de riscos no projeto de CRM. Dúvidas relativas às etapas de discussão de riscos e do processo de utilização de esquemas foram esclarecidas durante a realização do experimento. Quando o tempo disponível para realizar a discussão colaborativa de riscos chegou ao fim (aproximadamente 1h 30min), os participantes foram convidados a responder as questões/afirmações do questionário apresentado abaixo:

- Q1. Qual seu nível de experiência em gerenciamento de riscos?
- Q2. É importante executar a etapa de gerenciamento de riscos em projetos de desenvolvimento de software.
- Q3. É importante discutir riscos de projetos de desenvolvimento de software de modo colaborativo.
- Q4. Esquemas de argumentação auxiliam o processo de discussão de riscos.
- Q5. Esquemas de argumentação facilitam o entendimento do risco proposto.
- Q6. Esquemas de argumentação facilitam a atividade de propor um risco.
- Q7. Esquemas de argumentação (questões críticas) auxiliam na continuidade, fluência da discussão.
- Q8. Esquemas de argumentação são fáceis de usar durante a discussão de riscos.
- Q9. Os recursos de discussão colaborativa de riscos implementados nos sistemas são fáceis de usar.
- Q10. O sistema contempla as principais etapas de gerenciamento de riscos (completude).

8.1.2. Discussão do experimento na Cooperativa Médica – Unimed Ijuí/RS

As respostas obtidas no questionário foram bastante homogêneas (Figura 16) e, em geral, positivas em relação aos objetivos de avaliação sendo estudados.

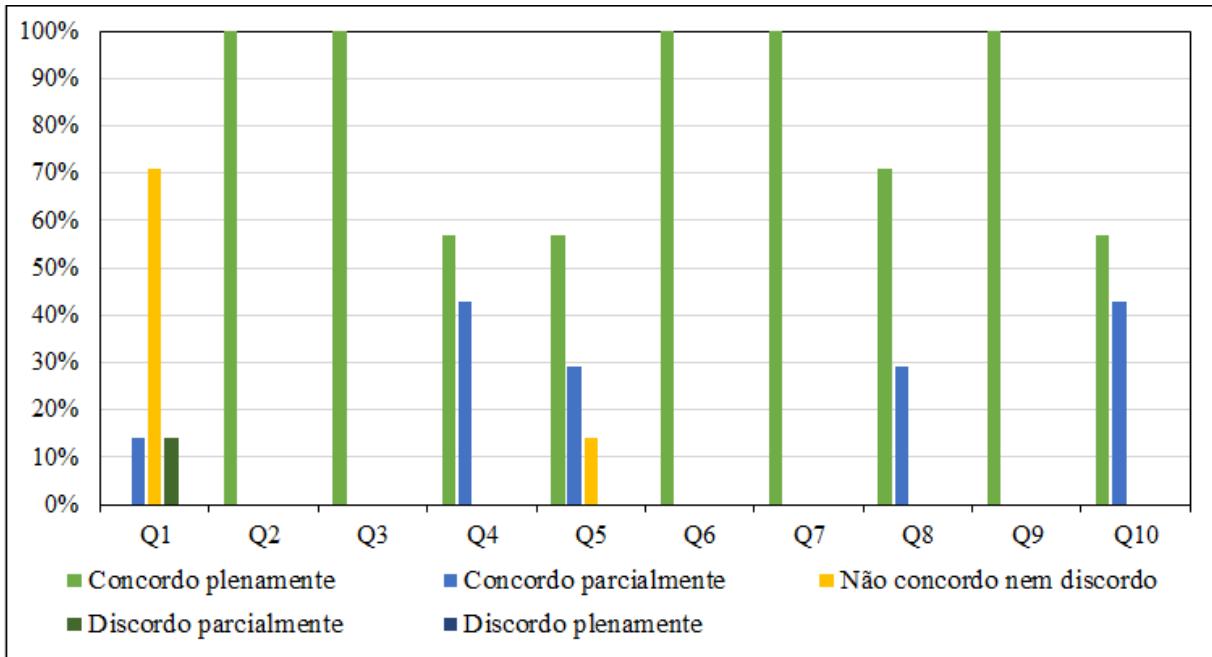


Figura 16 – Resultados do primeiro experimento.

Neste caso, os participantes deste estudo de caso afirmaram que eles acreditam que esquemas de argumentação podem melhorar o processo de discussão de riscos. Além disso, os participantes afirmaram que o uso de esquemas de argumentação permite que seja obtido um maior entendimento dos argumentos propostos neste tipo de discussão, uma vez que tais argumentos agora podem ser apresentados de forma mais completa. Os participantes também relataram que eles acreditam que esquemas podem facilitar a proposta de riscos em meio a uma discussão, visto que muitas vezes os participantes não consideram a proposta de riscos uma tarefa simples. Outro ponto importante percebido neste estudo de caso está relacionado à fluência do processo de discussão, onde tal fluência deve-se ao uso de um conjunto de questões críticas em conjunto com locuções do protocolo do tipo *Ask*, *Argument con* e *Argument pro*. Usando as questões críticas para dar continuidade à discussão, tende a acontecer uma etapa importante da gerência de riscos que é selecionar/priorizar os riscos mais prováveis de acontecerem ao longo do projeto, para assim serem propostos planos aos riscos selecionados. Isto ocorre no momento em que os participantes respondem as questões críticas, sendo que, se os participantes conseguirem responder as questões críticas estarão “apoiando” o risco proposto, caso contrário o risco será derrubado e abandonado na discussão.

Durante o estudo de caso, comentários, sugestões e questionamentos foram apresentados pelos participantes. Com relação ao nível de conhecimento sobre gerenciamento

de riscos, foi observado que muitos participantes deste estudo de caso não estavam familiarizados com etapas básicas de gerenciamento de riscos. Porém, este fato, comparando com a afirmação “Esquemas de argumentação auxiliam o processo de discussão de riscos”, pode-se observar, que mesmo os participantes não sendo experientes em gerenciamento de riscos, conseguem perceber que esquemas de argumentação auxiliam os mesmo no processo de discussão de riscos.

Questionamentos referentes à usabilidade e funcionalidade do *RD System v2.0* foram apresentados pelos participantes. Entre outros, os participantes perguntaram quais seriam os recursos de documentação disponíveis no sistema. Neste caso, participantes apresentaram a necessidade de obter um “relatório final” da discussão, o qual poderia ser colocado anexo à documentação do projeto. Por fim, comentários sobre o emprego de esquemas de argumentação foram apresentados pelos participantes. Entre outros, eles questionaram a necessidade de construir uma base de esquemas mais abrangente, assim claramente sugerindo a ampliação da biblioteca de esquemas de argumentação atualmente existente (no momento da realização do estudo de caso, apenas vinte esquemas de argumentação tinham sido cadastrados na memória do sistema).

8.2. Segundo experimento: alunos da graduação em computação da UFSM

Este segundo experimento foi realizado com alunos da graduação em computação da UFSM. O principal objetivo deste estudo de caso foi ressaltar que o uso de esquemas de argumentação auxilia usuários com pouca ou nenhuma experiência em gerenciamento de riscos a proporem riscos em meio a uma discussão colaborativas de gerenciamento de riscos.

8.2.1. Configuração do experimento envolvendo alunos de graduação em computação da UFSM

Este experimento envolveu vinte oito participantes, todos alunos da disciplina de gerenciamento de projetos de cursos de graduação em computação da UFSM.

A primeira etapa do experimento começou com um treinamento, por volta de cem minutos, sobre o desenvolvimento de atividades típicas de gerenciamento de riscos. Em uma segunda etapa deste treinamento, que teve igual duração, apresentou-se uma introdução sobre técnicas de jogos de diálogo e esquemas de argumentação, as quais seriam utilizadas para mediar uma discussão colaborativa de riscos desenvolvida durante o experimento. A apresentação destas técnicas de argumentação foi baseada em exemplos, onde os exemplos envolviam esquemas de argumentação da biblioteca de esquemas desenvolvida no nosso projeto, bem como locuções do protocolo de comunicação.

Em geral, a apresentação de esquemas e do protocolo foi bastante simplificada neste experimento. Neste caso, os esquemas apresentados estavam principalmente relacionados a riscos de requisitos, os quais deveriam ser analisados em um projeto de desenvolvimento de software proposto. Além disso, a apresentação do protocolo focalizou a utilização de algumas locuções particulares, tais como: *propose risk*, *ask*, *argument pro* e *argument con*. Em especial, estas locuções são normalmente selecionadas por participantes quando estes exploram o uso de esquemas de argumentação em uma discussão. Além disso, os principais recursos de discussão colaborativa do sistema *RD System v2.0* foram apresentados. Após este treinamento preliminar, os participantes receberam um material impresso de apoio contendo: a) um “dicionário” com as locuções que compunham o protocolo de comunicação (dicionário formado pela descrição e um exemplo de uso de cada locução do protocolo), b) um exemplo contendo uma discussão colaborativa de riscos realizada anteriormente no *RD System*, c) exemplos de esquemas de argumentação sobre riscos relacionados a mudanças de requisitos em projetos e esquemas de argumentação sobre riscos relacionados à incerteza de clientes sobre os requisitos de projetos e d) uma descrição de um projeto de software onde riscos de requisitos deveriam ser discutidos pelos participantes envolvidos no experimento.

A segunda etapa do experimento começou com a apresentação de um projeto de desenvolvimento de software onde os participantes deveriam realizar uma análise de riscos. O projeto proposto foi o desenvolvimento de um novo sistema financeiro para uma universidade. O principal objetivo deste projeto era solucionar os muitos problemas encontrados em um sistema financeiro atualmente sendo utilizado. Tais problemas estavam principalmente relacionados à correção de bugs em um sistema com conhecida baixa qualidade de implementação. Neste cenário, a maioria destes problemas estavam diretamente relacionadas às mudanças frequentes de requisitos e conseqüente manutenção do sistema atualmente sendo utilizado. Além disso, problemas estavam relacionados ao fato de que diferentes membros da universidade não tinham clareza e concordância sobre os recursos

necessários para este sistema financeiro. A equipe formada para desenvolver o novo sistema seria composta por um gerente de projetos, três/quatro desenvolvedores e um gerente sênior representando o corpo administrativo da universidade. Esta equipe poderia ser considerada como tendo alta habilidade em desenvolvimento de software, pois os desenvolvedores tinham experiência na linguagem de programação escolhida e no desenvolvimento de outros sistemas semelhantes para a universidade.

Após finalizar as etapas iniciais do experimento, os participantes foram divididos em quatro grupos (cada grupo contendo 4 participantes). Os integrantes destes grupos foram dispersos fisicamente em um laboratório, assim buscando criar um ambiente onde somente os recursos de colaboração disponíveis no *RD System* pudessem promover a comunicação entre eles. Em seguida, uma discussão dos riscos sobre requisitos do projeto proposto foi realizada, a qual durou aproximadamente 50 min. O processo de discussão foi parcialmente assistido, visto que perguntas sobre o uso de esquemas e do sistema *RD System* foram respondidas individualmente durante a realização da discussão. Ao final do processo de análise de riscos, os participantes responderam um questionário de avaliação contendo dez questões de múltipla escolha. As questões/afirmações deste questionário foram as seguintes:

- Q1. Qual seu nível de experiência em gerenciamento de riscos?
- Q2. É importante executar a etapa de gerenciamento de riscos em projetos de desenvolvimento de software.
- Q3. É importante discutir riscos de projetos de desenvolvimento de software de modo colaborativo.
- Q4. Esquemas de argumentação auxiliam o processo de discussão de riscos.
- Q5. Esquemas de argumentação facilitam o entendimento do risco proposto.
- Q6. Esquemas de argumentação facilitam a atividade de propor um risco.
- Q7. Esquemas de argumentação (questões críticas) auxiliam na continuidade, fluência da discussão.
- Q8. Esquemas de argumentação são fáceis de usar durante a discussão de riscos.
- Q9. Os recursos de discussão colaborativa de riscos implementados nos sistemas são fáceis de usar.
- Q10. O sistema contempla as principais etapas de gerenciamento de riscos (completude).

8.2.2. Resultados do experimento envolvendo alunos de graduação em computação da UFSM

Os resultados obtidos neste questionário são apresentados na Figura 17.

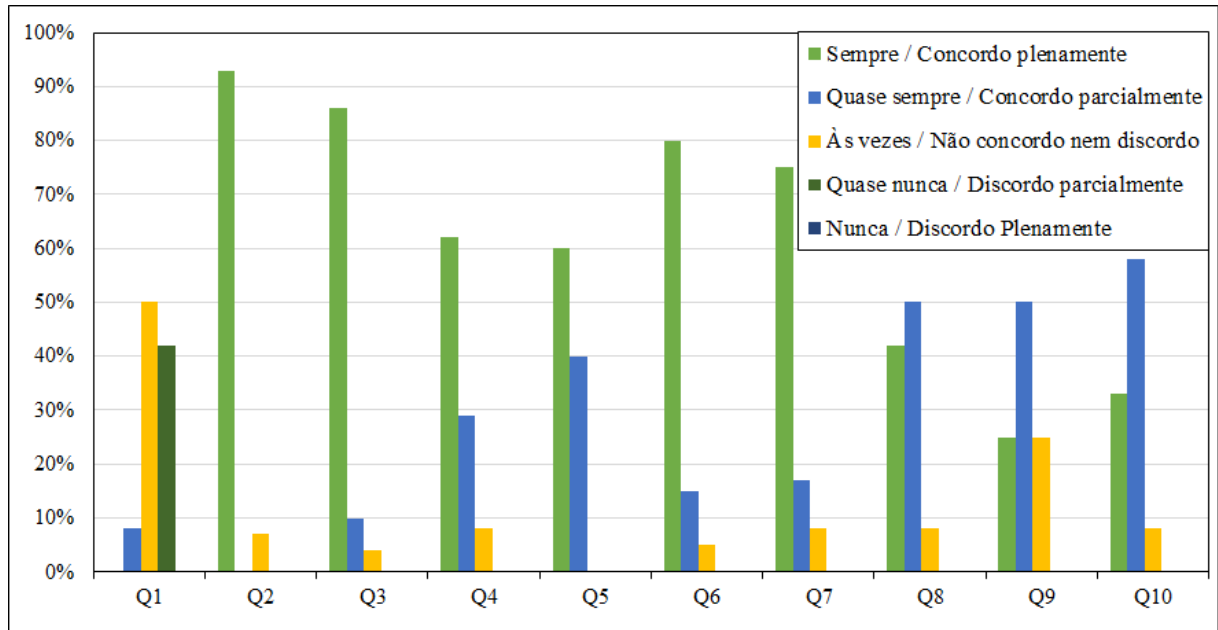


Figura 17 – Resultados do segundo experimento.

O objetivo das questões Q1, Q2 e Q3 é medir a experiência/conhecimento dos participantes envolvidos no experimento e avaliar a importância do gerenciamento de riscos no desenvolvimento de projetos. Considerando as respostas obtidas, os participantes não possuem experiência em gerenciamento de riscos (Q1). Mesmo diante desta falta de experiência, todos concordaram que é importante realizar atividades de gerenciamento de riscos em projetos de desenvolvimento de software (Q2), bem como explorar a colaboração no processo de gerenciamento de riscos (Q3).

O objetivo das questões Q4 à Q8 é avaliar se esquemas de argumentação podem auxiliar o desenvolvimento do processo de discussão de riscos e, em particular, a fluência e profundidade deste tipo de debate. Estas questões também buscaram avaliar se esquemas de argumentação podem facilitar a proposição de riscos, bem como a completude e clareza dos riscos propostos. A maioria (62%) dos participantes afirmou que esquemas de argumentação podem auxiliar o processo de discussão de riscos (Q4), como também podem facilitar

atividades de proposição de riscos (Q5) e facilitar o entendimento de riscos propostos (Q6). Em especial, os participantes concordaram que as questões críticas dos esquemas podem promover a continuidade de uma discussão (Q7). Contudo, a maioria dos participantes não concordou plenamente que esquemas de argumentação são fáceis de usar durante uma discussão colaborativa de riscos (Q8).

O objetivo das questões Q9 e Q10 é avaliar o quão usável é o *RD System v2.0* e verificar se esse sistema contempla as principais etapas do gerenciamento de riscos. Os resultados apresentados permitem observar que a maioria dos participantes concorda parcialmente com a afirmação que os recursos de discussão colaborativa de riscos implementados no *RD System v2.0* são fáceis de usar (Q9). A maioria dos participantes também relatou que o *RD System v2.0* ainda não contempla todas as necessidades de gerenciamento de riscos esperadas por eles (Q10).

8.2.3. Discussão do experimento envolvendo alunos de graduação em computação da UFSM

Os resultados obtidos demonstram a importância do gerenciamento de riscos em um projeto de software, visto que a maioria dos participantes concorda que a etapa de gerenciamento de riscos deve ser executada em um projeto, como também deve ser de modo colaborativo. De forma geral, o uso de esquemas de argumentação tende a trazer uma melhoria para uma discussão colaborativa de riscos. Neste caso, esquemas de argumentação são representados por uma estrutura completa, onde trazem informações que tendem a contextualizar e apoiar uma determinada conclusão, bem como eles conseguem auxiliar o processo de discussão de riscos.

Experimentos já realizados no passado com jogos de diálogo (SEVERO; FONTOURA; SILVA, 2013) demonstram evidência para a relevância da utilização desta técnica em uma discussão de riscos, visto que o mesmo organiza, e padroniza a sequência de análise que deve ser seguida por participantes de uma discussão de gerenciamento de riscos. Baseado nas discussões assistidas, é possível perceber que esquemas de argumentação e jogos de diálogos se complementam de maneira simples e usável por usuários neste domínio de aplicação

Esquemas de argumentação também trazem uma maior profundidade e detalhe de informações para uma discussão colaborativa de riscos. Esta maior profundidade é possível de

ser medida através de um comparativo da quantidade de locuções que são usadas em uma discussão colaborativa de gerenciamento de riscos. Esta comparação é realizada entre discussões onde os participantes poderiam utilizar esquemas de argumentação e discussões onde estes recursos não estariam disponíveis para os participantes. Podemos observar este fato da seguinte maneira: a) em uma discussão em que os participantes utilizam esquemas de argumentação, as locuções *Ask*, *Inform*, *Argument pro* e *Argument con* foram utilizadas 11%, 4%, 24% e 20%, respectivamente, no total de locuções empregadas no diálogo. Estas locuções são utilizadas na maior parte das vezes em conjunto com uma questão crítica de forma explícita através da locução *ask*, ou implicitamente pelo emprego das demais locuções do protocolo. Este fato faz com que os participantes apresentem questionamentos relevantes para uma discussão, fazendo com que os demais participantes tenham intuito de responder estas questões, assim trazendo informações adicionais que não estavam claras durante a discussão. b) em uma discussão que os participantes não utilizam os esquemas de argumentação, o mesmo conjunto de locuções é utilizado 4%, 5%, 2% e 1%, respectivamente. Estes números mostram que quando não se tem o auxílio dos esquemas de argumentação e de suas questões críticas, os participantes tendem a aceitar as proposições feitas, sejam elas de riscos, impactos, probabilidades ou planos, embora possivelmente estes participantes ainda tenham dúvidas sobre determinadas informações discutidas. Podemos concluir, portanto, que uma discussão colaborativa que use uma biblioteca de esquemas de argumentação tende a ter uma maior profundidade e detalhamento, como também, pode ter como resultado uma *checklist* dos riscos selecionados e priorizados mais próximo as necessidades de gerenciamento de riscos em diferentes projetos.

9. CONCLUSÃO E TRABALHOS FUTUROS

Para finalizar esta dissertação, este capítulo apresenta algumas considerações finais e sugestões para novas pesquisas nas áreas relacionadas a este trabalho. Por meio destas considerações finais e sugestões, o leitor pode obter uma visão panorâmica dos assuntos explorados na dissertação.

9.1. Considerações finais

O gerenciamento de riscos é uma das etapas mais importantes do gerenciamento de projetos. Mesmo diante deste fato, poucas empresas executam esta etapa, por diversos motivos. Neste caso, um dos principais motivos é a sua complexidade e a falta de técnicas que facilitem o processo de gerenciamento de riscos. Esta dissertação apresenta uma nova abordagem para apoiar o processo de gerenciamento colaborativo de riscos, a fim de torná-lo mais simples de ser executado de forma sistemática pelos diversos participantes de um projeto de software, desde gerentes de projetos até usuários finais.

Apesar de alguns trabalhos, como (LI; SLYNGSTAD, 2008; NEVES et al., 2013), terem sido propostos no intuito de facilitar processos de gerenciamento de riscos, a pesquisa no nosso projeto permitiu perceber que ainda existia a necessidade de abordar tópicos importantes como o gerenciamento colaborativo de riscos e a captura de dados e conhecimento de processos de discussão de riscos. Passos iniciais para atender tais necessidades foram dados em um projeto passado do nosso grupo de pesquisa, assim como proposto por Severo (SEVERO; FONTOURA; SILVA, 2013). Visando expandir o emprego de técnicas de diálogo empregadas no processo colaborativo de discussão de riscos proposto por Severo, esta dissertação discute a criação e utilização de esquemas de argumentação para o gerenciamento de riscos. Assim como descrito nesta dissertação, estes esquemas podem ser construídos a partir de uma série de passos, os quais visam apoiar o processo de construção de esquemas de argumentação voltados para gerenciamento de riscos. Em particular, os esquemas discutidos neste trabalho visam auxiliar o processo de proposição de riscos em meio a discussões de riscos de projetos de software, apoiando os diferentes participantes de uma discussão a expressar de maneira mais completa e compreensível os riscos de um projeto.

Um sistema de discussões de riscos (*RD system v2.0*) também é discutido neste trabalho. Este sistema oferece aos participantes dos projetos um ambiente onde estes podem amplamente colaborar no desenvolvimento das tarefas típicas de gerenciamento de riscos, como também registrar os diferentes argumentos apresentados pelos participantes envolvidos nos debates realizados. Usuários deste sistema também podem manter uma biblioteca de esquemas de argumentação, assim buscando ampliar a utilização destes recursos de argumentação no ambiente de discussão de riscos.

Dentre as contribuições apresentadas neste trabalho, pode-se ressaltar a proposição de uma forma estruturada de como construir uma biblioteca de esquemas de argumentação para proposição de riscos em tarefas colaborativas de gerenciamento de riscos. Neste caso, a dissertação propõe um processo simples para criação de esquemas de argumentação, como também a utilização de tipos de argumentos na indexação/categorização destes esquemas, os quais são particularmente voltados para a área de gerenciamento de riscos em projetos de software. A dissertação também descreve formas de como usuários podem utilizar estes esquemas de argumentação em meio à discussão de riscos, assim permitindo reutilizar o conhecimento capturado em diferentes argumentos apresentados nestas discussões.

Assim como descrito nesta dissertação, a utilização de esquemas de argumentação juntamente com jogos de diálogos (SEVERO et al., 2013) tem como objetivo auxiliar o processo de priorização dos riscos de um projeto. Isso se deve ao fato de que esquemas são compostos por um conjunto de questões críticas, onde estas questões auxiliam os usuários a raciocinarem e testarem a veracidade e/ou real relevância de riscos propostos em um projeto. Além disso, a técnica de jogos de diálogos permite estruturar o processo de discussão apontando o que pode ser feito, desta forma organizando os passos de gerencia de riscos que devem ser realizados, permitindo uma maior colaboração entre os participantes da discussão e resultando na construção de uma base de conhecimento reusável. Por sua vez, esquemas de argumentação auxiliam a formalizar e, portanto, tornar os argumentos apresentados mais compreensíveis, permitindo também que usuários com pouca ou nenhuma experiência em gerência de riscos possam expor suas opiniões e propor riscos que julguem relevantes para os seus projetos.

As contribuições deste trabalho foram testadas usando experimentos, envolvendo uma equipe de desenvolvimento de software de uma empresa e acadêmicos dos cursos de sistemas de informação e ciência da computação da UFSM. Tais experimentos apresentam evidência positiva para a relevância do uso de esquemas de argumentação para proposição de riscos no ambiente do *RD System v2.0*. Este trabalho também foi validado pela apresentação de partes

dele em dois eventos científicos das áreas de pesquisa envolvidas nesta dissertação. Uma versão preliminar deste trabalho foi apresentada no Simpósio Brasileiro de Sistemas de Informação (SBSI) na forma de um artigo, denominado *Argumentation Based Risk Management* (SEVERO et al., 2013). Este artigo propõe a abordagem de utilização de jogos de diálogo em conjunto com a utilização de esquemas de argumentação. A abordagem proposta nesta dissertação também foi aceita para publicação no evento IEEE 15th *International Conference on Information Reuse & Integration (IRI)*, na forma de um artigo, denominado *Argumentation Schemes for the Reuse of Argumentation Information in Collaborative Risk Management* (POZZEBON; SILVA; FONTOURA; CAMPBELL, 2014). Este artigo apresentou esquemas de argumentação para gerenciamento de riscos e modos de uso destes em conjunto com jogos de diálogos, bem como descreveu como estas técnicas foram implementadas no *RD System v2.0*.

Apesar da abordagem proposta nesta dissertação aprimorar e auxiliar o processo de gerenciamento de risco proposto e implementado no *RD System v2.0*, alguns trabalhos futuros ainda podem ser notados. Por exemplo, usuários podem usar o processo de construção de esquemas proposto nesta dissertação para expandir a biblioteca de esquemas já existente no *RD System*. Além disso, ainda existe a necessidade de investigar a extensão do uso de esquemas de argumentação para as demais etapas da gerencia de riscos, visto que esta dissertação focalizou a etapa de proposição de riscos.

9.2. Sugestões para pesquisas futuras

Esta dissertação não pretende finalizar as discussões sobre os tópicos de pesquisa sendo explorados no grupo de pesquisa onde esta dissertação esta inserida. Os resultados de pesquisas atuais evidenciam a necessidade de trabalhos futuros dentro em diferentes frentes.

Como perspectivas futuras, então, sugere-se o desenvolvimento de experimentos em empresas visando refinar o modo de desenvolvimento de esquemas de argumentação proposto nessa dissertação. Pode-se colocar como sugestão, também, o aprimoramento da integração de diferentes técnicas de IA para auxiliar o processo de gerenciamento de riscos, como também desenvolver um estudo e uma valiação aprofundada referente a usabilidade do *RD System v2.0*.

REFERÊNCIAS

ALBERTS, Christopher J.; DOROFEE, Audrey J. **Risk Management Framework**. Carnegie Mellon University Software Engineering Institute. August, 2010.

BENCH-CAPON, T.; DUNNE, D. Argumentation and dialogue in artificial intelligence. **IJCAI 2005 tutorial notes**, 2005.

BENCH-CAPON, Trevor JM.; DUNNE, Paul E. Argumentation in artificial intelligence. **Artificial intelligence**, v. 171, n. 10, p. 619–641, jul. 2007.

BEX, F. J.; REED, C. A. Schemes of Inference, Conflict, and Preference in a Computational Model of Argument. **Studies in Logic, Grammar and Rhetoric**, v. 23, n. 36, p. 39–58, 2011.

BEX, Floris; VERHEIJ, Bart. Solving a Murder Case by Asking Critical Questions: An Approach to Fact-Finding in Terms of Argumentation and Story Schemes. **Argumentation**, v. 26, n. 3, p. 325–353, 9 nov. 2011.

BOEHM, Barry W. Software risk management: principles and practices. **Software, IEEE**, v. 8, n. 1, p. 32-41, January, 1991.

BOEHM, Barry W.; BHUTA, Jesal. Balancing Opportunities and Risks in Component-Based Software Development. **Software, IEEE**, v. 25, n. 6, p. 56–63, 1 nov. 2008.

BOEHM, Barry W. A Risk-Driven Decision Table for Software Process Selection. In: MÜNCH, J.; YANG, Y.; SCHÄFER, W. (Eds.). **New Modeling Concepts for Today's Software Processes SE - 1**. Lecture Notes in Computer Science. [s.l.] Springer Berlin Heidelberg, 2010. v. 6195p. 1.

BOEHM, Barry W.; DEMARCO, Tom. Software risk management. **Software, IEEE**, v. 14, n. 3, p. 17–19, 1997.

CARR, Marvin J.; KONDA, S. L.; MONARCH, I.; ULRICH, F. C.; WALKER, C.F. **Taxonomy-based risk identification**. Carnegie Mellon University Software Engineering Institute, June, 1993.

VAN EEMEREN, Frans; GROOTENDORST, Rob. **A systematic theory of argumentation: The pragma-dialectical approach**. Cambridge University Press, 2004. v. 14

EL EMAM, Khaled; KORU, A. Gunes. A replicated survey of IT software project failures. **Software, IEEE**, v. 25, n. 5, p. 84–90, 2008.

GRASSO, Floriana; RAHWAN, I.; REED, C.; SIMARI, G. R. Introducing Argument & Computation. **Argument & Computation**, v. 1, n. 1, p. 1–5, mar. 2010.

GREER, D.; BUSTARD, D. Collaborative risk management. **IEEE International Conference on Systems Man and Cybernetics**, p. 5, v. 5, 2002.

- HADJINIKOLIS, Christos; MODGIL, S.; BLACK, E.; MCBURNEY, P.; LUCK, M. **Investigating Strategic Considerations in Persuasion Dialogue Games. STAIRS**, p. 137-148, 2012.
- HALL, Elaine, M. **Managing risk: Methods for software systems development**. Pearson Education, 1998.
- KARUNATILLAKE, Nishan, C.; JENNINGS, N. Dialogue games that agents play within a society. **Artificial intelligence**, v. 173, n. 9, p. 935–981, January, 2009.
- KAYIS, B.; ZHOU, M.; SAVCI, S. IRMAS—development of a risk management tool for collaborative multi-site, multi-partner new product development projects. **Journal of Manufacturing Technology Management**, v. 18, n. 4, p. 387–414, 2007.
- KWAN, T. W.; LEUNG, H. K. N. A Risk Management Methodology for Project Risk Dependencies. **IEEE Transactions on Software Engineering**, v. 37, n. 5, p. 635–648, set. 2011.
- LI, Jingyue; SLYNGSTAD, O. A state-of-the-practice survey of risk management in development with off-the-shelf software components. **IEEE Transactions on Software Engineering**, v. 34, n. 2, p. 271–286, 2008.
- LINDGREN, Helena. Towards using argumentation schemes and critical questions for supporting diagnostic reasoning in the dementia domain. **Proc. Computational Models of Natural Arguments (CMNA'09), Pasadena, CA**, p.10-14, 2009.
- MACHADO, Nielsen L. R.; SILVA, Luis A. L.; FONTOURA, Lisandra M.; CAMPBELL, John. Case-based Reasoning for Experience-based Collaborative Risk Management. **International Conference on Software Engineering and Knowledge Engineering**, p. 262-267, 2014.
- MANTARAS, Ramon. L. DE. Retrieval, reuse, revision and retention in case-based reasoning. **The Knowledge Engineering Review**, v. 20, n. 03, p. 215–240, 2005.
- MCBURNEY, Peter; HITCHCOCK, David; PARSONS, Simon. The Eightfold Way of Deliberation Dialogue: Research Articles. **International Journal of Intelligent Systems**, v. 22, n. 1, p. 95–132, 2007.
- MCBURNEY, Peter; PARSONS, Simon. Dialogue game protocols. **Communication in Multiagent Systems**. Springer Berlin Heidelberg, p. 269-283, 2003.
- MCLEOD, Kenneth; FERGUSON, Gus; BURGER, Albert. Using argumentation to resolve conflict in biological databases. **Proceedings of the IJCAI-09 workshop on Computational Models of Natural Argument (CMNA)**, v. 9, p. 15-23, 2009.
- MILER, Jakub; GÓRSKI, Janusz. Software support for collaborative risk management. **Advanced Computer Systems**. Springer US, p. 325–334, 2002.
- NEVES, Sandra. M.; Silva, C. E. S.; SALOMON, V. A. P.; SANTOS, A. L. A. Knowledge-Based Risk Management: Survey on Brazilian Software Development Enterprises. In:

Advances in Information Systems and Technologies. Springer Berlin Heidelberg, p. 55-65, 2013.

PAPADAKI, Ekaterini; POLEMI, Despina; DAMILOS, Dimitrios Kon. A Holistic, Collaborative, Knowledge-Sharing Approach for Information Security Risk Management. **The Third International Conference on Internet Monitoring and Protection, IEEE**, p. 125–130, 2008.

PARSONS, Simon; SIERRA, Carles; JENNINGS, Nick. Agents that reason and negotiate by arguing. **Journal of Logic and computation**, v. 8, n. 3, p. 261-292, 1998.

PARSONS, Simon. et al. Argument Schemes for Reasoning about Trust. **Proceedings of the 4th International Conference on Computational Models of Argument**, v. 245, p. 430, 2012.

PERSSON, John; MATHIASSEN, Lars. A process for managing risks in distributed teams. **Software, IEEE**, 2010.

PMI. PMSURVEY.ORG 2012 Edition. **Project Management Institute Chapters**, 2012.

PMI. A Guide to the Project Management Body of Knowledge (PMBOK Guide). **Project Management Institute**, v. Fifth edit, 2013.

PRESSMAN, Roger S. **Engenharia de Software: Uma Abordagem Profissional**. 7ª edição Ed. McGraw-Hill, 2011.

RAHWAN, Iyad. Mass argumentation and the semantic web. **Journal of Web Semantics: Science, Services and Agents on the World Wide Web.**, v. 6, n. 1., p. 29–37, December, 2008.

REED, Chris; WALTON, Douglas. Argumentation schemes in dialogue. **Dissensus and the Search for Common Ground (Proceedings of OSSA)**, p. 1–11, 2007.

REED, Chris; WALTON, Douglas; MACAGNO, Fabrizio. Argument diagramming in logic, law and artificial intelligence. **The Knowledge Engineering Review**, v. 22, n. 01, p. 87-109, 25 maio 2007.

REED, Chris; WELLS, Simon. Dialogical argument as an interface to complex debates. **Intelligent Systems, IEEE**, v. 22, n. 6, p. 60–65, 2007.

ROCHA, Ana R. C.; WEBER, Kival C.; NASCIMENTO, C. J. **Qualidade de Produtividade emSoftware**. 4ª edição renovada. São Paulo, Makron Books, 2001.

SEVERO, F. S. ; POZZEBON, Ruan C. B. ; FONTOURA, Lisandra Manzoni ; SILVA, Luís A. L. . Argumentation-Based Risk Management. **IX Simpósio Brasileiro de Sistemas de Informação (SBSI 2013)**, João Pessoa, Brasil, v. 01. p. 851-862, 2013.

SEVERO, F. S.; FONTOURA, L. M.; SILVA, L. A. L. A Dialogue Game Approach to Collaborative Risk Management. **The 25th International Conference on Software Engineering and Knowledge Engineering**, 2013.

SILVA, Luís. A. DE L. **Enhancement of Case-Based Reasoning through Informal Argumentation , Reasoning Templates and Numerical Taxonomy** Tese de Doutorado. UCL (University College London), 2010.

SILVA, Luís. A. L.; CAMPBELL, J. A.; EASTAUGH, N.; BUXTON, B. F. A Case for Folk Arguments in Case-Based Reasoning. **Case-Based Reasoning. Research and Development SE - 24**. Springer Berlin Heidelberg, p. 317–331, 2010.

SOMMERVILLE, I. **Engenharia de Software**. 9ª edição ed. PEARSON BRASIL, 2011.

SOUZA, Yóris L. et. al. A contribuição do compartilhamento do conhecimento para o gerenciamento de riscos em projetos: um estudo na indústria de software. **Revista da Gestão de Tecnologia da Informação**, v. 7, n. 1, p. 183-204, 2010.

STERN, R.; ARIAS, J. C. Review of Risk Management Methods. **Bussiness Intelligence Journal**, v. 4, p. 59–78, 2011.

SUN, Si. Study on Software Project Risk Priority Management and Framework Based on Information Management System. **Information Science and Engineering (ICISE), 2009 1st International Conference on. IEEE**, p. 2402–2405, 30 dez. 2009.

TOLCHINSKY, Pancho; CORTES, U. Increasing human-organ transplant availability: Argumentation-based agent deliberation. **Intelligent Systems, IEEE**, v. 21, n. 6, p. 30-37, 2006.

TOLCHINSKY, P.; MODGIL, S.; ATKINSON, K. Deliberation dialogues for reasoning about safety critical actions. **Autonomous Agents and Multi-Agent Systems**, v. 25, n. 2, p. 209–259, 11 maio 2012.

TOLCHINSKY, P.; MODGIL, S.; CORTÉS, U. Argument Schemes and Critical Questions for Heterogeneous Agents to Argue Over the Viability of a Human Organ for Transplantation. **AAAI Spring Symposium: Argumentation form Consumers of Healthcare**. p. 105-105, 2006.

TONIOLO, Alice; NORMAN, Timothy J.; SYCARA, Katia. On the benefits of argumentation schemes in deliberative dialogue. **Proceedings of the Eleventh International Conference on Autonomous Agents and Multiagent Systems**, v. 3, p. 1409-1410, 2012.

TOULMIN, Stephen E. **The Uses of Argument**. Cambridge University Press, 2003.

WALTON, D. **Fundamentals of critical argumentation**. Cambridge University Press, 2005.

WALTON, D. Argument Mining by Applying Argumentation Schemes. **Studies in Logic**, v. 4, p. 1–24, 2012.

WALTON, D.; REED, C.; MACAGNO, F. **Argumentation schemes**. New York: Cambridge University Press, 2008.

YUAN, Tangming; KELLY, Tim. Argument schemes in computer system safety engineering. **Informal Logic**, v. 31, n. 2, p. 89–109, 2011.

YUAN, Tangming; KELLY, Tim. Argument-based approach to computer system safety engineering. **International Journal of Critical Computer-Based Systems**, v. 3, p. 151–167, 2012.