

Resumo

Com o advento da globalização, empresas de desenvolvimento de software vem sofrendo concorrência de várias empresas em todo o âmbito mundial. Com isso, o mercado se torna cada vez mais acirrado e competitivo, “obrigando” as empresas a terem uma preocupação cada vez maior com as restrições de tempo, custos e qualidade do produto produzido. As organizações que desejam se destacar e obter sucesso, precisam saber gerenciar essas restrições para que o produto final seja entregue ao cliente dentro do prazo estipulado, dos custos estimados e satisfazendo plenamente os requisitos definidos. Diante desse contexto, percebe-se a grande importância da Gerência de Projetos nas empresas de desenvolvimento. Uma das atribuições do Gerente de Projetos é a atividade de gerenciar riscos. Os riscos são eventos incertos que podem ocorrer em um projeto e gerar alguns impactos ao desenvolvimento do mesmo, como atrasos, estouros de orçamento e até mesmo o cancelamento do projeto. Para gerenciar riscos é preciso, primeiramente, identificá-los. Existem diversas técnicas de identificação de riscos, e praticamente a totalidade delas utiliza-se de experiências passadas. Foi desenvolvido, no primeiro semestre de 2007, um método para identificação automática de riscos utilizando informações de projetos passados, o *CBR Risk Method*. Ainda em 2007, este método foi avaliado através de experimentos e foram sugeridas algumas melhorias ao modelo original. A implementação dessas melhorias é uma das contribuições do presente trabalho. O *CBR Risk* busca em uma base de dados projetos passados que sejam similares ao novo projeto, e essa similaridade é calculada através de alguns atributos de projetos. Dois desses atributos, “Tipo de Projeto” e “Plataforma Tecnológica”, possuem características diferenciadas em relação aos demais e, por isso, serão avaliados neste trabalho através de estudos experimentais e análise dos resultados.

Abstract

With the advent of globalisation, software development companies are suffering a strong competition from others around the world. This makes the market become increasingly competitive and tough, "forcing" companies to concern even more about the restrictions of time, cost and the quality of the final product. Organizations that wish to succeed, must learn to manage those restrictions in order to deliver the final product to the customer within the stipulated time, with the estimated costs and fully satisfying the requirements defined. In this context, it is realized how important the Project Management is in software development companies. One of the tasks of the Project Manager is the activity of managing risks. The risks are uncertain events that might occur on a project and generate some impacts on development, such as delays, bursts of budget and even the project cancellation. To manage risks, first, we must identify them. There are many techniques for identifying risks, and practically all of them are based on previous experience. In the first half of 2007, a method for automatic identification of risks using information from past projects was developed, the *CBR Risk Method*. Also in 2007, this method was evaluated through experiments and some improvements were suggested to the original model. The implementation of these improvements is one of the contributions of this work. The *CBR Risk* searches in a database past projects that are similar to the new project, and this similarity is calculated through some attributes of projects. Two of these attributes, "Project Type" and "Technology Platform", have different characteristics in relation to the other and, hence, they will be evaluated in this work through experimental studies and analysis of results.

Sumário

Índice de Figuras	v
Índice de Tabelas	vi
Índice de Equações	vii
Tabela de Símbolos e Siglas	viii
1 Introdução	10
1.1 Motivação	11
1.2 Objetivos	12
1.3 Metodologia	12
1.4 Estrutura do Documento	13
2 Gerenciamento de Riscos do Projeto	15
2.1 Gerência de Riscos – Visão Geral	15
2.2 Identificação de Riscos	17
2.3 Resumo do Capítulo	19
3 CBR Risk: Método para Identificação Automática de Riscos em Projetos de Software	20
3.1 Raciocínio Baseado em Casos	20
3.1.1 Elementos Básicos	21
3.2 Modelo	23
3.2.1 Representação de Casos	24
3.2.2 Cálculo de Similaridade	26
3.2.3 Recuperação de Casos	27
3.3 Protótipo	27
3.4 Resumo do Capítulo	29
4 Experimentação	31
4.1 Introdução	31
4.2 Coleta de Dados	32
4.3 Definição dos Experimentos	33
4.3.1 Experimento 1: Atributos “Tipo de Projeto” e “Plataforma Tecnológica” assumindo valores 0 e 4	37
4.3.2 Experimento 2: Atributos “Tipo de Projeto” e “Plataforma Tecnológica” assumindo valores 0 e 9	38
4.3.3 Experimento 3: Atributos “Tipo de Projeto” e “Plataforma Tecnológica” assumindo valores 0 e 16	39
4.3.4 Experimento 4: Atributos “Tipo de Projeto” e “Plataforma Tecnológica” com maior relevância	39
4.3.5 Experimento 5: Sem os atributos “Tipo de Projeto” e “Plataforma Tecnológica”	40
4.4 Critério de Avaliação do Desempenho	40
4.5 Resumo do Capítulo	40
5 Análise dos Resultados	41

	iv	
5.1	Introdução	41
5.2	Estrutura de Análise	42
5.2.1	Médias dos resultados %B	42
5.2.2	Desvio Padrão dos resultados %B	42
5.2.3	Riscos efetivamente identificados	43
5.3	Estudo Comparativo	44
5.4	Resumo do Capítulo	47
6	Conclusões e Trabalhos Futuros	48
6.1	Contribuições	48
6.2	Trabalhos Futuros	49

Índice de Figuras

Figura 1.	Visão geral do Gerenciamento de Riscos do Projeto [1]	17
Figura 2.	O ciclo do RBC [11]	23
Figura 3.	Tela Inicial do <i>CBR Risk Method</i>	28
Figura 4.	Médias dos Resultados %B	42
Figura 5.	Desvio Padrão dos Resultados %B	43
Figura 6.	Média percentual dos riscos efetivamente identificados	44
Figura 7.	Percentual de Riscos Identificados Vs. Percentual de Riscos que poderiam ter sido identificados (Modelo original)	46
Figura 8.	Percentual de Riscos Identificados Vs. Percentual de Riscos que poderiam ter sido identificados (Modelo sem atributos)	46

Índice de Tabelas

Tabela 1.	Possíveis valores numéricos de cada atributo no cálculo de similaridade	35
Tabela 2.	Resumo dos resultados para cada variação	45

Índice de Equações

Equação 1. Expressão para o cálculo da similaridade global do CBR Risk Method	27
Equação 2. Expressão para o cálculo da similaridade global do CBR Risk Method (dividido em duas fases)	35

Tabela de Símbolos e Siglas

%B – $100 * (\text{qtd. projetos retornados corretamente}) / (\text{qtd. projetos retornados após a filtragem})$

CBR – Case Based Reasoning

c_i - Valor do atributo para o projeto da base de dados

PMBOK - Project Management Body of Knowledge

PMI - Project Management Institute

q_i - Valor do atributo para o projeto atual

RBC – Raciocínio Baseado em Casos

Sg – Similaridade Global

S_1 – Valor numérico do atributo “Tamanho da Equipe”

S_2 – Valor numérico do atributo “Distribuição Geográfica”

$S_{3.1}$ – Valor numérico do atributo “Experiência no Processo”

$S_{3.2}$ – Valor numérico do atributo “Experiência no Domínio da Aplicação”

$S_{3.3}$ – Valor numérico do atributo “Experiência na Tecnologia”

S_4 – Valor numérico do atributo “Tamanho do Projeto”

S_5 – Valor numérico do atributo “Tipo de Projeto”

S_6 – Valor numérico do atributo “Plataforma Tecnológica”

w_n - peso de cada atributo

Agradecimentos

Gostaria de agradecer primeiramente a DEUS, é nele que eu sempre busco ajuda nos momentos difíceis por qual passo. Quero agradecer a todos os professores os quais me deram a oportunidade de aprender e ter a formação de engenheiro, em especial a minha orientadora, Cristine Gusmão.

Merecem grande destaque nos meus agradecimentos Thiago Trigo e Arthur Lins, que tiveram participação importante para a conclusão deste trabalho.

Um agradecimento todo especial aos grandes amigos que fiz durante o curso, os quais nem preciso citar nomes, eles já sabem quem são. Se não fosse eles eu não estaria aqui agora.

Gostaria de agradecer também a minha família, minha base de sustentação, sem os quais eu não teria conseguido chegar tão longe.

Obrigado a todos e um grande abraço.

Capítulo 1

Introdução

Nos dias atuais, empresas de desenvolvimento de software lidam cada vez mais com uma grande quantidade de informações e mudanças tecnológicas, que precisam ser bem geridas se a empresa pretende se destacar e obter sucesso no concorrido ramo de desenvolvimento de soluções em software. A concorrência de organizações de todo o mundo faz com que as empresas de desenvolvimento trabalhem com restrições maiores de tempo e custos, além de um maior rigor por parte dos clientes em relação à qualidade dos produtos gerados.

Por esse motivo, a Gerência de Projetos feita com consistência e eficiência vem se tornando um pré-requisito e um grande diferencial para uma empresa conseguir sucesso e visibilidade com seus projetos. O gerenciamento de projetos pode ser definido como a aplicação de conhecimento, habilidades, ferramentas e técnicas às atividades do projeto a fim de atender aos seus requisitos [1].

A atividade de gerenciar projetos é uma atividade bastante complexa e que exige muita experiência e conhecimentos do gerente de projetos. Para auxiliar as atividades da Gerência de Projetos existem diversos guias de boas práticas e um dos mais utilizados é o Guia PMBOK (*Project Management Body of Knowledge*), elaborado pelos membros do PMI (*Project Management Institute*) [1].

O Guia PMBOK é dividido em cinco grupos de processos: iniciação, planejamento, execução, controle e encerramento. Tais processos interagem entre si com o objetivo de que o projeto seja encerrado com qualidade e cumprindo os prazos e custos pré-definidos. A Gerência de Projetos é dividida nas seguintes áreas de conhecimento: Gerência de Integração, Gerência do Escopo, Gerência do Tempo, Gerência de Custos, Gerência da Qualidade, Gerência de Recursos Humanos, Gerência de Comunicação, Gerência de Riscos e Gerência de Aquisições [1].

O problema focalizado neste trabalho está inserido na área de Gerência de Projetos, mais especificamente Gerência de Riscos. Segundo o Guia PMBOK, o gerenciamento de riscos pode ser dividido em processos, que incluem: Planejamento do Gerenciamento de Riscos, Identificação de Riscos, Análise Qualitativa dos Riscos, Análise Quantitativa dos Riscos, Planejamento de Respostas a Riscos e Monitoramento e Controle de Riscos [1]. Dentre os processos da gestão de riscos apresentados, será abordado neste trabalho o processo de Identificação de Riscos.

O processo de Identificação de Riscos é um dos mais complexos, já que os riscos são eventos incertos. Existem diversas ferramentas e técnicas que auxiliam os gerentes a identificar os riscos que cercam um projeto. Os riscos precisam ser identificados para a realização de ações preventivas e definição de planos de contingência [2].

Nesse contexto de Identificação de Riscos foi desenvolvido por Arthur Lins, em seu Trabalho de Conclusão de Curso [3], o *CBR Risk Method*, que é um método para identificação de riscos em projetos que utiliza Raciocínio Baseado em Casos (do inglês *Case Based Reasoning* - CBR) - técnica que busca uma solução para um problema novo baseando-se em experiências passadas.

Com o modelo do *CBR Risk* definido foi possível criar uma ferramenta para identificação de riscos, mas, para a avaliação da mesma, são necessários experimentos. Tais experimentos visam aperfeiçoar cada vez mais a ferramenta, tornando-a mais confiável e útil para uso em ambientes de desenvolvimento [2].

1.1 Motivação

Como já observado, a atividade de Identificação de Riscos é crucial para o bom desenvolvimento do projeto. A ferramenta implementada por Arthur Lins é uma excelente proposta de suporte para tal atividade, mas são necessários experimentos sobre a mesma para avaliá-la e torná-la mais eficiente. Thiago Trigo, em seu Trabalho de Conclusão de Curso [2], realizou alguns desses experimentos e propôs pontos de melhoria, como um novo algoritmo para cálculo de similaridade entre projetos e a aplicação de um filtro. Mas novos experimentos são necessários, principalmente para a avaliação de dois atributos de projetos de caráter único que são utilizados pela ferramenta: “Tipo de Projeto” e “Plataforma Tecnológica”.

Os objetivos principais da execução de experimentos são a caracterização, avaliação, previsão, controle e melhoria a respeito de produtos, processos, recursos, modelos, teorias, entre outros [4].

1.2 Objetivos

Este trabalho visa aperfeiçoar o *CBR Risk*, desenvolvendo melhorias de acordo com o que foi constatado nos experimentos realizados por Thiago Trigo sobre a ferramenta [2]. As melhorias a serem implementadas são:

- Implementação de um novo algoritmo para o cálculo de similaridade entre projetos utilizando a Distância Euclidiana;
- Implementação de um filtro para evitar que muita informação inútil seja retornada na busca na base de projetos.

Além das melhorias citadas acima, foram realizados testes e avaliações na própria ferramenta para comprovação da eficiência do novo cálculo de similaridade baseado na Distância Euclidiana.

Outro objetivo a ser alcançado nesse trabalho é o estudo experimental de dois atributos utilizados para o cálculo de similaridade entre projetos no *CBR Risk*: “Tipo de Projeto” e “Plataforma Tecnológica”. Eles precisam ser estudados no sentido de sua influência no cálculo de similaridade e caracterização dos mesmos, já que diferentemente dos outros atributos, esses dois não possuem valores pré-definidos [3]. Após os experimentos com os atributos foi feita uma análise dos resultados obtidos.

1.3 Metodologia

A estratégia de ação utilizada no desenvolvimento deste trabalho foi dividida nas seguintes atividades:

1. Estudo da literatura:

– Estudo do Guia PMBOK na área de Gerência de Riscos, com foco na Identificação de Riscos [1];

– Estudo dos Trabalhos de Conclusão de Curso de Thiago Trigo (Avaliando um Modelo de Identificação de Projetos de Software Similares: Estudo de Caso) [2] e de Arthur Lins (Um Modelo para Identificação de Riscos de Projeto Utilizando Raciocínio Baseado em Casos) [3];

– Estudo da literatura na área de Raciocínio Baseado em Casos;

2. Estudo do Modelo *CBR Risk Method* [3], do seu código fonte e dos diversos experimentos realizados sobre o mesmo;

3. Implementação do novo cálculo de similaridade entre projetos utilizando Distância Euclidiana e do filtro no *CBR Risk*;

4. Atualização do site que hospeda o *CBR Risk* no endereço pma.dsc.upe.br/cbrrisk com o novo código com a Distância Euclidiana e o filtro implementados;

5. Realização de testes e avaliações para validação do cálculo de similaridade utilizando Distância Euclidiana através do protótipo do *CBR Risk*. Os dados utilizados nos testes são os projetos coletados por Thiago Trigo através de questionários em seu Trabalho de Conclusão de Curso [2];

6. Estudo dos atributos do *CBR Risk* “Tipo de projeto” e “Plataforma Tecnológica” no sentido de sua influência no cálculo de similaridade e caracterização dos mesmos;

7. Realização de experimentos, através de planilhas eletrônicas, com os atributos “Tipo de Projeto” e “Plataforma Tecnológica”, variando os valores dos mesmos;

8. Análise dos resultados obtidos dos experimentos com os atributos “Tipo de Projeto” e “Plataforma Tecnológica”.

1.4 Estrutura do Documento

Este trabalho está dividido em mais outros cinco capítulos, além deste introdutório, totalizando seis capítulos. Os próximos capítulos visam expor de forma detalhada como este trabalho foi desenvolvido para o alcance dos objetivos propostos, e estão divididos da seguinte forma:

Capítulo 2 – **Gerenciamento de Riscos do Projeto** – este capítulo trata da Gerência de Riscos, com um maior foco para o processo de Identificação de Riscos;

Capítulo 3 – **CBR Risk: Método para Identificação Automática de Riscos em Projetos de Software** – mostra todo o mecanismo e funcionamento da ferramenta *CBR Risk* [3] e sua evolução com os experimentos [2];

Capítulo 4 – **Experimentação** – este capítulo descreve os estudos experimentais realizados para a avaliação dos atributos “Tipo de Projeto” e “Plataforma Tecnológica”, utilizados para o cálculo de similaridade entre projetos no *CBR Risk*;

Capítulo 5 – **Análise dos Resultados** – a análise dos resultados dos experimentos é descrita nesse capítulo;

Capítulo 6 – **Conclusões e Trabalhos Futuros** – conclusões acerca do trabalho desenvolvido e sugestões de trabalhos futuros.

Capítulo 2

Gerenciamento de Riscos do Projeto

A Gerência de Riscos é a área focada neste trabalho. Todo o trabalho de aperfeiçoamento e experimentos feito sobre o *CBR Risk*, no trabalho aqui exposto, tem como objetivo melhorar o suporte à decisão na atividade de Identificação de Riscos. As práticas do Guia PMBOK [1] foram utilizadas como base neste trabalho. Neste capítulo será dada uma visão geral da disciplina Gerência de Riscos (Seção 2.1) e uma explanação maior do processo de Identificação de Riscos (Seção 2.2).

2.1 Gerência de Riscos – Visão Geral

O Gerenciamento de Riscos do projeto é uma das áreas de conhecimento da Gerência de Projetos. A atividade de Gerência de Riscos tem como objetivos planejar, identificar, analisar, controlar as respostas e monitorar os riscos pertencentes ao projeto em questão [1].

Risco pode ser definido como toda situação ou evento incerto com probabilidade de ocorrência conhecida, que pode gerar impactos positivos ou negativos aos objetivos do projeto. Cabe ao gerente de projetos, através de uma Gerência de Riscos bem feita, tentar minimizar a probabilidade e os impactos negativos e maximizar a probabilidade e os impactos positivos provenientes dos riscos.

O Gerenciamento de Riscos pode ser dividido em seis processos que interagem entre si e com processos relacionados a outras áreas de conhecimento da gestão de projetos. A esses processos estão associadas entradas, saídas, ferramentas e técnicas (como pode ser visto na Figura 1) e, apesar de serem representados pelo Guia PMBOK como interfaces bem-definidas e distintas, podem se sobrepor um ao outro e interagir de diversas maneiras. Outra característica importante dos processos da Gerência de Riscos é que eles ocorrem durante todo o

desenvolvimento do projeto pelo menos uma vez e, se o projeto estiver dividido em fases, ele vai ocorrer pelo menos em uma fase. Os processos da gestão de riscos incluem [1]:

- **Planejamento do Gerenciamento de Riscos** – Nesse processo é decidido como será a abordagem, o planejamento e a execução das diversas atividades do Gerenciamento de Riscos;
- **Identificação de Riscos** – Processo essencial em que os riscos ao projeto são identificados e é gerado um Registro de Riscos para posterior análise;
- **Análise qualitativa dos riscos** – Os riscos são priorizados através da análise da combinação da probabilidade de ocorrência com os impactos gerados pelos riscos;
- **Análise quantitativa dos riscos** – É feita uma classificação numérica para os riscos priorizados na análise qualitativa;
- **Planejamento de respostas a riscos** – Nesse processo é feito o planejamento de como os riscos serão tratados e são definidas as ações a serem tomadas com o intuito de aumentar as oportunidades e diminuir as ameaças aos objetivos do projeto;
- **Monitoramento e controle de riscos** – Processo responsável pelo constante acompanhamento dos riscos e identificação de novos riscos que venham a surgir. Além disso, nesse processo é executado os planos de respostas a riscos e é avaliado se essas respostas estão sendo eficientes.

A excelência na Gerência de Riscos está relacionada ao cumprimento desses processos/atividades com consistência e eficiência e, através de uma abordagem pró-ativa e não reativa. Apesar da extrema importância, a prática da Gerência de Riscos ainda é recente e muitas de suas técnicas e ferramentas são desconhecidas [5, 6, 7].

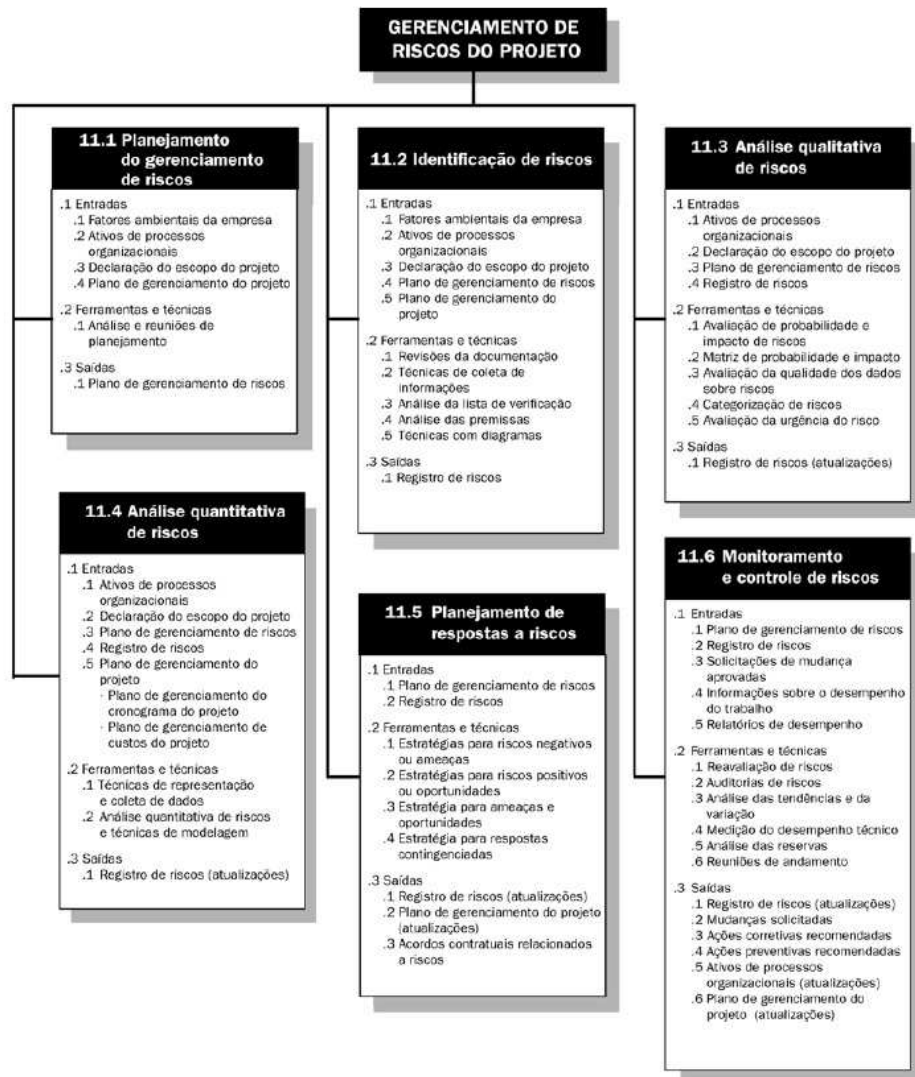


Figura 1. Visão geral do Gerenciamento de Riscos do Projeto [1]

2.2 Identificação de Riscos

A Identificação de Riscos é um processo essencial para a execução da atividade de Gerência de Riscos. Ou seja, não tem como gerenciar riscos sem a identificação dos mesmos. Essa atividade consiste no uso de técnicas ou ferramentas para a identificação e a documentação de características de riscos potenciais os quais afetariam os objetivos do projeto. Podem participar dessa atividade os gerentes de projetos, equipes de Gerenciamento de Riscos designadas, especialistas de fora da equipe do projeto, clientes, usuários finais, entre outros. Apesar das figuras citadas anteriormente serem os principais participantes do processo, é aconselhável o incentivo de toda a equipe do projeto para a participação na atividade de Identificação de Riscos [1].

Algumas das principais ferramentas e técnicas usadas no processo de Identificação de Riscos são [1]:

- **Revisões da documentação** – É feita uma análise sobre a documentação do projeto, incluindo requisitos, premissas, planos e outras informações. A consistência dos planos com as premissas e requisitos podem ser indicadores de riscos para o projeto;
- **Técnicas de coletas de informações** – Incluem algumas técnicas como Brainstorming (uso de especialistas que não fazem parte da equipe do projeto que dispõem idéias para a elaboração de uma lista de riscos), Técnica Delphi (especialistas dispõem suas idéias anonimamente através de um questionário, os questionários são redistribuídos e comentários são adicionados até se chegar num consenso), Entrevistas, Identificação da causa-raiz (permite o agrupamento de riscos por causas) e SWOT (Análise dos pontos fortes e fracos, oportunidades e ameaças);
- **Análise da lista de verificação** – São utilizadas listas de verificação elaboradas com base em informações históricas e de projetos anteriores semelhantes. As listas de verificação são aperfeiçoadas com a inclusão de novas informações ao término de um projeto;
- **Análise das premissas** – Todo o projeto a ser desenvolvido parte de premissas, as quais devem ser exatas, consistentes e completas. A análise das premissas é feita para a validação das mesmas e identificação de riscos;
- **Técnicas com diagramas** – São construídos diagramas para o estudo de riscos: Diagramas de causa e efeito, Fluxogramas e Diagramas de influência.

Um ponto importante a ser observado é que a experiência e informações passadas são de extrema importância na utilização de qualquer técnica de Identificação de Riscos. Inclusive no *CBR Risk* é necessário uma base de projetos passados para uma boa performance da ferramenta (quanto maior for essa base, melhor serão os resultados obtidos).

As saídas geradas pelo processo de Identificação de Riscos são compiladas em um documento chamado de “Registro de Riscos” que inclui informações como: os riscos

identificados, as respostas aos riscos (respostas podem ser identificadas durante a identificação de riscos), as causas dos riscos e categorias dos riscos [1].

2.3 Resumo do Capítulo

Este capítulo objetivou apresentar ao leitor uma visão geral da disciplina abordada durante todo o trabalho: a Gerência de Riscos. É mostrado que tal disciplina é dividida em processos que interagem entre si e, dentre esses processos, é focado neste capítulo o de Identificação de Riscos.

É constatado que o processo de Identificação de Riscos é vital para a execução da Gerência de Riscos e as principais técnicas utilizadas para tal atividade são descritas neste capítulo.

Capítulo 3

CBR Risk: Método para Identificação Automática de Riscos em Projetos de Software

Como constatado anteriormente, o processo de identificação de riscos é uma atividade vital para a gestão de riscos. Uma identificação de riscos bem feita aumenta as chances de sucesso no tratamento dos riscos e, por conseguinte, a diminuição ou aumento nos impactos aos objetivos do projeto [8].

Com o objetivo de suporte à atividade de identificação de riscos, foi desenvolvido por Arthur Lins, em seu Trabalho de Conclusão de Curso [3], um modelo para identificação automática de riscos usando uma técnica de inteligência artificial chamada Raciocínio Baseado em Casos (*Case-Based Reasoning*). O modelo foi evoluído em um protótipo cujo nome é: *CBR Risk Method*.

O capítulo que segue vai mostrar em detalhes o funcionamento do *CBR Risk Method*, dando uma introdução à técnica de Raciocínio Baseado em Casos (seção 3.1), passando pelo modelo (seção 3.2) até chegar ao protótipo (seção 3.3).

3.1 Raciocínio Baseado em Casos

Raciocínio Baseado em Casos (RBC) é uma técnica de inteligência artificial que tenta identificar soluções para um novo problema baseando-se em problemas semelhantes já resolvidos. O RBC está presente em muitos domínios de aplicação, podendo ser aplicado em diversas áreas como medicina, arquitetura, administração, engenharia, entre outras [9].

3.1.1 Elementos Básicos

Os sistemas RBC utilizam um processo interativo constituído genericamente por: identificação da situação atual, busca da experiência mais semelhante na memória e aplicação do conhecimento desta experiência na situação atual [10]. Para a execução desse processo, é preciso a definição de alguns elementos básicos de um sistema RBC. São eles [9]:

- **Representação do conhecimento** – Em um sistema RBC, a principal forma de representar conhecimento é através de casos que descrevem experiências passadas. Um caso é composto basicamente da descrição do problema e sua solução. No entanto, outros tipos de conhecimento podem ser armazenados dependendo do domínio de aplicação;
- **Medida de Similaridade** – Num sistema RBC é requerido que se defina fórmulas e medidas para o cálculo de similaridade entre casos. Só assim é possível a recuperação de casos relevantes para a solução do problema atual;
- **Adaptação** – Ao serem retornados os casos mais semelhantes ao novo problema, dificilmente os casos recuperados são idênticos ao problema atual. Por isso, pode ser necessária a adaptação da solução desses casos para o novo problema;
- **Aprendizado** – O sistema de RBC é capaz de evoluir e aprender à medida que é utilizado, já que os problemas resolvidos se transformam em novos casos que podem ser recuperados posteriormente.

Os casos em um sistema RBC podem ter diferentes conteúdos e representações dependendo da área de aplicação. Por isso, eles precisam ser representados adequadamente, ou seja, necessitam de uma linguagem definida. Existem diversos tipos de representação para casos no contexto do RBC, entre elas, destacam-se [9]:

- **Representações atributo - valor** – É a mais simples representação de casos. Nela, casos são representados através de uma série de atributos e seus respectivos valores. Os atributos utilizados para o cálculo de similaridade são chamados de índices e são escolhidos através de um criterioso método de indexação. Os valores que os atributos podem ter geralmente estão associados a um determinado tipo

como, por exemplo, inteiro, real, booleano, string, data e símbolos (ordenados, não-ordenados e taxonômicos);

- **Representação orientada a objetos** – Descreve os casos particionando-os com relação a seus objetos, tal qual as linguagens de programação orientada a objetos. Essas representações são importantes para domínios complexos, onde podem ocorrer casos com estruturas variáveis;
- **Árvores e Grafos** – Árvores e grafos podem ser usados para a representação de casos através de redes semânticas e árvores k-d.

Outros pontos cruciais a serem considerados num sistema de RBC são quanto ao cálculo de similaridade e recuperação de casos. Qualquer sistema de RBC deve possuir mecanismos de cálculo de similaridade entre os casos na base de dados e o problema atual. A similaridade entre dois casos pode ser calculada através da similaridade global, que é a semelhança entre dois casos como um todo, englobando todos os atributos, e a similaridade local, que é a semelhança entre atributos específicos.

Quanto à recuperação de casos, ela consiste na sugestão de soluções úteis para o problema atual. Para realizar essa recuperação, é necessário casar a descrição do problema atual com os problemas armazenados na base de casos, aplicando uma medida de similaridade [9]. Existem alguns métodos para a recuperação de casos, entre eles destacam-se a recuperação seqüencial, a recuperação de dois níveis, a recuperação orientada a índices, a recuperação com árvores k-d e as redes de recuperação de casos.

O ciclo de um sistema RBC pode ser visualizado na Figura 2:

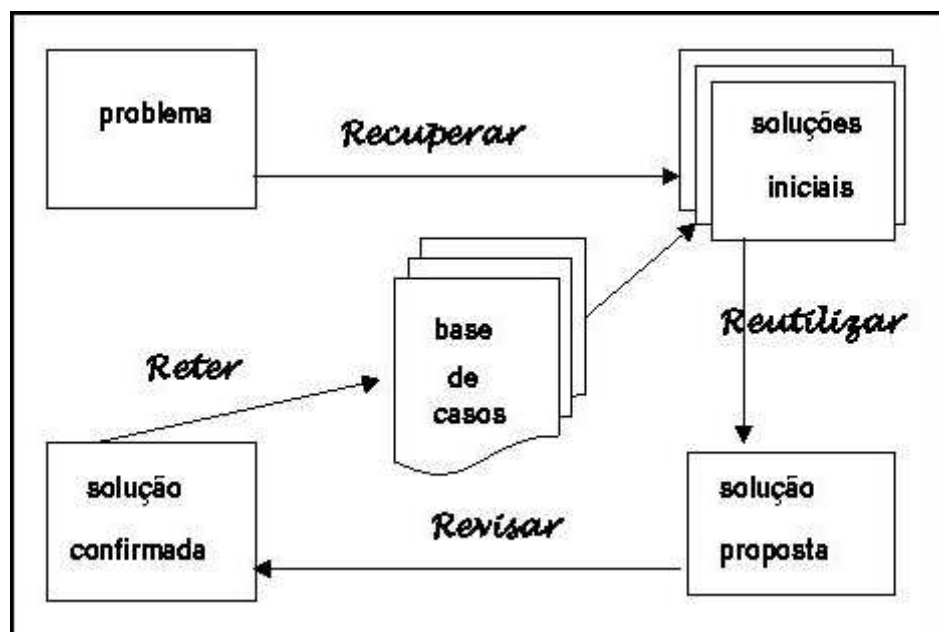


Figura 2. O ciclo do RBC [11]

Como pode ser visto, soluções iniciais úteis ao problema são recuperadas da base de casos e, a seguir, essas soluções podem ser reutilizadas, adaptando-as para a solução do problema atual (solução proposta). Essa solução proposta é então revisada e, caso apresente alguma falha ou não esteja correta, ela deve ser reparada. Caso esteja correta, a solução é confirmada e retida na base de casos (para poder ser recuperada posteriormente) com o objetivo de atualizar e estender a base de casos, tornando o sistema RBC mais poderoso [11].

3.2 Modelo

O *CBR Risk Method* foi desenvolvido como uma ferramenta de suporte à decisão para auxiliar os gerentes de projetos na identificação de riscos. O *CBR Risk Method* parte da premissa que “projetos de software semelhantes possuem riscos semelhantes” e funciona basicamente da seguinte forma: **Projetos Passados** (e seus respectivos riscos) são adicionados a uma base de dados e, para a identificação de riscos de um **Novo Projeto**, são definidas algumas características desse novo projeto (seção 3.2.1). Com base nessas características, é feito um cálculo de similaridade entre o novo projeto e os projetos na base de dados e, a seguir, são retornados os projetos mais semelhantes e seus respectivos riscos. De acordo com os riscos dos projetos mais semelhantes, estes são adaptados (ou não) para o novo projeto.

Como explicado anteriormente, essa técnica de solucionar um novo problema baseado em experiências passadas é conhecida como Raciocínio Baseado em Casos (RBC). No caso

específico do *CBR Risk Method*, o domínio de aplicação é o de desenvolvimento de softwares, os casos são os projetos de software e as informações retornadas dos projetos são os riscos identificados.

Tal qual todo sistema RBC, foi preciso definir três elementos essenciais para a construção do *CBR Risk Method*: a representação de casos, o cálculo de similaridade e a recuperação de casos.

3.2.1 Representação de Casos

A representação de casos utilizada no modelo do *CBR Risk Method* foi a representação atributo-valor [3]. No modelo *CBR Risk Method*, a representação atributo-valor define os casos. Com isso, foi necessário definir os atributos que caracterizariam os projetos de software, devendo ser capazes de fornecer informações suficientes para o cálculo de similaridade entre os casos (projetos). Esse processo de caracterização de projetos de software é uma atividade complexa devido à grande quantidade de características relacionadas a projetos no mundo real. Os atributos escolhidos, chamados de índices, foram seis:

- **Tamanho da equipe;**
- **Distribuição Geográfica;**
- **Experiência da equipe de desenvolvimento;**
- **Tamanho do projeto;**
- **Tipo de projeto;**
- **Plataforma tecnológica.**

A esses atributos expostos acima foram definidos intervalos de valores, exceto para os atributos “Tipo de Projeto” e “Plataforma Tecnológica” (o motivo será explicado mais adiante). Os valores dos atributos “Tamanho da Equipe”, “Distribuição Geográfica”, “Experiência da equipe de desenvolvimento” e “Tamanho do projeto” são do tipo símbolo ordenado, já que a sua similaridade é calculada baseada na distância entre os mesmos através de uma medida numérica que utiliza os valores de um (1) a cinco (5).

Os seguintes intervalos de valores foram utilizados:

- **Tamanho da equipe:**

1. Muito pequena: 1 – 6 pessoas;
2. Pequena: 7 – 20 pessoas;
3. Média: 21 – 50 pessoas;
4. Grande: 51 – 100 pessoas;
5. Muito grande: + de 100 pessoas.

- **Distribuição Geográfica:**

1. Mesma sala;
2. Mesmo prédio, diferentes salas;
3. Mesma cidade, mesma empresa, prédios diferentes;
4. Mesma cidade, empresas diferentes;
5. Cidades diferentes.

- **Tamanho do projeto:**

1. Até R\$ 50.000,00;
2. Entre R\$ 50.000,00 e R\$ 150.000,00;
3. Entre R\$ 150.000,00 e R\$ 1.000.000,00;
4. Entre R\$ 1.000.000,00 e R\$ 3.000.000,00;
5. Acima de R\$ 3.000.000,00.

O atributo “Experiência da equipe de desenvolvimento” foi dividido em três sub-atributos:

- **Experiência no processo;**
- **Experiência no domínio da aplicação;**
- **Experiência na tecnologia.**

Cada um dos três sub-atributos utiliza os seguintes intervalos de valores:

- I. Nenhum projeto;
- II. 1 projeto;

- III. 2 a 3 projetos;
- IV. 4 a 5 projetos;
- V. Mais de 5 projetos.

Quanto aos atributos “Tipo de Projeto” e “Plataforma Tecnológica”, como eles possuem uma grande quantidade de possíveis valores e novos tipos de projetos e plataformas tecnológicas surgem rapidamente no domínio de desenvolvimento de softwares, foi deliberado que o usuário do sistema é o responsável por definir esses valores (não há valores pré-definidos). Com isso, esses valores são do tipo símbolo não-ordenado, ou seja, não há nenhuma ordem de precedência entre eles [9]. Inclusive, um dos objetivos deste trabalho é uma avaliação mais detalhada da contribuição destes dois índices no cálculo de similaridade dos projetos, na base de casos.

3.2.2 Cálculo de Similaridade

Definidos os atributos (índices) que seriam utilizados no cálculo de similaridade entre projetos, o passo seguinte foi a definição da forma como seria calculado o grau de semelhança entre os projetos atual e da base de dados. No modelo original do *CBR Risk Method*, a similaridade entre casos era calculada em duas etapas: cálculo da similaridade local (entre atributos) e cálculo da similaridade global (entre projetos).

Mas Thiago Trigo, em seu Trabalho de Conclusão de Curso [2], realizou experimentos com o objetivo de validar a eficiência do modelo na prática. Esses experimentos foram realizados fazendo variações nas fórmulas de cálculo de similaridade e comparando o desempenho entre as mesmas. Planilhas eletrônicas foram utilizadas na execução dos experimentos e simulações, e todo o processo de coleta de dados acerca de projetos e riscos para serem utilizados nos experimentos foi feito através de questionários em versão escrita e digital.

Com a análise dos resultados dos experimentos executados por Thiago, chegou-se à conclusão que a fórmula para o cálculo de similaridade entre projetos seria baseada na Distância Euclidiana entre os atributos. Tal fórmula foi a que obteve melhor desempenho dentre todas as analisadas por Thiago e foi implementada efetivamente no trabalho aqui exposto. Essa fórmula não possui duas etapas como no modelo original, mas apenas uma:

$$S_g = \sqrt{\sum w_i (q_i - c_i)^2}$$

Equação 1. Expressão para o cálculo da similaridade global do *CBR Risk Method*

Na expressão apresentada acima, S_g representa a similaridade global, q_i representa os valores dos atributos do projeto atual, c_i representa os valores dos atributos do projeto da base de dados e w_i representa os pesos de cada atributo. Diferentemente do modelo original, com a Distância Euclidiana quanto menor o valor de S_g , maior será a similaridade entre projetos e quanto menor o valor dos pesos de cada atributo, maior será a sua relevância no cálculo de similaridade.

3.2.3 Recuperação de Casos

A técnica de recuperação de casos utilizada no *CBR Risk Method* é a mais simples de todas, a recuperação seqüencial. Nessa técnica, o cálculo de similaridade é feito com todos os projetos da base de dados e são retornados os projetos mais similares ao projeto atual e seus respectivos riscos [9].

As principais vantagens desse método são a implementação extremamente simples e a certeza de que todos os projetos da base de dados serão comparados com o projeto atual, o que garante que um projeto similar que tenha uma solução útil para o problema vai ser comparado.

Thiago Trigo, durante a execução dos seus experimentos, definiu valores de filtros com o objetivo de evitar que projetos com baixa similaridade global fossem retornados (sugeridos ao usuário), ou seja, evitar que a ferramenta recupere projetos muito diferentes do projeto submetido para identificação de riscos. Esses filtros foram calculados a partir de médias dos possíveis valores de similaridade global, e este trabalho tem como uma de suas contribuições a implementação do filtro mais rigoroso na ferramenta. O filtro mais rigoroso é o mais difícil de ser atingido e aquele que apresentou melhores resultados durante os experimentos. Tal filtro possui o valor 4,9497, o que significa que apenas os projetos com valor de similaridade global (S_g) em relação ao projeto atual menor ou igual a 4,9497 serão retornados pela ferramenta.

3.3 Protótipo

Após a explicação de todo o modelo do *CBR Risk Method* apresentada na seção anterior, foi desenvolvido um protótipo para o mesmo baseado em tecnologia WEB, utilizando a linguagem PHP e o banco de dados MySQL [12].

O objetivo principal da implementação do protótipo é a avaliação dos principais requisitos para validar o uso da técnica de identificação de riscos baseada em RBC. O protótipo desenvolvido possui a tela inicial mostrada abaixo:

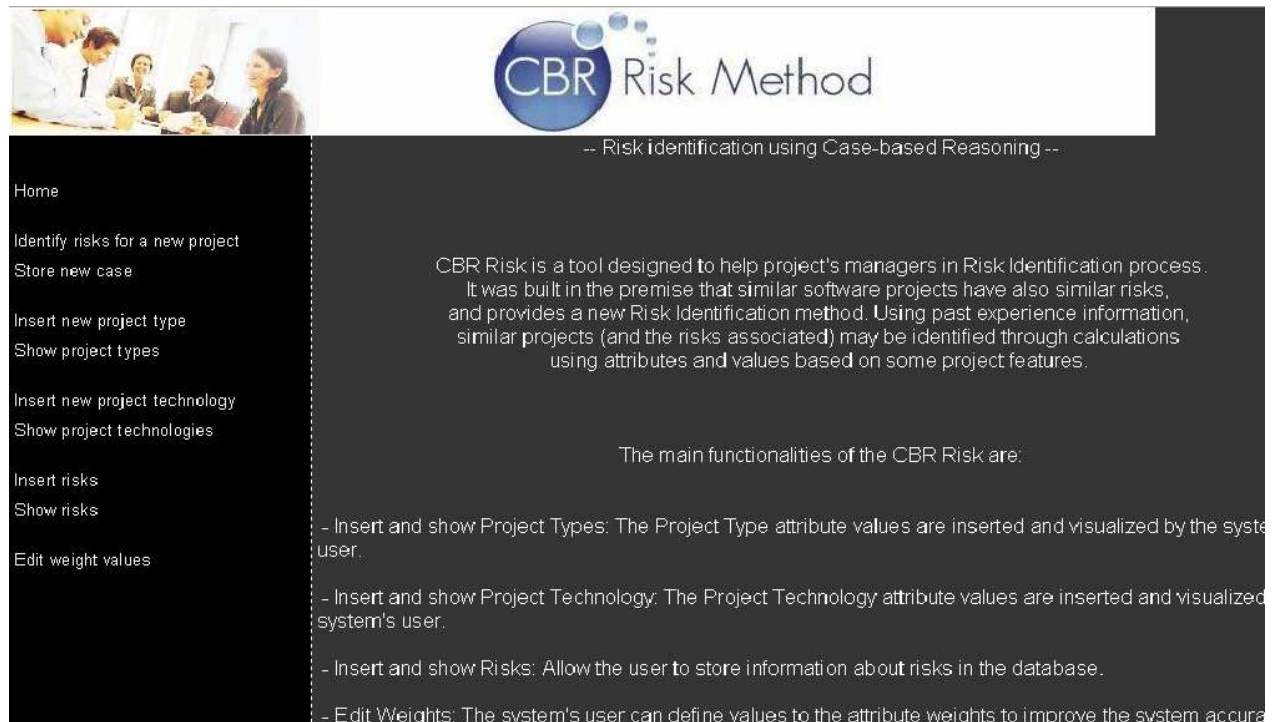


Figura 3. Tela Inicial do *CBR Risk Method*

O *CBR Risk Method* possui as seguintes funcionalidades:

- **Identificação de riscos para um novo projeto** (“*Identify risks for a new project*”) – Nessa opção o usuário vai selecionar as características (atributos expostos anteriormente) do novo projeto e clicar no botão correspondente (“*FIND RISKS!*”). O sistema vai retornar os projetos mais similares e seus respectivos riscos com os seguintes dados: se o risco já ocorreu no projeto, o plano de contingência adotado e informações importantes sobre o risco;
- **Armazenamento de projetos** (“*Store new case*”) – Nessa opção o usuário armazena novos projetos (casos) no banco de dados. O primeiro passo é escrever o nome do projeto e selecionar suas características. Logo depois o usuário seleciona os riscos associados ao projeto, informa se aquele risco ocorreu ou não, escreve informações sobre o risco e o plano de contingência adotado. Após esses passos o sistema mostra todos os projetos que estão armazenados na base de dados com seus respectivos riscos e informações;

- **Cadastramento de novo “Tipo de Projeto”** (“*Insert new project type*”) – Nessa opção o usuário informa ao sistema quais os valores que o atributo “Tipo de Projeto” poderá assumir;
- **Visualização de tipos de projeto** (“*Show project types*”) – Nessa opção é listado todos os valores de “Tipo de Projeto” cadastrados anteriormente;
- **Cadastramento de nova “Plataforma Tecnológica”** (“*Insert new project technology*”) – Nessa opção o usuário informa ao sistema quais os valores que o atributo “Plataforma Tecnológica” poderá assumir;
- **Visualização de plataformas tecnológicas** (“*Show project technologies*”) – Nessa opção é listado todos os valores de “Plataforma Tecnológica” cadastrados anteriormente;
- **Cadastramento de riscos** (“*Insert risks*”) – Nessa opção o usuário cadastra riscos no banco de dados para estes poderem, posteriormente, ser associados a projetos na opção de armazenamento de projetos. O usuário deve escrever o nome do risco, sua descrição e seu fator de risco. Após isso, todos os riscos armazenados no banco são listados com suas respectivas informações;
- **Visualização de riscos** (“*Show risks*”) – Nessa opção é listado todos os riscos cadastrados anteriormente com suas respectivas informações (nome, descrição e fator de risco);
- **Edição de valores dos pesos** (“*Edit weight values*”) – Nessa opção o usuário seleciona os valores dos pesos (1 a 6) para cada atributo, com o intuito de dar maior relevância a um determinado atributo ou atributos no cálculo de similaridade.

3.4 Resumo do Capítulo

Este capítulo aborda toda a metodologia usada no desenvolvimento da ferramenta para identificação automática de riscos *CBR Risk Method*. Ele apresenta uma visão geral da técnica de inteligência artificial usada na ferramenta, Raciocínio Baseado em Casos, e como a ferramenta foi modelada através dos elementos essenciais de um sistema RBC – representação de casos, cálculo de similaridade e recuperação de casos.

É visto que o *CBR Risk Method* utiliza a representação atributo-valor e o cálculo de similaridade implementado na ferramenta é baseado na Distância Euclidiana entre os atributos. A recuperação de casos na base de dados é seqüencial, ou seja, a similaridade é calculada entre todos os projetos na base e o novo projeto e existe um filtro para evitar a recuperação de casos não-similares.

O capítulo 3 também explica todas as funcionalidades presentes no *CBR Risk Method*.

Capítulo 4

Experimentação

A experimentação é um processo importante de verificação e validação de teorias. Realizando experimentos e analisando os resultados dos mesmos, é possível constatar, na prática, se uma teoria está correta ou não, e até mesmo dizer se um método apresenta um desempenho melhor ou pior que outro. Neste capítulo é descrito o processo experimental utilizado na avaliação dos atributos “Tipo de Projeto” e “Plataforma Tecnológica”, através da coleta de dados para execução dos experimentos (seção 4.2), definição dos experimentos (seção 4.3) e definição do critério de avaliação do desempenho (seção 4.4).

4.1 Introdução

Em Travassos e outros autores [4], são listados alguns objetivos da experimentação em Engenharia de Software, os quais se adequam ao intuito principal deste trabalho:

- A experimentação pode ajudar a construir uma base de conhecimento confiável e reduzir assim incerteza sobre quais teorias, ferramentas e metodologias são adequadas;
- A experimentação pode acelerar o processo eliminando abordagens inúteis e suposições errôneas. A experimentação ajuda também a orientar a engenharia e a teoria nas direções promissoras de pesquisa.

Existem quatro métodos principais para a condução dos experimentos na Engenharia de Software: científico, de engenharia, experimental e analítico. O método que mais se adequa para a realização dos experimentos neste trabalho é o **método experimental**. Esta abordagem sugere o modelo, desenvolve o método, aplica um experimento, mede e analisa, avalia o modelo e repete o

processo. O processo começa com o levantamento de um modelo e tenta estudar o efeito do processo ou produto sugerido por tal modelo [4].

Todo o processo experimental foi realizado no *CBR Risk* com o intuito de aprimorar a ferramenta e torná-la mais eficiente. As experimentações descritas a seguir foram realizadas em cima dos dois atributos de projetos do *CBR Risk* que não possuem valores pré-definidos: “Tipo de Projeto” e “Plataforma Tecnológica”. Os valores desses atributos são definidos pelo usuário do sistema e eles são do tipo símbolos não-ordenados, ou seja, diferentemente dos outros atributos (que são do tipo símbolos ordenados), esses valores não têm nenhuma ordem de precedência entre eles [9]. Portanto, podemos apenas dizer se dois projetos contêm o mesmo valor ou um valor diferente para esses atributos ao compará-los. Com isso, esses atributos só podem assumir dois valores numéricos para o cálculo de similaridade: 0 (zero), no caso dos valores comparados forem iguais, ou 1 (um), no caso dos valores comparados forem diferentes (lembrando que na Distância Euclidiana os projetos mais similares são os que possuem menor valor de similaridade global). É importante também ressaltar que esses dois atributos são muito importantes para se “medir” o grau de similaridade entre projetos de software, já que “Tipo de Projeto” e “Plataforma Tecnológica” são características das mais relevantes no que diz respeito a projetos de software. Esse caráter único dos atributos “Tipo de Projeto” e “Plataforma Tecnológica” motivou a realização dos experimentos. Após as experimentações realizadas, foi feita a análise dos resultados (Capítulo 5).

4.2 Coleta de Dados

Para a realização dos experimentos foi necessária a coleta de dados acerca de projetos de software. Os dados utilizados nos experimentos a seguir foram coletados através de questionários em versões escrita e digital e distribuídos em listas de discussões, grupos de estudos e turmas de Gerência de Projetos da QualiTi. Tais questionários e a coleta desses dados foram realizados por Thiago Trigo em seu Trabalho de Conclusão de Curso [2].

Ao todo, foram coletadas informações sobre 18 projetos diferentes. A opção de reaproveitar os dados já coletados por Thiago é a de evitar trabalho repetitivo e, conseqüentemente, otimizar o tempo.

4.3 Definição dos Experimentos

Analisando a fórmula do cálculo de similaridade descrita na **Equação 1** (Capítulo 3), pode-se dividir a mesma em partes para a verificação dos valores numéricos que cada atributo pode assumir no cálculo de similaridade. Para isso, o cálculo foi dividido em duas fases, que são descritas a seguir:

Na primeira fase cada atributo é calculado independentemente, como visto abaixo:

- **Tamanho da Equipe:**

$$S_1 = (q_i - c_i)^2$$

Onde S_1 corresponde ao valor que o atributo “Tamanho da Equipe” vai assumir, q_i é o valor do atributo do projeto atual e c_i é o valor do atributo do projeto da base de dados. As variáveis q_i e c_i podem assumir os valores 1, 2, 3, 4 e 5.

- **Distribuição Geográfica:**

$$S_2 = (q_i - c_i)^2$$

Onde S_2 corresponde ao valor que o atributo “Distribuição Geográfica” vai assumir, q_i é o valor do atributo do projeto atual e c_i é o valor do atributo do projeto da base de dados. As variáveis q_i e c_i podem assumir os valores 1, 2, 3, 4 e 5.

- **Experiência no Processo:**

$$S_{3,1} = (q_i - c_i)^2$$

Onde $S_{3,1}$ corresponde ao valor que o atributo “Experiência no Processo” vai assumir, q_i é o valor do atributo do projeto atual e c_i é o valor do atributo do projeto da base de dados. As variáveis q_i e c_i podem assumir os valores 1, 2, 3, 4 e 5.

- **Experiência no Domínio da Aplicação:**

$$S_{3,2} = (q_i - c_i)^2$$

Onde $S_{3,2}$ corresponde ao valor que o atributo “Experiência no Domínio da Aplicação” vai assumir, q_i é o valor do atributo do projeto atual e c_i é o valor do atributo do projeto da base de dados. As variáveis q_i e c_i podem assumir os valores 1, 2, 3, 4 e 5.

- **Experiência na Tecnologia:**

$$S_{3,3} = (q_i - c_i)^2$$

Onde $S_{3,3}$ corresponde ao valor que o atributo “Experiência na Tecnologia” vai assumir, q_i é o valor do atributo do projeto atual e c_i é o valor do atributo do projeto da base de dados. As variáveis q_i e c_i podem assumir os valores 1, 2, 3, 4 e 5.

- **Tamanho do Projeto:**

$$S_4 = (q_i - c_i)^2$$

Onde S_4 corresponde ao valor que o atributo “Tamanho do Projeto” vai assumir, q_i é o valor do atributo do projeto atual e c_i é o valor do atributo do projeto da base de dados. As variáveis q_i e c_i podem assumir os valores 1, 2, 3, 4 e 5.

- **Tipo de Projeto:**

$$\text{Se } q_i = c_i \text{ Então } S_5 = 0 \text{ Senão } S_5 = 1$$

Onde S_5 corresponde ao valor que o atributo “Tipo de Projeto” vai assumir, q_i é o valor do atributo do projeto atual e c_i é o valor do atributo do projeto da base de dados. As variáveis q_i e c_i correspondem às strings que foram previamente adicionadas ao sistema.

- **Plataforma Tecnológica:**

$$\text{Se } q_i = c_i \text{ Então } S_6 = 0 \text{ Senão } S_6 = 1$$

Onde S_6 corresponde ao valor que o atributo “Plataforma Tecnológica” vai assumir, q_i é o valor do atributo do projeto atual e c_i é o valor do atributo do projeto da base de dados. As variáveis q_i e c_i correspondem às strings que foram previamente adicionadas ao sistema.

Na segunda fase é feito o cálculo da similaridade global através da equação abaixo:

$$S_g = \sqrt{w_1S_1 + w_2S_2 + w_3(S_{3.1} + S_{3.2} + S_{3.3}) + w_4S_4 + w_5S_5 + w_6S_6}$$

Equação 2. Expressão para o cálculo da similaridade global do *CBR Risk Method* (dividido em duas fases)

Onde, S_g é a similaridade global entre dois projetos, w_n com $n = \{1,2,3,4,5,6\}$ é o peso de cada atributo, $S_1, S_2, S_{3.1}, S_{3.2}, S_{3.3}, S_4, S_5$ e S_6 são, respectivamente, os valores calculados (na primeira fase) dos atributos “Tamanho da Equipe”, “Distribuição Geográfica”, “Experiência no Processo”, “Experiência no Domínio da Aplicação”, “Experiência na Tecnologia”, “Tamanho do Projeto”, “Tipo de Projeto” e “Plataforma Tecnológica”.

Com a divisão do cálculo de similaridade em duas fases, é possível verificar todos os possíveis valores numéricos que cada atributo pode assumir e, conseqüentemente, sua influência no cálculo de similaridade. A Tabela 1 apresenta os possíveis valores de cada atributo.

Tabela 1. Possíveis valores numéricos de cada atributo no cálculo de similaridade

Atributo	Possíveis valores (Si)
Tamanho da Equipe	0, 1, 4, 9, 16
Distribuição Geográfica	0, 1, 4, 9, 16
Experiência no Processo	0, 1, 4, 9, 16
Experiência no Domínio da Aplicação	0, 1, 4, 9, 16
Experiência na Tecnologia	0, 1, 4, 9, 16
Tamanho do Projeto	0, 1, 4, 9, 16

Tipo de Projeto	0, 1
Plataforma Tecnológica	0, 1

Analisando a Tabela 1 percebe-se que os atributos “Tamanho da Equipe”, “Distribuição Geográfica”, “Experiência no Processo”, “Experiência no Domínio da Aplicação”, “Experiência na Tecnologia” e “Tamanho do Projeto”, ou seja, os atributos com valores pré-definidos, podem atingir até o valor máximo 16 (caso em que q_i e c_i possuem valores 1 e 5). Em relação aos atributos sem valores pré-definidos “Tipo de Projeto” e “Plataforma Tecnológica”, eles podem atingir o valor máximo 1.

Visando eliminar essa discrepância de valores pensou-se, antes da realização dos experimentos, em definir cinco valores para os atributos “Tipo de Projeto” e “Plataforma Tecnológica” da mesma forma que os demais atributos, para que todos os atributos pudessem ter a mesma influência no cálculo de similaridade. Mas, visto a grande quantidade de tipos de projeto e plataformas tecnológicas utilizadas em empresas de desenvolvimento de software e o grande dinamismo nessa área, em que novos tipos de projeto e plataformas tecnológicas surgem rapidamente, julgou-se impossível resumir esses valores em apenas cinco. Para ilustrar o que foi dito acima, abaixo é apresentado os valores de “Tipo de Projeto” e “Plataforma Tecnológica” coletados no questionário desenvolvido em Trigo [2].

Tipos de Projeto:

- **Jogo;**
- **Financeiro;**
- **Automação Comercial;**
- **Administrativo;**
- **ERP;**
- **Logística;**
- **Científico;**
- **Manutenção Corporativa.**

Plataformas Tecnológicas:

- **Móvel - J2ME;**
- **Web - J2EE;**
- **Desktop - Java;**
- **Ajax, VB, PHP;**
- **Delphi;**
- **.Net;**
- **Cobol;**
- **C;**
- **Oracle;**
- **Web – Asp C# .Net.**

Como pode ser visto, em apenas 18 projetos coletados, 8 tipos de projeto e 10 plataformas tecnológicas diferentes foram mencionadas. Muitos valores além desses existem no domínio de desenvolvimento de softwares e muitos ainda estão por vir, comprovando a impossibilidade de definir apenas cinco.

Visto tudo o que foi apresentado acima, cabe a pergunta: “Será que os atributos “Tipo de Projeto” e “Plataforma Tecnológica”, podendo assumir apenas os valores 0 e 1, têm uma influência relevante no cálculo de similaridade global?”. Para responder essa pergunta e na tentativa de buscar valores mais apropriados para esses atributos foram realizados os experimentos. Tais experimentos utilizaram como valores numéricos para os atributos “Tipo de Projeto” e “Plataforma Tecnológica” os valores numéricos possíveis dos outros atributos (4, 9, 16). Além disso, foram realizados experimentos eliminando esses atributos e dando uma maior relevância aos mesmos, através da variação dos pesos dos demais atributos. Todos os experimentos descritos nos próximos tópicos foram realizados utilizando planilhas eletrônicas.

4.3.1 Experimento 1: Atributos “Tipo de Projeto” e “Plataforma Tecnológica” assumindo valores 0 e 4

O primeiro experimento realizado foi com os atributos “Tipo de Projeto” e “Plataforma Tecnológica” podendo assumir os valores 0 e 4. Com isso, a primeira fase do cálculo de similaridade muda para os dois atributos:

- **Tipo de Projeto:**

Se $q_i = c_i$ Então $S_5 = 0$ Senão $S_5 = 4$

Onde S_5 corresponde ao valor que o atributo “Tipo de Projeto” vai assumir, q_i é o valor do atributo do projeto atual e c_i é o valor do atributo do projeto da base de dados. As variáveis q_i e c_i correspondem às strings que foram previamente adicionadas ao sistema.

- **Plataforma Tecnológica:**

Se $q_i = c_i$ Então $S_6 = 0$ Senão $S_6 = 4$

Onde S_6 corresponde ao valor que o atributo “Plataforma Tecnológica” vai assumir, q_i é o valor do atributo do projeto atual e c_i é o valor do atributo do projeto da base de dados. As variáveis q_i e c_i correspondem às strings que foram previamente adicionadas ao sistema.

Para a realização dos experimentos com os valores acima, foram considerados todos os atributos com mesmo peso 1.

4.3.2 Experimento 2: Atributos “Tipo de Projeto” e “Plataforma Tecnológica” assumindo valores 0 e 9

O segundo experimento realizado foi com os atributos “Tipo de Projeto” e “Plataforma Tecnológica” podendo assumir os valores 0 e 9. Com isso, a primeira fase do cálculo de similaridade muda para os dois atributos:

- **Tipo de Projeto:**

Se $q_i = c_i$ Então $S_5 = 0$ Senão $S_5 = 9$

Onde S_5 corresponde ao valor que o atributo “Tipo de Projeto” vai assumir, q_i é o valor do atributo do projeto atual e c_i é o valor do atributo do projeto da base de dados. As variáveis q_i e c_i correspondem às strings que foram previamente adicionadas ao sistema.

- **Plataforma Tecnológica:**

Se $q_i = c_i$ Então $S_6 = 0$ Senão $S_6 = 9$

Onde S_6 corresponde ao valor que o atributo “Plataforma Tecnológica” vai assumir, q_i é o valor do atributo do projeto atual e c_i é o valor do atributo do

projeto da base de dados. As variáveis q_i e c_i correspondem às strings que foram previamente adicionadas ao sistema.

Para a realização dos experimentos com os valores acima, foram considerados todos os atributos com mesmo peso 1.

4.3.3 Experimento 3: Atributos “Tipo de Projeto” e “Plataforma Tecnológica” assumindo valores 0 e 16

O terceiro experimento realizado foi com os atributos “Tipo de Projeto” e “Plataforma Tecnológica” podendo assumir os valores 0 e 16. Com isso, a primeira fase do cálculo de similaridade muda para os dois atributos:

- **Tipo de Projeto:**

$$\text{Se } q_i = c_i \text{ Então } S_5 = 0 \text{ Senão } S_5 = 16$$

Onde S_5 corresponde ao valor que o atributo “Tipo de Projeto” vai assumir, q_i é o valor do atributo do projeto atual e c_i é o valor do atributo do projeto da base de dados. As variáveis q_i e c_i correspondem às strings que foram previamente adicionadas ao sistema.

- **Plataforma Tecnológica:**

$$\text{Se } q_i = c_i \text{ Então } S_6 = 0 \text{ Senão } S_6 = 16$$

Onde S_6 corresponde ao valor que o atributo “Plataforma Tecnológica” vai assumir, q_i é o valor do atributo do projeto atual e c_i é o valor do atributo do projeto da base de dados. As variáveis q_i e c_i correspondem às strings que foram previamente adicionadas ao sistema.

Para a realização dos experimentos com os valores acima, foram considerados todos os atributos com mesmo peso 1.

4.3.4 Experimento 4: Atributos “Tipo de Projeto” e “Plataforma Tecnológica” com maior relevância

No quarto experimento realizado, os atributos “Tipo de Projeto” e “Plataforma Tecnológica” tiveram seus possíveis valores mantidos como originalmente (0 e 1). Seus pesos também foram mantidos com valor 1, mas os demais atributos tiveram seus pesos configurados para 6. Como

visto anteriormente, quanto menor o valor do peso para determinado atributo, maior será sua relevância para o cálculo de similaridade. Portanto, isso foi feito com o objetivo de dar maior relevância aos atributos “Tipo de Projeto” e “Plataforma Tecnológica” em relação aos demais atributos.

4.3.5 Experimento 5: Sem os atributos “Tipo de Projeto” e “Plataforma Tecnológica”

No quinto experimento realizado, os atributos “Tipo de Projeto” e “Plataforma Tecnológica” foram eliminados do cálculo de similaridade e, conseqüentemente, da ferramenta. O objetivo desse quinto experimento é verificar se esses atributos possuem alguma relevância para o cálculo de similaridade, comparando os resultados desse experimento com os demais. Para a realização desses experimentos foram considerados todos os atributos com mesmo peso 1.

4.4 Critério de Avaliação do Desempenho

O critério de avaliação de desempenho utilizado para a análise dos resultados dos experimentos foi o critério %B definido em Trigo [2]. Esse critério é a porcentagem da quantidade de projetos retornados corretamente em relação a quantidade de projetos retornados após a filtragem e quanto maior o seu resultado, melhor será a avaliação da variação sob estudo. Por exemplo: suponha que após a análise da descrição dos riscos de cada projeto na base de dados observamos que determinado Projeto X tem 8 projetos com riscos parecidos e que após utilizarmos o *CBR Risk* para identificar os projetos similares ao Projeto X, ele nos retornou 6 projetos, sendo que desses 6, apenas 4 estavam certos, logo, o Projeto X terá $\% B = 4/6 = 0,667$. Portanto, um valor de %B maior que 60% pode ser considerado um resultado satisfatório.

4.5 Resumo do Capítulo

Este capítulo apresenta todo o processo experimental utilizado na avaliação dos atributos “Tipo de Projeto” e “Plataforma Tecnológica”, desde a coleta de dados realizada, até a definição do critério de avaliação do desempenho para a análise dos resultados. Os experimentos são detalhadamente definidos e têm o objetivo principal de verificar a influência dos atributos sob estudo no cálculo de similaridade do *CBR Risk*.

Capítulo 5

Análise dos Resultados

Neste capítulo será apresentado uma análise dos resultados dos experimentos descritos no capítulo anterior. Através dessa análise (seção 5.2) será possível verificar o desempenho do *CBR Risk* para cada variação nos atributos “Tipo de Projeto” e “Plataforma Tecnológica”. Esta medição de desempenho possibilitará a execução de um estudo comparativo (seção 5.3) entre as diversas variações.

5.1 Introdução

Com a execução dos experimentos e a definição do critério de avaliação do desempenho, o próximo passo é a análise dos resultados. Esta análise segue a seguinte estrutura:

- Médias dos Resultados %B para o filtro mais rigoroso. O filtro mais rigoroso é aquele que vai retornar os projetos mais similares, já que é o mais difícil de ser atingido;
- Desvio padrão dos resultados %B para o filtro mais rigoroso;
- Médias dos riscos que foram efetivamente identificados para o filtro mais rigoroso.

Através da tripla <médias dos resultados %B, desvio padrão, médias dos riscos efetivamente identificados> será definida as melhores variações nos atributos “Tipo de Projeto” e “Plataforma Tecnológica”.

5.2 Estrutura de Análise

A estrutura de análise dos resultados experimentais é descrita detalhadamente a seguir e servirá como base para o estudo comparativo realizado posteriormente.

5.2.1 Médias dos resultados %B

As médias dos resultados do critério %B para cada projeto são mostrados abaixo para o filtro mais rigoroso, pois entende-se que o filtro mais rigoroso é aquele que nos dá melhores resultados e é o filtro efetivamente implementado na ferramenta.

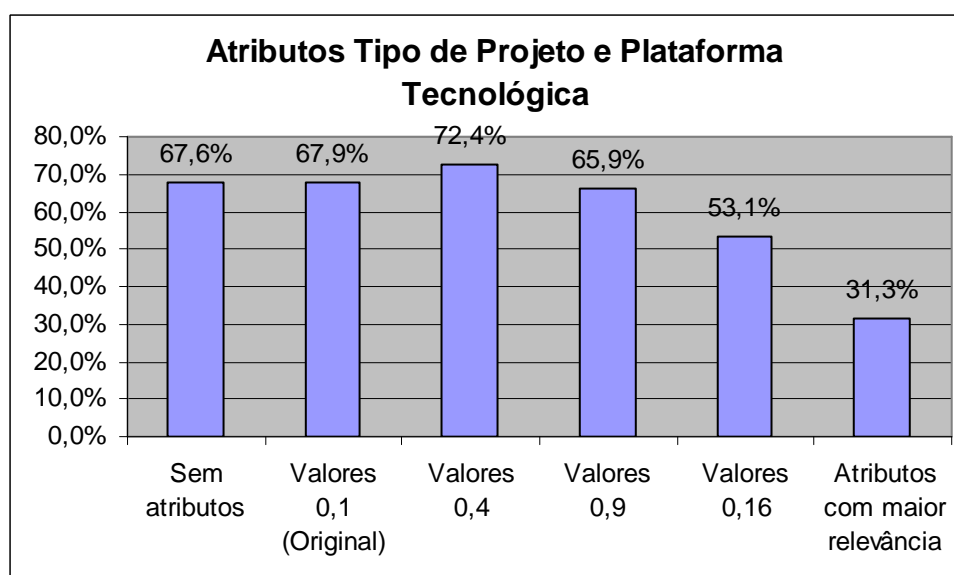


Figura 4. Médias dos Resultados %B

Observa-se no gráfico acima que os resultados com os atributos podendo assumir os valores 0 e 4 se sobressaíram sobre os demais (72,4%). A exclusão dos atributos praticamente não mudou nada os resultados em relação ao modelo original e foram próximos aos resultados com os atributos podendo assumir os valores 0 e 9. Os resultados do %B com os atributos podendo assumir os valores (0 e 16) e com uma maior relevância em relação aos demais atributos foram ruins.

5.2.2 Desvio Padrão dos resultados %B

O desvio padrão dos resultados foram calculados com o objetivo de medir a dispersão dos valores encontrados pelo critério %B para as diferentes variações nos atributos “Tipo de Projeto” e

“Plataforma Tecnológica”. Uma diferença muito grande nos resultados %B para os projetos implica um valor de desvio padrão alto, o que não é bom. O ideal é um valor de desvio padrão baixo, o que torna a variação mais confiável. Abaixo estão os resultados calculados:

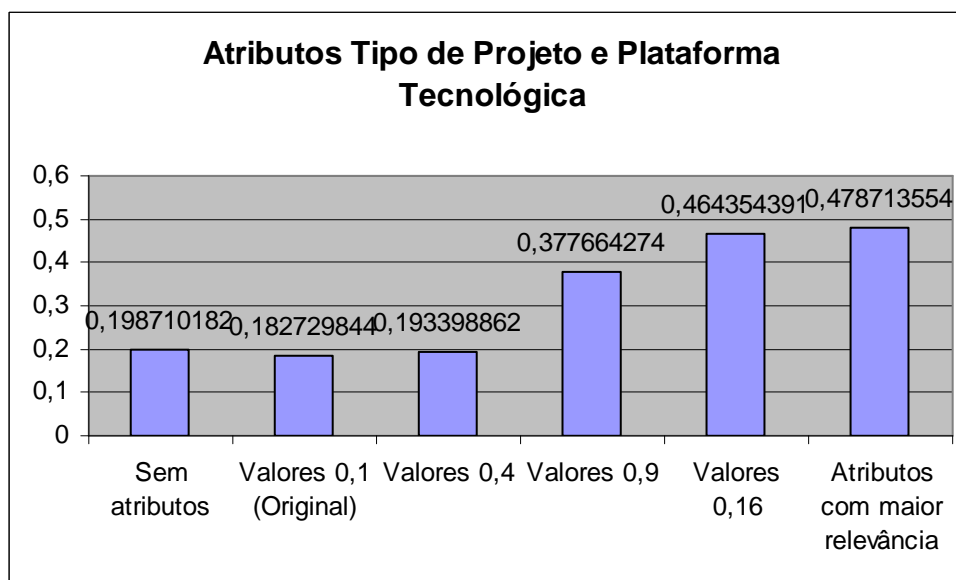


Figura 5. Desvio Padrão dos Resultados %B

Analisando os resultados acima, nota-se que o desvio padrão dos resultados para os atributos podendo assumir os valores (0 e 9), (0 e 16) e com uma maior relevância em relação aos demais atributos foram muito altos. Já para os atributos podendo assumir os valores (0 e 1), (0 e 4) e para a variação sem os atributos, os valores do desvio padrão foram bem próximo e bem menores em relação aos demais.

5.2.3 Riscos efetivamente identificados

O gráfico abaixo mostra a média dos riscos efetivamente identificados para os projetos coletados em cada variação dos experimentos:

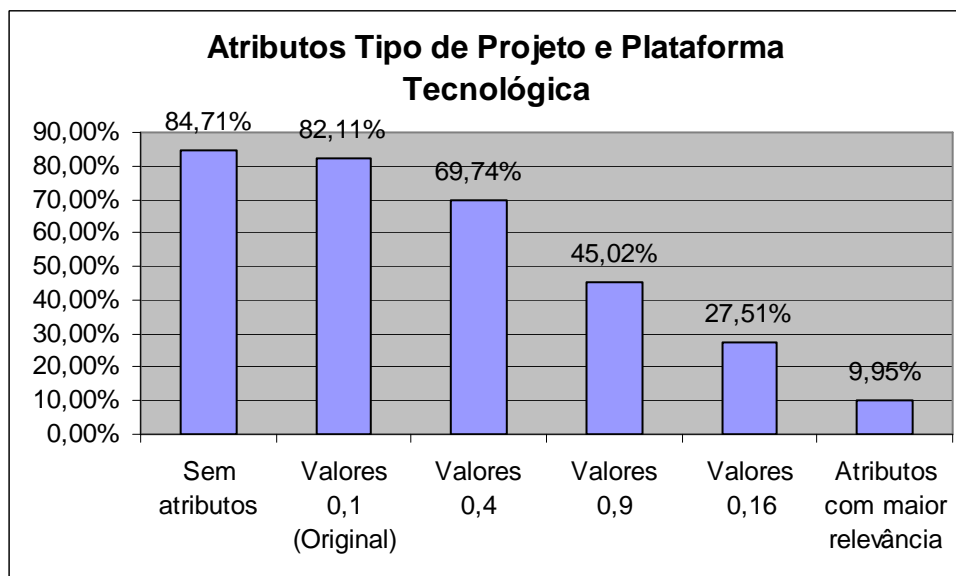


Figura 6. Média percentual dos riscos efetivamente identificados

Como podemos ver, os resultados para os atributos podendo assumir os valores (0 e 1) e os resultados para a variação sem os atributos foram melhores que os demais. Os resultados para os atributos podendo assumir os valores 0 e 4 foram razoáveis e para as outras variações foram ruins. A variação em que os atributos sob estudo tiveram maior relevância que os demais teve um resultado pífio (9,95%).

5.3 Estudo Comparativo

Com base nos resultados mostrados anteriormente, foi feito um estudo comparativo para se chegar às conclusões acerca dos experimentos realizados. O objetivo desse estudo é definir as variações de melhor desempenho para os atributos “Tipo de Projeto” e “Plataforma Tecnológica”. Essa definição será feita através da combinação <médias dos resultados %B, desvio padrão, médias dos riscos efetivamente identificados>.

Desse modo, cada variação possui a seguinte combinação:

Tabela 2. Resumo dos resultados para cada variação

Tipo de Projeto e Plataforma Tecnológica	Média dos Resultados %B	Desvio Padrão dos Resultados %B	Média dos riscos identificados
Sem atributos	67,6%	0,198710	84,71%
Valores 0 e 1	67,9%	0,182729	82,11%
Valores 0 e 4	72,4%	0,193398	69,74%
Valores 0 e 9	65,9%	0,377664	45,02%
Valores 0 e 16	53,1%	0,464354	27,51%
Maior Relevância (Menor Peso)	31,3%	0,478713	9,95%

Analisando a combinação de resultados resumidos na tabela acima com o objetivo de definir as melhores variações, elimina-se desse propósito, primeiramente, os experimentos com os atributos podendo assumir os valores (0 e 9), (0 e 16) e com uma maior relevância em relação aos demais atributos, visto que seus resultados destoam muito dos demais. Os experimentos com os atributos podendo assumir os valores (0 e 1) e (0 e 4) e os experimentos sem os atributos obtiveram resultados satisfatórios e se equivaleram, mas, percebe-se que, no geral, a variação sem os atributos apresentou a melhor combinação <médias dos resultados %B, desvio padrão, médias dos riscos efetivamente identificados>. A grande semelhança nos resultados para o modelo original e o modelo sem os atributos nos faz chegar à conclusão que, na implementação atual do *CBR Risk* com a Distância Euclidiana, os atributos “Tipo de Projeto” e “Plataforma Tecnológica” (que podem assumir os valores 0 e 1) não tem praticamente nenhuma influência no cálculo de similaridade e poderiam ser descartados.

Para ilustrar o que foi dito, abaixo segue os dois gráficos que representam o percentual de riscos identificados (atingiu) versus o percentual de riscos que poderiam ter sido identificados (alcançável) em cada um dos 18 projetos coletados, para o modelo original e para o modelo sem os atributos “Tipo de Projeto” e “Plataforma Tecnológica”.

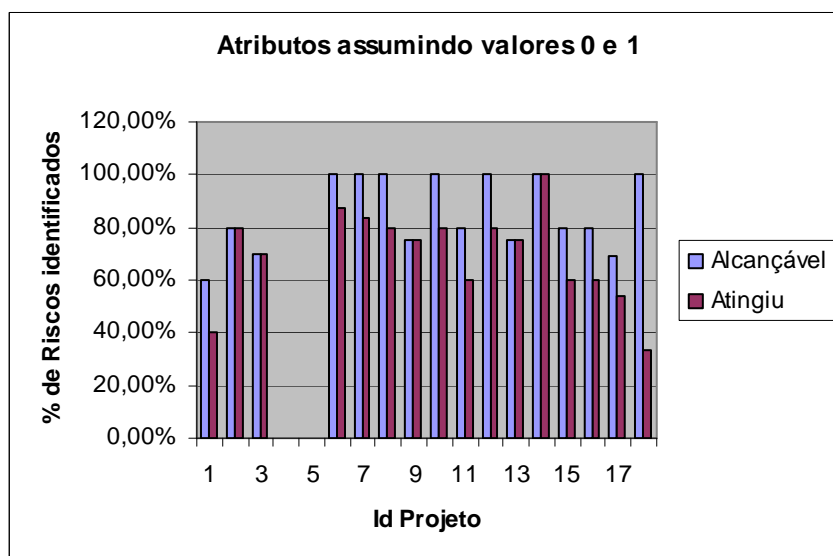


Figura 7. Percentual de Riscos Identificados Vs. Percentual de Riscos que poderiam ter sido identificados (Modelo original)

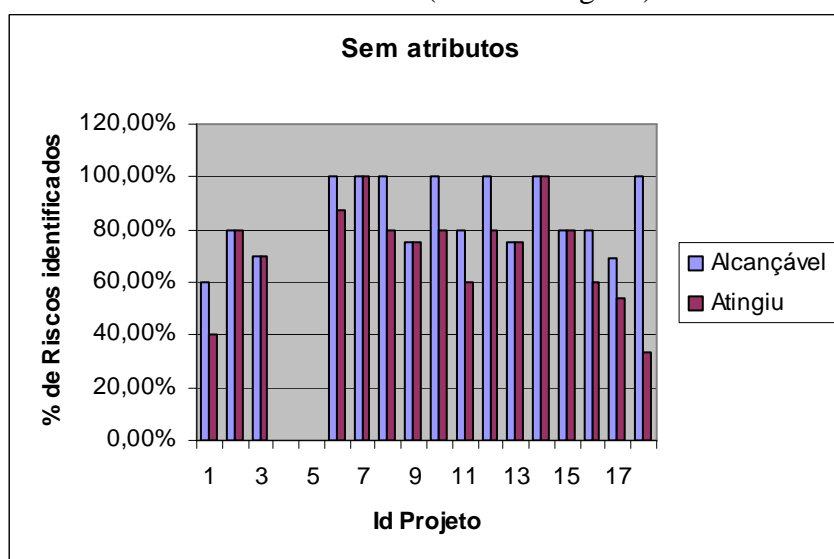


Figura 8. Percentual de Riscos Identificados Vs. Percentual de Riscos que poderiam ter sido identificados (Modelo sem atributos)

Podemos observar que os dois gráficos são quase idênticos, comprovando o que foi concluído anteriormente que os atributos “Tipo de Projeto” e “Plataforma Tecnológica” não têm influência relevante no cálculo de similaridade do *CBR Risk*.

Outra conclusão inferida da análise de resultados é que, visto o péssimo desempenho da variação em que os pesos dos atributos foram modificados, o filtro definido na ferramenta só é válido quando todos os atributos têm peso 1. Uma modificação nos pesos anula o filtro, já que tal filtro foi definido em cima de possíveis valores para a similaridade global [2] e uma mudança nos

pesos dos atributos altera esse intervalo de possíveis valores. Isso explica o desempenho bem abaixo em relação às demais variações.

5.4 Resumo do Capítulo

A análise de resultados dos experimentos é apresentada neste capítulo. A estrutura de análise segue o seguinte modelo:

- Médias dos Resultados %B;
- Desvio padrão dos resultados %B;
- Médias dos riscos que foram efetivamente identificados.

Através da combinação dos resultados expostos acima, é feito um estudo comparativo entre as diversas variações nos atributos “Tipo de Projeto” e “Plataforma Tecnológica”. Com esse estudo é inferida algumas conclusões, como a de que os atributos sob estudo, da maneira que estão implementados na ferramenta, não possuem influência relevante no cálculo de similaridade do *CBR Risk*. Outra conclusão é quanto a anulação do filtro quando os pesos dos atributos são modificados.

Capítulo 6

Conclusões e Trabalhos Futuros

Este trabalho executou as mudanças na implementação do *CBR Risk* propostas em Trigo [2]. Essas mudanças consistem numa nova fórmula para o cálculo de similaridade baseado em Distância Euclidiana e de um filtro, garantindo assim que projetos com grau de similaridade muito pequeno não sejam sugeridos ao usuário.

Além disso, foi feita, através de experimentos, a avaliação dos atributos de projetos “Tipo de Projeto” e “Plataforma Tecnológica”. Tais atributos possuem características peculiares para o cálculo de similaridade no *CBR Risk*.

Neste capítulo são tratadas as contribuições alcançadas por este trabalho (seção 6.1) e algumas sugestões de trabalhos futuros (seção 6.2).

6.1 Contribuições

Com a realização de experimentos por Thiago Trigo [2] no modelo original do *CBR Risk*, foi possível detectar pontos de melhoria a serem implementados na ferramenta, com o objetivo de torná-la mais eficiente. O trabalho aqui proposto implementou tais melhorias e realizou testes na própria ferramenta para validá-las.

Com a avaliação dos atributos “Tipo de Projeto” e “Plataforma Tecnológica” através de experimentos, chegou-se à conclusão que tais atributos, da maneira que estão implementados na ferramenta, não têm influência relevante no cálculo de similaridade, podendo ser descartados ou até mesmo substituídos por outros atributos. Chegou-se também a constatação que, visto a grande quantidade de tipos de projetos e plataformas tecnológicas existentes, não é coerente definir apenas cinco valores para tais atributos para torná-los compatíveis com os outros atributos.

A análise de resultados mostrou, também, que o filtro implementado na ferramenta só funciona quando os pesos dos atributos têm valor 1. Qualquer modificação nesses pesos anula o filtro.

Este trabalho produziu, até o momento, dois artigos:

- CBR Risk: uma ferramenta para identificação automática de riscos em projetos de software;
- Estudo sobre Similaridade de Projetos de Software.

Tais artigos aguardam o resultado da submissão no SBES (Simpósio Brasileiro de Engenharia de Software).

6.2 Trabalhos Futuros

Os experimentos realizados neste trabalho abriram novas perspectivas para trabalhos futuros a serem executados com o objetivo de melhorar ainda mais a ferramenta. Alguns destes trabalhos são:

- Definir uma nova forma de representação dos atributos “Tipo de Projeto” e “Plataforma Tecnológica”, com o objetivo de tornar esses atributos mais influentes no cálculo de similaridade;
- Visto a pouca influência dos atributos “Tipo de Projeto” e “Plataforma Tecnológica” comprovada pelos experimentos, substituir tais atributos por outros em que seja possível definir intervalos de valores compatíveis com os demais atributos;
- No cálculo de similaridade, adicionar um recurso em que informações podem ser requisitadas ao usuário a fim de selecionar o melhor projeto dentre um grupo de projetos considerados similares;
- Realizar estudos em ambientes reais de desenvolvimento de projetos através do uso da ferramenta *CBR Risk*, coletando dados reais e analisando as respostas obtidas;
- Aplicação da ferramenta *CBR Risk* em outras áreas de conhecimento através da adaptação da mesma;
- Estudos de ferramentas similares de gerenciamento de riscos para comparação com o *CBR Risk*.

Bibliografia

- [1] PMBOK Um Guia do Conjunto de Conhecimentos em Gerenciamento de Projetos (Guia PMBOK®) Terceira edição 2004 Project Management Institute. 2004.
- [2] Trigo, T. R.. Avaliando um Modelo de Identificação de Projetos de Software Similares: Estudo de Caso – Trabalho de conclusão de curso, Universidade de Pernambuco, 2007.
- [3] Lins, A. V.. Um Modelo para Identificação de Riscos de Projeto Utilizando Raciocínio Baseado em Casos – Trabalho de conclusão de curso, Universidade de Pernambuco, 2007.
- [4] Travassos, G. H., Gurov, D., Amaral, E. A. G.. Introdução à Engenharia de Software Experimental – Relatório Técnico, Universidade do Rio de Janeiro, 2002.
- [5] Branco Jr, E. C. e Belchior, A.D. (2001) Processos Gerenciais de Projetos de Software: Uma Abordagem Qualitativa. In: WQS'2001 – Workshop de Qualidade de Software. Trabalhos apresentados. pp 61- 72.
- [6] Coelho, P. G. (2004) Identificação das Estratégias de Aprendizado utilizadas pelos PMP's e Aspirantes a Certificação PMP. Projeto PMK – Environment Learning. CIn /UFPE – Centro de Informática – Universidade Federal de Pernambuco.
- [7] Leopoldino, C. B. (2004) Avaliação de Riscos em Desenvolvimento de Software. Dissertação de Mestrado. Universidade Federal do Rio Grande do Sul – Escola de Administração. Porto Alegre. Brasil.
- [8] Pauperio, L. F. S.. Proposta de um Modelo para a Gestão de Risco – Trabalho de conclusão de curso I, Centro Universitário Feevale, Novo Hamburgo, 2007.
- [9] Wangenheim, C. G e Wangenheim, A.. Raciocínio Baseado em Casos. Ed. Manole Ltda, São Paulo, Brasil. 2003.
- [10] da Costa, M. T. C.. Uma Arquitetura Baseada em Agentes para Suporte ao Ensino à Distância – Tese de Doutorado, Universidade Federal de Santa Catarina, Florianópolis, 1999.
- [11] Lee, R. W.. Pesquisa Jurisprudencial Inteligente – Tese de Doutorado, Universidade Federal de Santa Catarina, Florianópolis, 1998.
- [12] Niederauer, J.. Desenvolvendo Websites com PHP. Novatec Editora Ltda, São Paulo, Brasil. 2004.