



UNIVERSIDADE ESTÁCIO DE SÁ

ALVARO BATISTA AMARAL DA PURIFICAÇÃO

**USO DA TEORIA DOS CONJUNTOS *FUZZY* PARA ANÁLISE DO RISCO NO
DESENVOLVIMENTO DE *SOFTWARE***

Rio de Janeiro

2008

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

Dados Internacionais de Catalogação na Publicação (CIP)

P985

Purificação, Alvaro Batista Amaral da
Uso da teoria dos conjuntos Fuzzy para análise do risco no desenvolvimento de software. / Alvaro Batista Amaral da Purificação.- Rio de Janeiro, 2008.

114 f.

Dissertação (Mestrado em Administração e Desenvolvimento Empresarial) –
Universidade Estácio de Sá, 2008.

1. Risco (Economia). 2. Gerência de riscos. 3. Lógica fuzzy. 4. Software, Desenvolvimento. I. Título.

CDD 332.6

ALVARO BATISTA AMARAL DA PURIFICAÇÃO

**USO DA TEORIA DOS CONJUNTOS *FUZZY* PARA ANÁLISE DO RISCO NO
DESENVOLVIMENTO DE *SOFTWARE***

Dissertação apresentada à Universidade Estácio de Sá, como parte dos requisitos para a obtenção do título de Mestre em Administração.
Orientador: Prof. Jesús Domech Moré, D.Sc.

Rio de Janeiro

2008



UNIVERSIDADE
Estácio de Sá

PROGRAMA DE PÓS-GRADUAÇÃO EM
ADMINISTRAÇÃO E DESENVOLVIMENTO EMPRESARIAL

A dissertação

**USO DA TEORIA DOS CONJUNTOS FUZZY PARA ANÁLISE
DO RISCO NO DESENVOLVIMENTO DE SOFTWARE**

elaborada por

ALVARO BATISTA AMARAL DA PURIFICAÇÃO

e aprovada por todos os membros da Banca Examinadora foi aceita pelo Curso de Mestrado Profissional em Administração e Desenvolvimento Empresarial como requisito parcial à obtenção do título de

MESTRE EM ADMINISTRAÇÃO E DESENVOLVIMENTO EMPRESARIAL

Rio de Janeiro, 12 de dezembro de 2008.

BANCA EXAMINADORA

Prof. Dr. Jesús Domech Moré

Presidente

Universidade Estácio de Sá

Prof. Dr. Antônio Augusto Gonçalves

Universidade Estácio de Sá

Prof. Dr. Carlos Alberto Nunes Cozenza

Universidade Federal do Rio de Janeiro

Dedico este meu trabalho e o título de Mestre aos Deuses que me fizeram do pó e aos poucos que me amam e me ajudam a crescer e me transformar em uma rocha.

AGRADECIMENTOS

Ao meu pai (*in memoriam*) e a minha mãe, que com simplicidade, ensinaram-me que “o mais importante não é ter tudo que se quer, mas sim, querer bem tudo que se tem”.

A minha também mãe Maria José pelo carinho e interesse durante este curso.

Ao meu amigo, irmão e compadre Guilherme pelo incentivo e pela alegria contagiante que demonstrava a cada conquista neste curso.

Ao meu filho Guilherme pela compreensão durante a minha ausência, consequência da jornada dura dos trabalhos nos fins de semana na empresa e dos deveres deste curso.

Ao professor Dr. Jesús Domech Moré pela dedicação e incentivo com que orientou este trabalho, e também os dois artigos que escrevemos para diferentes congressos.

Ao colega de mestrado e hoje mestre em Administração, César Ribeiro de Araújo, funcionário da DATAPREV, pela fundamental colaboração no levantamento dos dados deste trabalho.

A Ana Lúcia da secretaria do MADE pelo apoio administrativo, atenção e carinho.

Aos colegas do Mestrado pelo companheirismo e aos professores do MADE pelos ensinamentos dados durante estes dois anos.

Ao colega de Mestrado e hoje mestre em Administração Júlio Martins e ao colega de trabalho Jonas Almeida pela ajuda na formatação deste trabalho.

A Cássia pela dedicação, incentivo e carinho durante esta jornada.

Aos amigos que me acompanham e me guardam nesta caminhada em busca do conhecimento.

RESUMO

PURIFICAÇÃO, A. B. A. **Uso da teoria dos conjuntos *fuzzy* para a análise do risco no desenvolvimento de *software***. 114 fls. Dissertação (Mestrado). Universidade Estácio de Sá, 2008.

O objetivo principal desta dissertação é tratar os riscos associados ao desenvolvimento de *software* na DATAPREV, através do uso da teoria dos conjuntos *fuzzy*. Segundo os estudos acadêmicos, o desenvolvimento de *software* pode ser considerado uma atividade de risco, e como tal, exige um gerenciamento voltado para um ambiente incerto, complexo e dinâmico. Um mundo que envolve percepções, como é o mundo da avaliação de riscos de um projeto, não apresenta fronteiras nítidas e bem definidas, e é repleto de ambigüidades e incertezas. Sendo assim, é razoável a idéia de que, ao se utilizar os conjuntos clássicos na formulação e modelagem de problemas de risco, criam-se fronteiras arbitrárias que se tornam, na verdade, zonas sobre as quais reinam os conflitos. Devido a esta natureza incerta e subjetiva, podem ocorrer divergências entre os profissionais no desenvolvimento de *software*, ambiente propício para a utilização da teoria dos conjuntos *fuzzy*. Esta teoria foi utilizada porque é a ferramenta indicada para dar forma matemática a expressões lingüísticas, qualitativas, derivadas dos fatores que influenciam no risco durante o desenvolvimento do *software*, e transformá-las em formato numérico, quantitativo, fundamental para a tomada de decisão destes gerentes desenvolvedores de *software*. A metodologia deste trabalho tem o objetivo de permitir ao gerente do projeto identificar quais os fatores de risco, aumentando as possibilidades de sucesso no desenvolvimento de *software*.

Palavras-chave: Risco, gerência de risco, conjuntos *fuzzy*, desenvolvimento de *software*.

ABSTRACT

PURIFICAÇÃO, A. B. A. **Uso da teoria dos conjuntos *fuzzy* para a análise do risco no desenvolvimento de *software***. 114 fls. Dissertação (Mestrado). Universidade Estácio de Sá, 2008.

The main objective of this dissertation is to deal with risks related to software development in DATAPREV, through the use of fuzzy sets theory. According to academic analysis, software development can be deemed a risk activity and, thereby, demands management directed to an uncertain, complex and dynamic environment. A world involving perceptions, which is the case in a project risk evaluation, does not present clear and sharp boundaries, being full of ambiguity and uncertainty. So, it is reasonable to think that, when using classic sets in risk problem formulation and modeling, arbitrary boundaries are raised and they become, indeed, zones over which conflicts rule. Owing to this uncertain and subjective nature, there may be divergences among those who are skilled in software development, making this environment propitious for the use of fuzzy set theory. This theory was used because it is the indicated tool for imparting mathematical form to linguistic and qualitative expressions that originate from factors that affect risk during software development, giving them numeric and quantitative format. This is essential for these software managers to make decisions. The methodology herein proposed intends that the project manager may be able to identify the risk factors, thus, increasing the success degree in software development.

Keywords: Risk, risk management, fuzzy set theory, software development.

LISTA DE FIGURAS

Figura 1 – Relação entre os temas do referencial teórico, autores e tópicos.....	24
Figura 2 – Contexto do desenvolvimento de um projeto.....	30
Figura 3 – As fases de um projeto	31
Figura 4 – Modelo de Identificação de Riscos	74
Figura 5 – Matriz de risco	81
Figura 6 – Riscos Inaceitáveis (Risco alto)	84
Figura 7 – Contribuição dos fatores por categoria à classificação do Risco alto..	86
Figura 8 – Composição <i>max-min</i> da influência do fator de risco 2 sobre o fator de risco 5.....	89

LISTA DE TABELAS

Tabela 1 – Avaliações obtidas na categoria “organização”	75
Tabela 2 – Normalização dos resultados	76
Tabela 3 – Cálculo da Cardinalidade, Nebulosidade, Incerteza e Qualidade da Informação	78
Tabela 4 – Identificação de riscos inaceitáveis (Risco alto).....	85
Tabela 5 – Contribuição dos fatores por categoria à classificação de risco inaceitável (Risco alto).....	86
Tabela 6 – Variáveis lingüísticas usadas na matriz de incidências	87
Tabela 7 – Matriz R de primeira geração	88
Tabela 8 – Matriz R_2 de segunda geração.....	90
Tabela 9 – Matriz $ R_2 - R $	90
Tabela 10 – Matriz $ R_2 - R + R$	91

LISTA DE ABREVIATURAS E SIGLAS

FMEA	<i>Failure Modes and Effective Analysis</i>
FTA	<i>Fault Tree Analysis</i>
INSS	Instituto Nacional do Seguro Social
ISO	<i>Internacional Organization for Standardization</i>
MPS	Ministério da Previdência Social
NASA	<i>National Aeronautics and Space Administration</i>
NBR ISO	Norma Brasileira ISO
PMI	<i>Project Management Institute</i>
PMBOK	<i>Project Management Body of Knowledge</i>
PSR	Probabilidade, Severidade do Risco
RUP	<i>Rational Unified Process</i>
SDLC	<i>Software Development Life Cycle</i>
SEI	<i>Software Engineering Institute</i>
SERIM	<i>Software Engineering Risk Model</i>
SI	Sistema de Informação
TI	Tecnologia da Informação
VME	Valor Monetário Esperado
VVT	Verificação, Validação e Teste

SUMÁRIO

1	INTRODUÇÃO	15
1.1	CONTEXTO	15
1.1.1.	Problema da pesquisa	21
1.1.2	Hipóteses	21
1.2	OBJETIVOS	21
1.2.1	Objetivo principal	21
1.2.2	Objetivos específicos	21
1.3	RELEVÂNCIA DO ESTUDO	21
1.3.1	Para a academia	22
1.3.2	Para a sociedade	22
1.3.3	Pessoal	23
1.4	LIMITAÇÕES DO ESTUDO	23
1.4.1	Limitação de tempo	23
1.4.2	Limitação geográfica	23
1.4.3	Limitação do método	23
2	REFERENCIAL TEÓRICO	24
2.1	TECNOLOGIA DA INFORMAÇÃO	25
2.1.1	Informação	25
2.2	DESENVOLVIMENTO DE SISTEMAS	28
2.2.1	Desenvolvimento de Sistemas	28
2.3	RISCO	34
2.3.1	Histórico	35
2.3.2	Probabilidade	37

2.3.3	Seguro, uma reação ao risco	40
2.3.4	Risco	41
2.4	GERENCIAMENTO DE RISCOS.....	44
2.5	RISCOS NO DESENVOLVIMENTO DE <i>SOFTWARE</i>	49
2.6	MODELOS QUALITATIVOS DE ANÁLISE DE RISCO	52
2.6.1	Modelo MOYNIHAN	53
2.6.2	Modelo KAROLAK	54
2.6.3	Modelo BRisk	57
2.7	TEORIA DOS CONJUNTOS <i>FUZZY</i>	58
2.7.1	Histórico	58
2.7.2	Teoria	59
2.7.3	Operações básicas	61
2.7.4	Aplicações da teoria dos conjuntos <i>fuzzy</i> no desenvolvimento de <i>software</i>	63
3	METODOLOGIA.....	67
3.1	ABORDAGEM	67
3.2	TIPOLOGIA DA PESQUISA.....	67
3.3	TIPO DE ARGUMENTO	68
3.4	AMOSTRA.....	69
3.4.1	Local da realização da pesquisa	69
3.4.2	Localização da empresa.....	70
3.4.3	DATAPREV	70
3.5	MÉTODO DE COLETA DE DADOS.....	70
3.5.1	Questionário.....	71

3.5.2	Apresentação do questionário	71
3.5.3	Determinação das variáveis lingüísticas e dos conjuntos <i>Fuzzy</i>	73
3.5.4	Descrição do Modelo de Identificação de Riscos	73
3.5.5	Identificação dos riscos	74
3.6	ANÁLISE QUALITATIVA DE RISCOS.....	80
3.6.1	Classificação dos riscos	80
3.6.2	Análise qualitativa de riscos.....	81
3.7	ANÁLISE QUANTITATIVA DE RISCOS	83
3.7.1	Identificação dos riscos altos.....	84
3.7.2	Análise quantitativa de riscos inaceitáveis (Risco alto).....	85
3.8	ESTUDO DA INFLUÊNCIA ENTRE OS FATORES DE RISCO ALTO	87
3.8.1	Matriz de influências.....	87
3.8.2	Influência entre os fatores de alto risco	88
4.	ANÁLISE DOS RESULTADOS	92
5.	CONCLUSÕES	94
6.	RECOMENDAÇÕES PARA FUTUROS ESTUDOS	95
	REFERÊNCIAS BIBLIOGRÁFICAS.....	96
	SITES CONSULTADOS	101
	APÊNDICE A – Questionário	102
	APÊNDICE B – Matriz de influência entre os fatores de risco alto	112
	APÊNDICE C – Matriz de influência modificada entre os fatores de risco alto	115

1 INTRODUÇÃO

1.1 CONTEXTO

No mundo atual, vivemos a “sociedade da informação”, onde estamos expostos a aceleradas e inúmeras mudanças no ambiente de TI, e por esta razão, um gerenciamento eficaz do projeto torna-se fundamental para o sucesso no desenvolvimento de *software*.

Para Davis e Meyer (1999, p. 6), “a comunicação e a computação quase instantâneas, por exemplo, estão reduzindo o tempo e nos concentrando no aspecto da velocidade”. Segundo estes autores, “a conectividade está colocando todo o mundo *online* de uma forma ou de outra, e tem provocado a “morte da distância”, um encolhimento do espaço”.

Para Benamarti e Lederer (1999), “a evolução da sociedade do conhecimento tem aumentado a importância do papel da tecnologia da informação”, e “os desafios do gerenciamento no futuro vão provavelmente continuar a crescer devido a uma alta taxa de mudanças”. Estes autores admitem que “TI é crítico para o sucesso das organizações atuais”, e consideram impossível a definição de qualquer estratégia sem o suporte desta área.

Davenport (1994, p. 6) entende que processo, “é simplesmente um conjunto de atividades estruturadas e medidas, destinadas a resultar num produto especificado para um determinado cliente ou mercado”. Para este autor, “um processo é, portanto, uma ordenação específica das atividades de trabalho no tempo e no espaço, com um começo, um fim, *inputs* e *outputs* claramente identificados: uma estrutura em ação”.

O desenvolvimento de um *software* é um processo com objetivos estabelecidos para atender as especificações do cliente, dentro de parâmetros de escopo, custo, tempo, qualidade, risco, suprimentos, recursos humanos e comunicação.

Segundo Schmitz, Alencar e Villar (2006, p. 2), o PMI (*Project Management Institute*) define projeto como “um empreendimento temporário almejando a criação de um produto ou serviço único”.

Para Vargas (2005, p. 13), as principais características dos projetos são a temporariedade e a individualidade do produto ou serviço a ser desenvolvido pelo projeto.

A temporariedade significa que todo o projeto possui um início e um fim definidos, ou seja, é um evento com duração finita determinada pelo seu objetivo.

Este autor define individualidade do produto ou serviço produzido pelo projeto, conforme o guia de conhecimento de gerenciamento de projetos do PMI, como realizar algo que não tenha sido realizado antes. Como o produto de cada projeto é único, suas características devem ser elaboradas de maneira progressiva, de modo a garantir as especificações do produto ou serviço a ser desenvolvido.

A partir destas características principais, Vargas (2005, p. 13) descreve as demais:

- Empreendimento não repetitivo: É um evento que não faz parte da rotina da empresa, algo novo que as pessoas irão realizar.
- Seqüência lógica de eventos: O projeto é caracterizado por atividades encadeadas logicamente de modo a permitir que, durante a execução, o acompanhamento e o controle sejam precisos.

- Início, meio e fim: Todo o projeto respeita um determinado ciclo de vida, uma característica temporal.
- Objetivo claro e definido: Todo o projeto tem metas e resultados bem estabelecidos a serem atingidos em sua finalização.
- Projetos utilizam recursos especificamente alocados a determinados trabalhos.
- Parâmetros predefinidos: Todo projeto necessita estabelecer valores para prazos, custos, pessoal, material e equipamentos envolvidos, apesar da impossibilidade de estabelecê-los com precisão. Os parâmetros iniciais atuam como referências para o projeto, e todos serão claramente identificados e quantificados no decorrer do plano deste projeto.

A aplicação de métodos quantitativos e técnicas de engenharia no desenvolvimento de *software* se faz necessário, para atender a estes parâmetros estabelecidos e gerenciar de forma adequada todo o projeto.

Para Borges e Falbo (2001, p. 1), a demanda no desenvolvimento de *software* vem aumentando, e devido a este momento, qualidade e produtividade são cada vez mais considerados fatores críticos de sucesso. Segundo estes autores, esta situação vem motivando a comunidade de engenharia de *software* a desenvolver pesquisas buscando garantir a qualidade de *software*, mantendo ou até mesmo aumentando a produtividade no desenvolvimento e diminuindo o tempo de resposta ao mercado. Estas pesquisas têm indicado que a qualidade do produto de software é fortemente dependente da qualidade de seu processo de desenvolvimento.

Para Espinha e Souza (2007, p. 1):

“um dos fatores importantes para a construção de um *software* de qualidade é o processo de desenvolvimento utilizado e como este é implantado na organização. A inexistência ou a não utilização de processos bem definidos e de boas práticas de

desenvolvimento, mesmo que informais, faz com que o desenvolvimento de *software* seja realizado de forma *ad-hoc*, ficando altamente dependente da experiência e do conhecimento das pessoas envolvidas. Este cenário resulta na realização de projetos cujos resultados são imprevisíveis, onde cada um realiza as suas atividades da forma que lhe convém, e dificulta a reutilização de boas práticas e de lições aprendidas.”

As sociedades de gerenciamento de projetos sediadas nos países desenvolvidos, dedicam espaço significativo à discussão e divulgação do gerenciamento de riscos em projetos, devido principalmente ao reconhecimento profissional e ao da academia.

Diante deste fato, podemos entender a importância do gerenciamento de projetos que não pode ser desconhecida pelos profissionais e acadêmicos que estejam envolvidos em qualquer atividade sujeita à subjetividade e incertezas.

A natureza incerta e subjetiva do desenvolvimento de *software* pode tornar-se uma importante fonte de inconsistências que precisa ser gerenciada, pois ocasiona divergências entre os profissionais do projeto.

Algumas das possíveis causas de inconsistências entre múltiplos desenvolvedores são: percepções diferenciadas de cada participante, linguagens diferentes usadas na comunicação, métodos de desenvolvimento e abordagens diferentes do estágio de desenvolvimento, áreas de interesse comuns, parcialmente comuns ou incomuns e, objetivos econômicos, políticos ou técnicos diferentes, entre outros.

O PMBOK (*Project Management Body of Knowledge*) é um modelo de referência, uma estrutura de conhecimento que organiza conceitos, práticas e padrões da área de gestão de projetos em geral, cobrindo assuntos como integração, escopo, tempo, custos, qualidade, recursos humanos, comunicações, aquisições e risco.

Segundo o guia PMBOK (2004), risco é definido como um evento ou condição incerta que, se ocorrer, provocará um efeito positivo ou negativo nos objetivos do projeto.

Segundo Vargas (2005, p. 93), os riscos associados aos grandes empreendimentos têm merecido uma atenção especial dos gerentes de projeto, devido, não só às grandes somas em dinheiro que estão em suas mãos, como também à reputação do time e dos patrocinadores desse projeto.

Com o objetivo de aproximar a engenharia de *software* e a economia, Costa, Barros e Travassos (2004, p. 3) comparam projetos de *software* a uma carteira de empréstimos em longo prazo. Segundo ele, instituições financeiras que fornecem empréstimos têm interesse pela probabilidade de perder ou ganhar, como nos projetos de *software* que também possuem probabilidades incertas de sucesso. Observa-se que os projetos de *software* também possuem um custo, que é o capital empenhado em seu desenvolvimento, como um montante financeiro concedido em um empréstimo.

Partindo-se destas premissas, podemos concluir que qualquer risco à qualidade e à institucionalização do processo, pode refletir em um aumento significativo no valor total do financiamento deste projeto.

Espinha e Souza (2007, p. 1) entendem que a qualidade do *software* está diretamente relacionada à qualidade dos processos utilizados na sua produção e ao conhecimento técnico que os usuários deste processo têm sobre as práticas institucionalizadas do processo.

A Gerência de Riscos em Engenharia de *Software* tem como objetivo aumentar a qualidade do *software* e de seu processo de desenvolvimento. Observa-se que os

projetos de desenvolvimento de *software*, em geral, apresentam atrasos de cronograma, custos além do planejado e funcionalidade aquém das expectativas.

O gerenciamento de riscos é o processo de identificação, análise, desenvolvimento de respostas e monitoramento dos riscos em projetos, com o objetivo de diminuir a probabilidade e o impacto de eventos negativos, e de aumentar a probabilidade e o impacto de eventos positivos. A determinação do risco pode ser qualitativa e quantitativa.

A questão central desta pesquisa é a de propor o uso deste método quantitativo para analisar e classificar os efeitos negativos e positivos que possam surgir durante o desenvolvimento de *software*.

A estrutura do trabalho foi dividida nos seguintes tópicos:

1. Introdução: apresenta o contexto do estudo onde o problema existe e suas hipóteses. Define os objetivos principal e intermediários, a sua relevância para o mundo acadêmico, para a sociedade e pessoal. Completa esta introdução apresentando as limitações de tempo, geográfica e do método usado;
2. Referencial teórico: revisão sobre a Tecnologia da informação e Desenvolvimento de sistemas, do conceito de risco, de gerenciamento de riscos, de riscos no Desenvolvimento de *software*, dos modelos qualitativos de análise de risco e da teoria dos conjuntos *fuzzy*;
3. Metodologia: abordagem, tipologia da pesquisa, método de abordagem, amostra, método de coleta de dados. Apresenta também a análise qualitativa e a análise quantitativa de riscos, a matriz de influência e a influência entre os fatores de Risco;

A teoria dos conjuntos *fuzzy* criada por Zadeh (1965) permite dar forma matemática às expressões próprias da linguagem natural, sem diminuir a potência

expressiva das mesmas. Ela é capaz de captar essas informações qualitativas, levá-las para um formato numérico e obter respostas úteis para a tomada de decisões da empresa, enquanto que, ao compararmos com outras metodologias atuais, estas apresentam lacunas.

1.1.1. Problema da pesquisa

Quais os riscos inerentes ao processo de desenvolvimento de *software* na empresa DATAPREV.

1.1.2 Hipóteses

Com a utilização da teoria dos conjuntos *fuzzy*, seremos capazes de identificar os riscos associados ao desenvolvimento de *software*.

1.2 OBJETIVOS

1.2.1 Objetivo principal

O objetivo desta dissertação consiste em:

Aplicar o método *fuzzy* para a identificação do risco no desenvolvimento de *software*.

1.2.2 Objetivos específicos

Para atingir o objetivo principal desta dissertação, necessitamos:

- a) Estudar os modelos qualitativos existentes de identificação de risco no desenvolvimento de *software*;
- b) Identificar quais os fatores influenciadores no risco mais importantes para os profissionais desenvolvedores de *software*;
- c) Utilizar a matriz de relação *fuzzy* na definição do risco de maior contribuição.

1.3 RELEVÂNCIA DO ESTUDO

1.3.1 Para a academia

Divulgar para o meio acadêmico o quanto é poderosa a teoria dos conjuntos *fuzzy*, ao captar a subjetividade do conhecimento do especialista e transformá-la em formato numérico com qualidade.

1.3.2 Para a sociedade

No referencial teórico abaixo, fica clara a necessidade do uso de ferramentas estruturadas para identificar o risco, onde, segundo as afirmações, por exemplo, de Reis e Albuquerque (2004, p. 1), menos da metade dos projetos dos setores públicos e privados são bem sucedidos.

A vasta literatura que procura determinar as razões deste elevado número de insucessos aponta para algumas das causas principais: as estimativas mal elaboradas baseadas em “experiência pessoal”, o projeto ter vários responsáveis e os objetivos mudarem com o andamento do projeto.

Em 2002 foi publicado um artigo por White e Fortune (2002), baseado numa pesquisa com gerentes de projetos, representantes dos setores público e privado, e tornou patente que 41% dos projetos foram bem sucedidos. Outro fato relevante divulgado por esta pesquisa mostra que apenas 35% dos projetos utilizaram alguma ferramenta de gerenciamento de riscos e que 46% dos projetos apresentaram efeitos colaterais inesperados. Diante deste fato podemos concluir que, é maior a chance de acertar “cara ou coroa” ao lançar uma moeda (50% de chance), do que ter um projeto bem sucedido que não produza efeitos imprevistos.

Como a DATAPREV não possui uma cultura voltada para o gerenciamento do risco, a relevância baseia-se no fato deste estudo apresentar uma metodologia para

identificar os riscos associados ao desenvolvimento de *software* e assim motivar o interesse dos seus profissionais para a criação desta cultura.

1.3.3 Pessoal

O Risco sempre esteve, implicitamente, presente na minha vida profissional, principalmente quando utilizava as ferramentas de qualidade FMEA (*Failure Modes and Effective Analysis*), FTA (*Fault Tree Analysis*), para gerenciar projetos de engenharia e na gestão de processos e produção.

A oportunidade de conhecer e estudar a teoria dos conjuntos *fuzzy* vem a complementar esta formação, pois me capacita a avaliar com qualidade as opiniões dos especialistas, para tratá-las de maneira estruturada com o uso destas ferramentas da qualidade.

1.4 LIMITAÇÕES DO ESTUDO

1.4.1 Limitação de tempo

O estudo será desenvolvido durante o curto período definido pelo curso de mestrado, de 2007 a 2008.

1.4.2 Limitação geográfica

Realizar esta pesquisa em uma única empresa do setor público na cidade do Rio de Janeiro.

1.4.3 Limitação do método

Todo o método possui uma limitação que é dada pela representação aproximada que este faz da realidade.

2 REFERENCIAL TEÓRICO

O referencial teórico foi construído a partir de uma pesquisa bibliográfica, e a relação entre os temas, autores e tópicos é mostrada no quadro a seguir:

Referencial teórico	Principais Autores	Tópicos
TECNOLOGIA DA INFORMAÇÃO	Setzer McGee e Prusak Aquino Oliveira, Mota e Alvarado Silva Barreto Albertin e Albertin	Informação
DESENVOLVIMENTO DE SISTEMAS	Turban, Mclean e Wetherbe Câmara Campos	Desenvolvimento de Sistemas
RISCO	Berntein Salles Jr., Soler, Valle e Rabechini Jr. Boorstin Lopes e Meirelles Gadelha Espinha e Souza Vargas Costa, Barros e Travassos Andrade Alencar e Schmitz	Histórico Probabilidade Seguro, uma reação ao Risco Risco
GERENCIAMENTO DE RISCOS	Salles Jr., Soler, Valle e Rabeschine Jr. Reis e Albuquerque Schmitz, Alencar e Villar Espinha e Souza Vargas Kroll e Kruchten Gusmão e Moura Hall	Gerenciamento de Riscos
RISCOS NO DESENVOLVIMENTO DE <i>SOFTWARE</i>	Junior e Belchior Gusmão e Moura Falbo Fernandes Costa, Barros e Travassos Machado Nascimento	Riscos no Desenvolvimento de <i>Software</i>
MODELOS QUALITATIVOS DE ANÁLISE DE RISCO	Pinho e Mendonça Xu, Khoshgoftaar e Allen Schmitz, Alencar e Villar	Modelo MOYNIHAN Modelo KAROLAK Modelo Brisa
TEORIA DOS CONJUNTOS <i>FUZZY</i>	Schmitz, Alencar e Villar Silva Moré Braga, Barreto e Machado Zadeh Lima e Amorim Cunha Kuchta	Histórico Teoria Operações básicas
APLICAÇÕES DA TEORIA DOS CONJUNTOS <i>FUZZY</i> NO DESENVOLVIMENTO DE <i>SOFTWARE</i>	Lee Xu, Khoshgoftaar e Allen Liu, Kane e Bamboon Engel e Last Kiran e Ravi	Aplicações da teoria dos conjuntos <i>fuzzy</i> no desenvolvimento de <i>software</i>

Figura 1 – Relação entre os temas do referencial teórico, autores e tópicos

2.1 TECNOLOGIA DA INFORMAÇÃO

2.1.1 Informação

Para Setzer (1999, p. 2), *Informação* é uma abstração informal (isto é, não pode ser formalizada através de uma teoria lógica ou matemática), que representa algo significativo para alguém através de textos, imagens, sons ou animação. A representação da informação pode eventualmente ser feita por meio de dados, que podem ser armazenados no computador.

Setzer (1999, p. 2) entende que:

“essa representação pode ser transformada pela máquina - como na formatação de um texto - mas não o seu significado, já que este depende de quem está entrando em contato com a informação. Por outro lado, dados, desde que inteligíveis, são sempre incorporados por alguém como informação, porque os seres humanos (adultos) buscam constantemente por significação e entendimento. Uma distinção fundamental entre dado e informação é que o primeiro é puramente sintático e o segundo contém necessariamente *semântica* (implícita na palavra "significado" usada em sua caracterização). É interessante notar que é impossível introduzir semântica em um computador, porque a máquina mesma é puramente sintática (assim como a totalidade da matemática)”.

Setzer (1999, p. 1) definiu *dado* como uma seqüência de símbolos quantificados ou quantificáveis, para ele:

“..., um texto é um dado. De fato, as letras são símbolos quantificados, já que o alfabeto por si só constitui uma base numérica. Também são dados, imagens, sons e animação, pois todos podem ser quantificados a ponto de alguém que entra em contato com eles ter eventualmente dificuldade de distinguir a sua reprodução, a partir da representação quantificada, com o original. É muito importante notar-se que qualquer texto constitui um dado ou uma seqüência de dados, mesmo que ele seja ininteligível para o leitor.”

McGee e Prusak (1994, p. 108) entendem que informação:

“... representa uma classe particular entre os ativos da organização, sendo sua administração sujeita a desafios específicos. É fácil perceber que a informação possui característica que a tornam bastante diferente de outros ativos organizacionais,...”.

A palavra informática vem do francês “*informatique*”, vocábulo derivado do verbo *informer* (informar) em analogia a matemática e a eletrônica. Esta palavra pode ser

substituída, em qualquer contexto, por tecnologia de informação ou sistemas de informação, apesar de originalmente, a palavra informática indicar a ciência do tratamento de informações por meio de processamento de dados.

Com o estudo da matemática e o desenvolvimento dos computadores, novos ramos do conhecimento foram sendo identificados: ciência da informação, tecnologia da informação, ciência da computação, matemática computacional e a computação gráfica, todos originados da informática, ramo tecnológico que trata do processamento de dados num computador.

Aquino (2002) entende que a ciência da informação teve origem no fenômeno da “explosão da informação”, ligado ao renascimento científico, após a 2ª Guerra Mundial, e no esforço subsequente de controle bibliográfico e de tratamento da documentação implícita neste processo.

Para Oliveira, Mota e Alvarado (2004) a ciência da informação nasceu para resolver um grande problema relacionado à documentação e à recuperação da Informação que é reunir, organizar e tornar acessível o conhecimento cultural, científico e tecnológico produzido em todo o mundo.

Segundo Silva (2007, p. 1), há um aceitável consenso quanto ao período histórico do surgimento da ciência da informação, situado entre as duas guerras mundiais, mas discorda das delimitações, de marcos históricos pontuais. Entretanto este autor admite que há um espaço de tempo compreendido entre o final do século XIX e a primeira metade do século XX que foi importante para o despertar de novas áreas de conhecimento centradas na informação.

Um ponto importante salientado por Meadows (1991 apud Oliveira, 2004, p. 2) foi a intensidade com que o computador afetou a estrutura dentro da qual a ciência da

informação opera, que como a ciência da computação, têm a sua origem na esteira da revolução científica e técnica que se seguiu ao pós-guerra.

O impacto dos computadores e das telecomunicações no gerenciamento da informação foi tão grande, que hoje a ciência da informação e a tecnologia da informação estão freqüentemente juntas na discussão sobre o percurso desta área.

Segundo Barreto (2008):

“as novas tecnologias de informação e transferência da informação ficaram muito perto do computador, suas linguagens e sua programação. Quando falamos em novas tecnologias de informação pensamos de imediato no computador, na telecomunicação e na convergência da base tecnológica. Contudo, este instrumental da técnica, apesar de imprescindível, representa uma pequena parte da conquista da liberdade individual sobre a informação. São efêmeros *gadgets* que acompanham a infra-estrutura de uma nova plataforma tecnológica, conjuntos, mutantes formando as hipervias em *backbones* que distribuem enunciados. Estes instrumentais da tecnologia da informação se modificam a cada vã momento reaparecendo sempre em melhor forma e mais potente.”

Este autor entende que as reais modificações advindas das tecnologias de informação são as condições de interatividade e interconectividade do receptor em relação à informação. Estas transformações estabeleceram um novo relacionamento entre o gerador, o receptor; uma grande idéia que se chocou com o tempo certo, pois a velocidade e modalidade de acesso à informação modificam nossa sensibilidade e competência cognitiva. Acompanhamos mudanças que trouxeram profundas modificações com a inclusão definitiva da imagem e do som no dia a dia da estrutura da informação em áreas como a medicina com o diagnóstico à distância, a engenharia com os projetos em realidade virtual e na educação à distância, além de outras.

A TI no interior das organizações é fruto da visão de sua utilidade, pois no momento em que a tecnologia da informação passou a exercer papéis mais estratégicos, ocorreu uma maior união entre a gerência empresarial e a TI.

Albertin e Albertin (2005) mostram que dentro de uma organização o uso da TI não é uma aplicação pura, mas fruto das interações sociais, culturais, técnicas e econômicas. Esta nova visão da Tecnologia da informação vem proporcionando às organizações melhor aproveitamento global de recursos, maior integração de objetivos e, conseqüentemente, melhores resultados para a empresa que tem de se manter e se destacar no mercado, diante de seus clientes, fornecedores e competidores.

É estratégico que uma organização moderna, mude a forma de encarar a TI, a fim de que não seja mais vista como despesa, e sim como um recurso essencial que permite a expansão estratégica do papel da própria TI nos negócios desta empresa, seja através de melhorias, da reestruturação, ou de outros ganhos de produtividade nos seus processos.

No mundo globalizado a importância da Tecnologia da Informação nas organizações tornou-se cada vez mais relevante frente às necessidades impostas pela sociedade moderna. O papel das organizações nesta sociedade relaciona-se principalmente às rápidas mudanças da economia, que fizeram com que a informação ganhasse proporções onde o principal recurso da organização é a utilização e a administração deste fluxo de informações, e para isso, é fundamental um bom projeto de desenvolvimento de sistemas.

2.2 DESENVOLVIMENTO DE SISTEMAS

2.2.1 Desenvolvimento de Sistemas

Segundo Turban, Mclean e Wetherbe (2004, p. 625), desenvolvimento de sistemas é a estruturação do *hardware* e do *software* para obter o processamento eficiente e eficaz da informação. Para a execução dos passos de um projeto como este,

é necessário um planejamento detalhado, onde cada tarefa deve ser identificada e designado cada responsável ou grupo responsável pelo seu cumprimento, dentro ou fora da empresa.

Turban, Mclean e Wetherbe (2004, p. 296) apresentam o *conceito de ciclo de vida do desenvolvimento de sistemas, SDLC-Software Development Life Cycle*, como um conjunto de categorias gerais que mostram os principais passos, dentro de um determinado prazo, de um projeto de desenvolvimento de sistemas de informação.

É importante entender que não há uma versão padronizada, universal, de *SDLC*.

Algumas tarefas estão presentes na maioria dos projetos, enquanto que outras só se aplicam a determinados tipos, como projetos menores que normalmente não requerem tantas tarefas se comparados aos maiores.

Câmara (2008, p. 2) entende que ao separarmos o desenvolvimento de um projeto em fases distintas, porém complementares, teremos um ambiente de estudo mais adequado para tratar os problemas dos projetos. O *SDLC* se propõe a formalizar dentro de uma equipe de projetos fases de construção, processos e normas que facilitarão a previsibilidade e, principalmente, a produtividade que poderão ser medidas nos resultados desta equipe.

Segundo este autor, alguns profissionais da área gostam de denominar *SDLC* como um *workflow* de construção de *software*. É comum encontrarmos referências a *SDLC* como uma espécie de linha fabril de desenvolvimento, como uma linha de produção de uma indústria.

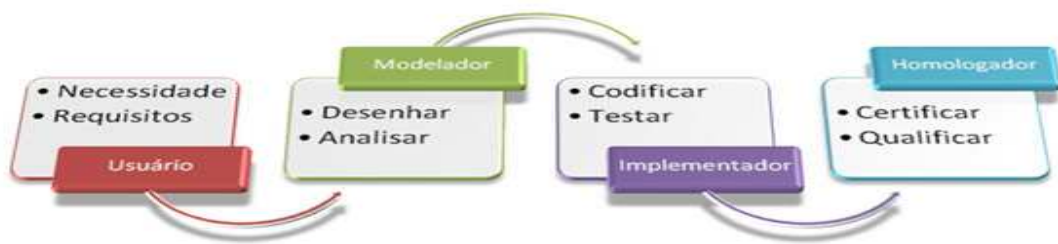


Figura 2 – Contexto do desenvolvimento de um projeto
Fonte: Câmara (2008)

Câmara (2008, p. 2) propõe que pensemos em um projeto, segmentando a idéia da forma representada na figura 1. Este contexto pode ser considerado bastante avançado quando buscamos por evidências práticas nas empresas que possuem equipes de desenvolvimento de *software*. Ao analisarmos os resultados dos estudos no "campo", verifica-se que na grande maioria dos casos o que se pratica é uma espécie de trabalho "oportunístico", ou seja, quando os problemas ocorrem todos se mobilizam em resolvê-los. A ausência de um modelo organizacional de trabalho proporciona dúvidas clássicas, como: este método de trabalhar é pouco produtivo, é reativo e proporciona aborrecimentos marcantes em usuários solicitantes e contratantes de serviços de *software*.

Segundo Câmara (2008, p. 3), fase é um conjunto de atividades afins necessárias e complementares, geralmente seqüenciais, para tratar uma determinada necessidade do conjunto de necessidades presentes na construção de um projeto de *software*. Estas fases são:

- Fase de início do projeto;
- Fase de planejamento do projeto;
- Fase de execução do projeto;

- Fase de fechamento do projeto.



Figura 3 – As fases de um projeto

Fonte: Câmara (2008)

Estes conjuntos de atividades necessárias à execução de um projeto, podem ser comparados ao método de gestão PDCA. A palavra método é a soma das palavras gregas *Meta* e *Hodos* (caminho), então o PDCA é o caminho para se atingir as metas, isto é, para realizar um projeto de *software*.

Segundo Campos (2001, p. 195), cada letra significa um momento deste método:

- O “P” representa *Plan*, fase da definição das metas e da determinação do método para alcançá-las;
- O “D” é a letra inicial de *Do*, momento de treinar e executar o trabalho;
- O “C” é a inicial de *Check*, onde verifica-se os efeitos do trabalho executado;
- O “A” origina-se da palavra *Action*, quando se verificam os efeitos do trabalho executado.

Turban, Mclean e Wetherbe (2004, p. 497) apresentam um modelo *SDLC* onde as principais tarefas foram agrupadas em oito categorias, são elas:

- Categoria 1: **O início do projeto.**

Alguém necessita iniciar um projeto para resolver um problema, ou como uma oportunidade de melhorar suas próprias operações.

b) Categoria 2: **Análise de sistemas e estudo de viabilidade.**

A **análise de sistemas** refere-se ao levantamento da situação atual e descreve o problema que se deseja que o sistema de informação solucione. Esta análise tem como objetivo proporcionar uma compreensão profunda da empresa, suas operações e da situação relevante para o sistema que se deseja construir.

O **estudo de viabilidade** é usado para determinar a probabilidade de êxito da solução proposta, podendo ser realizado várias vezes durante o ciclo de vida de desenvolvimento do sistema. As principais áreas de estudo de viabilidade são:

- **Tecnologia:** os níveis de desempenho podem ser alcançados com as atuais tecnologias de informação?
- **Economia:** os benefícios esperados serão maiores que os custos?
- **Fatores organizacionais:** o nível de capacitação necessária para usar o novo sistema é consistente com os funcionários que irão operá-lo?
- **Restrições legais, éticas e outras:** este processo é considerado ético pelos funcionários e clientes, atende às exigências legais?

Esta análise inicial precisa confirmar os resultados da primeira análise e desenvolver informações mais detalhadas sobre os custos e benefícios das análises financeiras, confirmando-se a sua viabilidade, e se ainda houver interesse, este projeto poderá passar para a fase seguinte.

c) Categoria 3: **Análise e projeto lógico.**

Nesta fase, dá-se ênfase ao projeto lógico, o desenho do Sistema de Informação do ponto de vista do usuário. O analista identifica as necessidades de informação e especifica processos e funções genéricas de SI, tais como entrada, saída e

armazenagem. A equipe de desenvolvimento precisa envolver os usuários no processo, pois além de ser uma importante fonte de informação, eleva o êxito na implantação.

Ao projeto lógico segue-se o projeto físico, que transforma o modelo lógico abstrato em um desenho técnico específico, a planta técnica do novo sistema.

d) Categoria 4: **Compra ou Desenvolvimento.**

O pessoal de sistemas de informação usa as especificações para comprar o *hardware* e o *software* necessários para o sistema e então os configura de acordo com as necessidades dos usuários. O pessoal de sistemas de informação testa o sistema, os usuários realizam alguns testes que identificam falhas e comparam o desempenho real do sistema com as especificações do projeto.

e) Categoria 5: **Implementação.**

Nesta fase a equipe deve planejar a implantação, minimizando os problemas que possam causar resistência junto aos usuários. Segundo Turban, Mclean e Wetherbe (2004, p. 499), na maioria das vezes a implantação de um novo sistema demanda uma conversão de um sistema já implantado. As abordagens para se fazer uma conversão incluem:

- **Conversão paralela:** os sistemas novo e antigo operam simultaneamente durante um período de testes, depois o antigo é desligado. Esta conversão é a mais segura e a mais cara.
- **Corte direto ou *big bang*:** o sistema antigo é desligado e o novo entra em operação. Esta conversão é a mais rápida, mais barata, mas também a mais arriscada.
- **Conversão piloto:** o novo sistema é implantado num número escolhido de pontos da empresa. A conversão piloto significa um corte direto ou *big bang*

para os pontos piloto, mas para a empresa como um todo é semelhante à conversão paralela. Os riscos e os custos são relativamente baixos.

- **Conversão modular:** os grandes sistemas são construídos em módulos distintos e cada um, se independente, pode ser convertido de cada vez. Esta conversão é mais segura que a direta, mas leva mais tempo e pode exigir mais testes, pois é necessário testar diversas partes do sistema toda vez que se implanta mais um módulo.

f) Categoria 6: **Operação.**

Após completar a implantação com êxito, o sistema pode operar por prazo indeterminado, até que não seja mais necessário.

g) Categoria 7: **Avaliação Pós-auditoria.**

Após a conclusão dos grandes projetos de sistemas, as empresas devem fazer uma avaliação, as pós-auditorias, que introduzem mais um elemento disciplinador no processo de desenvolvimento.

h) Categoria 8: **Manutenção.**

Cada sistema necessita de dois tipos regulares de manutenção: o conserto de *bugs* e atualização periódica. Essas correções e atualizações em geral não acrescentam nada à funcionalidade do sistema, sendo necessárias apenas para manter seu desempenho.

As categorias que compõem o *SDLC* estão sujeitas a riscos, isto é, problemas que surgem no desenvolvimento de *software* e devem ser tratados em cada uma destas categorias.

2.3 RISCO

2.3.1 Histórico

Bernstein (1997, p. 17) registra a mais antiga citação conhecida sobre risco, e a que mais se aproximou da sua quantificação, está contida no *Talmud*. Considerado como o livro da legislação judaica, o *Talmud*, palavra em hebraico que significa estudo e aprendizado, apresenta um raciocínio sobre a legitimidade de o homem separar-se da sua mulher por adultério, mas não se este ocorrer antes do casamento, e se foi o futuro marido o responsável. Diante desta dupla dúvida e das possibilidades, os filósofos concluíram que há apenas uma chance em quatro de que a mulher tenha cometido o adultério antes do casamento, logo se concluiu que o marido não podia divorciar-se dela por esta razão.

Salles Jr., Soler, Valle e Rabechini Jr. (2006, p. 20) entendem que:

“estabelecendo a ligação entre a medição e a previsão, temos também a contribuição da teoria das probabilidades. Os números são necessários para a administração dos riscos, que, sem eles, resume-se a um mero exercício de adivinhação. A história dos números no Ocidente remonta ao século XIII. O zero revolucionou o velho sistema de numeração, permitindo que se usassem apenas dez dígitos, de zero a nove, para realizar todos os cálculos imagináveis e para escrever qualquer número concebível’.

Para Bernstein (1997, p. 32), o ponto central do sistema indo-arábico foi a invenção do zero, chamado pelos indianos de *sunya*, pelos árabes de *cifr*. O termo chegou até nós como “cifra”, que significa vazio e se refere à coluna vazia do ábaco.

O conceito relacionado ao número zero era de difícil entendimento porque as pessoas usavam a contagem apenas para saber, por exemplo, a quantidade de dias decorridos, o número de animais, enfim, determinar alguma quantidade, e para a contagem deste número não fazia qualquer sentido. Segundo Bernstein (1997, p. 32), nas palavras do filósofo inglês do século XX, Alfred North Whitehead:

“o que caracteriza o zero é que não precisamos usá-lo nas operações do dia-a-dia. Ninguém sai à rua para comprar zero peixe. Em certo sentido, é o mais civilizado dos

cardinais, e seu uso é-nos forçado apenas pelas necessidades das formas cultas de pensamento”.

Segundo Salles Jr., Soler, Valle e Rabechini Jr. (2006, p. 21), sem o surgimento do zero não há probabilidade, e sem probabilidade não há como lidar com riscos. A palavra probabilidade deriva da conjunção de *probare*, que quer dizer testar, provar ou aprovar, e *ilis*, que significa capaz de ser. Após o surgimento dos números, os matemáticos foram capazes de calcular médias, seus desvios e fazerem medidas de amostragens, essenciais para se lidar com incertezas.

“a amostragem é essencial para se enfrentar riscos. Constantemente usamos amostras do presente e do passado para adivinhar o futuro.[...] O que é “normal”, de qualquer modo? Os estatísticos brincam sobre o homem com os pés no forno e a cabeça no refrigerador: na média, ele se sente muito bem”. (BERNSTEIN, 1997, p. 74)

Segundo estes autores, a amostragem é essencial quando desejamos trabalhar com a incerteza, mapeia de forma ordenada o existente.

Para Boorstin (1983, p. 598), o pioneiro da demografia moderna e também da estatística, foi um próspero comerciante, o londrino John Graunt (1620-1674). Este amador no mundo da matemática que se interessava pelo bem estar da sua comunidade ocupou vários cargos municipais, incluindo o de vereador.

A maior novidade apresentada pelo livro de Graunt é o uso da amostragem, a sua linha de análise é conhecida atualmente como inferência estatística, inferir uma estimativa global de uma amostra de dados. Segundo Bernstein (1997, p. 78), ele transformou o processo de uma simples coleta de informações em um instrumento poderoso e complexo de interpretação do mundo ao seu redor. O trabalho pioneiro de Graunt revelou os conceitos teóricos básicos necessários à tomada de decisões sob condições de incerteza.

“Amostragens, médias, e noções do que é normal compõem a estrutura que iria, mais à frente, abrigar a ciência da análise estatística, colocando a informação a

serviço da tomada de decisões e influenciando nossos graus de crença sobre as probabilidades de eventos futuros”. (BERNSTEIN, 1997, p. 83)

Segundo Salles Jr., Soler, Valle e Rabechini Jr. (2006, p. 20), os primeiros registros sobre riscos estão ligados à teoria das probabilidades, que foram desenvolvidas para a aplicação em jogos, notadamente de azar.

De acordo com Bernstein (1997, p. 12):

“Os soldados de Pôncio Pilatos sortearam o manto de cristo enquanto ele padecia na cruz. O imperador romano Marco Aurélio estava sempre acompanhado de seu *crupiê* pessoal. O conde de Sandwich inventou a refeição que tem o seu nome (sanduíche) para não precisar se afastar da mesa de jogo para comer. [...] O jogo de dados (*craps*), um invento norte-americano, origina-se de diferentes jogos de dados trazidos à Europa pelos cruzados. Esses jogos costumavam ser chamados de “azar”, de *al zahr*, a palavra árabe para dados”.

2.3.2 Probabilidade

A teoria da Probabilidade, para Lopes e Meirelles (2005, p. 1), surgiu como ramo da Matemática em meados do século XV, embora tenha se iniciado como ciência empírica muito antes, por volta de 3500 a.C. Suas raízes originam-se nos jogos e apostas. Bernstein (1997, p. 12) registra que pinturas de tumbas egípcias retratam jogos com o astrágalo, e vasos gregos mostram jovens atirando os ossos para o interior de um círculo. Este osso do calcanhar, ancestral dos dados atuais, retirado do tornozelo de carneiros e veados, era quadrado, duro e indestrutível, com dois lados estreitos e dois lados largos.

Este osso recebia valores para cada face, 4 e 3 para as faces largas, e 1 e 6 para as faces estreitas e menores. O astrágalo ao ser jogado podia cair sobre cada uma destas quatro faces, e baseado em experimentos, as probabilidades de ocorrência destes lados são:

$P(4)=0,39$, $P(3)=0,37$ e $P(1)=P(6)=0,12$.

“antigamente, um uso comum do jogo do osso era na previsão do futuro, para o que jogava-se cinco ossos de cada vez. Um exemplo sendo a seguinte adivinhação

grega, chamada “o lance do Zeus salvador”: foram um 1, dois 3 e dois 4 ... Os Deuses deram um argúrio favorável. Não tire-o da cabeça, pois nenhum mal cairá sobre ti.” (MAT.UFRGS, 2001)

Para calcularmos a probabilidade da ocorrência deste argúrio favorável devemos fazer as seguintes considerações:

- Este evento h pode ocorrer de cinco modos diferentes, pois são cinco ossos que serão jogados, então $h=5!$
- O total n são os modos possíveis, isto é, dois lados, dois a dois, $n=2!2!$
- A probabilidade da ocorrência de cada face, considerando a composição das faces no lance:

Uma face no lado de valor 1: $P(1)=0,12$

Duas faces no lado de valor 3: $P(3)^2=0,37*0,37$

Duas faces no lado de valor 4: $P(4)^2=0,39*0,39$

- A probabilidade da ocorrência deste lance do “Zeus salvador” é calculada conforme a expressão: $(5! / 2! 2!)(0,12)(0,37)^2(0,39)^2 = 0,075$.

Os jogos de cartas se desenvolveram na Ásia, mas popularizaram-se na Europa com a invenção da imprensa.

“originalmente, as cartas eram grandes e quadradas, sem as identificações nos cantos. As figuras de baralho eram impressas com uma só cabeça, ao invés de duas, obrigando muitas vezes aos jogadores a identificá-las pelos pés, virar a carta revelaria a posse de uma carta com figura. Os cantos quadrados facilitavam a fraude, pois os jogadores podiam dobrar uma parte minúscula do canto para identificar mais tarde as cartas do baralho”. (BERNSTEIN, 1997, p. 13)

As cartas de figuras com dupla cabeça e as cartas com cantos arredondados só foram adotadas no século XIX.

Para Gadelha (2004, p. 2), a aplicação sistemática de análise matemática e o estabelecimento de regras gerais para a solução de problemas, que originou uma teoria matemática da probabilidade, teve início em 1654 com os resultados obtidos por dois

franceses, Blaise Pascal e Pierre de Fermat, em resposta a um desafio para solucionar um problema proposto a cento e sessenta anos antes pelo monge franciscano Luca Paccioli.

Segundo Bernstein (1997, p. 43), este velho problema dos pontos estava relacionado ao jogo de *balla*, e a questão era a seguinte:

“dois jogadores disputavam um prêmio que seria dado a quem primeiro fizesse 6 pontos no jogo da *balla*. Quando o primeiro jogador tinha 5 pontos e o segundo tinha 3 pontos, foi preciso interromper o jogo. Como dividir o prêmio?”.

Gadelha (2004) apresenta Girolamo Cardano (1501-1576) como um famoso médico de Milão do século XVI que dedicava parte de seu tempo à matemática e era apaixonado por jogos de azar e apostas. Lopes (2005) considera Cardano como o iniciador da teoria da Probabilidade, pois foi o primeiro a fazer observações probabilísticas sobre um dado honesto, um dado em que todas as faces têm a mesma chance de sair, e a escrever um argumento teórico para calcular probabilidades.

Para Bernstein (1997, p. 45), Cardano era o rei da jogatina, um matemático e o mais famoso médico da época. O tratado de Cardano sobre o jogo intitula-se *Líber de Ludo aleae*, o livro dos jogos de azar.

“a palavra *aleae* refere-se aos jogos de dados. *Aleatorius*, da mesma raiz, refere-se a jogos de azar em geral. Essas palavras chegaram até nós através da palavra aleatório, que descreve eventos cujo resultado é incerto. Assim os Romanos, com sua linguagem elegante, involuntariamente associaram para nós os significados de jogo e incerteza”. (BERNSTEIN, 1997, p. 47)

Cardano pode ter sido o primeiro a introduzir o lado estatístico da teoria das probabilidades.

Pela primeira vez, ele definiu a forma convencional de expressar a probabilidade através de uma fração, onde o numerador está relacionado ao número de resultados favoráveis, e no denominador representa o total dos resultados possíveis.

Bernstein (1997, p. 51) afirma que Cardano deu um passo decisivo em sua análise sobre o que ocorre quando passamos de um para dois dados, afirmando que apesar dos dados terem doze faces, um jogador poderia obter um total de trinta e seis combinações possíveis. Para este autor, Cardano ao entender a importância das combinações dos números, deu o mais importante passo para o desenvolvimento das leis da probabilidade.

2.3.3 Seguro, uma reação ao risco

Para Salles Jr., Soler, Valle e Rabechini Jr. (2006, p. 104), as reações ao risco podem ter algum custo e deverão, obrigatoriamente, alterar ou a probabilidade, se atacarmos a sua causa raiz, ou o impacto, se atuarmos no seu efeito, gerando então um novo valor esperado para aquele risco.

Todo o risco tem uma probabilidade associada a ele que não é zero, pois se o fosse, teríamos a certeza de que nada ocorreria, e nem é cem por cento a certeza da ocorrência deste fato.

O seguro pode ser considerado como uma reação contrária ao risco, por exemplo, do roubo de um carro. Sabemos que esta reação não alterará a probabilidade do roubo de carros, pois ela continuará o mesmo valor, porém o impacto desta ocorrência será zero.

As primeiras modalidades de seguros surgiram a 2300 a.C. na Babilônia, quando as caravanas atravessavam o deserto para comercializar camelos em cidades vizinhas. Diante do elevado risco deste trajeto alguns animais morriam, e por esta razão, os cameleiros firmavam um acordo de pagar pelo camelo de quem o perdeu.

Os seguros atingiram seu pleno desenvolvimento como conceito comercial somente no século XVIII, mas segundo Bernstein (1997, p. 91), o negócio de seguros

remonta a além do século XVIII a.C. O código de Hamurábi, que surgiu ao redor de 1800 a.C., dedica 282 cláusulas ao tema “bodemeria”. A bodemeria era um empréstimo ou uma hipoteca contraída pelo proprietário de um navio para financiar sua viagem. Apesar de nenhum prêmio ser pago como nos moldes atuais, se houvesse algum acidente e o navio afundasse, o dinheiro não precisaria ser reembolsado.

A atividade seguradora ganhou impulso em torno do ano de 1600. O termo “apólice”, já de uso generalizado na época, deriva do italiano “*polizza*”, que significa uma promessa.

O café fundado por Edward Lloyd em 1687, próximo ao rio Tamisa, era segundo Bernstein (1997, p. 89), o ponto de encontro favorito dos marujos dos navios atracados nas docas de Londres. Este café serviu desde o início de sede para os seguradores marítimos, em parte devido às suas excelentes ligações mercantis e na navegação. Em 1771, quase 100 anos depois de Edward Lloyd abrir seu café na Tower Street, setenta e nove dos seguradores que negociavam no Lloyd’s subscreveu cem libras cada e foram os membros do Lloyd’s original, a mais famosa empresa seguradora da história.

Os seguradores emitiam apólices contra quase todo tipo de risco como roubo em residências, roubos nas estradas, morte por excesso de gim, morte de cavalos e “seguro da castidade feminina”, dos quais, exceto este último, ainda são seguráveis.

2.3.4 Risco

O risco pode ser considerado como um conceito essencialmente moderno, pois possibilita às pessoas a se reorientarem, a tomar diferentes atitudes frente a possíveis eventos futuros, como se tentassem “domesticar o futuro”.

Para Bernstein (1997, p. 1):

“a idéia revolucionária que define a fronteira entre os tempos modernos e o passado

é o domínio do risco: a noção de que o futuro é mais do que um capricho dos deuses e de que homens e mulheres não são passivos ante a natureza. Até os seres humanos descobrirem como transpor essa fronteira, o futuro era um espelho do passado ou o domínio obscuro e adivinhos que detinham o monopólio sobre o conhecimento dos eventos previstos. [...] a capacidade de definir o que poderá acontecer no futuro e de optar entre várias alternativas é central às sociedades contemporâneas”.

A palavra “risco” deriva do italiano antigo *risicare*, que significa “ousar”.

Bernstein (1999, p. 8) entende neste sentido, que o risco é uma opção, e não um destino. É das ações que ousamos tomar, que dependem de nosso grau de liberdade de opção, que a história do risco trata.

O tempo é o fator dominante do jogo. O risco e o tempo são faces opostas da mesma moeda, pois sem amanhã não haveria risco.

Espinha e Souza (2007, p. 2) definem risco como “exposição à chance de perdas ou danos”.

O conceito de risco resulta da incerteza quanto a eventos futuros e é parte de todas as atividades de desenvolvimento.

Um processo de desenvolvimento bem definido e institucionalizado tem o objetivo de reduzir a chance da ocorrência de ameaças, isto é, perdas e danos.

Sabemos que toda a atividade de desenvolvimento, por mais controlada e precisa que seja, carrega uma possibilidade de falha. Espinha e Souza (2007, p. 3) entende que isto ocorre pelo fato de ocorrerem um grande número de variáveis que podem influenciar no resultado final e muitas delas de natureza imprevisível.

A eliminação total dos riscos é um conceito utópico, pois para cada ação de identificação, acompanhamento e controle de um projeto, possui um custo associado, cabendo à organização identificar para cada projeto o ponto de equilíbrio entre o nível de exposição ao risco e o custo de sua redução.

Bernstein (1997, p. 21), afirma que “a perspectiva de enriquecer é altamente motivadora, e poucas pessoas ficam ricas sem correr riscos”.

Para Vargas (2003, p. 94), todo risco necessita ser avaliado segundo dois aspectos: probabilidade da ocorrência e gravidade das conseqüências.

Segundo a visão financeira, quando o risco não é bem tratado, ocasiona um prejuízo que é quantificado como o produto entre a probabilidade do risco ocorrer e a sua gravidade expressa em valores de prejuízo financeiro.

Consideraremos como prioritárias as respostas das ocorrências em que o maior produto seja obtido, isto é, o maior Valor Monetário Esperado (*Earned Monetary Value*):

$$VME = P * G$$

Onde: P = Probabilidade

G = Valor monetário da Gravidade

Segundo Costa, Barros e Travassos (2005, p. 1), a engenharia de *software* é uma atividade econômica como qualquer outra, e como tal, deve ser avaliada não só em relação aos seus problemas técnicos, mas como também quanto à possibilidade de gerar valor para a organização desenvolvedora, fato que justifica economicamente a utilização de recursos em projetos de *software*.

Para Andrade (2000), risco é definido como uma estimativa do grau de incerteza que se tem com respeito à realização de resultados futuros desejados. Então, quanto mais ampla for a faixa dos valores previsíveis para o retorno financeiro do investimento, maior será o valor do grau do risco do investimento.

Segundo Bernstein (1997, p. 337):

“a ciência da administração do risco cria, às vezes, novos riscos, ainda que leve o controle a antigos riscos. Nossa fé na administração do risco encoraja-nos a assumir riscos que, normalmente não assumiríamos. Na maioria das vezes, isso é benéfico,

mas precisamos ter cuidado para não aumentar a quantidade de risco do sistema. Pesquisas revelam que os cintos de segurança encorajam os motoristas a dirigir mais agressivamente. Por conseguinte, o número de acidentes aumenta, ainda que a gravidade dos danos em qualquer acidente individual diminua”.

Alencar e Schmitz (2005, p. 25) definem que uma maneira previsível para lidar com estes imprevistos, fazendo com que os possíveis cenários futuros fiquem dentro de uma faixa de variabilidade aceitável é o gerenciamento deste risco.

2.4 GERENCIAMENTO DE RISCOS

Segundo Salles Jr., Soler, Valle e Rebechini Jr. (2006, p. 26), gerenciar risco envolve a tomada de decisões em ambiente incerto, complexo e dinâmico. O efetivo gerenciamento de riscos em projetos ocorre quando sabemos lidar com as incertezas que surgem.

Para Reis e Albuquerque (2004, p. 2), apesar de existirem diferenças sutis na terminologia e na forma de organização e apresentação dos conceitos relacionados ao gerenciamento de riscos, existe consenso entre as sociedades de gerenciamento de projetos sobre este termo. Segundo ele, gerenciamento de risco é definido como um processo formal, integrado aos demais processos de gerenciamento de projetos, e dedicado à identificação, análise, definição das medidas de proteção contra os riscos, e controle sobre estes.

É um processo contínuo que deve acontecer durante todo o ciclo de vida do projeto.

Schmitz, Alencar e Villar (2006, p. 23) explicam que gerência de risco é um conjunto de atividades que tem por objetivo, de forma economicamente racional, maximizar o efeito dos fatores de risco positivos e minimizar o efeito dos negativos.

De acordo com Espinha e Souza (2007, p. 3), as ações de gerência de risco nos processos podem contribuir diretamente com a qualidade do produto final e fornecem dados que permitem identificar quais ações devem ser tomadas com maior urgência. As principais atividades da disciplina de gerência de risco são:

- Identificação: corresponde a identificação dos riscos inerentes a uma etapa do desenvolvimento (fase, processo, iteração). Isto é feito através do levantamento das ameaças presentes e do impacto que estas podem provocar, caso se realizem.
- Análise: corresponde à reflexão sobre os riscos identificados, a partir da identificação do nível de exposição de cada projeto. Nesta atividade é realizado um estudo de classificação dos riscos, com base no relacionamento entre a exposição e a consequência negativa do risco e o benefício da oportunidade. Neste momento, determinam-se quais serão eliminados, quais os que serão mitigados, os aceitáveis e os que serão acompanhados.
- Planejamento: observa e determina como e quando os riscos serão abordados ao longo do projeto. Comumente são elaborados planos de mitigação, eliminação e acompanhamento de riscos que serão utilizados como base para a gerência de riscos.
- Controle: corresponde a execução e ao acompanhamento dos planos elaborados para o projeto. Os riscos identificados são analisados constantemente para a identificação do seu estado atual, e na atualização dos planos, novos riscos podem ser identificados.

Para Vargas (2005, p. 95), o PMBOK subdivide o gerenciamento de riscos em seis processos:

- Planejamento do gerenciamento de riscos: decisão de como abordar, planejar e executar as atividades de gerenciamento de riscos de um projeto.
- Identificação de riscos: determinação dos riscos que podem afetar o projeto e a documentação de suas características.
- Análise qualitativa de riscos: priorização dos riscos para análise ou ação adicional subsequente, através de avaliação e combinação de sua probabilidade de ocorrência e impacto.
- Análise quantitativa de riscos: análise numérica do efeito dos riscos identificados nos objetivos gerais do projeto.
- Planejamento de respostas a riscos: desenvolvimento de opções e ações para aumentar as oportunidades e reduzir as ameaças aos objetivos do projeto.
- Monitoramento e controle de riscos: acompanhamento dos riscos identificados, monitoramento dos riscos residuais, identificação dos novos riscos, execução de planos de respostas a riscos e avaliação da sua eficácia durante todo o ciclo de vida do projeto.

Segundo ele, o gerenciamento de riscos possibilita a chance de melhor compreender a natureza do projeto e envolver os membros do time, de modo a identificar e responder às potenciais forças e riscos do projeto, geralmente associadas a tempo, qualidade e custos. A sobrevivência de qualquer empreendimento, atualmente, está intimamente vinculada ao conceito de aproveitar uma oportunidade, dentro de um espectro de incertezas.

O que torna a gestão dos riscos tão importante são estes fatores diversos, como o aumento da competitividade, o avanço tecnológico e as condições econômicas, que fazem com que os riscos assumam proporções muitas vezes incontroláveis.

Segundo Kroll e Kruchten (2003):

“o gerente de projeto de sucesso é aquele que se faz presente e curioso, conversa com os membros do time, investiga sobre a tecnologia, quer conhecer o porquê, o como e o porquê novamente, tudo com o objetivo de identificar novos e inesperados riscos”.

O *Rational Unified Process* (RUP) recomenda que os gerentes de projeto adotem os seguintes passos para gerenciar efetivamente os riscos envolvidos no projeto de *software*: identificar os riscos que ameaçam a qualidade do *software* final; analisar e definir prioridades para os riscos, estimando o impacto de cada um deles, caso venham a ocorrer; elaborar estratégias para evitar os riscos identificados; elaborar estratégias para diminuir os riscos e estratégias de contingência; e revisar os riscos frequentes entre as interações e nas fases subsequentes.

Segundo Gusmão e Moura (2004, p. 1), o SEI (*Software Engineering Institute*) define o processo de gerência de risco de *software* através de um modelo contínuo de gerenciamento de riscos composto por cinco fases distintas: Identificação de riscos, Análise de riscos, Plano de respostas aos riscos, Rastreamento e Controle de riscos.

Estes autores ainda consideram que a gerência de risco dá a gerência de projetos uma abordagem estruturada para a identificação e análise de riscos no início do planejamento do projeto e no decorrer das fases de desenvolvimento de *software*.

A NBR ISO 12207 apresenta os processos do ciclo de vida do *software* através da divisão destes em três grandes classes: Processos fundamentais, Processos de apoio e Processos organizacionais.

- Processos fundamentais: aquisição, fornecimento, desenvolvimento e operação, responsáveis pelo início e execução do desenvolvimento, operação e manutenção do *software* durante todo o seu ciclo de vida.
- Processos de apoio: documentação, gerência de configuração, garantia da qualidade, validação, auditoria e resolução de problemas.
- Processos organizacionais: processos gerenciais, de infra-estrutura, de melhoria e de treinamento com o objetivo aprimorar continuamente a estrutura e seus processos.

Na abordagem descrita por Espinha e Souza (2007, p. 3), a gerência de risco utiliza como base o conceito de exposição ao risco. Para cada ameaça ou possível fonte de problemas que possam causar perdas ao projeto, a exposição é definida como o produto da probabilidade da perda ocorrer pelo tamanho desta perda:

$$Exp. = P * I$$

Onde:

P = Probabilidade da perda ocorrer, caso o controle não esteja implementado na organização.

I = Impacto no projeto ou qualquer elemento foco da análise de risco.

Com o objetivo de obter maior precisão na distribuição e no impacto dos riscos, este conceito pode ser determinado para cada ativo da organização ou especialista participante, obtendo-se assim um índice de exposição balanceado. Este índice, denominado PSR, determina a exposição através da probabilidade da ocorrência do risco (P), a severidade desta ocorrência para o projeto ou para a organização(S) e a relevância do ativo da organização (pessoa ou processo) na qualidade do produto final.

Segundo o autor, assim é possível obter um retrato mais preciso da distribuição e do impacto dos riscos.

Os diversos estudos e autores atuais comprovam que muitos dos problemas envolvidos em projetos de grande porte estão mais associados à falhas em atividades de gerenciamento do que em atividades técnicas, por isso, o desenvolvimento de *software* pode ser considerado uma atividade de risco (HALL, 1998).

2.5 RISCOS NO DESENVOLVIMENTO DE SOFTWARE

Todos os projetos de desenvolvimento de *software* estão expostos a incertezas das mais variadas origens. Estas incertezas podem implicar em inúmeras perdas ou ganhos, demandando planejamento e acompanhamento através da gerência de riscos.

Segundo Junior e Belchior (2002, p. 1), um projeto de *software* é uma atividade que tem início e fim delimitados, com objetivos estabelecidos em parâmetros de escopo, custo, tempo, qualidade, risco, suprimentos, recursos humanos e comunicação.

Normalmente, um produto gerado por um projeto de *software* deve atender as especificações feitas por quem financiou seu desenvolvimento e realizar os objetivos que lhes foram propostos.

Segundo Junior e Belchior (2002, p. 1), muitos dos problemas que afetam os projetos de desenvolvimento de *software* são de ordem gerencial e não técnicos. Esta afirmação vem ao encontro do mito de que os problemas que causam falhas e atrasos no processo de produção de *software* são de natureza técnica.

A gerência de projetos tem o objetivo de suprir estas deficiências, pois abrange todo o processo de desenvolvimento de *software*, do início ao fim.

Para Gusmão e Moura (2004, p. 2):

“a gerência de risco em engenharia de *software* tem a finalidade de aumentar a qualidade do produto e do processo de desenvolvimento de *software*. [...] Os problemas que surgem embora considerados inerentes ao desenvolvimento de *software* por muitos autores, podem ser minimizados e controlados pelo contínuo gerenciamento de risco de projetos”.

Para Falbo (1998), num processo de desenvolvimento de *software*, devem ser consideradas as seguintes informações: atividades a serem realizadas, recursos necessários, artefatos requeridos e produzidos, procedimentos adotados e o modelo de ciclo de vida utilizado.

Fernandes (2003, p. 1) escreveu que:

“a prática do desenvolvimento de *software* está no cerne de uma relação humana de troca de planos, posses, desejos e necessidades entre três categorias de agentes coletivos: os que usam, os que adquirem e os que produzem.[...] o *software* não é, de fato, uma máquina, mas sim uma descrição de máquina.[...] *software* é um artefato virtual, incapaz de realizar trabalho a menos que exista uma máquina que carregue e interprete as instruções e informações contidas no mesmo, o que resulta na construção de outra máquina, de ordem superior, com a qual interage o usuário. Em outras palavras, na análise de qualquer sistema de computação estaremos sempre falando de duas máquinas”.

Segundo este autor, o desenvolvimento de *software* é um agente coletivo responsável por criar um plano de construção de máquina (*software*) que esteja dentro das condições estabelecidas pelo cliente.

Para Gusmão e Moura (2004, p. 1), “à medida que o tamanho e a complexidade dos sistemas de *software* crescem, aumenta a necessidade da utilização de metodologias para o gerenciamento de riscos”.

Podemos entender que as empresas que desenvolvem *softwares* estão se tornando cada vez mais dependentes do desempenho de seus produtos. Neste caso, a gerência de risco não pode se basear em boas práticas de desenvolvimento, mas sim, em boas práticas para gerir seu negócio.

Costa, Barros e Travassos (2005, p. 2), entendem que no desenvolvimento de projetos de *software*, os custos, compõem-se dos gastos com a equipe, *hardware*,

software e infra-estrutura necessária para a execução do projeto. Então poderemos considerar a engenharia de *software* como uma atividade econômica, e como tal, deverá ser avaliada não só em relação aos seus problemas técnicos, mas também quanto à possibilidade de gerar valor para a organização desenvolvedora.

Apesar de existirem muitos relatos individuais sobre sucessos no desenvolvimento de *softwares*, Gusmão e Moura (2004) afirmam que a indústria de *software* como um todo, parece ainda não aplicar ativamente e sistematicamente os métodos de gerência de risco. Uma metodologia de apoio aos projetistas e gerentes de projetos de tecnologia de informação no desenvolvimento e efetivação de suas atividades, garantiria o cumprimento das metas do projeto, os custos, prazos, e a qualidade do produto gerado.

Para Machado (2000), deve-se considerar a sua adequação às tecnologias envolvidas, ao tipo de *software* em questão, ao domínio de aplicação, ao grau de maturidade (ou capacitação) da equipe em engenharia de *software*, às características próprias da organização e às características do projeto e da equipe.

Os projetos de desenvolvimento de *software* têm tido inúmeros problemas relacionados à sua gerência, tais como gastos além do orçado, atrasos na entrega e não atendimento às necessidades dos clientes.

Segundo Nascimento (2000), os métodos para a detecção de defeitos eram utilizados mais intensamente do que os métodos de prevenção. Observa-se que o fracasso no desenvolvimento de *software* está relacionado com a deficiência em identificar as incertezas, fazendo-nos considerar o gerenciamento de risco estruturado, isto é, que conhece como lidar com estas incertezas que ocorrem em qualquer projeto, como um importante diferencial competitivo.

Para Salles Jr., Soler, Valle e Rebechini Jr. (2006, p. 60), as abordagens qualitativa e quantitativa se completam para o gerenciamento do risco, pois, enquanto a primeira qualifica a prioridade, a segunda quantifica o impacto deste risco.

Se olharmos para o mercado e para o gerenciamento de projetos, veremos que hoje em dia é muito mais comum o uso da qualificação do que da quantificação, por esta razão, existem modelos que possibilitam analisar qualitativamente o risco no desenvolvimento de *software*.

2.6 MODELOS QUALITATIVOS DE ANÁLISE DE RISCO

Para Pinho e Mendonça (2005, p. 1), a idéia da gerência de risco é a execução de atividades que viabilizem a identificação prévia, e o tratamento de potenciais problemas. Na literatura existem diversos modelos com o objetivo de incorporar esta gerência de risco ao processo de desenvolvimento de *software*.

Segundo Xu, Khoshgoftaar e Allen (2003), os métodos existentes de estimativa de risco bastante usados são o checklist e a análise de uma matriz de risco, onde os fatores de risco são pontuados de acordo com sua influência sobre o risco operacional potencial. Estas pontuações são então agregadas aritmeticamente em uma pontuação geral de risco. No entanto, como estas informações estão disponíveis durante fases muito precoces do *software*, estes métodos quantitativos, como a matriz de risco, trabalham com suposições para estimar o risco operacional.

Segundo Schmitz, Alencar e Villar (2006, p. 40), os modelos qualitativos são baseados em questionário sobre um conjunto predefinido de fatores de risco, ao respondê-lo, a aplicação destes modelos produz um valor numérico que representa o risco a que um projeto estará exposto.

São três os modelos de análise qualitativa de risco considerados:

2.4.1 Modelo Moynihan;

2.4.2 Modelo Karolak;

2.4.3 Modelo BRisk.

2.6.1 Modelo MOYNIHAN

Segundo Schmitz, Alencar e Villar (2006, p. 56), este modelo procura determinar a chance dos quatro critérios de sucesso serem atingidos, a partir dos valores estimados para os vinte e três fatores de risco, isto é, qual a probabilidade de ocorrência dos quatro critérios de sucesso: Custo, Terminação, Requisitos e Sem falhas, de serem afetados pelos fatores de risco.

Esta relação probabilística dá origem a uma matriz de impacto W contendo vinte e três linhas e quatro colunas.

Cada célula desta matriz pode assumir para a linha i os valores 0 ou 1, de acordo com a influência do fator de risco em relação ao critério de sucesso, isto é, se o valor da linha i é igual a 0, significa que este fator não afeta o critério de sucesso j , se este valor é igual a 1, significa que afeta este mesmo critério.

Seja:

I : Número do fatores de risco.

K : Índice do critério de sucesso em consideração;

$W[i, K]$: Impacto do fator de risco I no fator K ;

$N[i]$: Intensidade do fator i ;

$P(K)$: Probabilidade do critério de sucesso K falhar.

$$P(K) = \frac{\sum_{n=1}^{23} W[i, k] * N[i] - \sum_{n=1}^{23} W[i, k]}{3 * \sum_{n=1}^{23} W[i, k]}$$

A aplicação do modelo consiste em preparar um questionário com os vinte e três fatores de risco e quatro alternativas de respostas numeradas, cada uma relacionada com os critérios de sucesso. As respostas obtidas são separadas em grupos de mesmos valores de respostas, sendo os fatores de pior pontuação as avaliações maior ou igual a 3.

A análise dos resultados é realizada com base na probabilidade calculada para cada critério de sucesso, quanto maior esta probabilidade, maior as chances daquele critério de sucesso ser atingido.

2.6.2 Modelo KAROLAK

Segundo Schmitz, Alencar e Villar (2006, p. 80), karolak defende o fato de que os fatores de risco podem ser vistos sob duas óticas: a técnica ou tecnológica e a de negócios. Este método de gerenciamento do risco utiliza um modelo de estimativa de risco chamado SERIM (*Software Engineering Risk Model*), que utiliza três critérios de sucesso, dez categorias de risco e oitenta e um fatores de risco.

Os três critérios de sucesso são:

- Os fatores de risco técnicos que irão afetar a performance do final do produto, são eles: funcionalidade, qualidade, confiabilidade, usabilidade, temporal, manutenibilidade e reusabilidade (habilidade do software de poder ser usado novamente em aplicações similares).
- Os fatores de risco de custo irão impactar no custo do desenvolvimento do produto. O sucesso de custo depende de: orçamento, custos não recorrentes e recorrentes, custos fixos e variáveis, margem de lucro e realismo.
- Os fatores de risco de cronograma impactam no prazo do projeto, são eles:

flexibilidade, atingir metas e realismo.

As dez categorias de risco são: organização, estimativa, monitoramento, metodologia de desenvolvimento, ferramentas, cultura do risco, usabilidade, correção, confiabilidade e pessoal.

O modelo SERIM utiliza a seguinte avaliação da intensidade de atuação dos fatores de risco sobre o projeto:

- Nenhum: 0,0
- Um pouco: 0,2
- Algum: 0,5
- Maioria: 0,8
- Todos: 1,0

A probabilidade dos critérios de sucesso de cada uma das categorias de risco $P(C_k)$ é calculada, onde:

- $1 \geq k \geq 10$;
- $P(C)$ é a probabilidade de cada categoria de risco;
- i é métrica inicial e f é a métrica final;
- $P(Fr)$ é a resposta de cada fator de risco associado a categoria de risco;
- n é a quantidade de fatores de risco associados a categoria de risco.

$$P(Ck) = \frac{\sum_{j=1}^f P(Frj)}{n}$$

A probabilidade dos critérios de sucesso serem atingidos é obtido pela média ponderada das probabilidades de sucesso das categorias de risco.

O peso de cada categoria é determinado pela influência dela nos critérios de sucesso, para baixa influência, peso 1; para intermediária 2, e para uma influência alta peso 3.

A probabilidade dos critérios de sucesso ser atingida é calculada onde:

- $1 \leq k \leq 3$;
- $P(CS)$ é a probabilidade de cada critério de sucesso ser atingido;
- $P(C)$ é a probabilidade de sucesso em cada uma das categorias de risco;
- W é o peso da influência do fator de risco.

$$P(CSk) = \frac{\sum_{j=1}^{10} W[j, k] * P(Cj)}{\sum_{j=1}^{10} W[i, k]}$$

A probabilidade de sucesso do projeto é dada pela média aritmética da probabilidade de sucesso das categorias de risco.

A probabilidade de sucesso do projeto é calculada onde:

- $P(\text{Sucesso})$ é a probabilidade de sucesso do projeto de software;
- $P(C)$ é a probabilidade de cada uma das categorias de risco.

$$P(\text{Sucesso}) = \frac{\sum_{j=1}^3 P(Cj)}{3}$$

2.6.3 Modelo BRisk

Considerado por Schmitz, Alencar e Villar (2006, p. 117), como facilmente adaptável, o modelo BRisk é uma evolução do modelo karolak, em que é solicitado ao especialista sua avaliação sobre os fatores de risco, e informada através da estimativa de três pontos. Esta estimativa permite utilizar a função de distribuição de probabilidades triangular, contínua, com parâmetros: mínimo, mais provável e máximo, que representam melhor a subjetividade desta avaliação.

A probabilidade do valor de cada fator $P(Fr)$ é dada por uma distribuição normal, onde:

- j é o avaliador;
- i é o fator de risco em questão;
- n é o número de avaliadores de um fator de risco.

$$\mu(FRi) = \frac{\sum_{j=1}^n Waji \mu(Aji)}{\sum_{j=1}^n Waji}$$

$$\sigma^2(FRi) = \frac{\sum_{j=1}^n Waji^2 \sigma^2(Aji)}{\left(\sum_{j=1}^n Waji\right)^2}$$

Os modelos qualitativos de análise de risco citados, que se baseiam na teoria da probabilidade, na frequência e ocorrência do risco, muitas vezes não têm dados para avaliar com segurança estes eventos, pois contemplam a matemática e desconsideram um fator importantíssimo que é o peso da experiência e do conhecimento dos especialistas.

Diante desta realidade, a metodologia utilizada para este estudo será a quali-

quantitativa, para a identificação, análise e tratamento do risco no desenvolvimento de *software* utilizando a teoria dos conjuntos *fuzzy*. Comprovada a qualidade da informação, serão criadas regras *fuzzy* para obter a matriz de risco.

A partir da classificação do risco, serão criadas as condições necessárias para que se possa elaborar um plano de ações para mitigar os efeitos negativos que possam surgir durante o desenvolvimento do *software*.

2.7 TEORIA DOS CONJUNTOS FUZZY

2.7.1 Histórico

Podemos considerar que a evolução histórica da teoria dos conjuntos *fuzzy* iniciou-se com Aristóteles, filósofo grego (384-322 aC.), quando fundou a ciência da lógica, que estabeleceu um conjunto de regras rígidas para que as conclusões pudessem ser aceitas como logicamente válidas. O emprego desta lógica, baseada em premissas e conclusões, é binária, e classifica as afirmações como verdadeiras ou falsas.

Enquanto na teoria clássica, os conjuntos são denominados *crisp*, onde um dado elemento do universo em discurso (domínio) pertence ou não pertence ao referido conjunto, na teoria *fuzzy* existe um grau de pertinência de cada elemento a um determinado conjunto.

A lógica difusa viola estas suposições, o conceito de dualidade estabelecendo que algo possa e deva coexistir com o seu oposto, faz a lógica difusa parecer natural, pois muitas experiências humanas não podem ser classificadas simplesmente como sim ou não e preto ou branco.

As primeiras noções da lógica dos conceitos “vagos” foram desenvolvidas em 1920, por um lógico polonês Jan Lukasiewicz (1878-1956), quando apresentou

conjuntos com graus de pertinência 0, 1/2 e 1, e mais tarde expandiu este conjunto para um número infinito de valores no intervalo entre 0 e 1. Em 1930, Bertrand Russell afirmava que alguns problemas não poderiam ser resolvidos pela lógica aristotélica tradicional, mas apenas pela nebulosa.

A primeira publicação sobre lógica *fuzzy*, foi escrita em 1965 por Lotfi Asker Zadeh, onde ele combinava os conceitos da lógica clássica e os conjuntos de Lukasiewicz, definindo assim os graus de pertinência.

2.7.2 Teoria

Schmitz, Alencar e Villar (2006) consideram que a matemática nebulosa é uma tentativa que permite dar forma matemática às expressões próprias da linguagem natural, sem diminuir a sua potência expressiva. Segundo Silva (2001, p. 66), uma variável lingüística é uma variável cujos valores são palavras ou sentenças na forma de linguagens naturais. Ela é usada para representar a semântica de seus conjuntos. Por exemplo: velocidade, temperatura, peso, altura, aceleração, beleza, satisfação, etc.

Para Moré (2004):

“Lógica *fuzzy* é a ciência que se preocupa com os princípios formais do raciocínio aproximado. Procura modelar os modos imprecisos do raciocínio que têm um papel fundamental na habilidade humana de tomar decisões.”

Segundo Braga, Barreto e Machado (1995, p. 6):

“com a matemática nebulosa, pode-se realizar “operações com palavras”, onde os conjuntos nebulosos são os “valores” das palavras. [...] a matemática nebulosa é uma tentativa de aproximar a precisão característica da matemática à inerente imprecisão do mundo real, nascida no desejo profundo de conhecer melhor os processos mentais do raciocínio”.

Um conjunto *fuzzy* “A” é caracterizado por uma função de pertinência, relacionando os elementos de um domínio, espaço, ou universo de discurso X, ao intervalo unitário [0,1] (ZADEH, 1965). Para Lima e Amorim (2007, p. 4), a pertinência

de um elemento num determinado conjunto *fuzzy* é uma questão de grau. Este grau representa um nível de compatibilidade do elemento sobre o conjunto, no qual o zero significa a não participação, o valor um significa participação plena e os demais valores entre zero e um significam participação parcial. Os graus de pertinência são obtidos através de funções denominadas funções de pertinência, que podem ser representadas por formas geométricas diversas. As principais são: triangular, trapezoidal e gaussiana. O formato da função de pertinência é inerente à relação dos dados com o que eles representam.

Matematicamente, essa relação é descrita como: $\mu_A : X \rightarrow [0,1]$, em que o valor $\mu_A(X) \in [0,1]$, onde “X” é a variável do universo em estudo, e é uma função cuja imagem pertence ao intervalo $[0,1]$ e indica o grau com que o elemento x de X está no conjunto *fuzzy* A, com $\mu_A(X) = 0$ e $\mu_A(X) = 1$, indicando respectivamente, a não-pertinência e a pertinência completa de x ao conjunto *fuzzy* A. A variável “X” pode ser discreta ou contínua.

Em questão estão situações do tipo que se segue. Imagine que haja um fator que esteja sendo avaliado pelos desenvolvedores de *software* em relação ao risco no atraso do projeto (exemplo: *o número de pessoas para desenvolver e implementar o sistema no prazo agendado é insuficiente*). Um desenvolvedor pode avaliar este fator em termos de ocorrência (*raramente ocorre, ocorre com pouca frequência, ocorre às vezes, ocorre com frequência, sempre pode ocorrer*) e de grau de tolerabilidade (*não tolerável, pouco tolerável, moderadamente tolerável, bastante tolerável, totalmente tolerável*). A pergunta que se apresenta é: com que frequência, durante o

desenvolvimento do projeto, está presente esse fator? Quando a presença do fator deixa de ser *pouco tolerável* para se tornar *não tolerável*?

A teoria dos conjuntos *fuzzy* lida com esse problema atribuindo ao elemento (opinião de cada especialista entrevistado) graus de pertinência aos conjuntos *fuzzy* “*raramente ocorre, ocorre com pouca frequência, ..., bastante tolerável, totalmente tolerável*”. Quanto mais opiniões coincidirem em termos de frequência de respostas, maior será o seu grau de pertinência ao conjunto *fuzzy*.

Em relação à utilização da teoria dos conjuntos *fuzzy*, para traduzir a linguagem natural em um formato numérico, observa-se alguns trabalhos importantes na área de avaliação de riscos como em Cunha R. M. (2007), Braga M. J. F. (1995), Kuchta (2001) e outros.

2.7.3 Operações básicas

Para o entendimento da metodologia que será proposta, é importante o conhecimento das operações básicas desta teoria. Segundo Braga, Barreto e Machado (1995):

- Sejam os conjuntos $A(X_i)$ e $B(X_i)$ conjuntos *fuzzy* ($i=1\dots n$). A notação utilizada para a **união** é $A(X_i) \cup B(X_i)$. A união de dois conjuntos *fuzzy* $A(X_i)$ e $B(X_i)$ é dada por: $A(X_i) \cup B(X_i) = C(X_i) = \max\{\mu A(X_i), \mu B(X_i)\}$.
- Sejam os conjuntos $A(X_i)$ e $B(X_i)$ conjuntos *fuzzy* ($i=1\dots n$). A notação utilizada para a **interseção** é $A(X_i) \cap B(X_i)$. A interseção de dois conjuntos *fuzzy* $A(X_i)$ e $B(X_i)$ é dada por: $A(X_i) \cap B(X_i) = C(X_i) = \min\{\mu A(X_i), \mu B(X_i)\}$.
- Seja $A(X_i)$ um conjunto *fuzzy* ($i=1\dots n$). A notação utilizada para o **complemento** é $\bar{A}(X_i) = \{(1 - \mu_A(X_i))\} / X_i$.

- **Grau de Pertinência:** Diz-se que um elemento pertence a um conjunto se o grau de pertinência é 1, e se não pertence é 0. A notação de pertinência é “ μ ”. Como exemplo podemos considerar um objeto como “grande”, “médio”, “pequeno”, ... e podemos atribuir graus de pertinência que podem variar de 0,7 para “grande” e 0,2 para “pequeno”.
- **Cardinalidade:** A cardinalidade de um conjunto *fuzzy* $A(X_i)$ é definida como sendo a soma dos valores de pertinência de $A(X)$. Sua notação é: $|A| = \sum_{x \in U} \mu_A(x)$.
- **Nebulosidade:** A nebulosidade de um conjunto *fuzzy* $A(X)$ é dada pela área de $A(X) \cap \bar{A}(X)$. A notação utilizada é $B(A(X))$.

Caso a variável “X” seja discreta, teremos a seguinte expressão:

$$B(A(X)) = \frac{2}{N_{A(X)}} |A|, \text{ onde } N_{A(X)} \text{ representa o número de valores da variável X que}$$

estão em $A(X)$.

- **Incerteza:** A incerteza de um conjunto *fuzzy* $A(X)$ cuja notação utilizada é $I(A(X))$, é dada pela expressão:

$$I(A(X)) = \frac{1}{3} [B(A(X)) + \bar{\mu}_{\text{sup}}^{A(x)} + r_{A(x)}]^{-1}, \text{ onde}$$

$$\bar{\mu}_{\text{sup}}^{A(x)} = 1 - \underline{\mu}_{\text{sup}}^{A(x)}, \quad \underline{\mu}_{\text{sup}}^{A(x)} = \max \mu_{A(x)}, \quad r_{A(x)} = \frac{N_{A(X)}}{N_U}$$

$r_{A(x)}$ é o intervalo relativo do conjunto *fuzzy* $A(X)$

$N_{A(X)}$ é o número de valores da variável “X” que estão em $A(X)$.

N_U é o número de valores do Universo.

- **Qualidade da Informação:** Aceitando-se a medida de incerteza $I(A(X))$, pode se medir a qualidade de informação de um conjunto *fuzzy*, que será melhor quanto menor a incerteza $I(A(X))$ com índice $C(A(X))$.

$$C(A(X)) = [1 + I(A(X))]^{-1}.$$

$$0,50 \leq C(A(X)) \leq 1$$

$$I(A(X)) = 0 \Rightarrow \text{certeza} \Rightarrow (A(X)) = 1$$

$$I(A(X)) = 1 \Rightarrow \text{incerteza máxima} \Rightarrow (A(X)) = 0,50$$

- **Normalização:** Diz-se que um conjunto *fuzzy* é normal quando $\mu_{\text{sup}} = 1$. Quando isso não ocorre e se quer normalizar o conjunto *fuzzy*, dividem-se todos os μ por μ_{sup} .

2.7.4 Aplicações da teoria dos conjuntos *fuzzy* no desenvolvimento de *software*

No meio acadêmico encontramos inúmeros trabalhos na área da Tecnologia da Informação que utilizam a teoria dos conjuntos *fuzzy* no desenvolvimento de *software*.

Lee (1996) apresentou um novo algoritmo para avaliar a taxa de risco agregado no desenvolvimento de *software* usando a teoria de conjunto *fuzzy*. O algoritmo proposto tem as seguintes vantagens: (1) não precisa formar as matrizes de avaliação *fuzzy* para atributos; (2) não precisa realizar as complicadas operações de *defuzzyficação* de números *fuzzy* usando o método centróide. O algoritmo proposto é mais eficiente e mais rápido na avaliação da taxa de risco agregado no desenvolvimento de *software*.

Xu, Khoshgoftaar e Allen (2003), aplicaram um sistema *Fuzzy expert* na estimativa de risco operacional de *software*. Neste estudo os autores investigaram as

estimativas de risco no ciclo de vida do desenvolvimento de *software*. A NASA IV&V é uma instituição independente que conduz estimativas independentes para diversos projetos da NASA, e é responsável pela avaliação de riscos operacionais de *software*. Neste estudo, os autores investigaram as estimativas de risco no ciclo de vida do desenvolvimento de *software* e desenvolveram um sistema chamado *Research Prototype Early Assessment System* para dar suporte a estas estimativas dos projetos durante as fases iniciais do ciclo de vida do desenvolvimento de *software*. A lógica fuzzy proporciona um meio conveniente de representar variáveis lingüísticas, e para representar o risco, a probabilidade subjetiva é uma maneira mais indicada, se comparada com a probabilidade de falha objetiva quantitativa, e as categorias de severidade fuzzy têm mais credibilidade do que as pontuações numéricas. Este artigo ilustra como os sistemas *fuzzy expert* podem inferir resultados úteis ao utilizar os fatos sobre um projeto e regras sobre desenvolvimento de *software*.

Liu, Kane e Bamboon (2006), criaram um sistema inteligente de aviso antecipado para aperfeiçoamento da qualidade do *software* e para o gerenciamento do projeto. Para estes autores existem três grandes problemas em projetos de *software*: orçamento elevado, cronograma atrasado e baixa qualidade. Frequentemente, é muito tarde para corrigir estes problemas no momento em que eles são detectados em projetos de *software*. Neste artigo, discute-se o projeto, a implementação e a avaliação de um sistema de aviso antecipado de *software* baseado em lógica fuzzy que utiliza um conjunto integrado de características que consiste dos seguintes componentes: banco de dados de métrica de *software*, base de conhecimento de risco, estimativa inteligente de risco e rastreamento de risco. Este sistema ajuda a avaliar os riscos associados aos três problemas citados a partir de perspectivas do produto, processo e organização nas

primeiras fases do processo de desenvolvimento de *software*. Ele é capaz de agregar diversos resultados parciais de estimativa de risco em um indicador geral, mesmo que eles possam ser conflitantes, além de poder ser usado para analisar um risco por meio da identificação de sua causa raiz através de seu utilitário de rastreamento de risco.

Engel e Last (2007) apresentaram um modelo de verificação de risco e custo para o desenvolvimento de *software* utilizando a teoria dos conjuntos *fuzzy*. Segundo estes autores, o custo do ciclo de vida total associado às falhas do produto excede 10% do faturamento anual das empresas. Um fator relevante que contribui para esta perda é o desempenho ineficiente da Verificação, Validação e Teste (VVT) de *software* e sistemas. Os custos de qualidade em projetos de *software* consomem quase 60% do orçamento destinado ao seu desenvolvimento. O custo e a duração do projeto poderiam ser reduzidos por meio da otimização da estratégia VVT, o que ocasionaria a redução de cerca de 10 a 15% nos custos no desenvolvimento de *software*.

Diante desta realidade, os autores propõem um conjunto de modelos probabilísticos quantitativos para estimar custos e pesquisar os riscos oriundos da execução de qualquer estratégia de VVT. Para eles, um problema chave associado as estimativas de custo e de tempo é a imprecisão dos dados de entrada. Certos parâmetros são melhor capturados usando-se estruturas que utilizam valores máximos e mínimos, e outros parâmetros podem ser melhor encapsulados com o uso de termos linguísticos como “Alto” ou “Baixo”. Apresentam como exemplo, a estimativa do custo da qualidade que ocorre durante o desenvolvimento de *software* para uma aeronave de caça e demonstram que, ao aplicar a metodologia de lógica *fuzzy*, obtêm resultados comparáveis as estimativas baseadas em modelos que utilizam paradigma

probabilístico, isto é, menos de 4% em relação a cada uma das cinco categorias de custo VVT.

Segundo Kiran e Ravi (2008), em seu trabalho sobre predição da confiabilidade de *software*, foram desenvolvidos alguns modelos para prever, de maneira precisa, a confiabilidade do *software*. Os modelos apresentados utilizam diversas técnicas estatísticas (curvas de regressão linear múltipla e regressão adaptativa multi-variada) e técnicas inteligentes (rede neural com retro-propagação, sistema de inferência neuro-fuzzy expandido dinâmico e *TreeNet*). Os experimentos foram realizados em três conjuntos lineares e um conjunto não linear e com base nestes experimentos realizados e nos dados de confiabilidade de *software* obtidos da literatura, observou-se que o conjunto não linear superou todos os outros conjuntos e também as técnicas inteligentes e estatísticas utilizadas.

3 METODOLOGIA

Segundo Demo (1985, p. 19):

“metodologia é uma preocupação instrumental. Trata das formas de se fazer ciência. Cuida dos procedimentos, das ferramentas, dos caminhos. [...] é essencial entendermos a importância da metodologia para a formação do cientista. [...] a ciência propõe-se a captar e manipular a realidade assim como ela é. A metodologia desenvolve a preocupação em torno de como chegar a isto”.

Através do método se ordena e se hierarquiza as ações que serão adotadas para se chegar ao objetivo maior, a ciência.

3.1 ABORDAGEM

A abordagem da pesquisa apresentada nesta dissertação é considerada qualitativa e quantitativa, onde as informações de entrada deste modelo foram obtidas a partir da aplicação de um questionário a uma amostra composta por quinze especialistas em engenharia de *software* da empresa DATAPREV.

Segundo Dezin e Lincoln (2006, p. 23):

“a palavra qualitativa implica uma ênfase sobre as qualidades das entidades e sobre os processos e os significados que não são examinados ou medidos experimentalmente (se é que são medidos de alguma forma) em termos de quantidade, volume, intensidade ou frequência. [...] já os estudos quantitativos enfatizam o ato de medir e de analisar as relações causais entre variáveis, e não processos.”

Para Salles Jr., Soler, Valle e Rebechini (2006, p. 60), na abordagem qualitativa de riscos, estamos na verdade preocupados em qualificar a importância e a prioridade do risco, enquanto na abordagem quantitativa de riscos estaremos preocupados em quantificar o impacto dos riscos nos resultados do projeto.

As respostas obtidas a partir destes questionários foram apresentadas, através de dados mensuráveis e impessoais, que nos possibilita chegar a conclusões e assim abandonar a amostra e generalizar este resultado.

3.2 TIPOLOGIA DA PESQUISA

Quanto à tipologia, esta pesquisa é considerada um Estudo de Caso, investigação delimitada que é focada em um único caso, num tema específico, e estuda a realidade como ela se apresenta.

Segundo YIN (2001, p. 13), o Estudo de Caso é:

“uma investigação empírica que investiga um fenômeno contemporâneo dentro do seu contexto real de vida, especialmente quando as fronteiras entre o fenômeno e o contexto não são absolutamente evidentes.”

Neste Estudo de Caso, investigamos, por meio de um questionário, quais os fatores considerados de maior risco associados ao desenvolvimento de software na visão dos desenvolvedores da empresa DATAPREV.

Para Gil (1999, p. 72), o estudo de caso é caracterizado pelo estudo profundo e exaustivo de um ou de poucos objetivos, de maneira a permitir o seu conhecimento amplo e detalhado.

O estudo de caso vem sendo utilizado com freqüência cada vez maior pelos pesquisadores sociais, visto servir a pesquisas com diferentes propósitos, tais como:

- a) Explorar situações da vida real cujos limites não estão claramente definidos;
- b) Descrever a situação do contexto em que está sendo feita determinada investigação;
- c) Explicar as variáveis causais de determinado fenômeno em situações muito complexas que não possibilitam a utilização de levantamentos e experimentos.

3.3 TIPO DE ARGUMENTO

Neste trabalho, o tipo de argumento é o dedutivo.

Para Cooper e Schindler (2003, p. 48):

“ a dedução é uma forma de inferência que parece ser conclusiva, a conclusão deve necessariamente partir das razões dadas. Diz-se que essas razões implicam a conclusão e representam uma prova. [...] para que uma dedução seja correta ela deve ser verdadeira e válida: as premissas (razões) dadas para a conclusão devem concordar com o mundo real (verdadeiras); a conclusão deve necessariamente partir das premissas (válida). Uma dedução é válida desde que seja impossível que a conclusão seja falsa, se as premissas forem verdadeiras.”

Para Cooper e Schindler (2003, p. 49), a indução ocorre quando observamos um fato e perguntamos “Por que isso acontece?”. Ao respondermos esta pergunta, tentamos antecipar uma tentativa de explicação (hipótese). A dedução é o processo pelo qual testamos se a hipótese é capaz de explicar o fato.

3.4 AMOSTRA

Segundo Cooper e Schindler (2003, p. 150), a idéia básica de amostragem é que, ao selecionar alguns elementos em uma população, podemos tirar conclusões sobre toda a população. A unidade de estudo pode ser uma pessoa ou qualquer elemento que está sendo considerado para mensuração. Para estes autores, as razões principais para a amostragem são: a vantagem do custo mais baixo; maior acuidade dos resultados; maior velocidade na coleta de dados e a disponibilidade de elementos da população.

Segundo Braga, Barreto e Machado (1995, p. 35):

“as funções de pertinência ficam definidas após a consulta a pessoas especialistas no assunto, bastando 15 a 20 pessoas (comprovado experimentalmente por Zadeh). Acima de 20 pessoas, observou-se que a pertinência se mantém constante.”

A amostra considerada nesta pesquisa foi baseada nas respostas do questionário respondido por quinze profissionais da área de Tecnologia da informação da empresa DATAPREV.

3.4.1 Local da realização da pesquisa

A pesquisa foi realizada na área de TI da Divisão de administração de Dados na

DATAPREV, empresa de tecnologia de sistemas previdenciários do INSS.

3.4.2 Localização da empresa

A DATAPREV localiza-se na Rua Professor Álvaro Rodrigues, nº 460, em Botafogo, no Rio de Janeiro, RJ.

3.4.3 DATAPREV

Considerada como empresa pública, a DATAPREV é vinculada ao Ministério da Previdência Social (MPS), instituída há 33 anos, pela Lei nº 6.125 de 10 de março de 1975, com o estatuto aprovado pelo Decreto nº 75.463 de 10 de março de 1975, CNPJ 42.422.253/0001/01. Com personalidade jurídica de direito privado, a DATAPREV possui patrimônio próprio e autonomia administrativa e financeira, o seu cliente majoritário é o INSS, com o qual se vincula numa relação de interdependência.

A DATAPREV tem por objetivo principal estudar e viabilizar tecnologias de informática na área da Previdência Social, conforme publicado pelo Estatuto da Empresa DEC. 4.312 de 24 de julho de 2002. Esta empresa pública é responsável pelo processamento da maior folha de pagamento do país, alcançando mais de 20 milhões de beneficiários/mês.

A sede da Empresa fica localizada na SAS Quadra 1, Blocos E/F, Brasília/DF CEP 70070-931 PABX: (61) 3313-3076/3313-3077 FAX (61) 3321-4780, tendo como endereço na *Internet* **www.dataprev.gov.br**.

Segundo Araújo (2007, p. 51) DATAPREV, como parte integrante do contexto público, enfrenta o desafio de se transformar em uma empresa reconhecida pela excelência na prestação de serviços e, por via de consequência, pela contribuição às instituições públicas, à sociedade civil e ao mercado.

3.5 MÉTODO DE COLETA DE DADOS

3.5.1 Questionário

Para Gil (1999, p. 128), questionário é uma técnica de investigação composta por um número de questões apresentadas por escrito às pessoas que tem por objetivo o conhecimento de opiniões, crenças, sentimentos, interesses, expectativas, etc.

Segundo Gil (1999, p. 129):

“construir um questionário consiste basicamente em traduzir os objetivos da pesquisa em questões específicas, e as respostas a estas questões irão proporcionar os dados requeridos para testar as hipóteses ou esclarecer o problema da pesquisa.”

Para este autor, o questionário apresenta uma série de vantagens e limitações, são vantagens:

- a) Possibilita atingir grande número de pessoas;
- b) Implica em menores gastos;
- c) Garante o anonimato das respostas;
- d) Permite que as pessoas respondam o questionário no momento em que desejarem;
- e) Não expõe os pesquisados à influência das opiniões e ao aspecto pessoal do entrevistado.

O questionário também apresenta algumas limitações:

- a) Exclui as pessoas que não sabem ler e escrever;
- b) Impede o auxílio de informante;
- c) Impede o conhecimento das circunstâncias em que este foi respondido
- d) Não oferece garantias de todos devolvam-no devidamente preenchido;
- e) Proporciona resultados críticos em relação à objetividade, pois os itens podem ter significados diferentes para cada pesquisado.

3.5.2 Apresentação do questionário

A criação deste questionário foi baseada na categoria 2 do conceito de ciclo de vida do desenvolvimento de sistemas, *SDLC-Software Development Life Cycle*, apresentado por Turban, Mclean, Wetherbe (2004, p. 497), que corresponde a “Análise de sistemas e estudo de viabilidade”, categoria que tem como objetivo proporcionar uma compreensão profunda da empresa, suas operações e situação relevante para o sistema que se deseja construir.

O questionário foi elaborado com base nas dez categorias de risco conforme Alencar e Schmitz (2000, p. 82): organização, estimação, monitoramento, metodologia de desenvolvimento, ferramentas, cultura de riscos, usabilidade, correção, confiabilidade e pessoal, que envolverá oitenta e um fatores inibidores do desenvolvimento de *software* com o intuito de avaliar as opiniões dos quinze profissionais desenvolvedores de *software* participantes, quanto ao grau de tolerabilidade e ao grau de ocorrência destes fatores na DATAPREV.

Apesar de não existir uma metodologia para a criação de questionários, inúmeras recomendações de vários autores orientam para a formulação de um questionário de qualidade.

Para responder estas questões, foram convidados quinze profissionais da DATAPREV, desenvolvedores de *software*, e como se trata de colher opiniões pessoais destes profissionais, a confiabilidade é fundamental, pois esta empresa é considerada estratégica para o governo federal.

O objetivo deste questionário é de levantar as opiniões dos desenvolvedores sobre o grau de tolerabilidade e a frequência em que este risco ocorre.

As possibilidades de respostas foram divididas em dois grupos, são eles:

a) Grau de tolerabilidade: Não tolerável(NT); Pouco tolerável(PT); Moderadamente tolerável(MT); Bastante tolerável(BT) e Totalmente tolerável(TT).

b) Frequência de ocorrência na DATAPREV: Raramente ocorre(RO); Ocorre com pouca frequência(OPF); Ocorre às vezes(OAV); Ocorre com frequência(OCF) e Sempre podem ocorrer(SPO).

Este questionário foi apresentado em meio digital, em arquivo excel, com um texto explicativo preparado para eliminar possíveis dúvidas relacionadas às questões que compõem cada critério avaliado.

3.5.3 Determinação das variáveis lingüísticas e dos conjuntos *Fuzzy*

Como conjunto *fuzzy*, referente ao grau de tolerabilidade, foram escolhidos os termos *fuzzy* “*não tolerável (NT), pouco tolerável (PT), moderadamente tolerável (MT), bastante tolerável (BT) e totalmente tolerável (TT)*”.

Como conjuntos *fuzzy*, referente ao grau de ocorrência, foram escolhidos os termos *raramente ocorre (RO), ocorre com pouca frequência (OPF), ocorre às vezes (OAV), ocorre com frequência (OCF) e sempre pode ocorrer(SPO)*.

3.5.4 Descrição do Modelo de Identificação de Riscos

O modelo apresentado na Figura 1 está representado por elementos de aplicação de questionários, avaliação lingüística por parte dos desenvolvedores de *software* em relação aos fatores inibidores ao desenvolvimento de *software* na instituição estatal estudada, pela aplicação da matemática *fuzzy* para tratar e processar os dados subjetivos (normalização, cálculo da cardinalidade, cálculo da nebulosidade, cálculo da incerteza, cálculo da qualidade da informação), aceitação dos dados, pela criação da matriz de riscos e identificação dos riscos.

Após a identificação dos riscos, é feita uma análise qualitativa dos riscos considerados inaceitáveis, de risco alto e uma análise quantitativa descritiva.

O modelo aqui apresentado utiliza o conjunto de fatores inibidores considerados por categorias de risco em Alencar e Schmitz (2000, p. 82) e alguns conceitos da matemática *fuzzy* tratados por Braga, Barreto e Machado em 1995.

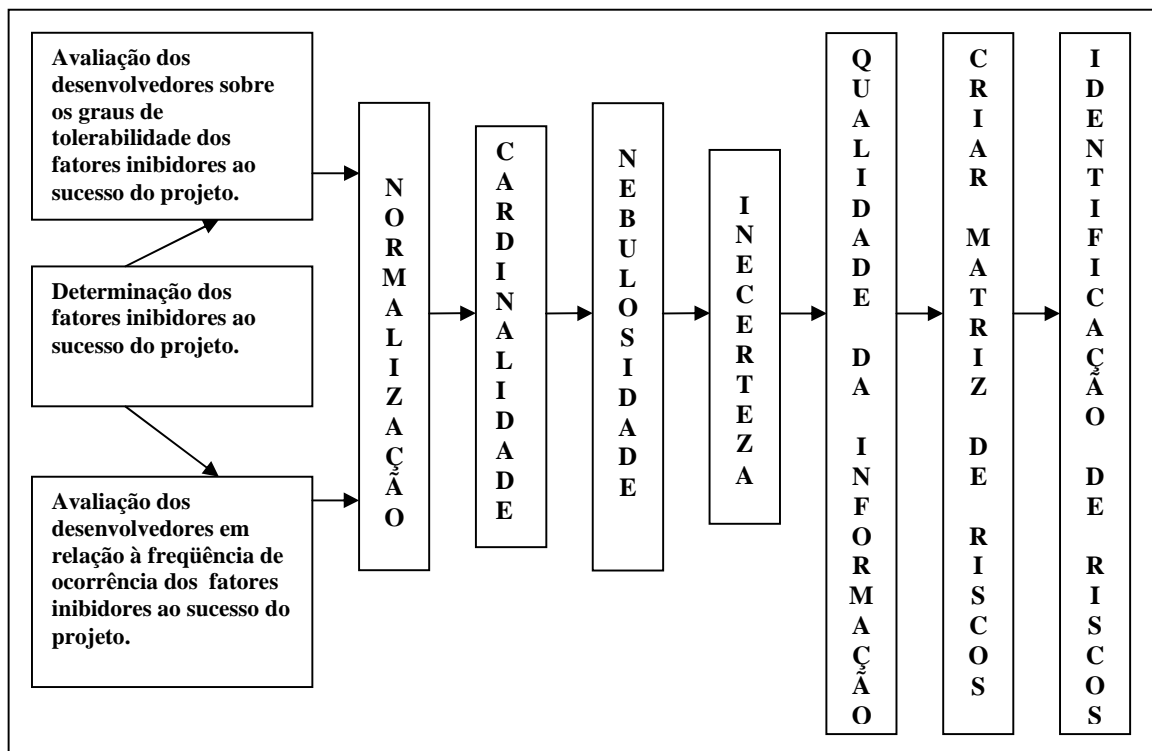


Figura 4 – Modelo de Identificação de Riscos
 Fonte: Elaboração própria

3.5.5 Identificação dos Riscos

Com o objetivo de tornar comparáveis os diferentes fatores, procederemos à normalização dos resultados, ou seja, $\mu=1$ (nível de pertinência = 1) para o conjunto *fuzzy* que somar um maior número de subjetividade (maior frequência de respostas).

Para realizar a normalização simplesmente, dividimos as respostas de cada alternativa (*raramente ocorre, ocorre com pouca freqüência,..., bastante tolerável, totalmente tolerável*) pela maior freqüência de respostas ou moda. Assim obtemos os dados que aparecem na tabela 2.

QUESTÕES	Grau de tolerabilidade					Freqüência de ocorrência				
	NT	PT	MT	BT	TT	RO	OPF	OAV	OCF	SPO
1 - Quanto à categoria "organização".										
1.1 - Planeja-se utilizar gerentes experientes em desenvolvimento de software?										
Só gerentes com pouca experiência são utilizados.	6	8	1	0	0	2	3	6	4	0
Uma mistura de gerentes com pouca e muita experiência.	0	1	3	6	5	1	3	4	6	1
Só gerentes com grande experiência são utilizados.	0	0	1	7	7	3	3	5	3	1
1.2 - Sua empresa já fez algum software similar ao que está sendo considerado?										
Nenhuma vez.	0	5	7	2	1	1	5	4	3	2
Alguma vez.	0	0	9	5	1	0	2	10	3	0
Muitas vezes.	0	2	1	3	9	2	2	4	4	3
1.3 - Existe uma estrutura organizacional formal?										
Não.	8	6	0	1	0	8	4	1	1	1
Estrutura organizacional não documentada.	3	8	3	1	0	6	2	5	2	0
Estrutura organizacional formal.	1	0	0	2	12	0	0	1	8	6
1.4 - A estrutura organizacional é estável?										
Muda freqüentemente.	1	9	4	0	0	0	2	4	4	5
Muda esporadicamente.	0	1	6	7	0	0	1	6	6	2
Estrutura estável, sem mudanças.	0	4	2	7	1	8	4	1	2	0
1.5 - Qual o grau e confiança da equipe de gerentes do projeto na entrega do software em dia, dentro do orçamento e de acordo com a										
Sem confiança.	9	4	2	0	0	0	3	5	5	2
Alguma confiança.	1	9	2	3	0	0	2	7	6	0
Alta confiança.	0	0	1	7	7	2	6	5	2	0
1.6 - Existe boa comunicação entre as diferentes organizações internas envolvidas no desenvolvimento do software?										
Comunicação pobre.	8	4	3	0	0	0	3	3	7	2
Comunicação regular.	0	5	4	6	0	0	0	7	6	2
Muito boa comunicação.	0	0	0	3	12	2	5	3	4	1
1.7 - Existe gerência de configuração de software?										
Muito pouca ou nenhuma.	5	9	1	0	0	1	2	4	8	0
Gerência de abrangência intermediária.	0	1	11	3	0	0	3	9	3	0
Gerência abrangendo todas as funcionalidades.	0	0	1	6	8	5	6	4	0	0
1.8 - Existe controle de qualidade de software?										
Muito pouco ou nenhum.	9	4	2	0	0	0	2	4	9	0
Algum controle de qualidade.	0	7	4	4	0	1	4	6	3	1
Amplio controle de qualidade.	0	0	0	3	12	7	7	1	0	0

Tabela 1 – Avaliações obtidas na categoria “organização”

Por exemplo, a categoria “organização” apresenta 8 fatores inibidores (1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7 e 1.8). No caso do fator inibidor 1.6 (Comunicação entre as diferentes organizações internas envolvidas no desenvolvimento do *software*) pode se apresentar de 3 formas de atendimento. Essa comunicação pode ser *Pobre, Regular* ou

Muito Boa. As funções de pertinências do subconjunto fuzzy “Comunicação pobre” vão ser representadas por (1) e (2), onde o sinal “+” representa *união* e não *soma*.

$$\text{Comunicação pobre} = \left(\frac{NT}{1}\right) + \left(\frac{PT}{0,5}\right) + \left(\frac{MT}{0,38}\right) + \left(\frac{BT}{0}\right) + \left(\frac{TT}{0}\right) \quad (1)$$

$$\text{Comunicação pobre} = \left(\frac{RO}{0}\right) + \left(\frac{OPF}{0,43}\right) + \left(\frac{OAV}{0,43}\right) + \left(\frac{OCF}{1}\right) + \left(\frac{SPO}{0,29}\right) \quad (2)$$

QUESTÕES	Grau de tolerabilidade					Frequência de ocorrência				
	NT	PT	MT	BT	TT	RO	OPF	OAV	OCF	SPO
1 - Quanto à categoria "organização".										
1.1 - Planeja-se utilizar gerentes experientes em desenvolvimento de software?										
Só gerentes com pouca experiência são utilizados.	0,75	1,00	0,13	0,00	0,00	0,33	0,50	1,00	0,67	0,00
Uma mistura de gerentes com pouca e muita experiência.	0,00	0,17	0,50	1,00	0,83	0,17	0,50	0,67	1,00	0,17
Só gerentes com grande experiência são utilizados.	0,00	0,00	0,14	1,00	1,00	0,60	0,60	1,00	0,60	0,20
1.2 - Sua empresa já fez algum software similar ao que está sendo considerado?										
Nenhuma vez.	0,00	0,71	1,00	0,29	0,14	0,20	1,00	0,80	0,60	0,40
Alguma vez.	0,00	0,00	1,00	0,56	0,11	0,00	0,20	1,00	0,30	0,00
Muitas vezes.	0,00	0,22	0,11	0,33	1,00	0,50	0,50	1,00	1,00	0,75
1.3 - Existe uma estrutura organizacional formal?										
Não.	1,00	0,75	0,00	0,13	0,00	1,00	0,50	0,13	0,13	0,13
Estrutura organizacional não documentada.	0,38	1,00	0,38	0,13	0,00	1,00	0,33	0,83	0,33	0,00
Estrutura organizacional formal.	0,08	0,00	0,00	0,17	1,00	0,00	0,00	0,13	1,00	0,75
1.4 - A estrutura organizacional é estável?										
Muda freqüentemente.	0,11	1,00	0,44	0,00	0,00	0,00	0,40	0,80	0,80	1,00
Muda esporadicamente.	0,00	0,14	0,86	1,00	0,00	0,00	0,17	1,00	1,00	0,33
Estrutura estável, sem mudanças.	0,00	0,57	0,29	1,00	0,14	1,00	0,50	0,13	0,25	0,00
1.5 - Qual o grau e confiança da equipe de gerentes do projeto na entrega do software em dia, dentro do orçamento e de acordo com a especificação?.										
Sem confiança.	1,00	0,44	0,22	0,00	0,00	0,00	0,60	1,00	1,00	0,40
Alguma confiança.	0,11	1,00	0,22	0,33	0,00	0,00	0,29	1,00	0,86	0,00
Alta confiança.	0,00	0,00	0,14	1,00	1,00	0,33	1,00	0,83	0,33	0,00
1.6 - Existe boa comunicação entre as diferentes organizações internas envolvidas no desenvolvimento do software?										
Comunicação pobre.	1,00	0,50	0,38	0,00	0,00	0,00	0,43	0,43	1,00	0,29
Comunicação regular.	0,00	0,83	0,67	1,00	0,00	0,00	0,00	1,00	0,86	0,29
Muito boa comunicação.	0,00	0,00	0,00	0,25	1,00	0,40	1,00	0,60	0,80	0,20
1.7 - Existe gerência de configuração de software?										
Muito pouca ou nenhuma.	0,56	1,00	0,11	0,00	0,00	0,13	0,25	0,50	1,00	0,00
Gerência de abrangência intermediária.	0,00	0,09	1,00	0,27	0,00	0,00	0,33	1,00	0,33	0,00
Gerência abrangendo todas as funcionalidades.	0,00	0,00	0,13	0,75	1,00	0,83	1,00	0,67	0,00	0,00
1.8 - Existe controle de qualidade de software?										
Muito pouco ou nenhum.	1,00	0,44	0,22	0,00	0,00	0,00	0,22	0,44	1,00	0,00
Algum controle de qualidade.	0,00	1,00	0,57	0,57	0,00	0,17	0,67	1,00	0,50	0,17
Amplio controle de qualidade.	0,00	0,00	0,00	0,25	1,00	1,00	1,00	0,14	0,00	0,00

Tabela 2 – Normalização dos resultados

Isto significa que podemos medir a posição dos desenvolvedores em um intervalo de confiança (um espaço definido por números reais de 0 a 1), onde situamos

a posição dos desenvolvedores e um nível de pertinência, ou seja, a possibilidade de que a opinião dos desenvolvedores esteja situada num determinado *subconjunto fuzzy* referente ao “*grau de tolerabilidade*” e à “*freqüência de ocorrência*”.

O fator inibidor “Comunicação pobre” está considerado pelos desenvolvedores num intervalo de confiança entre “*não tolerável (NT)*” e “*moderadamente tolerável (MT)*” para o risco do projeto. Para os desenvolvedores, o fator “Comunicação pobre” possui um nível de pertinência 1 ao subconjunto *fuzzy* “*não tolerável (NT)*” para o risco de desenvolvimento de *software* e ao mesmo tempo possui nível de pertinência 0 aos subconjuntos *fuzzy* “*bastante tolerável (BT)*” e “*totalmente tolerável (TT)*”(ver função de pertinência 1) .

Assim podemos indicar que é provável que o fator inibidor “Comunicação pobre” seja considerado *não tolerável (NT)* para o risco de desenvolvimento de *software* do que *pouco tolerável (PT)*. Da mesma forma analisamos os dados referentes à “*freqüência de ocorrência*” do fator “Comunicação pobre” durante o desenvolvimento de *software*. Neste caso, e observando a função de pertinência 2, indicamos que é provável que o fator inibidor “Comunicação pobre” *ocorra com freqüência (OCF)* durante o desenvolvimento de *software*, do que *ocorra com pouca freqüência (OPF)*, *ocorra às vezes (OAV)* ou *sempre possa ocorrer(SPO)*.

Antes de identificar os riscos devemos validar os dados. Para isto procedemos a calcular a cardinalidade (Card T, Card F), a nebulosidade (Neb T, Neb F), a incerteza (Inc T, Inc F) e a qualidade da informação (Qual T, Qual F). Em todo caso T se refere à *tolerabilidade* e F à *freqüência de ocorrência*.

Na tabela 3 aparecem representados os valores correspondentes.

Vamos verificar agora se o conjunto *fuzzy* que é o resultado das opiniões dos desenvolvedores (ver equação 1 e equação 2) é aceitável.

1 - Quanto à categoria "organização".								
1.1 - Planeja-se utilizar gerentes experientes em desenvolvimento de software?								
Só gerentes com pouca experiência são utilizados.	1,875	2,500	0,083	0,259	0,307	0,264	0,765	0,791
Uma mistura de gerentes com pouca e muita experiência.	2,500	2,500	0,185	0,259	0,281	0,264	0,781	0,791
Só gerentes com grande experiência são utilizados.	2,143	3,000	0,032	0,311	0,323	0,254	0,756	0,797
1.2 - Sua empresa já fez algum software similar ao que está sendo considerado?								
Nenhuma vez.	2,143	3,000	0,159	0,266	0,287	0,263	0,777	0,792
Alguma vez.	1,667	1,500	0,123	0,111	0,296	0,300	0,771	0,769
Muitas vezes.	1,667	3,750	0,148	0,278	0,290	0,261	0,775	0,793
1.3 - Existe uma estrutura organizacional formal?								
Não.	1,875	1,875	0,083	0,194	0,307	0,279	0,765	0,782
Estrutura organizacional não documentada.	1,875	2,500	0,194	0,185	0,279	0,281	0,782	0,781
Estrutura organizacional formal.	1,250	1,875	0,056	0,083	0,315	0,307	0,760	0,765
1.4 - A estrutura organizacional é estável?								
Muda frequentemente.	1,556	3,000	0,123	0,178	0,296	0,283	0,771	0,780
Muda esporadicamente.	2,000	2,500	0,063	0,111	0,313	0,300	0,762	0,769
Estrutura estável, sem mudanças.	2,000	1,875	0,190	0,194	0,280	0,279	0,781	0,782
1.5 - Qual o grau e confiança da equipe de gerentes do projeto na entrega do software em dia, dentro do orçamento e de acordo com a especificação?								
Sem confiança.	1,667	3,000	0,148	0,178	0,290	0,283	0,775	0,780
Alguma confiança.	1,667	2,143	0,148	0,095	0,290	0,304	0,775	0,767
Alta confiança.	2,143	2,500	0,032	0,185	0,323	0,281	0,756	0,781
1.6 - Existe boa comunicação entre as diferentes organizações internas envolvidas no desenvolvimento do software?								
Comunicação pobre.	1,875	2,143	0,194	0,254	0,279	0,266	0,782	0,790
Comunicação regular.	2,500	2,143	0,111	0,095	0,300	0,304	0,769	0,767
Muito boa comunicação.	1,250	3,000	0,056	0,266	0,315	0,263	0,760	0,792
1.7 - Existe gerência de configuração de software?								
Muito pouca ou nenhuma.	1,667	1,875	0,123	0,194	0,296	0,279	0,771	0,782
Gerência de abrangência intermediária.	1,364	1,667	0,081	0,148	0,308	0,290	0,764	0,775
Gerência abrangendo todas as funcionalidades.	1,875	2,500	0,083	0,111	0,307	0,300	0,765	0,769
1.8 - Existe controle de qualidade de software?								
Muito pouco ou nenhum.	1,667	1,667	0,148	0,148	0,290	0,290	0,775	0,775
Algum controle de qualidade.	2,143	2,500	0,190	0,259	0,280	0,264	0,781	0,791
Amplio controle de qualidade.	1,250	2,143	0,056	0,032	0,315	0,323	0,760	0,756

Tabela 3 – Cálculo da Cardinalidade, Nebulosidade, Incerteza e Qualidade da Informação

A **cardinalidade** nos ajuda a avaliar quão vago se apresenta cada posicionamento dos desenvolvedores. Assim, à maior cardinalidade teremos um posicionamento mais ambíguo, ou seja, menos rígido.

Segundo Braga, Barreto e Machado (1995, p. 29), pode-se medir a qualidade de informação a partir da medida de incerteza de um conjunto *fuzzy* $A(X)$ cuja notação utilizada é $I(A(X))$:

Para $I(A(X))=0$, certeza ($A(X))=1$;

Para $I(A(X))=1$, incerteza máxima ($A(X))=0,5$.

A **incerteza** calculada tem como maior valor 0,315, abaixo da incerteza máxima apresentada como referência por Braga, Barreto e Machado (1995, p. 29).

A **qualidade** da informação adquire valores aceitáveis acima de 0,75 considerado-se também o índice de incerteza encontrado.

Valores similares foram obtidos nas outras nove categorias de risco avaliadas.

Para realizar a normalização, dividimos as respostas de cada alternativa (*raramente ocorre, ocorre com pouca freqüência,..., bastante tolerável, totalmente tolerável*) pela maior freqüência de respostas ou moda da mesma alternativa considerada no questionário.

Podemos interpretar os resultados normalizados como se esta representação do posicionamento dos desenvolvedores a conhecemos como um subconjunto *fuzzy*, que representa a pertinência de um fator inibidor a um primeiro referencial de grau de tolerabilidade e a um segundo referencial de freqüência de ocorrência.

Podemos então medir a posição dos desenvolvedores em um intervalo de confiança, um espaço definido por números reais de 0 a 1, onde situamos a posição dos desenvolvedores e um nível de pertinência, ou seja, a possibilidade de que a opinião dos desenvolvedores esteja situada num determinado *subconjunto fuzzy* referente ao “*grau de tolerabilidade*” e a “*freqüência de ocorrência*”.

O fator inibidor “Comunicação pobre” está considerado pelos desenvolvedores num intervalo de confiança entre “*não tolerável (NT)*” e “*moderadamente tolerável (MT)*” para o risco do projeto.

Para os desenvolvedores, o fator “Comunicação pobre” possui um nível de pertinência 1 ao subconjunto *fuzzy* “*não tolerável (NT)*” para o risco de desenvolvimento

de *software* e ao mesmo tempo possui nível de pertinência 0 aos subconjuntos *fuzzy* “*bastante tolerável*” (BT) e “*totalmente tolerável*” (TT).

Assim podemos indicar que é provável que o fator inibidor “Comunicação pobre” seja considerado *não tolerável* (NT) para o risco de desenvolvimento de *software* do que *pouco tolerável* (PT).

Da mesma forma analisaremos os dados referentes à “*freqüência de ocorrência*” do fator “Comunicação pobre” durante o desenvolvimento de *software*. Neste caso, e observando a função de pertinência 2, indicamos que é provável que o fator inibidor “Comunicação pobre” *ocorra com freqüência* (OCF) durante o desenvolvimento de *software*, do que *ocorra com pouca freqüência* (OPF) ou *ocorra às vezes* (OAV) e também *sempre possa ocorrer*(SPO).

Antes de identificar os riscos devemos validar os dados. Para isto, procedemos a calcular a cardinalidade (Card T, Card F), a nebulosidade (Neb T, Neb F), a incerteza (Inc T, Inc F) e a qualidade da informação (Qual T, Qual F). Em todo caso T se refere à *tolerabilidade* e F à *freqüência de ocorrência*.

Verificaremos se o conjunto *fuzzy* resultado das opiniões dos desenvolvedores é aceitável.

A cardinalidade nos ajuda a avaliar quão vago se apresenta cada posicionamento dos desenvolvedores. Assim, a maior cardinalidade terá um posicionamento mais ambíguo, ou seja, menos rígido.

3.6 ANÁLISE QUALITATIVA DE RISCOS

3.6.1 Classificação dos riscos

Os riscos foram classificados como:

- a) Risco Alto (RA);
- b) Risco Médio Alto (RMA);
- c) Risco Médio (RM);
- d) Risco Baixo (RB).

3.6.2 Análise qualitativa de riscos

Para a análise qualitativa dos riscos foi criada a matriz de risco indicada na Figura 4. Esta matriz surgiu a partir da opinião dos desenvolvedores de *software*.

		MATRIZ DE RISCO				
Frequência	SPO	RM	RM	RM	RA	RA
	OCF	RM	RM	RM	RA	RA
	OAV	RM	RM	RM	RMA	RMA
	OPF	RB	RB	RM	RM	RM
	RO	RB	RB	RM	RM	RM
	Tolerabilidade	TT	BT	MT	PT	NT

Figura 5 – Matriz de risco
 Fonte: Elaboração própria

A opinião dos desenvolvedores é representada através das vinte e cinco regras *fuzzy* que aparecem à seguir, obtendo-se assim quatro níveis de risco.

Para a análise qualitativa dos riscos será criada uma matriz de risco. Esta matriz surgirá a partir da opinião dos desenvolvedores de *software*.

A opinião dos desenvolvedores será representada através das vinte e cinco regras *fuzzy*, obtendo-se assim quatro níveis de risco.

SE *Freqüência de Ocorrência SPO* **E** *Grau de Tolerabilidade PT* **ENTÃO** Risco Alto

SE *Freqüência de Ocorrência OCF* **E** *Grau de Tolerabilidade PT* **ENTÃO** Risco Alto

SE *Freqüência de Ocorrência SPO* **E** *Grau de Tolerabilidade NT* **ENTÃO** Risco Alto

SE *Freqüência de Ocorrência OCF* **E** *Grau de Tolerabilidade NT* **ENTÃO** Risco Alto

SE *Freqüência de Ocorrência OAV* **E** *Grau de Tolerabilidade PT* **ENTÃO** Risco Médio
Alto

SE *Freqüência de Ocorrência OAV* **E** *Grau de Tolerabilidade NT* **ENTÃO** Risco Médio
Alto

SE *Freqüência de Ocorrência SPO* **E** *Grau de Tolerabilidade TT* **ENTÃO** Risco Médio

SE *Freqüência de Ocorrência OCF* **E** *Grau de Tolerabilidade TT* **ENTÃO** Risco Médio

SE *Freqüência de Ocorrência OAV* **E** *Grau de Tolerabilidade TT* **ENTÃO** Risco Médio

SE *Freqüência de Ocorrência SPO* **E** *Grau de Tolerabilidade BT* **ENTÃO** Risco Médio

SE *Freqüência de Ocorrência OCF* **E** *Grau de Tolerabilidade BT* **ENTÃO** Risco Médio

SE *Freqüência de Ocorrência OAV* **E** *Grau de Tolerabilidade BT* **ENTÃO** Risco Médio

SE *Freqüência de Ocorrência SPO* **E** *Grau de Tolerabilidade MT* **ENTÃO** Risco Médio

SE *Freqüência de Ocorrência OCF* **E** *Grau de Tolerabilidade MT* **ENTÃO** Risco Médio

SE *Freqüência de Ocorrência OAV* **E** *Grau de Tolerabilidade MT* **ENTÃO** Risco Médio

SE *Freqüência de Ocorrência OPF* **E** *Grau de Tolerabilidade MT* **ENTÃO** Risco Médio

SE *Freqüência de Ocorrência RO* **E** *Grau de Tolerabilidade MT* **ENTÃO** Risco Médio

SE *Freqüência de Ocorrência OPF* **E** *Grau de Tolerabilidade PT* **ENTÃO** Risco Médio

SE *Freqüência de Ocorrência RO* **E** *Grau de Tolerabilidade PT* **ENTÃO** Risco Médio

SE *Freqüência de Ocorrência OPF* **E** *Grau de Tolerabilidade NT* **ENTÃO** Risco Médio

SE *Freqüência de Ocorrência RO* **E** *Grau de Tolerabilidade NT* **ENTÃO** Risco Médio

SE *Freqüência de Ocorrência OPF* **E** *Grau de Tolerabilidade TT* **ENTÃO** Risco Baixo

SE *Freqüência de Ocorrência RO* **E** *Grau de Tolerabilidade TT* **ENTÃO** Risco Baixo

SE *Freqüência de Ocorrência OPF* **E** *Grau de Tolerabilidade BT* **ENTÃO** Risco Baixo

SE *Freqüência de Ocorrência RO* **E** *Grau de Tolerabilidade BT* **ENTÃO** Risco Baixo

Nesta pesquisa apresentaremos somente os riscos inaceitáveis, isto é, Risco Alto.

3.7 ANÁLISE QUANTITATIVA DE RISCOS

Esta análise será fundamental para ajudar a empresa estatal na tomada de decisões.

Os fatores que definiremos após a análise dos resultados da pesquisa constituirão os problemas potenciais que deverão ser atacados de forma a minimizar os efeitos destes eventos negativos no desenvolvimento deste projeto.

Poderemos também prever quais as ações de contingência na hipótese dos esforços para mitigar o risco alto que poderá surgir e, como consequência, anular o seu efeito indesejado.

Observa-se que é importante que este tipo de análise seja também realizada com os fatores que contribuem para o Risco Médio Alto, Risco Médio e Risco Baixo.

A identificação dos riscos de um projeto envolve conhecimentos e experiências somente disponíveis com as pessoas, isto é, os especialistas. Esse conhecimento é subjetivo e sujeito a incertezas.

Na realização bem sucedida desta identificação, a teoria dos conjuntos *fuzzy* serve de auxílio para captar essas informações vagas, levá-las para um formato numérico e tratá-las de forma sólida, matematicamente.

O resultado almejado com a proposta permitirá ao gerente do projeto avaliar quais os fatores mais críticos (de maior risco) que deverão exigir maior supervisão.

A metodologia apresentada servirá de ferramenta para melhor considerar os efeitos que um risco pode acarretar ao projeto e conhecer quão vulnerável o projeto

está ao risco. De igual forma permite indicar qual a ordem de importância para o acompanhamento destes eventos de risco.

3.7.1 Identificação dos riscos altos

Como modo de explicação, apresentamos somente os fatores de risco considerados inaceitáveis (risco alto) que constituem 31 fatores de risco.

Estes fatores de risco estão representados na Figura 5.

SPO	1.4.1, 9.9.1	
OCF	1.7.1, 2.3.1, 2.4.1, 5.3.1, 5.4.1, 6.2.1, 6.3.1, 6.5.1, 6.9.3, 9.12.1	1.6.1, 1.8.1, 2.1.1, 3.4.1, 3.5.1, 3.6.1, 3.7.1, 4.4.1, 4.5.1, 4.7.1, 5.5.1, 5.6.1, 5.7.1, 5.8.1, 8.5.1, 9.2.1, 9.4.1, 9.9.2, 9.11.1
	PT	NT

Figura 6 – Riscos Inaceitáveis (Risco alto)

Listamos abaixo estes fatores de risco inaceitáveis, organizados por categoria, que contribuem para o atraso do projeto.

Para o tratamento destes fatores de risco, fazem-se necessárias ações mitigadoras, ou seja, deve-se decidir quais as ações preventivas permanentes para a não ocorrência dos possíveis eventos.

Identificação de riscos inaceitáveis (risco alto) por categoria.
Categoria “Organização”
1.4.1.- A estrutura organizacional muda frequentemente
1.6.1.-Existe comunicação pobre entre as organizações internas envolvidas no projeto
1.7.1.-Existe muito pouca ou nenhuma gerência de configuração de <i>software</i>
1.8.1.-Existe muito pouco ou nenhum controle de qualidade de <i>software</i>
Categoria “Estimação”
2.1.1.-O método utilizado para estimar custos e prazos é a intuição
2.3.1.- A estimativa de produtividade não é baseada em dados históricos
2.4.1.- A estimativa de prazos não é baseada em dados históricos
Categoria “Monitoramento”
3.4.1.-Não existem relatórios atualizados de custos, prazos e valor adicionado
3.5.1.-Não existe atualização periódica de relatórios de custos, prazos e valor adicionado
3.6.1.-Não existe registro normal de nenhum problema
3.7.1.-Não existem recursos específicos para resolver e registrar problemas técnicos
Categoria “Metodologia de Desenvolvimento”
4.4.1.-A metodologia inclui pouca ou nenhuma revisão e inspeção de código, projeto e requisitos
4.5.1.- A metodologia não requer planos de teste para todos os componentes de <i>software</i>
4.7.1.- Teste de regressão não são executados
Categoria “Ferramentas”
5.3.1.-Não são utilizadas ferramentas automatizadas para teste
5.4.1.- Não são utilizadas ferramentas para geração automatizadas de casos de teste
5.5.1.- Não são utilizadas ferramentas automatizadas para teste de regressão
5.6.1.- Não são utilizadas ferramentas automatizadas para teste de requisitos
5.7.1.- Não são utilizadas ferramentas automatizadas para reengenharia
5.8.1.-O ambiente de desenvolvimento é pouco estável
Categoria “Cultura de Riscos”
6.2.1.-A empresa está almejando assumir riscos de prazos adicionais por uma melhor margem
6.3.1.- A empresa está almejando assumir riscos técnicos adicionais por uma melhor margem
6.5.1.- A empresa está almejando assumir riscos de cronograma adicionais por uma melhor margem
6.9.3.-Os investimentos em tecnologia por parte da empresa são baixos
Categoria “Correção”
8.5.1.-Esperam-se muitas mudanças por mês nos requisitos especificados
Categoria “Confiabilidade”
9.2.1.-O <i>software</i> não continua quando houver um erro detectado
9.4.1.-As entradas de dados ao <i>software</i> não são validadas antes do processamento ser iniciado
9.9.1.- Nenhum modelo é utilizado para predição do nível de confiabilidade de <i>software</i>
9.9.2.-Um modelo para predição do nível de confiabilidade é utilizado sem ser o mais adequado
9.11.1.-Os testes de <i>stress</i> não são executados
9.12.1.-Os testes são executados pelos desenvolvedores

Tabela 4 – Identificação de riscos inaceitáveis (Risco alto)

3.7.2 Análise quantitativa de riscos inaceitáveis (Risco alto)

Na tabela 5 apresentamos a contribuição dos fatores de risco inaceitável (risco alto), organizados conforme as dez categorias de risco apresentadas.

Categorias	Fatores de risco alto	Total das respostas	Porcentagem que representam	Frequência Relativa em porcentagem
1-Organização	4	24	16,7	9,80
2-Estimação	3	23	13,0	9,39
3-Monitoramento	4	21	19,0	8,57
4-Metodologia de desenvolvimento	3	21	14,3	8,57
5-Ferramentas	6	27	22,2	11,02
6-Cultura de riscos	4	33	12,1	13,47
7-Usabilidade	0	18	0,0	7,35
8-Correção	1	27	3,7	11,02
9-Confiabilidade	6	36	16,7	14,69
10-Pessoal	0	15	0,0	6,12
Total	31	245	12,7	100

Tabela 5 – Contribuição dos fatores por categoria à classificação de risco inaceitável (Risco alto)

Percebe-se que as categorias “ferramentas” (22,2%), “monitoramento” (19,0%), “organização” (16,7%) e “confiabilidade” (16,7%) são as que mais contribuem para o tipo de risco alto, conforme a Figura 6.

Dos 245 fatores analisados, 12,7% foram considerados risco alto.

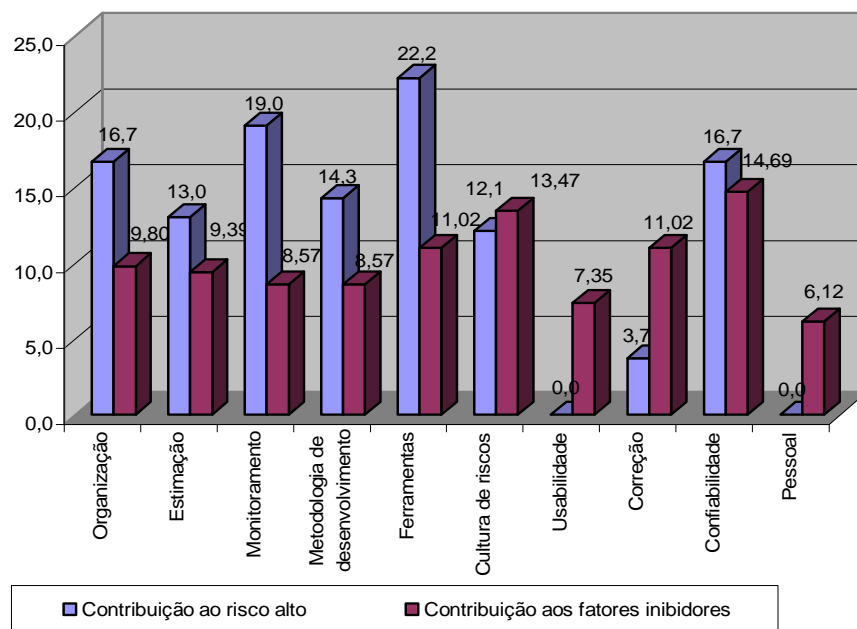


Figura 7 – Contribuição dos fatores por categoria à classificação do risco alto

Estas informações ajudam à empresa estatal na tomada de decisões. Estes 31 fatores constituem os problemas potenciais que devem ser atacados de forma a

minimizar os efeitos de eventos negativos ao projeto e prever as ações de contingência para mitigar o risco alto não surtirem o efeito desejado.

3.8 ESTUDO DA INFLUÊNCIA ENTRE OS FATORES DE RISCO ALTO

3.8.1 Matriz de influências

Como continuação desta pesquisa, buscamos conhecer qual a influência entre os fatores de risco alto. Este conhecimento é obtido através da criação da matriz de influência entre estes fatores de risco alto, e para isso, solicitamos a um dos especialistas para classificar este grau de influência conforme as variáveis lingüísticas representadas a seguir:

Variável lingüística	Avaliação
Grande influência	1,00
Muita influência	0,75
Moderada influência	0,50
Pequena influência	0,25
Nenhuma influência	0,00

Tabela 6 – Variáveis lingüísticas usadas na matriz de incidências

É importante entender que os valores de cada variável lingüística arbitrado pelo especialista é uma aproximação, e por esta razão, valores próximos aos limites de cada intervalo podem sofrer influência dos intervalos vizinhos.

A matriz de influência (Apêndice B) preenchida a partir das opiniões de um especialista é uma matriz quadrada que possui trinta e uma colunas por trinta e uma linhas e representa o nível de influência entre os fatores de risco alto. Segundo Garcia, Lazzari e Machado (1998, p. 16), consideramos que cada elemento desta matriz incide sobre si mesmo com a maior incidência, então a diagonal desta matriz estará formada pelo valor 1,00 (um), que representa a maior influência.

3.8.2 Influência entre os fatores de alto risco

A matriz de influência (Apêndice B), também conhecida como matriz de efeitos de primeira geração foi modificada (Apêndice C) usando como critério a soma de todas as influências exercidas por cada fator de risco nos demais. Esta matriz modificada está hierarquizada em ordem decrescente pelos resultados do somatório destas influências.

Para o cálculo da influência entre os fatores de alto risco utilizamos uma matriz formada pelos seis maiores somatórios retirados da matriz de efeitos de primeira geração modificada do apêndice C. Este critério foi escolhido com base no princípio de Pareto onde pode ser observado que aproximadamente 80% dos problemas são originados de aproximadamente 20% das causas, como são trinta e um fatores de alto risco, esta matriz é uma matriz quadrada formada por seis linhas e seis colunas.

<i>R</i>	5.8.1.	6.9.3.	1.6.1.	4.4.1.	1.8.1.	6.3.1.
5.8.1.-O ambiente de desenvolvimento é pouco estável	1,0	1,0	0,8	0,8	0,8	0,8
6.9.3.-Os investimentos em tecnologia por parte da empresa são baixos	1,0	1,0	0,8	0,8	0,3	0,8
1.6.1.- Existe comunicação pobre entre as organizações internas envolvidas no projeto	0,5	0,3	1,0	0,8	0,8	0,5
4.4.1.- A metodologia inclui pouca ou nenhuma revisão e inspeção de código, projeto e requisitos	0,5	0,8	0,8	1,0	0,8	0,8
1.8.1.- Existe muito pouco ou nenhum controle de qualidade de <i>software</i>	0,8	0,8	0,8	0,3	1,0	0,8
6.3.1.- A empresa está almejando assumir riscos técnicos adicionais por uma melhor margem	1,0	1,0	0,5	0,8	0,8	1,0

Tabela 7 – Matriz *R* de primeira geração

Conforme a matriz na Tabela 7, temos a influência do fator de risco alto 6.9.3-Os investimentos em tecnologia por parte da empresa são baixos, sobre o fator de risco

alto 1.8.1-Existe muito pouco ou nenhum controle de qualidade de *software*. A relação prevista entre os fatores 6.9.3 e 1.8.1 é 0,3.

A partir da matriz R obtivemos a composição *max-min*, isto é, o valor máximo do mínimo das influências, conforme apresentado na Figura 8.

Segundo Izard (2008, p. 69), para este cálculo utilizamos a composição *max-min* $\mu_{p \circ Q}(x_1, x_3) = \max (\min (\mu_p(x_1, x_2), \mu_Q(x_2, x_3)))$ onde x_1, x_2 e x_3 , são as variáveis observadas e P e Q as matrizes de efeitos de primeira e segunda geração (união dos efeitos da primeira geração e os efeitos intermediários).

Para o cálculo da composição *max-min* os valores da influência do fator de risco alto 2 (6.9.3-Os investimentos em tecnologia por parte da empresa são baixos), são comparados com os valores de influência de todos os seis fatores de risco alto no fator de risco alto 5 (1.8.1-Existe muito pouco ou nenhum controle de qualidade de *software*). Desta forma foram obtidos os valores mínimos e 0,8 é o maior valor dentre todos os valores mínimos encontrados.

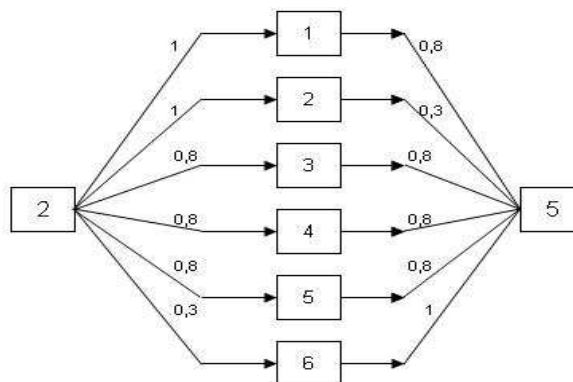


Figura 8 – Composição *max-min* da influência do fator de risco 6.9.3 sobre o fator de risco 1.8.1

Apresentamos a matriz de segunda geração, a partir da qual podemos identificar quais as relações de influência não previstas corretamente pelo especialista.

<i>R2</i>	5.8.1.	6.9.3.	1.6.1.	4.4.1.	1.8.1.	6.3.1.
5.8.1.-O ambiente de desenvolvimento é pouco estável	1,0	1,0	0,8	0,8	0,8	0,8
6.9.3.-Os investimentos em tecnologia por parte da empresa são baixos	1,0	1,0	0,8	0,8	0,8	0,8
1.6.1.- Existe comunicação pobre entre as organizações internas envolvidas no projeto	0,8	0,8	1,0	0,8	0,8	0,8
4.4.1.- A metodologia inclui pouca ou nenhuma revisão e inspeção de código, projeto e requisitos	0,8	0,8	0,8	1,0	0,8	0,8
1.8.1.- Existe muito pouco ou nenhum controle de qualidade de <i>software</i>	0,8	0,8	0,8	0,8	1,0	0,8
6.3.1.- A empresa está almejando assumir riscos técnicos adicionais por uma melhor margem	1,0	1,0	0,8	0,8	0,8	1,0

Tabela 8 – Matriz *R2* de segunda geração

Obtivemos a matriz dos efeitos esquecidos a partir da subtração da matriz *R2* pela matriz *R*.

$ R2 - R $	5.8.1.	6.9.3.	1.6.1.	4.4.1.	1.8.1.	6.3.1.
5.8.1.-O ambiente de desenvolvimento é pouco estável	0	0	0	0	0	0
6.9.3.-Os investimentos em tecnologia por parte da empresa são baixos	0	0	0	0	0,5	0
1.6.1.- Existe comunicação pobre entre as organizações internas envolvidas no projeto	0,3	0,5	0	0	0	0,3
4.4.1.- A metodologia inclui pouca ou nenhuma revisão e inspeção de código, projeto e requisitos	0,3	0,0	0	0	0	0
1.8.1.- Existe muito pouco ou nenhum controle de qualidade de <i>software</i>	0	0	0	0,5	0	0
6.3.1.- A empresa está almejando assumir riscos técnicos adicionais por uma melhor margem	0	0	0,3	0	0	0

Tabela 9 – Matriz $|R2 - R|$

É importante observar que nesta matriz $|R2 - R|$ qualquer valor diferente de zero indica que alguma correlação foi “esquecida”, isto é, não foi percebida pelo especialista

quando no momento de preenchimento destas informações, enquanto o valor zero indica que a avaliação realizada foi precisa.

No nosso exemplo o valor referente à influência esquecida é $|0,8 - 0,3| = 0,5$

A nova matriz de influência $|R2 - R| + R$, considerando os seis fatores de maior influência, está completa considerando-se toda a influência exercida por cada fator de risco alto.

Para o nosso exemplo o valor referente a toda a influência exercida entre o fator de risco alto 6.9.3 e o fator de risco alto 1.8.1 é $|0,8 - 0,3| + 0,3 = 0,8$.

$ R2 - R + R$	5.8.1.	6.9.3.	1.6.1.	4.4.1.	1.8.1.	6.3.1.	SOMA
5.8.1.-O ambiente de desenvolvimento é pouco estável	1,0	1,0	0,8	0,8	0,8	0,8	5,2
6.9.3.-Os investimentos em tecnologia por parte da empresa são baixos	1,0	1,0	0,8	0,8	0,8	0,8	5,2
1.6.1.-Existe comunicação pobre entre as organizações internas envolvidas no projeto	0,8	0,8	1,0	0,8	0,8	0,8	5,0
4.4.1. A metodologia inclui pouca ou nenhuma revisão e inspeção de código, projeto e requisitos	0,8	0,8	0,8	1,0	0,8	0,8	5,0
1.8.1.-Existe muito pouco ou nenhum controle de qualidade de <i>software</i>	0,8	0,8	0,8	0,8	1,0	0,8	5,0
6.3.1.-A empresa está almejando assumir riscos técnicos adicionais por uma melhor margem	1,0	1,0	0,8	0,8	0,8	1,0	5,4

Tabela 10 – Matriz $|R2 - R| + R$

4. ANÁLISE DOS RESULTADOS

A empresa DATAPREV está exposta a um risco alto de falhar no processo de desenvolvimento de software devido à presença dos fatores de risco identificados por categorias. Percebe-se que a categoria “ferramentas” contribui com seis fatores (22,2%), “monitoramento” com quatro fatores (19,0%), “organização” também com quatro (16,7%) e “confiabilidade” com seis (16,7%). São as categorias que mais contribuem para o tipo de risco alto (ver Tabela 5).

Em relação à categoria com a maior contribuição ao risco alto, “ferramentas” com 22,2%, os fatores de risco são seis:

5.3.1.- Não são utilizadas ferramentas automatizadas para teste;

5.4.1.- Não são utilizadas ferramentas para geração automatizadas de casos de teste;

5.5.1.- Não são utilizadas ferramentas automatizadas para teste de regressão;

5.6.1.- Não são utilizadas ferramentas automatizadas para teste de requisitos;

5.7.1.- Não são utilizadas ferramentas automatizadas para reengenharia;

5.8.1.-O ambiente de desenvolvimento é pouco estável.

Ao consultar as respostas dos questionários, observamos que os especialistas entendem que o fato de não utilizar estas ferramentas é não tolerável ou pouco tolerável, demonstrando o conhecimento que eles possuem na eficiência destas ferramentas no tratamento do risco.

Observamos também que na Tabela 10, referente à influência real entre os fatores, a maior influência no risco alto, é a do fator 6.3.1 (A empresa está almejando assumir riscos técnicos adicionais por uma melhor imagem) da categoria “Cultura de riscos”. Nesta categoria, todas as questões relacionadas com a possibilidade da empresa assumir algum tipo de risco, recebeu a resposta “pouco tolerável” da maioria

dos especialistas, demonstrando que eles têm consciência da vulnerabilidade da DATAPREV em relação ao risco.

5. CONCLUSÕES

Atualmente, a demanda no desenvolvimento de *software* vem aumentando na DATAPREV, conseqüentemente, aumentam também as exigências de qualidade, tempo e custos. Daí a importância para a empresa contar com uma técnica capaz de identificar os riscos no desenvolvimento de *software*.

A identificação dos riscos de um projeto envolve conhecimentos e experiências somente disponíveis em pessoas. Esse conhecimento é subjetivo e sujeito a incertezas. Na realização bem sucedida desta identificação, a teoria dos conjuntos *fuzzy* serviu de auxílio para captar essas informações vagas, levá-las para um formato numérico e tratá-las de forma sólida, matematicamente.

Neste trabalho foi aplicado o método *fuzzy* com o objetivo de identificar e analisar os fatores que podem provocar efeitos negativos durante o desenvolvimento de *software* na DATAPREV. O método *fuzzy* se mostrou eficaz ao tratar as informações subjetivas e imprecisas obtidas a partir de desenvolvedores de *software* nesta empresa, levá-las para um formato numérico obtendo os fatores que contribuem para o risco alto, risco médio alto, risco médio e risco baixo.

A partir da matriz de relação *fuzzy* utilizada no trabalho foi possível conhecer que a empresa está almejando assumir riscos técnicos adicionais, os quais explicam a baixa cultura em relação ao risco durante o desenvolvimento de *software* na DATAPREV.

6. RECOMENDAÇÕES PARA FUTUROS ESTUDOS

Para trabalhos futuros recomendamos o seguinte:

- 1) Realizar o mesmo tipo de análise com os fatores que contribuem para o Risco Médio Alto, Risco Médio e Risco Baixo.
- 2) Fazer o mesmo estudo em outra empresa da iniciativa privada para que possamos compará-la com esta empresa pública e conhecer a visão que cada uma tem sobre o Risco.
- 3) Pesquisar uma empresa em que a principal atividade, o *Core Business*, seja o desenvolvimento de *software* para que possamos comparar com esta empresa pública e conhecer a visão que cada uma tem sobre o Risco.
- 4) Utilizar outras ferramentas de análises para comparar os resultados com os obtidos neste trabalho.

REFERÊNCIAS BIBLIOGRÁFICAS

ALBERTIN, A.; ALBERTIN, R. M. de M. (Org). **Tecnologia da informação**: desafios da tecnologia de informação aplicada aos negócios. São Paulo: Atlas, 2005.

ALENCAR, A. J.; SCHMITZ, E. A. **Análise de Risco em Gerência de Projetos**. Rio de Janeiro: Brasport, 2005.

ANDRADE, E. L. **Introdução à Pesquisa operacional**: Métodos e Modelos para análise de Decisão, 2. ed. Rio de Janeiro: LTC, 2000.

AQUINO, M. A. **O Campo da Ciência da informação**: gênese, conexões e especificidade, João Pessoa: Editora Universitária/UFPB, p.9-24, 2002.

ARAUJO, C. R. de. **Avaliação dos graus de importância dos critérios ergonômicos para a interação homem-computador**. 127 f. Dissertação (Mestrado em Administração)-Universidade Estácio de Sá, Rio de Janeiro, 2007.

BARRETO, A. A. Uma quase história da ciência da informação. **Datagrama – Revista da Ciência da Informação**, Rio de Janeiro, RJ, v. 9, n. 2, abril 2008. Disponível em: www.datagramazero.org.br. Acesso em 30 agosto 2008.

BENAMARTI, J.; LEDERER, A. An impirical study of IT management and rapid IT change. In: 1999 ACM SIGCPR conference on Computer personnel research. Disponível em: www.acm.org. Acesso em 15 maio 2008.

BERNSTEIN, P. L. **Desafio aos deuses**: a fascinante história do risco, 20. ed. Rio de Janeiro: Elsevier, 1997.

BOORSTIN, D. J. **Os descobridores**: de como o homem procurou conhecer-se a si mesmo e ao mundo. Rio de Janeiro: Editora Civilização Brasileira, 1983.

BORGES, L. M. S.; FALBO, R. A. **Gerência de Conhecimento sobre Processos de Software**. UFES, 2001.

BRAGA, M. J. F.; BARRETO, J. M.; MACHADO, M. A. S. **Conceitos da Matemática Nebulosa na Análise de Risco**. Rio de Janeiro: Artes & Rabiscus, 1995. p. 15-31.

CAMPOS, V. F. **Gerenciamento da rotina do trabalho do dia-a-dia**. Belo Horizonte: Editora de Desenvolvimento Gerencial, 2001.

CÂMARA, F. **SDLC – Software Development Life Cycle**, 2008. Disponível em: www.linhadecodigo.com.br. Acesso em 15 setembro 2008.

COOPER, D. R.; SCHINDLER, P. S. **Métodos de pesquisa em administração**. 7. ed. Porto Alegre: Bookman, 2003.

COSTA, H. R.; BARROS, M. O.; TRAVASSOS, G. H. Uma abordagem econômica baseada em riscos para avaliação de uma carteira de projetos de software. In: 19º Congresso Brasileiro de Engenharia de software – SBES 2005. Uberlândia - Minas Gerais, 2005.

CUNHA, R. M. Avaliação de Riscos na Emissão de Propostas Comerciais usando Lógica *Fuzzy*. In: Aguiar H. O. J. et al. Inteligência Computacional Aplicada à Administração, Economia e Engenharia em Matlab São Paulo: Ed. Thomson Learning, p.281-319, 2007.

DAVENPORT, T. **Reengenharia de Processos**: como inovar a empresa através da tecnologia da informação. Rio de Janeiro: Campus. 1999.

DAVIS, S; MEYER, C. **BLUR**: a velocidade da mudança na economia integrada. Rio de Janeiro: Campus, 1999.

DEMO, P. Introdução à metodologia da ciência. São Paulo: Atlas, 1985.

DEZIN, N. K.; LINCOLN, Y. S. **O Planejamento da Pesquisa Qualitativa**: Teorias e Abordagens, 2. ed. Porto Alegre: Artmed, 2006.

ENGEL A.; LAST M. Modeling software testing costs and risks using fuzzy logic paradigm. Journal of Systems and Software, v. 80, Issue 6, June 2007. p. 817-835.

ESPINHA, R.; SOUZA, J. Melhorando processos através da análise de risco e conformidade. Revista engenharia de software, Rio de Janeiro, ano 1, ed. Especial, p.10-21, 2007.

FALBO, R. A. Integração de Conhecimento em um Ambiente de Desenvolvimento de Software. Tese de doutorado, COPPE/UFRJ, Rio de Janeiro, 1998.

FERNANDES, J. H. C. Qual a Prática do Desenvolvimento de Software? *Ciência e Cultura*, Vol. 55, n. 2, p. 29-33, abr./jun. 2003.

GADELHA, A. Uma pequena história da probabilidade. Curso de Pós-graduação em Estatística, DME/IM/UFRJ, Rio de Janeiro, mar. 2004.

GARCIA, P. S.; LAZZARI, L. L.; MACHADO, E. A. M. Una propuesta *fuzzy* para definir indicadores de pobreza. In: V Congresso de SIGEF. Lausanne – Suíça, 1998.

GIL, A. C. Métodos e Técnicas de Pesquisa Social, 5. ed. São Paulo: Atlas, 1999.

GUSMÃO, C. M. G.; MOURA, H. P. **Gerência de Risco em Processos de Qualidade de Software**: uma Análise Comparativa. In: III Simpósio Brasileiro de Qualidade de Software – SBQS 2004. ISBN 858844276-0. Brasília - Distrito Federal, 2004.

HALL, E. M., Managing Risk. 2ª Ed. Addison Wesley, 1998.

IZARD, I. R. S. **Indicação das ações empresariais a partir da percepção do consumidor:** uso da lógica *fuzzy*. 117 f. Dissertação (Mestrado em Administração)- Universidade Estácio de Sá, Rio de Janeiro, 2008.

JUNIOR, E. C. B.; BELCHIOR, A. D. Um modelo para a avaliação da qualidade da gerência de projetos de software. In: I Simpósio Brasileiro de Qualidade de Software – SBQS 2002. Gramado – Rio Grande do Sul, 2002.

KIRAN, N. R.; RAVI, V. Software reliability prediction by soft computing techniques. *Journal of Systems and Software*, v. 81, Issue 4, April 2008. p. 576-583.

KROLL, P.; KRUCHTEN, P. *The Rational Unified Process Made Easy: A Practitioner's Guide to the RUP*. The Addison-Wesley Object Technology Series, 2003.

KUCHTA, D. Use of fuzzy numbers in project risk (criticality) assessment. *International Journal of Project Management*. Elsevier Science Ltd and IPMA, v. 19, p. 305-310, 2001.

LEE, H. Group decision making using fuzzy sets theory for evaluating the rate of aggregative risk in software development. *Fuzzy Sets and Systems*, v.80, Issue 3, June 1996. p. 261-271.

LIMA, M.P.; AMORIM, C. A. Proposta de um modelo fuzzy para apoio à tomada de decisão no Controle de Tráfego Aéreo do Aeroporto Internacional de Salvador. UNEB, 2007. Disponível em: www.sbc.org.br. Acesso em 08 novembro 2008.

LIU, F.; KANE, G.; BAMBOON, M. An intelligent early warning system for software quality improvement and project management. *Journal of Systems and Software*, v. 79, Issue 11, November 2006. p. 1552-1564.

LOPES, C. E.; MEIRELLES, E. O Desenvolvimento da Probabilidade e da Estatística. In: XVIII Encontro Regional de Professores de Matemática – LEM/IMECC/UNICAMP 2005. Campinas – São Paulo, 2005

MACHADO, L. F. D. C. Modelo para definição de processos de software na estação TABA. Tese de Mestrado, COPPE/UFRJ, Rio de Janeiro, 2000

MAT.UFRGS. Instituto de Matemática da UFRGS. Início da matematização das probabilidades, 2001. Disponível em: www.mat.ufrgs.br. Acesso em 13 julho 2008.

McGEE, J.; PRUSAK, L. **Gerenciamento estratégico da informação:** aumente a competitividade e a eficiência de sua empresa utilizando a informação como uma ferramenta estratégica. Rio de Janeiro: Campus, 1994.

MORÉ, J. D. Aplicação da lógica fuzzy na avaliação da confiabilidade humana nos ensaios não destrutivos por ultra-som. Tese de Doutorado, COPPE/UFRJ, Rio de Janeiro, 2004.

NASCIMENTO, C. J. **A Evolução da Qualidade no Setor de Software Brasileiro: Quatro Biênios Medindo e Acompanhando Indicadores de Gestão.** MCT - Secretaria de Política de Informática, 2000. Disponível em: www.mct.gov.br/sepin. Acesso em 15 maio 2008.

NBR ISO 12207, 1998. ISO 12207 – Tecnologia da Informação – Processos de ciclo de vida de software. Associação Brasileira de Normas Técnicas – ABNT. Rio de Janeiro.

OLIVEIRA, M.; MOTA, F. R. L.; ALVARADO, R. U. Comunidade Científica e Cientificidade da Ciência da informação. In: 8º Congresso Nacional de Bibliotecários, Arquivistas e Documentalistas – VIII ABAD. Estoril – Portugal, 2004.

PINHO, V. D. M., MENDONÇA, M. G. Análise do tratamento de riscos em projetos de desenvolvimento de software de uma organização, 2005. Disponível em: www.sbbd-sbes2005.ufu.br Acesso em 03 maio 2008.

PMBok- *A Guide to the Project Management Body of Knowledge*. 3rd Edition: Project Management Institute, 2004.

REIS, A. F., ALBUQUERQUE, A. R. P. O estado da arte em gerenciamento de riscos em projetos. XI SIMPEP, Bauru, São Paulo, 08-10 novembro 2004.

SALLES JR., C. A. C.; SOLER, A. M.; VALLE, J. A. S.; RABECHINI JR. Gerenciamento de Riscos em projetos. Rio de Janeiro: FGV, 2006.

SCHMITZ, E. A.; ALENCAR, A. J.; VILLAR, C. B. Modelos Qualitativos de análise de Risco para Processos de tecnologia da informação. Rio de Janeiro: Brasport, 2006.

SETZER V. W. Data, Information, Knowledge and Competency. **Datagrama – Revista de Ciência da Informação**, Rio de Janeiro, RJ, n. 0, dezembro 1999. Disponível em: www.datagramazero.org.br. Acesso em 30 agosto 2008.

SILVA, A. N. Gerenciamento Pró-Ativo de Redes de Computadores baseado em Agentes Móveis e Lógica Fuzzy. Universidade Federal de Pernambuco, Recife, 2001.

SILVA, F. M. **INFORMATION SCIENCE: a science for Scientific and Technological information?** Texto para disciplina Formas de Estruturação e Mediação da Informação Institucionalizada do programa de pós-graduação em Ciência da Informação, USP, São Paulo, 2007.

TURBAN, E., MCLEAN E.; WETHERBE, J. **Tecnologia da Informação para Gestão: Transformando os negócios na economia digital.** 3. ed. Porto Alegre: Bookman, 2004.

VARGAS, R.V. **Gerenciamento de Projetos**: Estabelecendo diferenciais competitivos. 6 ed. Rio de Janeiro: Brasport, 2005.

WHITE, D.; FORTUNE, J. **Current Practice in Project Management**: An Empirical Study. *International Journal of Project Management* 20, Elsevier Science Ltd and IPMA, p. 1-11, 2002.

XU, Z.; KHOSHGOFTAAR, T. M.; ALLEN, E. B. Application of fuzzy expert systems in assessing operational risk of software. *Information and Software Technology*, v. 45, Issue 7, May 2003. p. 373-388.

YIN, R. K. **Estudo de caso**: planejamento e métodos, 2. ed. São Paulo: Bookman, 2001.

ZADEH L.A., Fuzzy Sets. *Information and Control*, 8, 1965. p. 338-353.

SITES CONSULTADOS

BRDESCOSAÚDE. Disponível em: www.Brdescosaude.com.br. Acesso em 01 julho 2008.

CAPES. Disponível em: www.capes.org.br. Acesso em 10 outubro 2008.

EDUCAR. Disponível em: www.educar.sc.usp.br. Acesso em 13 junho 2008.

FECAP. Fundação Escola de comércio Álvares Penteado. O questionário na pesquisa científica, 2000. Disponível em: www.fecap.br. Acesso em: 19 julho 2008.

IME-UFRJ. Instituto de Matemática e Estatística. Disponível em: www.ime.ufrj/informatica.htm. Acesso em: 31 agosto 2008

PUC-SP. Pontifícia Universidade Católica de São Paulo. Histórico da Lógica “Fuzzy”. www.pucsp.br. Acesso em 01 maio 2008.

PUC-RGS. Pontifícia Universidade Católica do Rio Grande do Sul. Biografia de Vilfredo Pareto. www.pucrs.br/famat/statweb/historia/daestatistica/biografias/pareto.htm. Acesso em 15 novembro 2008.

NEO/USP – Núcleo de estudos em Otimização. Universidade de São Paulo. www.lps.usp.br/neo/fuzzy. Acesso em 03 maio 2008.

SEPIN – Relatório da qualidade e produtividade de software. Ministério da Ciência e Tecnologia, 2002. Disponível em: www.mct.gov.br. Acesso em: 03 maio 2008

QUESTÕES	Grau de tolerabilidade					Frequência de ocorrência na DATAPREV				
	NT	PT	MT	BT	TT	RO	OPF	OAV	OCF	SPO
2 - Quanto à categoria " estimacão "										
2.1 - Qual o método de estimacão utilizado?										
Intuição										
Analogia										
Preço para vencer.										
Ditado pelas circunstâncias.										
Outros.										
2.2 - Utiliza-se um modelo de custo de software ?										
Nenhum modelo é utilizado.										
Utiliza um modelo, porém não de forma totalmente correta.										
Utiliza um modelo de forma adequada.										
2.3 - A estimativa de produtividade é baseada em dados históricos?										
Não										
Sim, mas dados são de projetos não similares.										
Sim, com dados de projetos similares.										
2.4 - A estimativa de prazos é baseada em dados históricos?										
Não										
Sim, mas dados são de projetos não similares.										
Sim, com dados de projetos similares.										
2.5 - A estimativa são revistas mensalmente ou mais frequentemente?										
Nenhuma atualização periódica										
Atualizadas, porém não mensalmente.										
Atualizadas mensalmente ou mais frequentemente.										
2.6 - Quão acuradas são as estimativas de custos passadas comparando com os custos atuais?										
Mais ou menos 100% de diferença.										
Mais ou menos 50% de diferença.										
Mais ou menos 5% de diferença.										
2.7 - Quão acuradas são as estimativas de prazo passadas comparando com os prazos atuais?										
Mais ou menos 100% de diferença.										
Mais ou menos 50% de diferença.										
Mais ou menos 5% de diferença.										

QUESTÕES	Grau de tolerabilidade					Frequência de ocorrência na DATAPREV				
	NT	PT	MT	BT	TT	RO	OPF	OAV	OCF	SPO
3 - Quanto à categoria " monitoração".										
3.1 - Existem metas intermediárias (milestones) definidas para cada fase?										
Nenhuma meta.										
Algumas, mas não para todas as fases.										
Existem metas intermediárias que permitem acompanhar todas as fases.										
3.2 - Existe WBS (Work Breakdown Structure) para acompanhar cada parte do desenvolvimento do software ?										
Não										
Utiliza WBS mas não para acompanhar custos nem orçamento.										
Utiliza WBS adequadamente.										
3.3 - Existe sistema de monitoramento para custo, prazo e valor adicionado?										
Não										
Existe sistema de monitoramento, porém ineficiente.										
Sim.										
3.4 - Existem relatórios atualizados de custos, prazos e valor adicionado?										
Não										
Sim., sem atualização mensal.										
Sim.										
3.5 - Os relatórios de custo, prazo e valor adicionado são atualizados mensalmente?										
Nenhuma atualização periódica.										
Atualizadas, porém não mensalmente.										
Atualizadas mensalmente ou mais frequentemente.										
3.6 - Existe registro formal de problemas?										
Nenhum registro formal.										
Registro formal com atualização acima de semanal.										
Registro formal com atualização semanal ou mais frequentemente.										
3.7 - Existem recursos para resolver e registrar problemas técnicos?										
Nenhum recurso específico.										
Existem recursos, porém sem atualização quinzenal.										
Existem recursos com atualização quinzenal ou mais frequentemente.										

QUESTÕES	Grau de tolerabilidade						Frequência de ocorrência na DATAPREV					
	NT	PT	MT	BT	TT	RO	OPF	OAV	OCF	SPO		
5 - Quanto à categoria "ferramentas" .												
5.1 - Os desenvolvedores estão treinados nas ferramentas utilizadas?												
Não.												
Algum treinamento.												
Totalmente treinados.												
5.2 -São utilizadas ferramentas automatizadas para projeto do software?												
Não.												
Alguma ferramenta é utilizada ou a que é utilizada não corresponde adequadamente.												
Ferramentas são utilizadas e são adequadas.												
5.3 -São utilizadas ferramentas automatizadas para teste?												
Não.												
Algumas ferramentas ou a que é utilizada não corresponde adequadamente.												
Ferramentas são utilizadas e são adequadas.												
5.4 -São utilizadas ferramentas para geração automatizadas de casos de teste?												
Não.												
Algumas ou a que é utilizada não corresponde adequadamente.												
Ferramentas são utilizadas e são adequadas.												
5.5 -São utilizadas ferramentas automatizadas para testes de regressão?												
Não.												
Algumas ou a que é utilizada não corresponde adequadamente.												
Ferramentas são utilizadas e são adequadas.												
5.6 -São utilizadas ferramentas automatizadas para testes de requisitos?												
Não.												
Algumas ou a que é utilizada não corresponde adequadamente.												
Ferramentas são utilizadas e são adequadas.												
5.7 -São utilizadas ferramentas automatizadas para reengenharia?												
Não.												
Algumas ou a que é utilizada não corresponde adequadamente.												
Ferramentas são utilizadas e são adequadas.												
5.8 - Qão estável é o ambiente de desenvolvimento?												
Pouco estável.												
Estável, com problemas conhecidos e documentados.												
Muito estável e sem problemas.												
5.9 - As ferramentas estão disponíveis a qualquer hora?												
Não.												
Algumas.												
Todas.												

QUESTÕES	Grau de tolerabilidade						Frequência de ocorrência na DATAPREV					
	NT	PT	MT	BT	TT	RO	OPF	OAV	OCF	SPO		
8 - Quanto à categoria "correção".												
8.1 - Todos os requisitos foram identificados e documentados?												
Não, nenhum.												
Sim, paulatinamente.												
Sim, totalmente.												
8.2 - Todos os requisitos foram identificados no projeto de software?												
Não.												
Algum.												
Todos.												
8.3 - Os requisitos serão identificados na codificação?												
Não.												
Sim, alguns.												
Sim, todos.												
8.4 - Os requisitos serão identificados nos procedimentos de teste?												
Não.												
Sim, alguns.												
Sim, todos.												
8.5 - Esperam-se muitas mudanças nos requisitos especificados?												
Sim, muitas (mais de 10 por mês).												
Sim, moderadas (5 por mês)												
Sim, poucas (1 por mês)												
8.6 - O projeto de software será rastreado no código?												
Não.												
Algumas das características serão rastreadas.												
Todas as características serão rastreadas.												
8.7 - O projeto de software será rastreado nos procedimentos de teste?												
Não.												
Algumas das características serão rastreadas.												
Todas as características serão rastreadas.												
8.8 - Todos os itens de ações do software serão implementados antes da entrega ao cliente?												
Não.												
Alguns.												
Sim, todos.												
8.9 - Todos os procedimentos de teste serão executados antes da entrega ao cliente?												
Não, nenhum.												
Paulatinamente.												
Sim, todos.												

QUESTÕES	Grau de tolerabilidade					Frequência de ocorrência na DATAPREV				
	NT	PT	MT	BT	TT	RO	OPF	OAV	OCF	SPO
10 - Quanto à categoria "pessoal".										
10.1 - A equipe está identificada e disponível?										
Não										
Algum.										
Todos.										
10.2 - Como é a experiência da equipe no software a ser desenvolvido?										
Nenhuma experiência.										
Alguns possuem experiência.										
Todos são experientes.										
10.3 - Como é a experiência da equipe no ambiente de desenvolvimento do software?										
Nenhuma experiência.										
Alguns possuem experiência.										
Todos são experientes.										
10.4 - Como é a experiência da equipe na linguagem de implementação?										
Nenhuma experiência.										
Alguns possuem experiência.										
Todos são experientes.										
10.5 - Quantas pessoas estarão trabalhando em determinado momento durante o desenvolvimento do software?										
Número muito grande (> 1000).										
Intermediário (30 a 50).										
Pequeno (menos de 20).										

Identificação dos trinta e um riscos considerados inaceitáveis (risco alto):

- 1.4.1.- A estrutura organizacional muda freqüentemente;
- 1.6.1.- Existe comunicação pobre entre as organizações internas envolvidas no projeto;
- 1.7.1.- Existe muito pouca ou nenhuma gerência de configuração de software;
- 1.8.1.- Existe muito pouco ou nenhum controle de qualidade de software;
- 2.1.1.- O método utilizado para estimar custos e prazos é a intuição;
- 2.3.1.- A estimativa de produtividade não é baseada em dados históricos;
- 2.4.1.- A estimativa de prazos não é baseada em dados históricos;
- 3.4.1.- Não existem relatórios atualizados de custos, prazos e valor adicionado;
- 3.5.1.- Não existe atualização periódica de relatórios de custos, prazos e valor adicionado;
- 3.6.1.- Não existe registro normal de nenhum problema;
- 3.7.1.- Não existem recursos específicos para resolver e registrar problemas técnicos;
- 4.4.1.- A metodologia inclui pouca ou nenhuma revisão e inspeção de código, projeto e requisitos;
- 4.5.1.- A metodologia não requer planos de teste para todos os componentes de software;
- 4.7.1.- Testes de regressão não são executados;
- 5.3.1.- Não são utilizadas ferramentas automatizadas para teste;
- 5.4.1.- Não são utilizadas ferramentas para geração automatizadas de casos de teste;
- 5.5.1.- Não são utilizadas ferramentas automatizadas para teste de regressão;
- 5.6.1.- Não são utilizadas ferramentas automatizadas para teste de requisitos;
- 5.7.1.- Não são utilizadas ferramentas automatizadas para reengenharia;
- 5.8.1.-O ambiente de desenvolvimento é pouco estável;
- 6.2.1.-A empresa está almejando assumir riscos de prazos adicionais por uma melhor margem;
- 6.3.1.- A empresa está almejando assumir riscos técnicos adicionais por uma melhor margem;
- 6.5.1.- A empresa está almejando assumir riscos de cronograma adicionais por uma melhor margem;
- 6.9.3.-Os investimentos em tecnologia por parte da empresa são baixos;

- 8.5.1.-Esperam-se muitas mudanças por mês nos requisitos especificados;
- 9.2.1.-O software não continua quando houver um erro detectado;
- 9.4.1.-As entradas de dados ao software não são validadas antes do processamento ser iniciado;
- 9.9.1.- Nenhum modelo é utilizado para predição do nível de confiabilidade de software;
- 9.9.2.-Um modelo para predição do nível de confiabilidade é utilizado sem ser o mais adequado;
- 9.11.1.-Os testes de stress não são executados;
- 9.12.1.-Os testes são executados pelos desenvolvedores.

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)