



SENAI CIMATEC

**PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM
COMPUTACIONAL E TECNOLOGIA INDUSTRIAL**
Mestrado em Modelagem Computacional e Tecnologia Industrial

Dissertação

**MODELO PARA AVALIAÇÃO DE RISCOS EM
PROJETOS DE SOFTWARE**

Apresentada por: Alex de Oliveira Coelho
Orientador: Dr. Hernane Borges de Barros Pereira
Co-orientador: Prof. Dr. Marcelo Albano Moret Simões Gonçalves

Fevereiro/2014

Alex de Oliveira Coelho

MODELO PARA AVALIAÇÃO DE RISCOS EM PROJETOS DE SOFTWARE

Dissertação apresentada ao Programa de Pós-graduação em Modelagem Computacional e Tecnologia Industrial, Curso de Mestrado em Modelagem Computacional e Tecnologia Industrial do SENAI CIMATEC, como requisito parcial para a obtenção do título de **Mestre em Modelagem Computacional e Tecnologia Industrial**.

Área de conhecimento: Interdisciplinar

Orientador: Dr. Hernane Borges de Barros Pereira
SENAI CIMATEC

Co-orientador: Prof. Dr. Marcelo Albano Moret
SENAI CIMATEC

Salvador
SENAI CIMATEC
2014

Nota sobre o estilo do PPGMCTI

Esta dissertação foi elaborada considerando as normas de estilo (i.e. estéticas e estruturais) propostas, aprovadas pelo colegiado do Programa de Pós-graduação em Modelagem Computacional e Tecnologia Industrial e disponíveis em formato eletrônico (*download* na Página Web http://ead.fieb.org.br/portal_faculdades/dissertacoes-e-teses-mcti.html ou solicitação via e-mail à secretaria do programa) bem como no formato impresso somente para consulta.

Ressalta-se que o formato proposto considera diversos itens das normas da Associação Brasileira de Normas Técnicas (ABNT), entretanto opta-se, em alguns aspectos, seguir um estilo próprio elaborado e amadurecido pelos professores do Programa supracitado.

SENAI CIMATEC

Programa de Pós-graduação em Modelagem Computacional e Tecnologia Industrial

Mestrado em Modelagem Computacional e Tecnologia Industrial

A Banca Examinadora, constituída pelos professores abaixo listados, leram e recomendam a aprovação [com distinção] da Dissertação , intitulada “MODELO PARA AVALIAÇÃO DE RISCOS EM PROJETOS DE SOFTWARE”, apresentada no dia 10 de fevereiro de 2014, como requisito parcial para a obtenção do título de **Mestre em Modelagem Computacional e Tecnologia Industrial**.

Orientador:

Prof. Dr. Dr. Hernane Borges de Barros Pereira
SENAI CIMATEC

Co-orientador:

Prof. Dr. Marcelo Albano Moret
SENAI CIMATEC

Agradecimentos

Agradeço a minha família pelo apoio ao longo dessa caminhada, aos profissionais do SENAI BAHIA, pelo apoio na realização dessa pesquisa. Agradeço também aos professores da Faculdade SENAI CIMATEC, especialmente ao professor Dr. Hernane Pereira pelo conhecimento compartilhado nessa etapa, ao colega Diego Stéfano e ao Professor Dr. Eduardo Manuel de Freitas Jorge.

Salvador, Brasil
10 de fevereiro de 2014

Alex de Oliveira Coelho

Resumo

Para gestores de projetos de software, significa um grande desafio controlar de forma eficaz os seus recursos, assim como gerenciar pessoas e analisar os riscos envolvidos nos projetos. Apesar dos avanços na área de Engenharia de Software, nota-se que o processo de produção ainda impõe uma série de desafios. Essa pesquisa busca contribuir com a gestão de projetos de software, propondo um modelo para calcular riscos em projetos de software. No trabalho foram realizadas as etapas de (i) seleção de projetos para a pesquisa, (ii) definição de variáveis de influência nos riscos e (iii) aplicação do modelo para calcular os graus de risco do projeto, considerando variáveis humanas e de tecnologia.

Palavras-Chave: Avaliação de Riscos, Desenvolvimento de Software, Lógica Fuzzy.

Abstract

For managers of software projects, significant challenges effectively control their resources as well as manage people and analyze the risks involved in the projects. Despite advances in software engineering, we note that the production process also imposes a number of challenges. This research seeks to contribute to the management of software projects, and proposed a model to calculate the degree of risk in software projects. At work the steps of (i) selection of projects for research, (ii) definition of variables influence the risks and (iii) application of the model to calculate the degree of risk of the project were conducted. Keywords: Risk Assessment, Software Development, Fuzzy Logic.

Sumário

1	Introdução	1
1.1	Definição do Problema	1
1.2	Objetivo	2
1.3	Metodologia	2
1.4	Organização	3
2	Métricas	4
2.1	Métricas, Medição e Cálculo	4
2.2	Métricas Técnicas de Software	5
2.3	Coleta e Métricas Funcionais de Software	6
2.3.1	Análise de Pontos de Função (Function Point Analysis - FPA)	8
2.3.2	Tamanho Funcional do Software	8
2.3.3	Contagem de Pontos de Função	9
2.4	Fronteira da Aplicação, Escopo e Contagem	9
2.4.1	Fronteira da Aplicação	9
2.4.2	Escopo da Aplicação	10
2.4.3	Contagem de Funções de Dados	10
2.4.4	Análise de Pontos de Função Extendida - (Extend Function Point Analysis - XFPA)	10
3	Gestão de Projetos de Software	12
3.1	Gestão de Projetos	12
3.1.1	Processos no PMBOK	13
3.1.2	Ferramentas para Gestão de Projetos	13
3.2	Gerenciamento de Riscos	14
3.2.1	Identificação de Riscos	15
3.2.2	Avaliação dos Riscos	17
3.2.3	Riscos de Software	18
3.3	Trabalhos Correlatos - Riscos e Lógica Fuzzy	18
3.4	Engenharia de Software	21
3.4.1	Modelos de Desenvolvimento de Software	21
3.5	Projetos de Software	22
3.5.1	Pessoas	22
3.5.2	Escopo do Software	23
3.5.3	Produto	23
3.5.4	Estrutura de Divisão do Trabalho	23
3.5.5	Estimativas	24
3.5.6	Elaboração de Cronograma	25
3.5.7	Custos do Projeto	25
4	Lógica Difusa (Fuzzy)	26
4.1	Conjuntos Fuzzy	27
4.1.1	Teoria de Conjuntos Fuzzy	28
4.2	Modelos Linguísticos Fuzzy	29
4.2.1	Sistemas Fuzzy	30

4.3	Controladores Fuzzy	31
4.4	O Modelo Fuzzy Mamdani	32
4.4.1	Processo de Fuzzificação	32
4.4.2	Processo de Defuzzificação	32
4.4.3	Sistemas Lógicos Fuzzy	33
4.5	Exemplos	34
4.5.1	Primeiro exemplo: uma variável de entrada e uma de saída	35
4.5.2	Segundo exemplo: duas variáveis de entrada e uma de saída	35
5	Modelo Proposto	38
5.1	Informações do Modelo	38
5.2	Variáveis Linguísticas do Sistema e Base de Regras	41
5.2.1	Variável Linguística Mudanças no Escopo ou Objetivos Mal Definidos	42
5.2.2	Variável Linguística Complexidade do Software	43
5.2.3	Variável Linguística Volatilidade dos Requisitos	44
5.3	Variáveis Linguísticas de Entrada dos Recursos Humanos	45
5.3.1	Variável Linguística Terceirização de Pessoal	45
5.3.2	Variável Linguística Equipe em Projetos Paralelos	46
5.3.3	Variável Linguística Rotatividade de Pessoal	47
5.4	Cálculo dos Graus de Risco e Função de Pertinência	48
5.5	Resultados e Discussão	50
5.5.1	Projeto 1 - Rede Social para o Governo do Estado da Bahia	50
5.5.2	Projeto 2 - Sistema Gestor de Informação	59
5.5.3	Projeto 3 - Sistema de Capacitação em Educação Ambiental	67
5.5.4	Projeto 4 - Treinamento Online em Informática Básica Para Deficientes Visuais	75
5.5.5	Pesquisa com Profissionais e Índice de Acerto do Modelo	83
6	Considerações finais	85
6.1	Conclusões	85
6.2	Contribuições	86
6.3	Atividades Futuras de Pesquisa	87
	Referências	88

Lista de Tabelas

2.1	Exemplo em métricas e produtividade. Fonte: Demarco (1991)	6
2.2	Exemplo de métrica de software. Fonte: Kriesers (2011)	7
3.1	Exemplo de tabela de risco para aplicação funcionando na Internet - Fonte: Autor	18
5.1	Grau de risco e semântica equivalente - Fonte: Autor	41

Lista de Figuras

3.1	Fluxo de monitoramento e planejamento de riscos - Fonte: Autor.	16
4.1	Funções de Pertinência para Conjuntos Fuzzy: triangular (topo), trapezoidal (centro) e Gaussiana. Fonte: Autor.	29
4.2	Controlador Fuzzy	32
4.3	Métodos usados na defuzzificação - Fonte: (KLIR; YUAN, 1996)	33
4.4	Sistema Fuzzy com Base em um Modelo Linguístico. Fonte: Zadeh (1965).	33
4.5	Motor de Inferência Fuzzy. Fonte: Zadeh (1965).	34
4.6	Sistema Mamdani com associação “min” (esquerda) de graus de pertinência (direita). Fonte: Zadeh (1965).	35
4.7	Variáveis linguísticas e curvas de entrada-saída para o Exemplo 1. Fonte: Jan (2004).	36
4.8	Variáveis linguísticas e curvas de entrada-saída para o exemplo 2. Fonte: Jan (2004).	37
5.1	Equipe participante da pesquisa. Fonte: Autor.	39
5.2	Esquema do sistema Fuzzy - As variáveis de entrada são inseridas, o processamento é realizado, atendendo as regras definidas e o grau de risco é indicado - Fonte: Autor	40
5.3	Forma para grupo de regras Fuzzy - Fonte: Autor.	41
5.4	Regras para Escopo ou Objetivos Mal Definidos - Fonte: Autor.	43
5.5	Regras para Complexidade do Software - Fonte: Autor.	44
5.6	Regras para Complexidade do Software - Fonte: Autor.	45
5.7	Regras para Terceirização de Pessoal - Fonte: Autor.	46
5.8	Regras para Equipe em Projetos Paralelos - Fonte: Autor.	47
5.9	Regras para Rotatividade de Pessoal - Fonte: Autor.	48
5.10	Função de pertinência para Grau de Risco das Variáveis do Sistema. Fonte: Autor.	49
5.11	Função de pertinência para Grau de Risco das Variáveis dos Recursos Humanos. Fonte: Autor	49
5.12	Função de pertinência Triangular. Fonte: Autor	49
5.13	Gráfico de Cenário Previsto para Variável do Sistema Escopo ou Objetivos Mal Definidos. Fonte: Autor.	51
5.14	Gráfico de Cenário Previsto para Variável do Complexidade do Software. Fonte: Autor.	52
5.15	Gráfico de Cenário Previsto para Variável Requisitos Voláteis. Fonte: Autor.	52
5.16	Gráfico de Cenário Realizado para Variável Escopo ou Objetivos Mal Definidos. Fonte: Autor.	53
5.17	Gráfico de Cenário Previsto para Variável do Complexidade do Software. Fonte: Autor.	54
5.18	Gráfico de Cenário Previsto para Variável Requisitos Voláteis. Fonte: Autor.	54
5.19	Gráfico de Cenário Previsto para Variável dos Recursos Humanos - Terceirização de Pessoal. Fonte: Autor.	55
5.20	Gráfico de Cenário Previsto para Variável dos Recursos Humanos - Equipe em Projetos Paralelos. Fonte: Autor.	56

5.21	Gráfico de Cenário Previsto para Variável dos Recursos Humanos - Rotatividade da Equipe. Fonte: Autor.	56
5.22	Gráfico de Cenário Realizado para Variável dos Recursos Humanos - Terceirização de Pessoal. Fonte: Autor.	57
5.23	Gráfico de Cenário Realizado para Variável dos Recursos Humanos - Equipe em Projetos Paralelos. Fonte: Autor.	58
5.24	Gráfico de Cenário Realizado para Variável dos Recursos Humanos - Rotatividade da Equipe. Fonte: Autor.	58
5.25	Gráfico de Cenário Previsto para Variável do Sistema Escopo ou Objetivos Mal Definidos. Fonte: Autor.	60
5.26	Gráfico de Cenário Previsto para Variável do Complexidade do Software. Fonte: Autor.	60
5.27	Gráfico de Cenário Previsto para Variável Requisitos Voláteis. Fonte: Autor.	61
5.28	Gráfico de Cenário Realizado para Variável Escopo ou Objetivos Mal Definidos. Fonte: Autor.	62
5.29	Gráfico de Cenário Previsto para Variável do Complexidade do Software. Fonte: Autor.	62
5.30	Gráfico de Cenário Previsto para Variável Requisitos Voláteis. Fonte: Autor.	63
5.31	Gráfico de Cenário Previsto para Variável dos Recursos Humanos - Terceirização de Pessoal. Fonte: Autor.	64
5.32	Gráfico de Cenário Previsto para Variável dos Recursos Humanos - Equipe em Projetos Paralelos. Fonte: Autor.	64
5.33	Gráfico de Cenário Previsto para Variável dos Recursos Humanos - Rotatividade da Equipe. Fonte: Autor.	65
5.34	Gráfico de Cenário Realizado para Variável dos Recursos Humanos - Terceirização de Pessoal. Fonte: Autor.	65
5.35	Gráfico de Cenário Realizado para Variável dos Recursos Humanos - Equipe em Projetos Paralelos. Fonte: Autor.	66
5.36	Gráfico de Cenário Realizado para Variável dos Recursos Humanos - Rotatividade da Equipe. Fonte: Autor.	66
5.37	Gráfico de Cenário Previsto para Variável do Sistema Escopo ou Objetivos Mal Definidos. Fonte: Autor.	68
5.38	Gráfico de Cenário Previsto para Variável do Complexidade do Software. Fonte: Autor.	68
5.39	Gráfico de Cenário Previsto para Variável Requisitos Voláteis. Fonte: Autor.	69
5.40	Gráfico de Cenário Realizado para Variável Escopo ou Objetivos Mal Definidos. Fonte: Autor.	70
5.41	Gráfico de Cenário Previsto para Variável do Complexidade do Software. Fonte: Autor.	70
5.42	Gráfico de Cenário Previsto para Variável Requisitos Voláteis. Fonte: Autor.	70
5.43	Gráfico de Cenário Previsto para Variável dos Recursos Humanos - Terceirização de Pessoal. Fonte: Autor.	72
5.44	Gráfico de Cenário Previsto para Variável dos Recursos Humanos - Equipe em Projetos Paralelos. Fonte: Autor.	72
5.45	Gráfico de Cenário Previsto para Variável dos Recursos Humanos - Rotatividade da Equipe. Fonte: Autor.	73
5.46	Gráfico de Cenário Realizado para Variável dos Recursos Humanos - Terceirização de Pessoal. Fonte: Autor.	73
5.47	Gráfico de Cenário Realizado para Variável dos Recursos Humanos - Equipe em Projetos Paralelos. Fonte: Autor.	74

5.48	Gráfico de Cenário Realizado para Variável dos Recursos Humanos - Rotatividade da Equipe. Fonte: Autor.	74
5.49	Gráfico de Cenário Previsto para Variável do Sistema Escopo ou Objetivos Mal Definidos. Fonte: Autor.	76
5.50	Gráfico de Cenário Previsto para Variável do Complexidade do Software. Fonte: Autor.	77
5.51	Gráfico de Cenário Previsto para Variável Requisitos Voláteis. Fonte: Autor.	77
5.52	Gráfico de Cenário Realizado para Variável Escopo ou Objetivos Mal Definidos. Fonte: Autor.	78
5.53	Gráfico de Cenário Previsto para Variável do Complexidade do Software. Fonte: Autor.	79
5.54	Gráfico de Cenário Previsto para Variável Requisitos Voláteis. Fonte: Autor.	79
5.55	Gráfico de Cenário Previsto para Variável dos Recursos Humanos - Terceirização de Pessoal. Fonte: Autor.	80
5.56	Gráfico de Cenário Previsto para Variável dos Recursos Humanos - Equipe em Projetos Paralelos. Fonte: Autor.	80
5.57	Gráfico de Cenário Previsto para Variável dos Recursos Humanos - Rotatividade da Equipe. Fonte: Autor.	81
5.58	Gráfico de Cenário Realizado para Variável dos Recursos Humanos - Terceirização de Pessoal. Fonte: Autor.	82
5.59	Gráfico de Cenário Realizado para Variável dos Recursos Humanos - Equipe em Projetos Paralelos. Fonte: Autor.	82
5.60	Gráfico de Cenário Realizado para Variável dos Recursos Humanos - Rotatividade da Equipe. Fonte: Autor.	83
5.61	Pesquisa com Profissionais de T.I. de Salvador. Fonte: Autor.	84

Lista de Siglas

ASSESPRO .	Associação das Empresas Brasileiras de Tecnologia da Informação
BFPUG	Brazilian Function Point Users Group
BRASSCOM	Associação Brasileira das Empresas de Tecnologia da Informação e Comunicação
CAPM	Certied Associate in Project Management
CMMI	Capability Maturity Model Integration
COBIT	Control Objectives for Information and Related Technology
ERP	Enterprise Resource Planning
ES	Engenharia de Software
FAPESB	Fundação de Amparo à Pesquisa do Estado da Bahia
FINEP	Financiadora de Estudos e Projeto
FLS	Sistemas Lógicos Fuzzy
FPA	Function Point Analysis (análise de pontos de função)
IEEE	Institute of Electrical and Electronic Engineers
IFPUG	Grupo Internacional de Usuários de Pontos de Função
ISO	International Organization for Standardization
ITIL	Information Technology Infrastructure Library
PCU	Use Case Points
PMBok	Project Management Body of Knowledge
PMI	Project Management Institute
PPGMCTI ..	Programa de Pós-graduação em Modelagem Computacional e Tecnologia Industrial
SWOT	Strengths, Weaknesses, Opportunities, Threats
TI	Tecnologia da Informação
XFPA	Extended Function Point Analysis (Análise de Pontos de Função Estendida)
XP	eXtreming Programming
WWW	World Wide Web

Introdução

O processo de criação de um software é uma atividade complexa e influenciada por uma série de variáveis. Questões como tempo de produção de uma aplicação, produtividade de uma linguagem de software, metodologias de desenvolvimento envolvidas ou participação da equipe em projetos simultâneos merecem maior atenção dos gestores de projetos. Essas dificuldades podem ser ignoradas por gestores de projetos de T.I. (tecnologia da informação), os quais são pressionados regularmente com prazos apertados, equipes reduzidas e problemas relacionados a escassez de mão de obra em alguns Estados do Brasil.

De acordo com uma pesquisa recente realizada pelo [IDC \(2012\)](#), o Brasil tem uma carência de 39,9 mil profissionais de tecnologia. Até 2015 esse número pode crescer para 117 mil vagas. Segundo a pesquisa, as principais razões para o déficit são a rápida expansão das empresas do setor, adoção acelerada de serviços de T.I., Copa do Mundo e Olimpíadas no Brasil. Em nosso Estado o quadro não é diferente. Dados da [ASSESPRO \(2013\)](#), Associação das Empresas Brasileiras de Tecnologia da Informação, indicam grande carência de programadores na Bahia, especialmente nas linguagens como JAVA,.NET e na área de games.

A qualidade de um produto de software requer adequada gestão de pessoas, acompanhamento constante de metas, pessoal qualificado e acompanhamento dos custos. O projeto pode ser influenciado por dificuldades técnicas que afetam negativamente a produção das aplicações, os chamados riscos, a exemplo de: prazos mal estimados, falhas na identificação de requisitos ou custos estimados de forma errada.

1.1 Definição do Problema

Dentro do contexto apresentado e iniciando o estudo de gestão de projetos de software, observa-se que existem problemas no processo que não são claramente identificados pelos gestores. Diversos esforços são realizados para construção de projetos de software, buscando atender às necessidades dos clientes, aos requisitos de qualidade e aos objetivos previamente especificados. Desde a década de 80, empresas e estudiosos do setor tentam criar métodos e modelos para quantificar, reduzir riscos e produzir softwares mais alinhados com as necessidades dos clientes., No início da década de 90, [Kemerer \(1993\)](#), realizou pesquisas utilizando Análise por Pontos de Função (FPA). Os estudos tinham

como objetivo quantificar de forma mais eficiente os custos do software.

No trabalho de Avaliação de Riscos em Desenvolvimento de Software ([LEOPOLDINO, 2004](#)), estudos sobre riscos confrontaram a percepção de riscos entre programadores e gerentes de projetos, fazendo uma análise crítica sobre o estado da arte da gerência de riscos. A referida pesquisa, no entanto, não definiu variáveis para aplicação de modelo computacional e não teve como proposta realizar o cálculo de grau de riscos de um software, escopo principal da presente pesquisa.

O desenvolvimento dessa pesquisa nos remete a uma reflexão que pode auxiliar os gestores de projetos. Como avaliar o grau de risco de um projeto e identificar potenciais fatores de insucesso? A análise de riscos pode ser utilizada em projetos de software para reduzir custos e dar mais eficiência ao processo. Segundo [Vieira \(2005\)](#), conhecer os conceitos e princípios básicos do gerenciamento de custos ajuda o gestor a apresentar e discutir melhor os benefícios dos projetos em termos financeiros e técnicos.

1.2 Objetivo

O objetivo deste trabalho é: (i) propor um modelo para calcular os graus de risco de um projeto de software, visando auxiliar empresas e gestores que atuam neste segmento. A pesquisa propõe um estudo de temas relacionados à tecnologia e aos recursos humanos envolvidos nos projetos. A pesquisa objetiva, ainda, (ii) definir um grupo de variáveis para auxílio dos gestores na identificação de riscos do projeto.

O modelo poderá servir como apoio à tomada de decisão, ao indicar os graus de risco nos projetos de software e possíveis ajustes que podem ser realizados para minimizar esses riscos. Sabe-se que os riscos em projetos são inerentes ao processo de criação de software, no entanto devem ser tratados e validados. Segundo [Pressman \(2005\)](#), do ponto de vista técnico, o processo de software tem um conjunto de tarefas que são necessárias para se construir um software com alta qualidade e isso inclui também métodos técnicos, avaliação de riscos, bem como o uso de ferramentas automatizadas.

1.3 Metodologia

Para aplicação e validação do modelo, foram selecionados quatro projetos de software realizados entre os anos de 2011 e 2012. As informações foram coletadas junto ao Núcleo de Educação à Distância do SENAI Bahia (Serviço Nacional de Aprendizado Industrial). A unidade é responsável pela produção de sistemas educacionais, aplicações

de mídia center e games eletrônicos. A equipe desenvolvedora era formada por profissionais de diversas áreas de tecnologia da informação, como analistas, programadores e web designers. O grupo também incluía pessoal sem formação técnica na área de software, a exemplo de pedagogos, engenheiros e jornalistas. Os projetos usados na pesquisa foram selecionados de forma aleatória.

A pesquisa foi desenvolvida em três etapas. A primeira fase compreendeu a revisão bibliográfica sobre Projetos de Software, Engenharia de Software e Lógica Difusa. A segunda fase definiu variáveis para aplicação na presente pesquisa, incluindo variáveis do sistema e variáveis humanas. Na última etapa, foi realizado o cálculo dos graus de risco dos projetos.

1.4 Organização

Esta dissertação foi estruturada em seis capítulos, organizados da seguinte forma:

- Introdução;
- Métricas de Software: apresenta conceitos de métricas de software, modelos de desenvolvimento e contagem por pontos de função;
- Gestão de Projetos de Software: gestão de projetos, gestão de riscos e engenharia de software;
- Lógica Fuzzy: conceitos de lógica difusa, modelos linguísticos, controladores e modelos fuzzy;
- Modelo Proposto: descrição do modelo a ser utilizado, definição das variáveis de entrada, cálculo dos graus de risco e resultados encontrados e
- Considerações Finais: análise dos resultados alcançados e perspectivas de pesquisas futuras indicadas.

Métricas

2.1 Métricas, Medição e Cálculo

Métrica de software é a medição de um atributo (propriedades ou características) de uma entidade (produto, processo ou recursos). [Sommerville \(2003\)](#) afirma que: “uma métrica de software é qualquer tipo de medição que se refira a um sistema de software, processo ou documentação relacionada.”. As métricas indicam quantidade através de uma unidade de medida padronizada. Já a medição, é o processo de medir ou de obter uma métrica. As métricas podem ser usadas para avaliar a qualidade de um produto ou serviço.

As medições podem ser aplicadas ao software para propor melhorias e aperfeiçoar a eficiência no desenvolvimento do produto. São exemplos de medições: tempo gasto em horas para ir de uma cidade a outra, a quantidade de patentes registradas em um determinado período do ano ou a quantidade de notebooks vendidos em uma cidade. Tratando-se de software, pode-se estimar a quantidade de registros de um banco de dados, o tempo gasto no processamento de uma informação na internet, o número de linhas de código de um sistema ou a quantidade de erros encontrados em uma aplicação.

A medição é caracterizada por cinco atividades, conforme listadas a seguir ([ROCHE, 1994](#)):

- Formulação: derivação de medidas e métricas de software que são adequadas para representar o software em análise;
- Coleta: possibilidade de acumular dados para originar as métricas;
- Análise: usar métricas e ferramentas matemáticas;
- Interpretação: avaliar os resultados das métricas, visando qualidade na representação;
- Realimentação: aplicar recomendações de interpretação das métricas que são transmitidas à equipe de criação.

O cálculo representa o processo de medir e chegar a um resultado final. Pode-se, por exemplo, calcular o número de imóveis residenciais construídos na cidade do Salvador, no período de janeiro a dezembro de 2012. Essas informações podem ser usadas pela Administração Pública para medir o crescimento populacional, calcular índice de pobreza ou estabelecer implantação de novos postos de saúde.

As medidas de software e as métricas não são absolutas. Segundo [Fenton e Kaposi \(1987\)](#), “medição é o processo pelo qual os números ou símbolos são associados aos atributos de entidades do mundo real de modo a determiná-los de acordo com regras claramente definidas.”

Empresas que atuam com desenvolvimento de software podem aplicar técnicas de medição para identificar oportunidades de melhorias em seus processos. Essas empresas buscam diferencial no mercado através de certificações de qualidade, a exemplos de CMMI (*Capability Maturity Model Integration*) [CMMI \(2013\)](#) ou ISO (*International Organization for Standardization*) [ISO \(2013\)](#), atendendo aos requisitos solicitados pelos clientes.

2.2 Métricas Técnicas de Software

Segundo [Peters e Pedrycs \(2001\)](#), as métricas técnicas de software servem como um guia para os gestores de projeto, pois permitem avaliar a qualidade do software, verificar o atendimento aos requisitos e concentrar esforços no escopo do projeto. Essas informações podem apoiar os gestores e projetistas de software, permitindo analisar dados e verificar os problemas na gestão do processo.

Para alcançar níveis satisfatórios de qualidade, é necessário melhorar as etapas do ciclo de desenvolvimento ([OMAN; PFLEEGER, 1997](#)). A realização de medições foi reconhecida como pré-requisito indispensável para se introduzir a disciplina da engenharia ao desenvolvimento, manutenção e uso de produtos de software ([BASILI; WEISS, 1984](#)).

Para empresas de desenvolvedoras de software, é um desafio constante atender a prazos cada vez mais reduzidos e obter bons índices de qualidade. Elaborar produtos nessas condições mencionadas é o objetivo principal de muitas empresas do setor. Apesar da evolução no uso de métricas, o mercado ainda busca metodologias para torná-las mais eficientes e com implementação mais simplificada.

As métricas descritas a seguir podem ser usadas para caracterizar um projeto, realizar medição e propor melhorias em um processo de software. O uso dessas informações possibilita identificar ajustes necessários para otimizar recursos, realocar membros da equipe, identificar treinamentos necessários ou precificar o software. As métricas relacionadas a seguir foram extraídas da literatura da área ([CARLETON et al., 1992](#); [TAJIMA; MATSUBARA, 1984](#)):

- **M1 - Tempo:** quantidade de horas utilizadas na modelagem de um banco de dados, reuniões de alinhamento de projetos, homologação de versões iniciais, tempo

destinado a retrabalho por erros no código ou tempo usado em conversão do código para uma linguagem orientada a objetos

- **M2 - Esforço** - tempo demandando para execução do projeto nas fases de análise, testes de protótipos ou reuniões não planejadas.
- **M3 - Tamanho do Sistema**: quantidade de linhas de códigos ou número de registros de um banco de dados.
- **M4 - Número de Erros**: número de erros encontrados em um sistema, total de erros registrados durante a análise de versões de protótipos.
- **M5 - Produtividade**: quantidade de linhas de código produzida por unidade de esforço, número de bancos de dados modelados durante um período de tempo.
- **M6 - Rotatividade do time**: número percentual de colaboradores que saem da empresa durante as etapas de desenvolvimento.
- **M7 - Experiência**: tempo de experiência em programação para Internet.

Ao selecionar algumas dessas métricas, os resultados podem atender a diversos objetivos que buscam melhoria no processo de software. Isso pode ser feito analisando tamanho do sistema ou número de erros encontrados em versões iniciais. Dessa forma, os gestores podem comparar os resultados com projetos anteriores e, assim, corrigir falhas para processos futuros. Um exemplo sobre a indicação de métricas e produtividade é apresentado na tabela 2.1. Nessa tabela, observa-se alguns pontos que podem ser medidos nos projetos, a exemplo de, custo por hora, quantidade de páginas acessadas, número de pessoas envolvidas no projeto e o número de erros encontrados.

Tabela 2.1: Exemplo em métricas e produtividade. Fonte: Demarco (1991)

Projeto	Esforço	Custo	KLOC	Págs	Erros	Pessoas
AAA-01	24	168	12,1	365	29	3
CCC-04	62	440	27,2	1224	86	5
FFF-03	43	314	20,2	1050	64	6

2.3 Coleta e Métricas Funcionais de Software

A coleta das métricas que serão usadas pode ser obtida com o histórico dos projetos realizados. O processo de coleta não é um atividade simples, pois exige dedicação da equipe, criação de um histórico do projeto e coleta de dados como horas gastas na execução, quantidade de erros no código ou produtividade dos programadores.

Para aproveitamento das métricas aplicadas, torna-se necessária a criação de um relatório histórico do projeto, o qual deve conter as atividades realizadas por cada membro e pela gerência envolvida. Segundo [Maldonado, Rocha e Weber \(2001\)](#), é possível medir adequadamente usando métricas, ao se utilizar uma planilha de atividades que devem considerar os diferentes tipos de tarefas, tais como análise, projeto, codificação, testes ou retrabalho.

A análise dos dados coletados deve atender a dois propósitos: um é tentar melhorias no processo, fazendo adaptações na gestão e execução dos projetos de software. O outro objetivo é comparar métricas obtidas em projetos anteriores, realizando estudos empíricos e propondo modificações nos novos valores. Esse estudo comparativo deve resultar em ações que permitam melhorias no processo, revisões de etapas de produção e ajustes no desenvolvimento.

A tabela [2.2](#) relaciona exemplos de métricas de software. Nessa tabela são definidas a variável e sua métrica equivalente. Na primeira linha, por exemplo, o custo por membro do time é identificado com o valor por hora de cada um dos profissionais envolvidos.

Tabela 2.2: Exemplo de métrica de software. Fonte: [Kriesers \(2011\)](#)

Tópico	Métrica
Custo por membro do time	Custo por hora de um profissional do time, como programador, DBA, Analista, Revisor ou profissional de suporte. É possível relacionar experiência profissional e titulação por colaborador.
Erros de código	Quantidades de erros encontrados durante o processo de testes ou após a revisão ou utilização no cliente.
Qualidade Subjetiva	Facilidade de uso, facilidade de operação, aceitação do cliente. Embora sejam confusas, as maneiras de tentar a quantificação foram planejadas.
Facilidade de Manutenção	Custo por linha de código ou ponto de função por ano.
Capacidade Tecnológica	<ul style="list-style-type: none"> • Ferramentas - uma organização deve saber quais são mais usadas e o nível de conhecimento dos funcionários sobre as que não são tão usadas. • Maturidade do Processo - onde se encontra a taxa da organização na escala SEI CMM, por exemplo? • Capacidade do Domínio - em que domínios do aplicativo a organização é capaz de atuar?
Medidas de Melhoria do Processo	<ul style="list-style-type: none"> • Tempo e esforço da execução do processo. • Taxas de defeito, estatística de análise causal, taxas fixas, retalhamento e retrabalho.

2.3.1 *Análise de Pontos de Função (Function Point Analysis - FPA)*

A necessidade de quantificar e mensurar dados relacionados ao processo de produção do software sempre foi tema de estudo de diversos autores. Desde a década de 60, profissionais que atuavam no desenvolvimento de sistemas buscavam formas mais eficientes de medir aquilo que o cliente pretendia e estabelecer de forma mais justa os preços dos projetos. Estudos nesse sentido começaram a surgir em 1970, quando um grupo de pesquisadores da IBM dos Estados Unidos, liderado por Alan Albrecht, criou uma metodologia de contagem para estimar o tamanho de um software.

Nesse período, foi decidido que o modelo de Análise de Pontos de Função (FPA) seria testado na própria IBM. Essa metodologia foi aplicada em empresas de software em diversos países e até hoje é utilizada. No início da década de 80, a técnica de contagem de pontos de função resultou em um manual, patrocinado na época pela IBM e foi usado internamente nos projetos da empresa.

O objetivo da contagem dos pontos de função é poder mensurar o tamanho de um determinado software, usando uma contagem de linhas de código. A metodologia não avalia produtividade, metodologia ou outro tipo de análise relacionada ao programa. A contagem dos pontos de função independe da linguagem, metodologia de desenvolvimento ou capacidade de processamento dos computadores, ou seja, mede somente o tamanho funcional do software.

2.3.2 *Tamanho Funcional do Software*

O tamanho funcional de um sistema é método de medição para medir um software. Atualmente existem diversas formas de medição, mas a contagem de pontos de função é a mais antiga e segue uma série de requisitos lógicos apresentados no manual de Práticas de Contagem, elaborado pelo Grupo Internacional de Usuários de Pontos de Função ([IFPUG, 2010](#)). O primeiro trabalho publicado pelo grupo foi chamado de Guidelines to Software Measurement, lançado em 1994. A organização atua em diversos países, oferecendo apoio através de livros, manuais e ferramentas para contagem de pontos, além de estabelecer normas e parâmetros. No Brasil, a entidade é representada pelo BFPUG - Brazilian Function Point Users Group ([BFPUG, 2012](#)).

A Análise de Pontos de Função visa também medir projetos de construção, manutenção e funcionalidades do software que serão entregues aos usuários. Esses objetivos devem atender ao propósito principal da metodologia, que é verificar (a) o tamanho e (b) o custo do projeto.

2.3.3 Contagem de Pontos de Função

Para realizar a contagem de pontos, o manual (IFPUG, 2010) classifica os seguintes elementos funcionais:

- E1 - Entrada Externa (EI, *external input*) - são as transações lógicas onde os dados entram e mantem os dados internos;
- E2 - Saída Externa (EO, *external output*) - transações lógicas onde os dados saem da aplicação para fornecer informações aos usuários;
- E3 - Consulta externa (EQ, *external query*) - transação lógica onde uma entrada solicita uma resposta da aplicação;
- E4 - Arquivos lógicos internos (ILF, *internal logical files*) - grupo lógico de dados mantidos pela aplicação;
- E5 - Arquivos de interface externa (EIF, *external interface files*) - grupo lógico de dados mantidos pela aplicação, porém mantidos por outra aplicação.

Para realizar a contagem dos pontos de função, recomenda-se determinar qual tipo será utilizada. São possíveis três tipos de contagem, segundo o (IFPUG, 2010):

- T1 - Contagem de projeto de desenvolvimento - contagem com as informações fornecidas pelos usuários na primeira versão do software. Esta contagem inclui as funções de conversões de dados que são necessárias para implantar um software.
- T2 - Contagem de projeto de melhoria (manutenção) - conta as atualizações necessárias e trata das funcionalidades de uma aplicação, para aplicar melhorias.
- T3 - Contagem de aplicação (produção) - conta as funções de uma aplicação existente. É realizada no final do projeto de desenvolvimento (VAZQUEZ; SIMÕES; ALBERT, 2005).

2.4 Fronteira da Aplicação, Escopo e Contagem

2.4.1 Fronteira da Aplicação

A fronteira da aplicação delimita a aplicação envolvida no processo de contagem. Nessa área conceitual estão envolvidas a visão do usuário do negócio e a definição

da aplicação. Isto independe de considerações técnicas ou da forma como a aplicação está sendo desenvolvida pela equipe. A fronteira da aplicação é elaborada de acordo com a visão do usuário, ou seja, de acordo com sua perspectiva enquanto utilizador do sistema.

2.4.2 Escopo da Aplicação

O escopo da aplicação indica os conjuntos ou subconjuntos do software onde a medição será realizada. Tal procedimento identifica as funcionalidades da aplicação que estarão presentes na análise. A identificação do escopo não altera as regras da contagem, no entanto pode influenciar no seu tamanho, motivo pelo qual a espécie de contagem deve ser identificada e, posteriormente, definida.

2.4.3 Contagem de Funções de Dados

As funções de dados representam grupos lógicos de dados mantidos ou referenciados pela aplicação através dos requisitos funcionais dos usuários. São exemplos de grupo lógico de dados um arquivo de cadastro com dados de clientes de uma empresa.

Os dois tipos de funções de dados em FPA são:

- ALIs - Arquivos Lógicos Internos (*Internal Logical Files*): grupos de dados logicamente relacionados ou informações de controle identificáveis pelo usuário, mantido dentro da fronteira da aplicação, tais como tabelas que armazenam dados mantidos pela aplicação ou arquivos de configuração.
- AIEs - Arquivos de Interface Externa (*External Interface Files*): são grupos de dados logicamente relacionados ou informações de controle identificáveis pelo usuário, mantido fora da fronteira da aplicação, tais como tabelas que armazenam dados lidos pela aplicação, mas que são alimentadas por outra aplicação, ou seja, dados de referência externa consultadas pela aplicação.

2.4.4 Análise de Pontos de Função Extendida - (*Extend Function Point Analysis - XFPA*)

Na pesquisa de [Sousa \(2006\)](#), foi proposta uma nova métrica de dimensionamento de software, onde novos aspectos tecnológicos e ambientais/contextuais foram

incorporados à métrica FPA, Análise de Pontos de Função. A pesquisa foi denominada de Análise de Pontos de Função Estendida (XFPA). O trabalho visava obter melhores estimativas de esforço e custo para o desenvolvimento de sistemas, propôs também uma classificação dos fatores nas dimensões tecnológicas e ambiental/contextual para melhor avaliação dos usuários e da equipe técnica envolvida.

O trabalho identificou fatores técnicos e ambientais candidatos a integrar a nova métrica, através da suspeita de interferirem no esforço de desenvolvimento. A pesquisa buscou unir aspectos tecnológicos e ambientais/contextuais selecionados em conjunto com aspectos funcionais, formando a métrica XFPA. Uma das etapas incluía a coleta de informações de projetos realizados e a realizar, considerando a métrica proposta e contrastando com os resultados produzidos pela métrica FPA.

Conforme apresentado neste capítulo, o uso de métricas como a de Análise por Ponto de Função FPA pode ser utilizada para mensurar sistemas e estabelecer procedimentos para realização de medidas do software. Desde a década de 70, empresas de software procuram desenvolver novos modelos de contagem, visando melhorar o processo de produção de software e ter estimativas mais concretas de tamanho, custo e cronograma. Assim como no trabalho de [Sousa \(2006\)](#), com a definição da métrica XFPA, a presente pesquisa busca definir um modelo computacional para indentificar variáveis que podem gerar riscos em um projeto. O modelo aqui proposto, visa permitir que os gestores identifiquem as variáveis que mais influenciam nos riscos, de acordo com os resultados obtidos.

Gestão de Projetos de Software

O gerenciamento de projetos é cada vez mais reconhecido pelo mercado pela sua importância, tendo seus conceitos e premissas aplicados em diversos projetos de software. Um dos principais difusores do gerenciamento de projetos é o Instituto de Gerenciamento de Projetos (PMI, [s. d.]). O PMI é uma instituição sem fins lucrativos criada em 1969, que tem como principal objetivo contribuir para melhoria contínua da gestão de projetos. Para tanto, o PMI criou o guia PMBOK, Project Management Body of Knowledge (PMBOK, 2000), um manual contemplando diretrizes e estrutura de projetos, decomposta em nove áreas de conhecimento.

O documento pode ajudar a suprir uma carência na formação técnica de muitos programadores, analistas, e profissionais de computação que frequentemente ocupam postos de liderança. Com isso, a gestão dos projetos de software pode ser aprimorada e pode possibilitar redução de riscos.

3.1 *Gestão de Projetos*

As empresas reconhecem, cada vez mais, a importância da gestão de projetos, para reduzir custos, melhorar desempenho da equipe ou atender às exigências do mercado. A criação de um software, por exemplo, precisa ser controlada em detalhes, com a correta alocação de recursos e acompanhamento de prazos. Segundo Menezes (2001), alguns fatores internos demandam a execução de projetos nas organizações, tais como:

- Melhoria em um produto;
- Melhoria de processos;
- Criação de um produto ou serviço;
- Gestão estratégica da empresa;
- Compartilhamento de recursos;
- Mudanças organizacionais;
- Prazos ou recursos limitados.

Planejar e gerenciar de forma adequada um projeto pode reduzir substancialmente as chances de um fracasso. De um modo geral, os gestores de projeto trabalham com recursos e prazos limitados. Os erros são pouco toleráveis e os resultados são importantes para manter as organizações ativas e preparadas para um mercado competitivo. As organizações têm descoberto a importância de pensar de forma estruturada, de gerenciar custos, pessoas, tempo e recursos, com o objetivo de atingir metas e resultados previamente traçados.

3.1.1 *Processos no PMBOK*

O (PMBOK, 2000) formaliza diversos conceitos em gerenciamento de projetos. Conceitua projeto, define seu ciclo de vida, reconhece cinco grupos de processos e nove áreas de conhecimento.

A gestão de um projeto exige o cumprimento de etapas para alcançar o objetivo final. Tais etapas são constituídas por identificação de requisitos, onde a organização faz o seu planejamento, analisa as etapas iniciais do projeto, verifica os insumos e planeja a execução dos trabalhos.

No documento são classificadas nove áreas de conhecimento. Cada área abrange os seguintes processos: escopo, tempo, custos e qualidade, os quais são os principais elementos para atingir o objetivo de um projeto. É preciso fornecer o produto ou serviço de acordo com o escopo solicitado, ter prazo e custo claramente definidos e atender aos requisitos de qualidade. Recursos humanos e aquisições são os insumos que movem um projeto. Comunicação e riscos são elementos que merecem atenção constante de toda a equipe. Já a integração engloba todos esses aspectos.

3.1.2 *Ferramentas para Gestão de Projetos*

As empresas que atuam com desenvolvimento de software utilizam diversos programas para gerir os projetos. Os gerentes de projetos sabem que estes raramente são executados da forma como são planejados, pois podem haver imprevistos, riscos não calculados e atrasos. O uso de ferramentas para gestão do projeto é uma medida recomendada para acompanhar custos, prazos e riscos envolvidos no processo de produção de um software.

São exemplos de programas de gestão de projetos existentes no mercado:

- **1. KM Project:** software em versão Internet com suporte a controle de riscos, custos e cronograma dos projetos. Permite comunicação entre os membros da equipe, por usar uma base de conhecimento compartilhada, ([KM Project, 2012](#)).
- **2. ACE Project:** permite gerenciar grande número de projetos, customização de estrutura e estilo, gerenciar permissões para os membros e acompanhar o cronograma através de gráficos de Gantt, ([AceProject, 2012](#)).
- **3. RIQTek Manager:** desenvolvido para o ambiente Internet, ajuda no gerenciamento de problemas complexos do ciclo de vida de um produto. O projeto pode ser acompanhado desde o momento de sua concepção até após a sua implantação e tem suas informações centralizadas, ([RIQTek, 2012](#)).
- **4. Microsoft Project:** o Microsoft Project é uma das ferramentas mais utilizadas pelo mercado. Oferece recursos tanto para o processo de seleção e priorização de projetos quanto ao processo de controle dos mesmos. A versão Server permite um repositório central com dados acessíveis a todos os membros e com possibilidade de envio de relatórios online, ([MS Project, 2012](#)).

3.2 Gerenciamento de Riscos

Os estudos sobre Gerenciamento de Riscos em projetos de software ganharam força na década de 80, com algumas iniciativas para minimizar e reduzir as incertezas que envolvem a atividade de produção do software. À época, um grupo da Força Aérea Americana criou um documento que buscava tratar, identificar e reduzir riscos nos projetos de software. Segundo este, era uma exigência básica que o gerente identificasse os fatores de riscos que afetavam os principais componentes do software, como desempenho, custo, apoio da equipe e cronograma.

O documento acima mencionado, relacionava os seguintes componentes de risco:

- Risco de custo: grau de incerteza quanto à manutenção do orçamento do projeto;
- Risco de desempenho: grau de incerteza quanto ao atendimento aos requisitos de desempenho;
- Risco de apoio: grau de incerteza de que o software será de fácil manutenção, correção e aperfeiçoamento;
- Risco de cronograma: grau de incerteza de que os prazos do projeto serão cumpridos bem como se o produto será finalizado no tempo acertado.

Gerenciar riscos em projetos envolve (i) identificação, (ii) avaliação e (iii) resposta aos riscos, visando com isso minimizar seus impactos, para que os objetivos sejam atingidos. As atividades de acompanhamento e tratamento dos riscos consomem grande parte das atribuições do gestor, que concentra esforços para que estes não ocorram ou para que sejam reduzidos ao máximo. Um risco é uma probabilidade de alguma circunstância adversa acontecer [Sommerville \(2003\)](#).

3.2.1 Identificação de Riscos

Identificar adequadamente os riscos de um projeto de software ajuda a minimizar as perdas para as empresas que atuam nesse segmento. O processo de tratar, analisar e criar planos para contenção dos riscos ajuda a prevenir a ocorrência destes, além de reduzir chances de perdas financeiras. Segundo [Barki \(1993\)](#), “a quantificação é um dos passos mais importantes no processo de avaliação de risco”.

Para identificação dos riscos, podem ser utilizadas ferramentas ou técnicas que podem auxiliar na execução dos projetos, dentre as quais se destacam:

- **Revisão da documentação:** análise estruturada da documentação existente em projetos similares;
- **Técnicas de reunião de informação:** reuniões em grupo para construção de cenários;
- **Checklists:** verificação de dados históricos de projetos similares e identificações de impactos;
- **Análise das premissas:** permite comparar as premissas adotadas na fase de concepção do projeto com os dados realizados e levantar riscos em caso de divergências;
- **Técnicas de diagramação:** fluxogramas e diagramas de influência.

As ferramentas para análise e tratamento de riscos estão presentes no mercado, incluindo metodologias de gestão como o [ITIL \(2013\)](#) - Information Technology Infrastructure Library, o qual é definido como um conjunto de boas práticas em projetos de tecnologia da informação.

A gerência de riscos do projeto inclui processos referentes ao (P1) planejamento da gerência de riscos, (P2) à identificação, (P3) à análise, (P4) planejamento de respostas e (P5) controle e monitoração dos riscos do projeto. Esses processos interagem entre si e com os processos das outras áreas do conhecimento. A gerência de risco deve

diminuir ou eliminar a probabilidade de ocorrência dos riscos e responder aos eventos adversos ao projeto. Os processos da gerência de risco são assim definidos no (PMBOK, 2000):

- *P1 - Planejamento da gerência de riscos*: planejar as atividades de gerência de risco a serem realizadas para o projeto;
- *P2 - Identificação dos riscos*: identificar os riscos e documentar suas características;
- *P3 - Análise qualitativa dos riscos*: analisar qualitativamente os riscos, priorizando seus efeitos no projeto;
- *P4 - Planejamento de resposta aos riscos*: gerar procedimentos e técnicas para avaliar oportunidades, com o objetivo de mitigar as ameaças no projeto;
- *P5 - Monitoração e controle dos riscos*: monitorar os riscos residuais, identificar novos riscos, executar os planos de mitigação de riscos e avaliar sua efetividade durante todo o ciclo de vida do projeto.



Figura 3.1: Fluxo de monitoramento e planejamento de riscos - Fonte: Autor.

Os projetos precisam ser monitorados e controlados durante todo o ciclo de vida do software. É importante salientar que nenhum especialista é capaz de prever todas as possibilidades de riscos. Deste modo, buscar informações é fundamental na fase de identificação. São tarefas do gestor:

T1 - Identificar os riscos inerentes a uma etapa do desenvolvimento (fase, processo, iteração). Verificar ameaças presentes e impactos que podem provocar.

Determinar riscos e suas consequências. Em um processo de criação de software, a escolha de uma linguagem de programação, por exemplo, pode representar um risco, pois a mesma pode deixar de ser usada ou ter falhas de segurança. Ao analisar o risco, o gestor deve prioritariamente relacionar as possíveis consequências e os atrasos que podem ser gerados.

T2 - Avaliar os riscos a partir da identificação do nível de exposição do projeto. Verificar e decidir quais riscos serão eliminados, quais serão aceitáveis e quais serão acompanhados. A avaliação do risco determina a probabilidade de que um evento ocorra e sua influência no projeto. O gestor, juntamente com a equipe, pode classificar o risco como baixo, médio ou alto.

T3 - Controlar a execução e ao acompanhamento dos planos elaborados para o projeto. Analisar constantemente a identificação o estado atual do risco e os impactos que estes podem provocar.

3.2.2 Avaliação dos Riscos

Após relacionar os riscos, a etapa seguinte será a de classificação de cada um, atribuindo pontos ou características de acordo com a probabilidade de sua ocorrência. Esse processo deve levar em consideração o impacto que podem gerar em outras etapas do projeto. Na avaliação dos riscos, um grande número de variáveis podem ser encontradas, dificultando o seu tratamento, assim, o gestor deve definir prioridades. Essa análise pode ser feita criando-se uma tabela conhecida como Matriz de Tolerabilidade de Riscos, classificando aqueles que são toleráveis e os que não podem ocorrer. Essa matriz deve levar em consideração as dimensões do projeto: escopo, prazos e recursos (orçamento).

O gestor do projeto deve ter atenção com os riscos envolvidos nas etapas iniciais e criar um plano de ação, de acordo com as probabilidades de ocorrência e os respectivos responsáveis. Uma tabela pode ser criada pontuando os riscos e classificando-os de acordo com os dados que esses podem gerar. Este processo é conhecido como o plano de contingência, o qual pode ser implementado usando protótipos e tem como objetivo evitar, diminuir ou aceitar os riscos. Será aplicado sempre que o risco se converter em um problema.

Durante a fase de avaliação, o gestor deve analisar constantemente quais riscos foram modificados, excluídos ou exigem maior atenção. Em projetos de software, recomenda-se que a equipe discuta periodicamente com os gestores, tente classificar os riscos, faça revisões e avalie como os riscos podem afetar o projeto.

Tabela 3.1: Exemplo de tabela de risco para aplicação funcionando na Internet - Fonte: Autor

Risco	Probabilidade	Impacto
Mais usuários acessando o sistema do que o planejado	50%	médio
Equipe de segurança da informação inexperiente	30%	baixo
Alta rotatividade no grupo de programação do sistema	60%	alto

3.2.3 Riscos de Software

Estimar e acompanhar corretamente os riscos envolvidos no projeto é um dos procedimentos mais importantes no processo de produção de um software. Segundo [Charette \(2005\)](#), o risco é algo que pode impactar acontecimentos futuros, envolve mudanças e gerenciamento de incertezas. Os projetos de software devem analisar e tratar de forma adequada os riscos envolvidos, administrar incertezas e evitar problemas que possam comprometer o andamento dos trabalhos. O risco em um projeto de software mede a probabilidade e os prejuízos relacionados à ocorrência de um evento negativo que podem gerar problemas no produto final.

O risco do projeto relaciona-se com aspectos operacionais, organizacionais e de contrato. A gestão e acompanhamento dos riscos devem ser feitos pela equipe com coordenação do Gerente do Projeto. É necessário criar um plano de riscos, incluindo limitações de recursos, interfaces externas, relacionamento com fornecedores e restrições contratuais.

Os riscos, de forma geral, envolvem duas características: incerteza e perda. A incerteza é algo que pode ou não acontecer e a perda é consequência não desejável, que podem ocorrer caso o risco se concretize. Os riscos impactam diretamente no projeto e implicam em consequências negativas, como atrasos de cronograma, aumento de custos e problemas na alocação de pessoal.

3.3 Trabalhos Correlatos - Riscos e Lógica Fuzzy

Em Avaliação de Riscos em Desenvolvimento de Software, [Leopoldino \(2004\)](#), foram analisadas as percepções sobre os riscos, de acordo com a visão de desenvolvedores e de gestores de projeto. O trabalho incluiu pesquisas realizadas em diversos Estados do

Brasil. Buscou-se identificar os riscos dos projetos de software e como esses impactavam no processo de desenvolvimento de software. Foram entrevistados 81 profissionais de tecnologia da informação. Fizeram parte da amostra 56 gerentes de projetos e 25 desenvolvedores. Todos os entrevistados tinham experiência na área, formação acadêmica e experiência em desenvolvimento de software.

Os resultados descritos a seguir foram coletados com pesquisas respondidas pelos gestores e algumas dessas informações serão comparadas com o modelo da presente pesquisa no Capítulo “Modelo Proposto”. De acordo com esse grupo, foram considerados riscos de alto impacto:

- Requisitos mal entendidos;
- Escopo ou objetivos pouco claros;
- Prazos mal estimados;
- Volatilidade dos requisitos e
- Planejamento inadequado;

Dentre os riscos relacionados na pesquisa, os gestores consideraram como os de maior gravidade: requisitos mal entendidos, volatilidade dos requisitos, escopo e objetivos pouco claros. Esses três riscos foram incorporados à presente pesquisa e serão abordados mais adiante.

A pesquisa identificou diversos riscos, segundo a visão dos desenvolvedores e gestores e usou métricas para avaliar seus impactos. Não foi utilizada lógica Fuzzy para realização do cálculo e não foram analisados projetos em separado, como proposto no presente trabalho.

Segundo a visão dos profissionais, foram considerados como maior gravidade os seguintes riscos:

- Requisitos mal entendidos;
- Custos mal estimados;
- Adoção de nova tecnologia;
- Definição imprópria de papéis;
- Prazos mal estimados;

- Pessoal envolvido insuficiente;
- Falhas em identificar os papéis dos usuários finais;

Os resultados do trabalho demonstram os riscos considerados mais graves pelos dois grupos, desenvolvedores e gestores. A produção do software com prazo irreal, foi informada pelos desenvolvedores como maior fator de risco, assim como a complexidade do software. Os gestores, no entanto, identificaram como o risco mais grave a estimativa inadequada de custos. O trabalho deixa claro a importância da gestão de riscos para maiores chances de sucesso dos projetos, mostrando visões diferentes, de gestores e desenvolvedores, para problemas relacionados a criação de software, assim como mostrado na presente pesquisa.

Em Gerenciamento de Riscos de Software [Goncalves \(2006\)](#), desenvolveu-se um modelo de gerenciamento de risco contendo um conjunto de informações sobre riscos, divididos em classes, alocados na fase de elaboração de proposta e nas fases do ciclo de vida de desenvolvimento de software. A pesquisa resultou em uma ferramenta que disponibiliza um módulo para o gerenciamento de risco de projetos em desenvolvimento, no qual foram registrados os riscos, impactos e controles das fases e das atividades do respectivo projeto.

A pesquisa apresentou um modelo de processo de gerenciamento de risco e uma ferramenta de gestão denominada GRisk-Tool, usando como base uma série de entrevistas e dados coletados junto a empresas fabricantes de software de Piracicaba (SP). O modelo foi desenvolvido com base na literatura da área e a partir da experiência de diretores, gerentes e analistas de sistemas seniores de fábricas de software do Brasil.

No trabalho sobre Lógica Fuzzy Para Controle de Tráfego Aéreo, [Lima \(2008\)](#), foi realizado um estudo de caso para ilustrar a aplicabilidade da lógica fuzzy na tomada de decisão, com o objetivo de solucionar um problema no controle de tráfego aéreo da cidade de Salvador (Ba). A pesquisa criou uma metodologia e propôs o desenvolvimento de um framework para a validação do modelo.

Os resultados da pesquisa acima mencionada apontam a potencialidade da modelagem Fuzzy, principalmente no desenvolvimento de sistemas que sirvam como ferramenta de apoio à tomada de decisão e controle de riscos, conforme implementado neste trabalho. Foi utilizada a modelagem Fuzzy para tratar questões envolvendo atividades de controle de tráfego aéreo. O trabalho confirma a eficácia e a adequação da lógica difusa para solucionar questões complexas e que tratam de incertezas. De igual modo, a presente pesquisa analisa difíceis questões, todavia relacionadas ao desenvolvimento de software.

3.4 Engenharia de Software

Software é um conjunto de instruções que, ao ser executadas, produzem a função e o desempenho desejados, Pressman (2005). Segundo Sommerville (2003), o software é o conjunto de vários artefatos e não apenas o código fonte. Nos anos 40, iniciando-se a evolução dos sistemas computadorizados, grande parte dos esforços e custos era concentrada no desenvolvimento do hardware, principalmente devido às dificuldades encontradas na época. Ao longo dos anos, a tecnologia de hardware foi melhorando e, no início dos anos 50, iniciou-se o desenvolvimento dos sistemas operacionais. Nessa época, surgiram as primeiras versões destes sistemas, assim como das chamadas linguagens de programação de alto nível, como FORTRAN e COBOL, além dos respectivos compiladores.

A Engenharia de Software (ES) surgiu na década de 70 com o intuito contornar a crise do software e dar um tratamento de engenharia, mais sistemático e controlado, ao desenvolvimento de sistemas de software complexos, assim como melhorar os pequenos aplicativos que começavam a surgir no mercado, os chamados softwares de prateleira. Nesse período, um grande número de empresas multinacionais de desenvolvimento de software começou a surgir, o que colaborou para criação de novas metodologias para reduzir erros nos programas e otimizar o processo de elaboração de sistemas. Os fundamentos de engenharia de software envolvem o uso de modelos que permitem ao engenheiro, programadores e analistas especificarem, projetarem, implementarem e manterem sistemas de software.

3.4.1 Modelos de Desenvolvimento de Software

O modelo de desenvolvimento de software corresponde a uma representação do seu processo de criação, de suas etapas, tentando atingir o objetivo do desenvolvimento, ou seja, produção de um software com qualidade, dentro de prazos aceitáveis e com custos relativamente baixos. Dentre os modelos clássicos do processo de desenvolvimento de software, são relacionados os modelos cascata, espiral e o incremental, conforme descrito a seguir:

MODELO CASCATA - O modelo cascata (*waterfall*) tornou-se conhecido na década de 70 e é referenciado na maioria dos livros de engenharia de software ou manuais de padrões de software, sendo um dos mais conhecidos. Neste modelo, os processos são executados em sequência e demarcados pontos de controle bem definidos. As atividades do processo de desenvolvimento são estruturadas em uma cascata onde a saída de uma é a entrada para a próxima. Neste modelo, o cliente tem baixa visibilidade do projeto e conhece o produto no final do ciclo.

MODELO ESPIRAL - Foi proposto em 1988 por [Boehm \(1988\)](#), envolvendo uma série de iterações com atividades em espiral, através de diversos ciclos. Neste modelo, diversas versões incrementais do software são criadas. Cada ciclo na espiral se inicia com a identificação dos objetivos e alternativas para resolução dos problemas, assim como as restrições que serão impostas. O próximo passo no ciclo é a avaliação das diferentes alternativas com base nos objetivos fixados, permitindo definir incertezas e riscos de cada uma delas. No passo seguinte são desenvolvidas estratégias para resolver ou eliminar as incertezas levantadas anteriormente.

MODELO INCREMENTAL - Combina os elementos da prototipagem e fundamentos do modelo linear. O modelo de processo incremental é iterativo, assim como no processo de prototipagem, mas tem como objetivo apresentar um produto operacional a cada incremento realizado. O modelo incremental é útil quando a empresa não possui equipe disponível no momento para uma implementação completa, em atenção ao prazo estipulado.

3.5 *Projetos de Software*

[Pressman \(2005\)](#) define a gestão de projeto de software como a junção de planejamento, monitoração, controle do pessoal, processos e eventos que ocorrem na criação de um software. Dessa forma, a Gerência de Projetos de Software envolve monitoria, planejamento e acompanhamento do grupo, do produto desenvolvido e do processo seguido para a evolução do software de um conceito preliminar para sua implementação. A gestão de um projeto de software focaliza quatro Ps, ou seja, pessoas, produto, projeto e processo e a ordem desses não é arbitrária. Um plano de trabalho deve ser realizado, relacionando prazos, custos envolvidos e metas por etapa ([PRESSMAN, 2005](#)).

3.5.1 *Pessoas*

Em um projeto de software, há diversas pessoas envolvidas durante todo o trabalho que exercem diferentes papéis, tais como: o Gerente de Projeto, Desenvolvedor (Analistas, Projetistas, Programadores, Gerente da Qualidade e Clientes). Em alguns momentos, os usuários poderão participar do desenvolvimento, de testes, de entrevistas ou avaliar versões do produto. Os grupos de trabalho são organizados em equipes e, dessa forma, o conceito de equipe pode ser visto como um conjunto de pessoas que atuam em diferentes tarefas, voltadas para o mesmo objetivo. A definição de equipes e pessoal alocadas nos projetos é importante e pode evitar problemas no processo de criação do software, pois os profissionais podem não ter o conhecimento necessário para atender ao

escopo do trabalho

Para adequada formação das equipes de trabalho, devem ser considerados aspectos fundamentais em todos os projetos: liderança, organização (estrutura da equipe) e coordenação. Na organização dos grupos de trabalho, há diversos tipos de equipes, tais como os citados por [Pressman \(2005\)](#):

- Controlada descentralizada: existe um líder do projeto, mas a comunicação ainda é horizontal;
- Democrática descentralizada: sem líder permanente e as decisões são tomadas por consenso do grupo. A comunicação é entre os membros da equipe é horizontal;
- Controlada centralizada: há um líder de projeto e a sua comunicação com os demais membros da equipe é vertical.

3.5.2 *Escopo do Software*

A primeira etapa na criação de um projeto de software é a determinação do escopo a ser desenvolvido, ou seja, aquilo que a aplicação deseja produzir. O escopo do produto é composto pela especificação de um conjunto de funcionalidades - chamado de requisitos funcionais -, associada a outras características desejadas - requisitos não funcionais -, tais como desempenho, segurança e confiabilidade. Para que o escopo do software seja determinado, um levantamento preliminar de requisitos deve ser realizado e informado ao cliente, com o objetivo de evitar problemas de entendimento.

3.5.3 *Produto*

Na gerência de projetos, o gestor precisa ter conhecimento sobre gerenciamento do tempo e de custos envolvidos. É necessário definir o escopo do software, realizando um levantamento de requisitos alinhados com a necessidade do cliente, para que o produto final atenda às especificações, prazos definidos e exigências solicitadas.

3.5.4 *Estrutura de Divisão do Trabalho*

A gerência deve analisar e especificar como as etapas do projeto serão realizadas e como as equipes devem trabalhar para atender aos requisitos especificados no

escopo dentro de prazos aceitáveis e com recursos alocados de forma adequada. Esse processo envolve as visões de produto e de processo. O módulo a ser desenvolvido pela equipe deve passar pelas etapas definidas no processo de software, incluindo especificação, desenvolvimento e testes.

O processo de produção do software é composto das seguintes etapas:

- **Planejamento:** no início do projeto é criado um plano organizado de como ele será conduzido, que deve incluir informações sobre o escopo, definição do processo, realização de estimativas, elaboração de um cronograma e tratamento dos riscos;
- **Acompanhamento:** é fundamental o acompanhamento e o progresso do trabalho, refinar o escopo e as estimativas, alterar cronograma quando necessário, além de monitorar riscos e tomar ações corretivas;
- **Encerramento:** ao final do projeto, a gerência deve realizar uma análise crítica de pontos positivos e negativos, registrando informações para melhoria e ajustes futuros. É necessário comparações entre valores estimados e realizados bem como a identificação dos eventuais problemas ocorridos.

3.5.5 *Estimativas*

Antes de iniciar as atividades técnicas de um projeto, a gestão deve estimar o trabalho a ser realizado, os recursos necessários, a duração e os custos envolvidos. As práticas adequadas para realizar estimativas de prazo, esforço e custo incluem:

- Prever eventuais atrasos;
- Usar técnicas de decomposição, com separação de etapas;
- Usar um ou mais modelos empíricos para estimativas de custo e esforço;
- Analisar projetos similares que já tenham sido concluídos.

As estimativas de esforço são realizadas com a ajuda dos especialistas, a partir de dados históricos armazenados pelo grupo, além de contar com a experiência dos membros do projeto. A empresa deve coletar dados de vários projetos e estabelecer, por exemplo, quantos homens-hora (uma unidade de esforço) são necessários para desenvolver determinados módulos especificados no sistema.

3.5.6 *Elaboração de Cronograma*

Após realizar as estimativas de esforço e alocação de recursos, é possível calcular a duração em meses ou horas para cada atividade. Com essas informações, o cálculo do orçamento e a criação de um cronograma tornam-se mais precisos. Se a estimativa de esforço tiver sido realizada para o projeto como um todo, então ela deverá ser distribuída de acordo com as atividades listadas para execução do projeto. O cronograma deve ser acompanhado durante todo o processo, assim como os custos relacionados a cada uma das tarefas propostas.

3.5.7 *Custos do Projeto*

Com base nas estimativas iniciais e experiência da equipe em projetos anteriores, conhecimentos dos requisitos, metas e prazos estabelecidos, é possível estimar os custos do projeto. Os seguintes itens devem ser considerados nas estimativas de custos:

- Esforço realizado pelos membros da equipe no projeto;
- outros custos relacionados ao projeto, tais como custos de viagens e treinamentos realizados;
- aquisição de hardware ou de software;
- custos com água, luz, telefone, pessoal de apoio, suporte e outros custos fixos.

Conforme visto neste capítulo, a gestão dos riscos nos projetos de software permite reduzir perdas e identificar situações que podem prejudicar a execução dos trabalhos. A gestão de riscos requer a criação de um plano de contingência, com a finalidade de aumentar as chances de sucesso do trabalho e reduzir incertezas. O processo de classificar, identificar e responder aos riscos, proporciona aprendizado constante para equipe desenvolvida e melhorias na qualidade do produto. Essas ações permitem atendimento aos prazos estabelecidos, redução de custos e eficiência no processo de produção. [Pressman \(2005\)](#) afirma que a gerência de riscos é crucial para o bom gerenciamento dos projetos de software.

Lógica Difusa (Fuzzy)

A primeira publicação sobre lógica “Fuzzy” data de 1965 e recebeu esta denominação por Lotfi Asker Zadeh (ZAH-da), professor em Berkeley, Universidade da Califórnia (ZADEH, 1965). Ele foi um dos pioneiros na utilização da lógica Fuzzy, combinando os conceitos da lógica clássica e os graus de incerteza. A formalização tinha como objetivo principal fornecer informações que ajudassem no tratamento de dados para informações imprecisas ou consideradas vagas. O entendimento dos sistemas Fuzzy não é algo complexo, mas aqueles que utilizam seus conceitos precisam conhecer de modo aprofundado incertezas e imprecisões e assim utilizar os benefícios que a lógica Fuzzy pode oferecer.

Ao mesmo tempo em que os experimentos de Zadeh geravam ceticismo e resistência entre os pesquisadores da época, a seriedade dos seus estudos levou outros pesquisadores a testar os conceitos estabelecidos naquele momento. Em 1974, um outro pesquisador, Ebrahim Mamdani, professor na Universidade de Londres realizou diversas experiências quando, finalmente conseguiu comprovar os conceitos de lógica Fuzzy no controle de uma máquina a vapor. A partir daí, outros pesquisadores iniciaram estudos sobre lógica Fuzzy na área da saúde e no controle de máquinas industriais.

Entre os anos de 1970 e 1980 as aplicações industriais da lógica “fuzzy” aconteceram com maior importância na Europa e, após 1980, com o crescente número de aplicações indústria em todo mundo, principalmente na Ásia. Empresas japonesas como a Mitsui, desenvolveram aplicações computacionais com métodos de controle Fuzzy ou Fuzzificação para melhoria de processos de trabalho e de suas plantas industriais. Os modelos fuzzy permitem métodos de trabalho que utilizam informações vagas, imprecisas e qualitativas convertendo-as em informações numéricas que podem ser utilizadas na melhoria e análise de inúmeras aplicações, a exemplo de dados sobre violência no Brasil, casos de epidemias em locais com pouco acesso a saneamento básico ou mesmo aplicações para detectar cidades com índice de nanismo elevado.

O sucesso na utilização da lógica fuzzy em processos industriais despertou a atenção de outros países. A partir de 1980, grandes corporações americanas e outras na Inglaterra iniciaram o uso dos modelos Fuzzy em suas plantas industriais, resultando em práticas de sucesso e aceitação na área acadêmica desse novo modelo lógico. A teoria de conjuntos fuzzy é também descrita como teoria das possibilidades (AGUIAR; JUNIOR, 1999).

A lógica Fuzzy está associada ao uso da teoria de conjuntos Fuzzy proposto por Lukasiewicz e é uma área de conhecimento que utiliza modelos matemáticos para tratamento de incertezas (ZADEH, 1965). Muitos autores preferem relacionar a Lógica Fuzzy como uma lógica não verdadeira. Uma das suas características interessantes é permitir representar de forma inovadora o manuseio de informações imprecisas de modo distinto da teoria de probabilidades (SHAW; SIMÕES, 1999).

A presente pesquisa utilizou a lógica Fuzzy para aplicação do modelo proposto, avaliando os riscos de projetos de software. O estudo e tratamento de dados incertos ou imprecisos proporcionados pelo Fuzzy pode contribuir com o trabalho de gestão de projetos de software, escopo principal dessa pesquisa, auxiliando empresas a definir variáveis que exigem maior atenção durante a execução do projeto.

4.1 Conjuntos Fuzzy

De acordo com McNeill e Thro (1994), na modelagem de um determinado problema utilizando conjuntos clássicos, o conjunto universo conteria todos os elementos possíveis dentro do escopo do problema. Chamando o conjunto universo de \mathbf{S} , uma união de elementos de \mathbf{S} é chamada de subconjunto de \mathbf{S} . Dado um subconjunto S de \mathbf{S} , escreve-se $S \subset \mathbf{S}$ para denotar que S está contido em \mathbf{S} . Se S pode inclusive ser igual a \mathbf{S} , escreve-se $S \subseteq \mathbf{S}$. Para expressar a pertinência de um elemento s a \mathbf{S} , escreve-se $s \in \mathbf{S}$. O caso contrário é expresso por $s \notin \mathbf{S}$ (isto é, s não pertence a \mathbf{S}). Entre as operações que podem ser executadas sobre um número de conjuntos, destacam-se a interseção (denotada por “ \cap ”), que retorna o conjunto composto pelos elementos em comum dos conjuntos envolvidos e a união (denotada por “ \cup ”), que retorna o conjunto composto pelos elementos de todos os conjuntos envolvidos. Em termos de lógica, a interseção representaria a operação lógica “E”, enquanto a união representaria um “OU” lógico.

Os autores também definem uma função característica:

$$C_A(x) = \begin{cases} 1, & \text{se } x \in A \\ 0, & \text{se } x \notin A \end{cases}$$

Assim, C_A é uma função cujo domínio é U e a imagem está contida no conjunto $\{0, 1\}$, com $C_A(x) = 1$ indicando que o elemento x está em A , enquanto $C_A(x) = 0$ indica que x não é um elemento de A . Assim, a função característica descreve completamente o conjunto A , indicando quais elementos do conjunto universo são elementos também de A .

4.1.1 Teoria de Conjuntos Fuzzy

Em computação são inúmeras aplicações que utilizam ferramentas para classificar e tratar informações. Algumas aplicações exigem a utilização de parâmetros subjetivos e imprecisos que constantemente geram dúvidas nos grupos de desenvolvimento de software, por exemplo (ZADEH, 1965). Há situações onde os gestores de projetos de software precisam conhecer os riscos envolvidos na execução da aplicação. Pode-se, por exemplo, identificar qual linguagem é a mais indicada para resolução de um problema, ou qual grau de dificuldade os usuários terão no uso do sistema. São questões difíceis de ser respondidas, mas, com o uso das Teorias dos Conjuntos Fuzzy, questionamentos que envolvem incertezas e imprecisões podem ser respondidos.

O termo “Fuzzy”, de origem inglesa, tem como tradução nebuloso, impreciso e incerto. Tentando a formalização matemática clássica para classificar pessoas com alta renda, por exemplo, uma das possibilidades seria estipular uma variável de renda mensal (valor), considerando desta forma a pessoa como rica, caso esta tenha altos rendimentos. Outra possibilidade, como exemplo de classificação usando o modelo Fuzzy, seria identificar cidades brasileiras que podem ser “mais ou menos” sujeitas ao vírus da dengue. Desafios como estes, tratando de incertezas ou de dados imprecisos, fez com que a Teoria dos Conjuntos Fuzzy ganhasse credibilidade.

Na teoria dos conjuntos “fuzzy” há um grau de pertinência de cada elemento a um determinado conjunto. No exemplo a seguir, temos os dois grupos:

- Conjunto das pessoas com alta renda;
- Conjunto das pessoas baixas.

Nos dois exemplos, verifica-se que há dificuldades em estabelecer conceitos que classifiquem se a pessoa é verdadeiramente rica ou verdadeiramente baixa, pois inexistente classificação exata para esses dois conjuntos. A utilização dos conjuntos Fuzzy permite que situações imprecisas possam ser tratadas e examinadas, verificando assim o seu grau de pertinência.

Um conjunto *fuzzy* A em X é expresso como um conjunto de pares ordenados:

$$A = \{(x, \mu_A(x)) | x \in X\} \quad (4.1)$$

onde $\mu_A(x)$ é a função de pertinência que associa os elementos x pertencentes a X a um número real $\mu_A(x)$ no intervalo $[0, 1]$ (i.e. representa o grau de pertinência do elemento x

ao conjunto A) e X representa o universo ou universo de discurso.

Considerando o exemplo “conjunto das pessoas baixas”, pode-se perguntar se uma pessoa com altura $x = 1.67m$ é baixa ou alta. Para esse exemplo, o conjunto *fuzzy* é igual a **baixo**; $\mu_A(x)$ é a função de pertinência de x em A ; e temos as seguintes opiniões: 0,85 para os suecos, 0,50 para os americanos e 0,25 para os brasileiros.

As funções de pertinência mais aplicadas são a triangular, a trapezoidal e a Gaussiana, mostradas na Figura 5.1. Nessa figura, a reta vertical vermelha denota o valor de um dado elemento do conjunto fuzzy (determinado pela abscissa do gráfico, de 0 a 5), e a interseção da reta vermelha com o gráfico da função de pertinência produz o grau de pertinência daquele elemento ao conjunto fuzzy em questão.

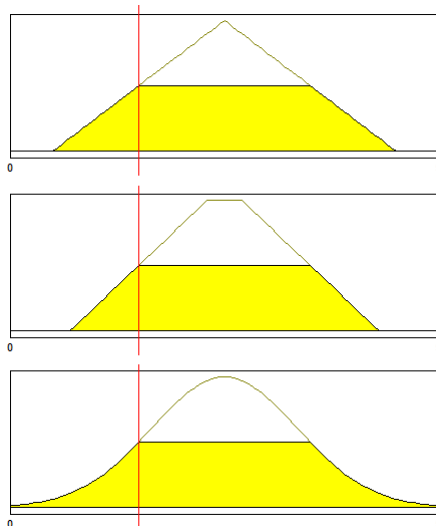


Figura 4.1: Funções de Pertinência para Conjuntos Fuzzy: triangular (topo), trapezoidal (centro) e Gaussiana. Fonte: Autor.

4.2 Modelos Linguísticos Fuzzy

A utilização de modelos e simulações para retratar o mundo real é um desafio constante para pesquisadores de todo o mundo. Algumas pesquisas precisam ter como variável de entrada valores subjetivos, imprecisos e muitas vezes incertos. Os modelos e simulações são necessários nesses casos, pois podem proporcionar estudos e experimentos que podem auxiliar profissionais e empresas de diversas áreas (SHAW; SIMÕES, 1999).

A utilização da lógica fuzzy é fundamentada no conceito de variáveis linguísticas, estabelecido por Zadeh (1965). Segundo o supracitado autor, uma variável linguística se-

ria uma variável que, ao invés de receber valores numéricos exatos, receberiam valores linguísticos imprecisos e se traduziriam em conjuntos fuzzy. Assim, por exemplo, “temperatura” poderia ser uma variável linguística que, ao invés de ter valores numéricos de temperatura atribuídos a ela, teria valores linguísticos como “frio”, “morno” ou “muito quente”. Estes valores linguísticos seriam matematicamente representados por conjuntos fuzzy.

4.2.1 *Sistemas Fuzzy*

Antes de criar um sistema fuzzy, é necessário escolher o modelo mais apropriado para o sistema. Os dois modelos mais conhecidos são o Mamdani e o de Takagi-Sugeno-Kang (MENDEL, 1995).

O primeiro modelo é o Mamdani, criado em 1975 por Ebrahim Mamdani. Seus esforços foram baseados nas publicações já existentes sobre algoritmos fuzzy para sistemas complexos e processos de decisão. O segundo modelo é o Takagi-Sugeno-Kang ou TSK, bem similar ao de Mamdani, mas com algumas diferenças. Diferentemente da abordagem proposta por Mamdani, o TSK compreende modelos lineares, sendo que a saída é determinada diretamente pelos valores reais das entradas, não existindo conjuntos fuzzy para a variável de saída.

A abordagem TSK pode ser usada para modelar sistemas de inferência cujas funções de pertinência de saída sejam lineares ou constantes. O modelo Sugeno dispensa a definição de uma função de implicação específica, sendo que a resposta final do controlador é obtida pela média ponderada das respostas das regras individuais. Neste tipo de controlador não cabe processo de defuzzificação, considerado importante etapa no estudo de dados imprecisos e um dos pontos centrais do presente trabalho. Por isso, foi escolhido o modelo Mamdani para esta pesquisa.

No modelo Mamdani, o algoritmo fuzzy do controlador tem cada regra como proposição condicional fuzzy e diferentes relações fuzzy podem ser dela derivadas. Na proposta desse trabalho, usando o modelo Mamdani, foram analisados quatro projetos de software, conforme demonstrado no capítulo “Modelo Proposto”. A implementação das regras é feita mediante a definição de operadores para o processamento do antecedente da regra e da função de implicação. Nessa pesquisa, a saída efetiva do controlador é obtida por meio de um processo de defuzzificação, que atribui um grau de risco para o projeto, considerando as entradas de variáveis do sistema e variáveis dos recursos humanos.

O modelo fuzzy Takagi-Sugeno constitui uma abordagem alternativa para modelagem fuzzy. Este modelo também possui uma estrutura baseada em regras. Con-

tudo, os consequentes das regras não são conjuntos fuzzy como nos modelos linguísticos. Os consequentes das regras fuzzy Takagi-Sugeno são formados por funções não-fuzzy que mapeiam as entradas do modelo em relação à sua saída. Assim, o modelo fuzzy Takagi-Sugeno é capaz de aproximar um sistema não-linear com uma combinação de vários sistemas lineares afins pela decomposição de todo o espaço de entrada em vários espaços parciais e representar cada espaço entrada/saída com uma equação linear.

4.3 Controladores Fuzzy

A composição básica de um controlador Fuzzy é formada pelos seguintes blocos (SHAW; SIMÕES, 1999):

- (a) Interface de fuzzificação: a interface de fuzzificação permite que valores de entrada, geralmente vindos de grandezas físicas ou de dispositivos computadorizados, sejam inseridos, verificando a base de conhecimento e convertendo sinais em um intervalo que pode variar de $[0,1]$.
- (b) Base de conhecimento: a base de conhecimento trata do modelo do sistema que será controlado, tendo como informações a base de dados utilizada e as regras fuzzy linguísticas (SHAW; SIMÕES, 1999). Essa base de dados possui as informações das funções de pertinência que estarão no conjunto de regras fuzzy.
- (c) Lógica de tomada de decisões: a tomada de decisões é realizada através de uma estrutura de inferência da base de regras, usando o modelo fuzzy para tomar as decisões.
- (d) Interface de defuzzificação: a defuzzificação trata do processo que gera um valor único discreto, sendo esse usado a partir de valores fuzzy de saída obtidos pelo controlador.

Essa estrutura demonstra um domínio real que pode se transformar em um domínio Fuzzy, usando os número Fuzzy e inferências para tomada de decisões que podem ser aplicadas para diferentes propósitos. Com os números fuzzy é possível converter dados e ações para o mundo real usando os algoritmos fuzzy.

A Figura 4.2 a seguir mostra um exemplo de controlador Fuzzy proposto por Mandani. Fonte: (SHAW; SIMÕES, 1999)

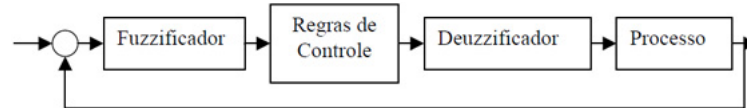


Figura 4.2: Controlador Fuzzy

4.4 O Modelo Fuzzy Mamdani

4.4.1 Processo de Fuzzificação

O processo de fuzzificação consiste em transformar as variáveis qualitativas escolhidas usando as regras de pertinência. Desse modo, o processo permite que esses dados sejam entendidos pelo computador. A fuzzificação realiza o mapeamento do domínio de números reais em um domínio fuzzy, usando pré processamento de categorias ou de classes de entrada, otimizando os valores que serão processados. Com menos valores sendo processados há mais rapidez no processo de análise das variáveis de entrada.

Para a combinação lógica de conjuntos fuzzy através da operação “E”, o operador min pode ser aplicado sobre os graus de pertinência dos dados de entrada. Uma outra opção para o “E” lógico em conjuntos fuzzy é o produto dos graus de pertinência, mas este pode produzir valores extremamente pequenos à medida que o número de conjuntos envolvidos na operação cresce, uma vez que os graus de pertinência estão entre 0 e 1 (KLIR; YUAN, 1996).

4.4.2 Processo de Defuzzificação

No processo de defuzzificação, as variáveis utilizadas devem “decifrar” ações vagas ou imprecisas, como, por exemplo, “ele é muito alto” ou “preciso perder peso”. Neste processo, conflitos entre regras que possam parecer contraditórias devem ser resolvidas. Dentre os métodos mais utilizados na defuzzificação estão:

- Centróide - centro de gravidade ou massa;
- Primeiro dos máximos;
- Média dos máximos.

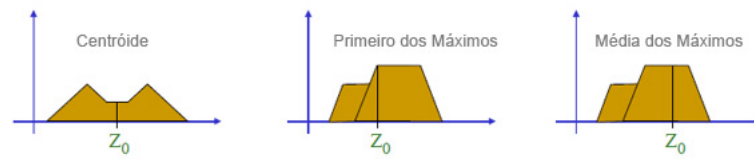


Figura 4.3: Métodos usados na defuzzificação - Fonte: (KLIR; YUAN, 1996)

4.4.3 Sistemas Lógicos Fuzzy

Os Sistemas Lógicos Fuzzy (FLS) são os únicos que permitem trabalhar com dados numéricos e linguísticos (MENDEL, 1995). As teorias do conjunto fuzzy e a lógica fuzzy estabelecem as especificidades de um mapeamento não linear. O uso da lógica fuzzy combinada com as teorias dos conjuntos fuzzy permitem a criação de sistemas lógicos que podem ser utilizados para diversas aplicações, usando matemática e modelos linguísticos com finalidades variadas.

Um sistema lógico fuzzy pode ser definido com um mapeamento não linear de um vetor de entrada de dados dentro de uma saída de uma grandeza escalar. O grande diferencial desse sistema é permitir que um grande número de possibilidades tenha diversas formas de mapeamento, facilitando o entendimento do problema e suas diferentes resoluções. O desafio, neste caso, é tratar esses dados, validando-os de forma correta. A Figura 4.4 mostra um sistema fuzzy usando modelo linguístico.

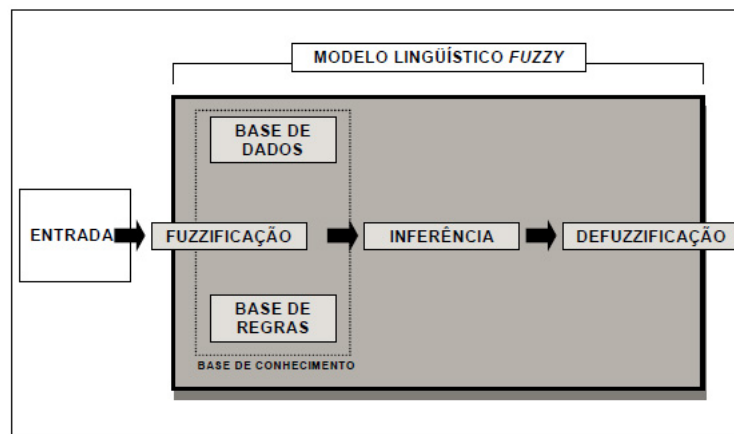


Figura 4.4: Sistema Fuzzy com Base em um Modelo Linguístico. Fonte: Zadeh (1965).

O processo de inferência fuzzy passa por diferentes etapas, que serão detalhadas a seguir:

REGRAS - as **regras** que farão parte do conjunto fuzzy, são estabelecidas por especialistas no assunto ou extraídas de dados numéricos. O tratamento destes dados é feito pelo sistema Fuzzy que realiza a conversão destas informações para que possam

ser usadas da melhor forma possível. A figura 4.5 mostra um sistema lógico fuzzy e o tratamento de dados processados.

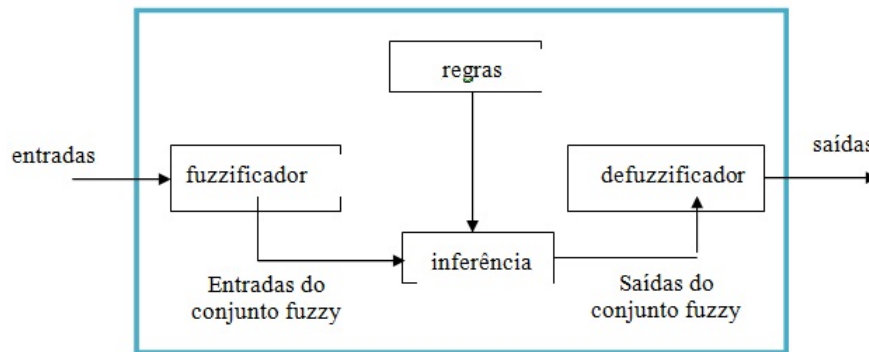


Figura 4.5: Motor de Inferência Fuzzy. Fonte: Zadeh (1965).

FUZZIFICADOR - o **fuzzificador** mapeia números na entrada, dentro de um conjunto Fuzzy. Isso torna-se necessário para que as regras sejam ativadas usando termos das variáveis linguísticas que, na verdade, estão associados a outros conjuntos fuzzy.

INFERÊNCIA - a **inferência** mapeia os conjuntos fuzzy atribuindo informações dentro dos mesmos conjuntos. Nesse momento, as regras são combinadas para que as decisões sejam processadas de forma correta.

DEFUZZIFICADOR - o **defuzzificador** mapeia números dos conjuntos de saída dentro de números fuzzy. Esse modelo poderia ser usado, por exemplo, para previsões climáticas ou para mapeamento de estações do ano com maior índice de epidemias.

A figura 4.6 mostra como ocorre o processo de inferência em um modelo Mamdani com associação através do operador *min* e através do produto dos graus de pertinência.

4.5 Exemplos

Para demonstrar os conceitos expostos até então, dois exemplos de sistemas fuzzy serão aqui apresentados. Os dois exemplos constam em Jan (2004). O primeiro mostra um caso com apenas uma variável de entrada e uma variável de saída, enquanto o segundo mostra duas variáveis de entrada e uma de saída. Nos dois casos, segundo o autor, a operação *max - min* é utilizada como interseção e a defuzzyficação é feita pelo método do centróide.

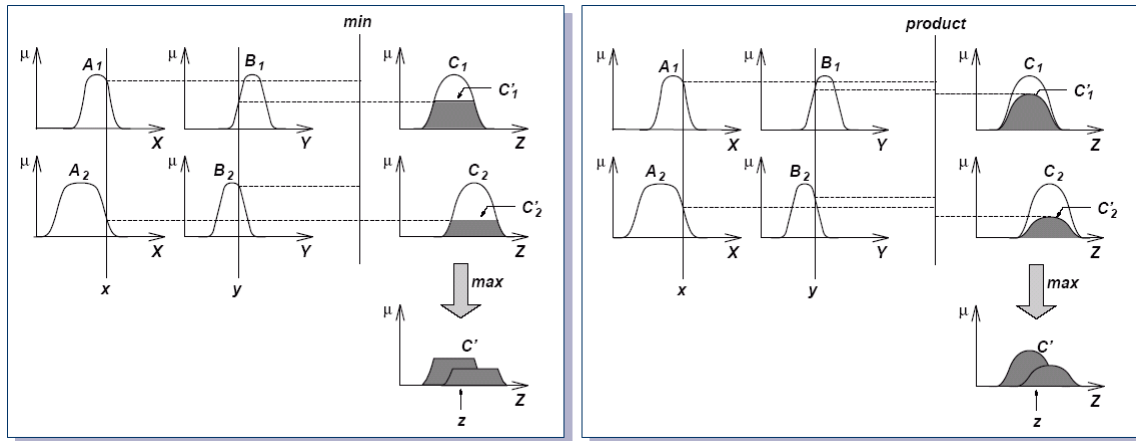


Figura 4.6: Sistema Mamdani com associação “min” (esquerda) de graus de pertinência (direita).
 Fonte: Zadeh (1965).

4.5.1 Primeiro exemplo: uma variável de entrada e uma de saída

Neste exemplo, a variável de entrada é rotulada "X" e a de saída "Y". Ambas possuem três valores linguísticos, a saber "pequeno", "médio" e "grande", mas os conjuntos fuzzy utilizados para representar estes valores linguísticos são diferentes para cada variável, conforme ilustra a Figura 4.7. Como a variável de entrada só pode assumir três valores, a base de regras será formada por:

- Se **X** é *Pequeno*, então **Y** é *Pequeno*.
- Se **X** é *Médio*, então **Y** é *Médio*.
- Se **X** é *Grande*, então **Y** é *Grande*.

A saída do sistema também é mostrada na figura 4.7, através da curva de entrada-saída. Esta curva mostra qual o valor da saída do sistema para um dado valor da variável de entrada. O valor da saída, conforme já foi mencionado anteriormente, é calculado utilizando uma base de regras.

4.5.2 Segundo exemplo: duas variáveis de entrada e uma de saída

Neste segundo exemplo, X e Y são as variáveis de entrada e Z é a variável de saída, conforme mostrado na figura a seguir. Na entrada, existem duas variáveis linguísticas, chamadas de "pequeno" e de "grande". As duas são representadas por conjuntos fuzzy distintos, conforme ilustra a Figura 4.8. A variável de saída Z, por sua vez, possui quatro valores linguísticos: "Grande Negativo", "Pequeno Negativo", "Pequeno

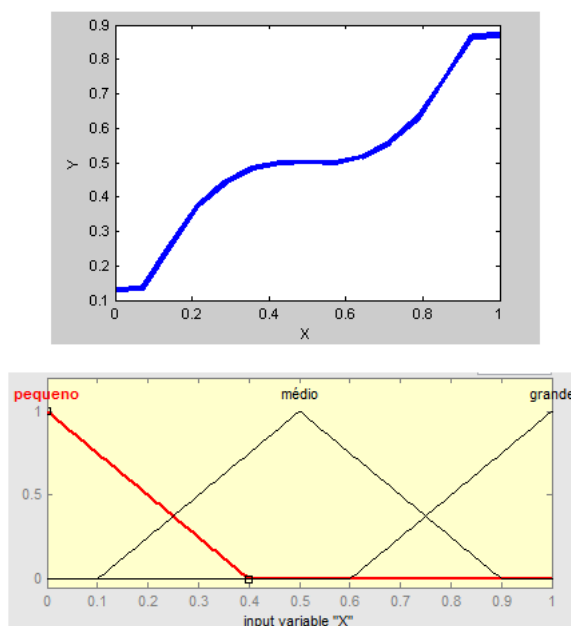


Figura 4.7: Variáveis linguísticas e curvas de entrada-saída para o Exemplo 1. Fonte: Jan (2004).

Positivo” e “Grande Positivo”, também lustrados na figura 4.8. Dessa forma, se existem duas variáveis de entrada, cada uma com dois valores possíveis, a base de regras terá todas as combinações possíveis de entrada, incluindo as quatro regras listadas a seguir:

- Se X é *Pequeno* e Y é *Pequeno*, então Z é *Grande Negativo*.
- Se X é *Pequeno* e Y é *Grande*, então X é *Pequeno Negativo*.
- Se X é *Grande* e Y é *Pequeno*, então X é *Pequeno Positivo*.
- Se X é *Grande* e Y é *Grande*, então X é *Grande Positivo*.

No gráfico de entrada-saída, mostrado na Figura 4.8, demonstra-se que a superfície representa os valores da saída do sistema para diferentes combinações das entradas.

Este capítulo discutiu as potencialidades e aplicações de Modelos Fuzzy, como nas áreas de tecnologia, indústria e saúde. Pereira e Boness (2009) utilizaram a modelagem Fuzzy para mensurar os níveis de violência no Estado da Bahia entre os anos de 2007 e 2010. A pesquisa analisou dados sociológicos da população como, por exemplo, cidade, idade, nacionalidade e hora das ocorrências. Dentre os benefícios proporcionados pela Lógica Fuzzy podem ser citados: possibilidade de criar soluções mais eficientes para problemas tratados com técnicas não-fuzzy, modelar sistemas não-lineares complexos e utilizar variáveis linguísticas para tratar dados imprecisos ou difíceis de serem medidos. A proposta abordada neste trabalho, utilizou os conceitos da lógica Fuzzy para definir um

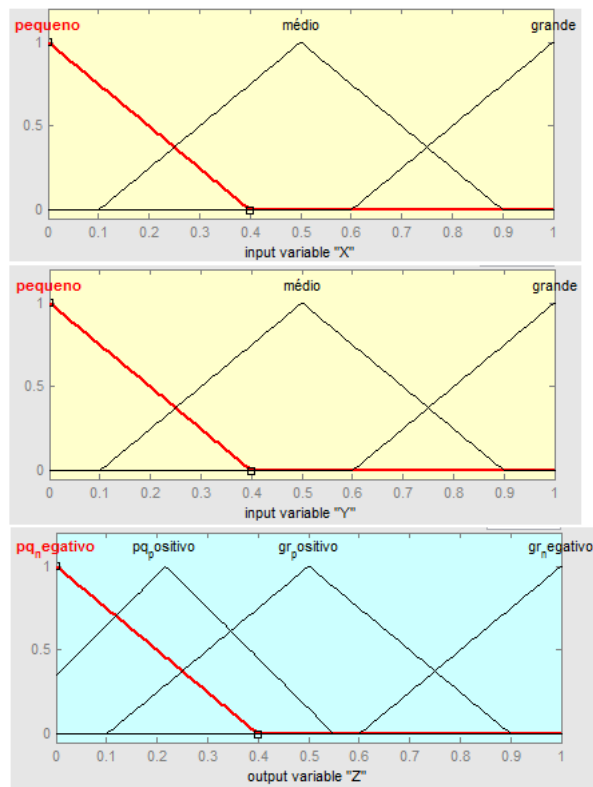


Figura 4.8: Variáveis linguísticas e curvas de entrada-saída para o exemplo 2. Fonte: Jan (2004).

modelo e analisar projetos de software que, em geral, são complexos e envolvem variáveis diversas. Neste trabalho foram determinadas variáveis linguísticas e regras Fuzzy, conforme será visto no próximo capítulo.

Modelo Proposto

Uma pesquisa realizada pelo [Gartner \(2011\)](#), empresa americana especializada em estudos sobre tecnologia, indicou números preocupantes sobre o mercado de produção de software. O estudo revelou os seguintes resultados:

- 31% de todos os projetos são cancelados antes do seu término, representando um desperdício da ordem de US\$ 81 bilhões;
- 53% de todos os projetos chegam ao final tendo custado 189% do valor estimado, representando US\$ 59 bilhões em custo adicional e mais: atrasam em até 222% da estimativa original, além de serem entregues com apenas 61% das características originalmente especificadas;
- Somente 16% dos projetos são entregues no prazo e dentro do orçamento.

No Brasil, ainda são poucos os estudos envolvendo avaliação de riscos em projetos de software. Em nosso país, pouco se sabe a respeito do desempenho dos projetos de software e sobre os fatores de risco que interferem no atendimento aos prazos do projeto, pois não há muitos estudos sobre o tema ([MACHADO, 2002](#)). Atento a essas dificuldades e reconhecendo a importância em indentificar e tratar os riscos, essa pesquisa apresenta uma proposta para calcular os graus de risco de projetos de software. Para aplicação do modelo, foi escolhida a Lógica Fuzzy, pois permite o tratamento de dados incertos ou imprecisos, algo comum em projetos de software, que variáveis linguísticas para tratar a complexidade de produção de aplicações computacionais, conforme será apresentado mais adiante.

O modelo contou com o apoio de profissionais atuantes no setor de produção de software, os quais ajudaram na escolha das variáveis que fazem parte desta pesquisa. O grupo participante atuava no Núcleo de Educação a Distância do SENAI e tem o perfil exemplificado na figura 5.1.

5.1 Informações do Modelo

Para execução da pesquisa e aplicação do modelo foi definido um grupo de variáveis envolvendo temas presentes na produção do sistema e na gestão dos recursos

CARGO	FORMAÇÃO SUPERIOR NA ÁREA DE ATUAÇÃO	EXPERIÊNCIA MAIOR QUE TRÊS ANOS EM PROJETOS DE SOFTWARE	CONHECIMENTO DE NORMAS DE QUALIDADE DE SOFTWARE
Analista de Sistemas (6)	80%	95%	80%
Programadores (6)	80%	90%	50%
Web Designs (6)	50%	100%	50%
Especialista 3D (2)	50%	50%	50%
Design (2)	50%	50%	50%
Jornalistas (2)	100%	50%	50%
Pedagogos (4)	100%	50%	25%
Engenheiro Químico (1)	100%	100%	100%
Desenhista (1)	100%	0	100%
Gestor (1)	100%	100%	100%

Figura 5.1: Equipe participante da pesquisa. Fonte: Autor.

humanos. O grau de risco das variáveis foi criado usando a Lógica Fuzzy, e esse pode variar de 0 até 1. Neste trabalho foram definidos os termos risco baixo, médio ou alto para qualificar as análises de cada projeto.

A primeira etapa do processo foi selecionar, de forma aleatória, quatro projetos de software, finalizados em no máximo 12 meses, entre o período de 2010 e 2011. A segunda etapa definiu variáveis de entrada de dados para posterior utilização no modelo. A etapa final realizou o cálculo dos graus de risco de cada variável, usando Lógica Fuzzy com o software MatLAB. A pesquisa definiu também as regras que fazem parte do modelo Fuzzy, usando uma descrição linguística, procedimento comum para aplicação da Lógica Fuzzy.

Durante a seleção dos projetos desta pesquisa, foram encontradas diversas informações que poderiam ser usadas no trabalho, a exemplo de linguagem de programação utilizada, uso de normas de qualidade, clareza dos requisitos, formação acadêmica da equipe e outras. Essas informações podem gerar riscos no projeto, quando não tratadas de forma adequada. No trabalho sobre Avaliação de Riscos em Projetos de Software [Leopoldino \(2004\)](#), foram identificados os seguintes riscos classificados como graves: volatilidade dos requisitos, custos ou prazos irrealistas e mudanças frequentes nos objetivos do projeto. São temas complexos e que devem ser verificados cuidadosamente pelos gestores, pois podem prejudicar a produção do software e fazem parte desta pesquisa.

Para implementação do trabalho, foram estabelecidas variáveis que envolvem questões relacionadas à gestão de pessoas e dificuldades técnicas de produção. Para simplificar o modelo e possibilitar a avaliação da pesquisa, foram selecionadas três variáveis para questões de sistema e três para os recursos humanos, assim classificadas:

- (a) VARIÁVEIS DE ENTRADA DE SISTEMA - são aquelas envolvendo questões do sis-

tema. Para o estudo dessas variáveis foram escolhidas: mudança de escopo/objetivos, complexidade do software e volatilidade dos requisitos. A variável “mudanças de escopo ou nos objetivos” são aquelas realizadas de forma frequente, após início dos trabalhos. A variável “complexidade do software trata do nível de dificuldade ou de complexidade do problema que será resolvido. A “volatilidade dos requisitos” trata do uso de requisitos voláteis, ou seja, aqueles que mudam com frequência e podem aumentar os riscos.

- (b) VARIÁVEIS DE ENTRADA DE RECURSOS HUMANOS - são as variáveis que tratam da equipe participante do projeto. Em recursos humanos foram escolhidas: terceirizados no processo, atuação da equipe em projetos paralelos e taxa de rotatividade dos colaboradores (*turnover*). A variável “terceirizados no processo”, analisa o percentual de uso de profissionais terceirizados que fazem parte do projeto. A “atuação da equipe em projetos paralelos” indica se os profissionais estão trabalhando em outros projetos na organização, ou seja, em mais de um projeto simultaneamente. A última variável definida foi taxa de “rotatividade da empresa”, a qual verifica o percentual de rotatividade no setor de produção de software.

As variáveis aqui mencionadas definem o grau de influência que podem exercer no projeto, podendo aumentar ou diminuir o grau de risco do projeto, de acordo com os dados fornecidos pelo gestor. Para verificar o valor do grau do risco deve-se inserir no modelo os valores de entrada para cada variável, isso refletirá a realidade de produção do software analisado. As definições dos termos linguísticos levou em consideração situações comuns vivenciadas com maior frequência por profissionais da área. O grau de risco de cada variável ficou definido com os seguintes termos linguísticos: baixo, variando de 0 até 0,4; risco médio entre 0,41 e 0,7 e risco alto de 0,71 a 1. O sistema de inferência Fuzzy para esse trabalho é mostrado na figura 5.2).

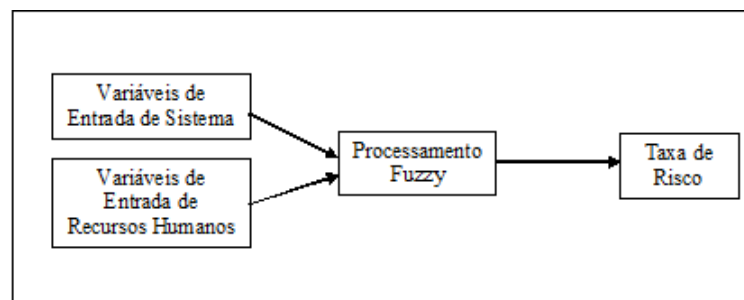
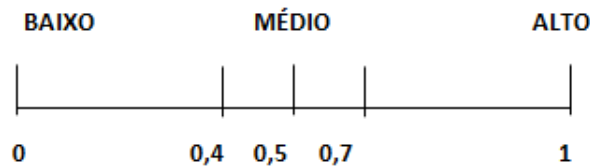


Figura 5.2: Esquema do sistema Fuzzy - As variáveis de entrada são inseridas, o processamento é realizado, atendendo as regras definidas e o grau de risco é indicado - Fonte: Autor

A Tabela 5.1, mostra o grau de risco que o projeto apresenta e a semântica equivalente. Fonte: Autor.

Tabela 5.1: Grau de risco e semântica equivalente - Fonte: Autor

GRAU DO RISCO	SEMANTICA
Até 0,4	Grau de risco da variável é Baixa
Entre 0,41 e 0,7	Grau de risco da variável é Média
Entre 0,71 e 1	Grau de risco da variável é Alta



5.2 Variáveis Linguísticas do Sistema e Base de Regras

Esta seção apresenta as variáveis de entrada para sistemas e recursos humanos, base de regras com suas descrições e os riscos associados. A base de regras envolve todas as possibilidades envolvendo as variáveis linguísticas que foram definidas. Os termos linguísticos serão usados para para transcrever a base de regras Fuzzy. Para exemplificar o uso da base de regras Fuzzy nesse modelo, podem ser definidos para um dado projeto: se equipe é inexperiente em gestão de requisitos, o risco é definido pelo gestor como 0,9 (risco alto); se o grupo é inexperiente em projetos complexos, o risco será 0,8 e, se os requisitos são associados às preferências de usuários, o risco informado é de 0,9. As regras aplicadas definem a variável linguística Requisitos Voláteis com grau de risco Alto. O conjunto de regras Fuzzy tem a forma:

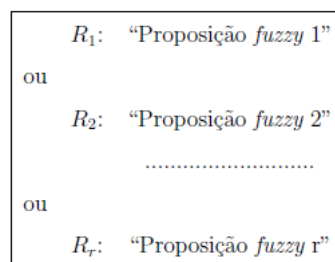


Figura 5.3: Forma para grupo de regras Fuzzy - Fonte: Autor.

Para aplicação do modelo, o gestor deve pontuar cada uma das sentenças com valores que podem variar entre 0 (risco baixo) e 1 (risco alto). Ficou assim definido as variáveis linguísticas do sistema:

- ESCOPO OU OBJETIVO MAL DEFINIDOS: pouco entendimento do problema; usuários envolvidos não são os indicados para apoio no projeto; a equipe envolvida é inexperiente na gestão do escopo.

- **COMPLEXIDADE DO SOFTWARE:** uso de novas tecnologias; dificuldades técnicas em implementar a solução; o assunto é novo ou não é familiar para a equipe.
- **REQUISITOS VOLÁTEIS:** há mudanças constantes na definição dos requisitos; equipe é inexperiente em engenharia de requisitos; há preferências de usuários ou há políticas governamentais envolvidas.

As variáveis linguísticas dos Recursos Humanos foram assim classificadas:

- **TERCEIRIZAÇÃO DE PESSOAL:** até metade da equipe é terceirizada; terceirizados com boa experiência em projeto de software; terceirizados seguem normas de qualidade reconhecidas pelo mercado.
- **EQUIPE EM PROJETOS PARALELOS:** atua em outros projetos que estão sendo finalizados; equipe alocada em outros projetos o que prejudica o início de outros; equipe alocada em outros projetos com prazos vencidos.
- **ROTATIVIDADE DA EQUIPE:** índice de rotatividade; perfil com dificuldades de reposição; políticas de retenção de pessoal.

5.2.1 *Variável Linguística Mudanças no Escopo ou Objetivos Mal Definidos*

As mudanças frequentes no escopo e nos objetivos do trabalho causam dificuldades ao projeto, pois geram dúvidas na equipe e podem criar atrasos no cronograma. O software é orçado e gerenciado levando em consideração objetivos e escopo previamente analisados, os quais não devem sofrer alterações críticas durante a execução. Alterações no escopo e nos objetivos do projeto criam problemas na alocação de pessoal e incremento nos custos, aumentando dessa forma as chances de insucesso. Definição de objetivos mal especificados pode levar a equipe a decisões equivocadas e comprometer o êxito do trabalho

A figura a seguir mostra a base de regras usada para analisar a variável Mudanças no Escopo ou Objetivos Mal Definidos.

Descrição da variável e o risco associado:

- O projeto não sofre mudanças frequentes no escopo ou nos objetivos, os usuários envolvidos são indicados para participar do projeto e a equipe possui boa experiência em gestão de escopo - **risco baixo**

Variáveis de Sistema	Entendimento do problema	Usuários envolvidos	Equipe envolvida	Risco Escopo
ESCOPO OU OBJETIVOS MAL DEFINIDOS	BAIXO	BAIXO	BAIXO	BAIXO
	BAIXO	BAIXO	MÉDIO	
	BAIXO	MÉDIO	BAIXO	
	MÉDIO	BAIXO	BAIXO	
	MÉDIO	MÉDIO	MÉDIO	MÉDIO
	MÉDIO	MÉDIO	BAIXO	
	BAIXO	MÉDIO	MÉDIO	
	MÉDIO	BAIXO	MÉDIO	
	MÉDIO	MÉDIO	ALTO	
	MÉDIO	ALTO	MÉDIO	
	ALTO	MÉDIO	MÉDIO	ALTO
	ALTO	ALTO	ALTO	
	ALTO	ALTO	BAIXO	
	BAIXO	ALTO	ALTO	
	ALTO	BAIXO	ALTO	
MÉDIO	ALTO	ALTO		
ALTO	MÉDIO	ALTO		
ALTO	ALTO	MÉDIO		

Figura 5.4: Regras para Escopo ou Objetivos Mal Definidos - Fonte: Autor.

- O projeto sofre poucas mudanças no escopo ou nos objetivos, parte dos usuários envolvidos são indicados para participar do projeto e a equipe possui experiência em gestão de escopo - **risco médio**
- O projeto sofre mudanças frequentes no escopo ou nos objetivos, os usuários envolvidos não são indicados para participar do projeto e a equipe não possui experiência em gestão de escopo - **risco alto**

5.2.2 Variável Linguística Complexidade do Software

A variável Complexidade do Software permite analisar o nível de dificuldade que o software possui e seu grau de influência. Essa complexidade pode ocorrer devido a dificuldades do entendimento do problema, uso de tecnologias não dominadas pela equipe ou por ser algo desafiador e difícil de ser implementado pelo grupo de desenvolvimento.

A figura a seguir mostra a base de regras usada para analisar a variável Complexidade do Software:

Descrição da variável e o risco associado:

- O software possui baixa complexidade de produção, usa tecnologias conhecidas e dominadas pelo time e o tema do trabalho é conhecido pela equipe - **risco baixo**
- O software possui média complexidade de produção, usa tecnologias pouco conhecidas e dominadas pelo time e o tema do trabalho é conhecido somente por parte da equipe - **risco Médio**

Variáveis de Sistema	Novas tecnologias	Dificuldades em implementar a solução	Assunto novo ou não familiar	Risco complexidade
COMPLEXIDADE DO SOFTWARE	BAIXO	BAIXO	BAIXO	BAIXO
	BAIXO	BAIXO	MÉDIO	
	BAIXO	MÉDIO	BAIXO	
	MÉDIO	BAIXO	BAIXO	
	ALTO	BAIXO	BAIXO	
	BAIXO	BAIXO	ALTO	
	BAIXO	ALTO	BAIXO	MÉDIO
	MÉDIO	MÉDIO	MÉDIO	
	MÉDIO	MÉDIO	BAIXO	
	BAIXO	MÉDIO	MÉDIO	
	MÉDIO	MÉDIO	ALTO	
	MÉDIO	ALTO	MÉDIO	
	ALTO	MÉDIO	MÉDIO	ALTO
	ALTO	ALTO	ALTO	
	ALTO	ALTO	BAIXO	
	BAIXO	ALTO	ALTO	
	ALTO	BAIXO	ALTO	
	MÉDIO	ALTO	ALTO	
ALTO	ALTO	MÉDIO		

Figura 5.5: Regras para Complexidade do Software - Fonte: Autor.

- O software possui alta complexidade de produção, usa tecnologias desconhecidas e dominadas pelo time e o tema do trabalho é novo para a equipe - **risco alto**

5.2.3 Variável Linguística Volatilidade dos Requisitos

Requisitos voláteis prejudicam um projeto de software, (PRESSMAN, 2005). Fatores como sonegação de informação, imaturidade do cliente ou inexperiência em gestão de requisitos são fatores que implicam na volatilidade de requisitos. Projetos de software relacionados a política governamentais podem ter dificuldades de execução, aumentando a volatilidade dos requisitos, devido a instabilidade na gestão ou pela influência de fatores externos. A volatilidade pode aumentar as chances de riscos do projeto e gerar atrasos na entrega, retrabalho e aumento de custos de operação. Segundo (SOMMERVILLE, 2003), requisitos voláteis são prejudiciais ao projeto e devem ser gerenciados de forma cuidadosa.

A figura a seguir mostra a base de regras usada para analisar a variável Volatilidade dos Requisitos:

Descrição da variável e o risco associado:

- Os requisitos do projeto são estáveis, a equipe tem boa experiência em gestão de requisitos e o tema não envolve políticas governamentais - **risco baixo**
- Os requisitos do projeto são pouco voláteis, a equipe tem pouca experiência em gestão de requisitos e o tema envolve políticas governamentais- **risco médio**

Variáveis de Sistema	Mudanças constantes	Equipe inexperiente	Preferências de usuários	Risco requisitos
REQUISITOS VOLÁTEIS	BAIXO	BAIXO	BAIXO	BAIXO
	BAIXO	BAIXO	MÉDIO	
	BAIXO	MÉDIO	BAIXO	
	MÉDIO	BAIXO	BAIXO	
	ALTO	BAIXO	BAIXO	
	BAIXO	BAIXO	ALTO	MÉDIO
	MÉDIO	MÉDIO	MÉDIO	
	MÉDIO	MÉDIO	BAIXO	
	BAIXO	MÉDIO	MÉDIO	
	MÉDIO	BAIXO	MÉDIO	
	MÉDIO	MÉDIO	ALTO	ALTO
	MÉDIO	ALTO	MÉDIO	
	ALTO	MÉDIO	MÉDIO	
	ALTO	ALTO	ALTO	
	BAIXO	ALTO	ALTO	
ALTO	BAIXO	ALTO	ALTO	
MÉDIO	ALTO	ALTO		
ALTO	MÉDIO	ALTO		
ALTO	ALTO	MÉDIO		

Figura 5.6: Regras para Complexidade do Software - Fonte: Autor.

- Os requisitos do projeto são muito voláteis, a equipe não possui experiência em gestão de requisitos e o tema central do projeto envolve políticas governamentais - **risco alto**

5.3 Variáveis Linguísticas de Entrada dos Recursos Humanos

Esta seção apresenta as variáveis relacionadas aos recursos humanos do projeto. Assim como na escolha de variáveis de sistema, as variáveis dessa seção foram definidas pelo autor e visam contribuir com os gestores de projetos de software, permitindo identificar fatores humanos que podem gerar riscos no projeto.

5.3.1 Variável Linguística Terceirização de Pessoal

Visando reduzir prazos e diminuir custos, cada vez mais empresas de software decidem terceirizar parte de seu processo de produção. Terceirizar pode trazer benefícios como aumento de produtividade, foco em áreas estratégicas e possibilita agilizar etapas de criação. No entanto, a terceirização de pessoal, pode implicar em redução de qualidade e atrasos no cronograma. O desconhecimento de padrões ou normas da empresa por parte dos terceirizados pode criar problemas no processo e aumentar as chances de riscos do projeto. Essa variável foi proposta para verificar o impacto da terceirização de pessoal no projeto.

A figura a seguir mostra a base de regras usada para analisar a variável Terceirização de Pessoal:

Variáveis de RH	Até 50% da equipe ou não usa terceiros	Terceirizados com boa experiência em projeto de software	Terceirizados seguem normas de qualidade reconhecidas pelo mercado	Risco Escopo
Terceirização de Pessoal	BAIXO	BAIXO	BAIXO	BAIXO
	BAIXO	BAIXO	MÉDIO	
	BAIXO	MÉDIO	BAIXO	
	MÉDIO	BAIXO	BAIXO	
	MÉDIO	ALTO	BAIXO	
	MÉDIO	MÉDIO	MÉDIO	MÉDIO
	MÉDIO	MÉDIO	BAIXO	
	BAIXO	MÉDIO	MÉDIO	
	MÉDIO	BAIXO	MÉDIO	
	MÉDIO	MÉDIO	ALTO	
	MÉDIO	ALTO	MÉDIO	
	ALTO	MÉDIO	MÉDIO	
	BAIXO	ALTO	MÉDIO	ALTO
	ALTO	ALTO	ALTO	
	ALTO	ALTO	BAIXO	
	BAIXO	ALTO	ALTO	
	ALTO	BAIXO	ALTO	
	MÉDIO	ALTO	ALTO	
ALTO	MÉDIO	ALTO		
ALTO	ALTO	MÉDIO		

Figura 5.7: Regras para Terceirização de Pessoal - Fonte: Autor.

Descrição da variável e risco associado:

- O projeto não requer terceirização de pessoal ou até metade da equipe é terceirizada, a equipe terceirizada possui boa experiência em desenvolvimento de projetos de software e segue normas de qualidade de reconhecidas pelo mercado - **risco baixo**
- Até metade da equipe é terceirizada, a equipe terceirizada possui pouca experiência em desenvolvimento de projetos de software e segue normas de qualidade de reconhecidas pelo mercado - **risco médio**
- Acima da metade da equipe será terceirizada no projeto, a equipe terceirizada não possui boa experiência em desenvolvimento de projetos de software e não utiliza normas de qualidade - **risco alto**

5.3.2 Variável Linguística Equipe em Projetos Paralelos

Prazos apertados, pessoal reduzido e busca constante de produtividade podem implicar em alocação de profissionais em diversos projetos paralelos. Em atividades do processo de software, a exemplo de etapas de programação, a atuação desses profissionais em projetos paralelos podem gerar dificuldades, exigindo do gestor atenção e acompanhamento constante. As atividades de programação do sistema exigem foco, dedicação e atenção do profissional. Alocar profissionais em diversos projetos ao mesmo tempo costuma ser uma realidade nas empresas, mas isso pode levar a atrasos no cronograma e

dificultar o trabalho do gestor. A variável foi escolhida devido à constante alocação de profissionais em diversos projetos paralelos.

A figura a seguir mostra a base de regras usada para analisar a variável Equipe em Projetos Paralelos:

Variáveis de RH	Atuando em outros projetos sendo finalizados	Em outros projetos e impactando na alocação de pessoal	Em outros projetos e com prazos já estourados	Risco Projetos Paralelos
Equipe em Projetos Paralelos	BAIXO	BAIXO	BAIXO	BAIXO
	BAIXO	BAIXO	MÉDIO	
	BAIXO	MÉDIO	BAIXO	
	MÉDIO	BAIXO	BAIXO	MÉDIO
	MÉDIO	MÉDIO	MÉDIO	
	MÉDIO	MÉDIO	BAIXO	
	BAIXO	MÉDIO	MÉDIO	
	MÉDIO	BAIXO	MÉDIO	
	MÉDIO	MÉDIO	ALTO	
	MÉDIO	ALTO	MÉDIO	ALTO
	MÉDIO	MÉDIO	MÉDIO	
	ALTO	MÉDIO	MÉDIO	
	BAIXO	ALTO	MÉDIO	
	ALTO	ALTO	ALTO	
	ALTO	ALTO	BAIXO	
BAIXO	ALTO	ALTO	ALTO	
ALTO	BAIXO	ALTO		
MÉDIO	ALTO	ALTO		
ALTO	MÉDIO	ALTO		
ALTO	ALTO	MÉDIO	ALTO	

Figura 5.8: Regras para Equipe em Projetos Paralelos - Fonte: Autor.

Descrição da variável e risco associado:

- A equipe está dedicada somente a um projeto - **risco baixo**
- A equipe está atuando em poucos projetos paralelos, gerando pouco impacto na alocação de pessoal - **risco médio**
- A equipe está atuando em muitos projetos paralelos, gerando dificuldades na alocação de pessoal e os prazos foram estourados - **risco alto**

5.3.3 Variável Linguística Rotatividade de Pessoal

A Rotatividade de Pessoal pode prejudicar os projetos de software, por essa razão foi um tema escolhido para a pesquisa. A saída de profissionais no decorrer do projeto gera problemas que podem aumentar os riscos, principalmente quando a organização não possui pessoal disponível para substituição. A rotatividade implica também em mais custos, pois é necessário contratação de pessoal e treinamento de novos colaboradores, além disso, há um intervalo de tempo necessário para que os esses profissionais assimilem

a cultura da empresa ou aprenda normas de qualidade existentes. A empresa analisada nessa pesquisa tinha como meta aceitável de rotatividade até seis por cento ao ano.

A figura a seguir mostra a base de regras usadas para analisar a variável Rotatividade de Pessoal:

Variáveis de RH	Índice de Rotatividade	Perfil com dificuldades de reposição	Políticas de retenção de pessoal	Risco Rotatividade
Rotatividade de Pessoal	BAIXO	BAIXO	BAIXO	BAIXO
	BAIXO	BAIXO	MÉDIO	
	BAIXO	MÉDIO	BAIXO	
	MÉDIO	BAIXO	BAIXO	MÉDIO
	MÉDIO	MÉDIO	MÉDIO	
	MÉDIO	MÉDIO	BAIXO	
	BAIXO	MÉDIO	MÉDIO	
	MÉDIO	BAIXO	MÉDIO	
	MÉDIO	MÉDIO	ALTO	
	MÉDIO	ALTO	MÉDIO	ALTO
	ALTO	MÉDIO	MÉDIO	
	ALTO	ALTO	ALTO	
	ALTO	ALTO	BAIXO	
	BAIXO	ALTO	ALTO	
	ALTO	BAIXO	ALTO	
MÉDIO	ALTO	ALTO	ALTO	
ALTO	MÉDIO	ALTO		
ALTO	ALTO	MÉDIO		

Figura 5.9: Regras para Rotatividade de Pessoal - Fonte: Autor.

Descrição da variável e risco associado:

- A rotatividade da equipe é menor que 6% nos últimos 12 meses - **risco baixo**
- Índice médio de rotatividade, perfil com dificuldades de reposição, necessidade de revisão na política de retenção de pessoal - **risco médio**
- Índice alto de rotatividade, perfil com grandes dificuldades de reposição, política de retenção de pessoal inadequadas para o mercado - **risco alto**

5.4 Cálculo dos Graus de Risco e Função de Pertinência

Para calcular os graus de risco do projeto, foram analisadas inicialmente duas possibilidades, a primeira seria utilizar as variáveis em um processamento único, ou seja, as variáveis do sistema e dos recursos humanos estariam juntas e teriam um cálculo unificado. A segunda possibilidade, escolhida para esse modelo, separou as variáveis (i) sistema e (ii) recursos humanos. Cada uma delas teve o cálculo realizado com modelagem Fuzzy.

O cálculo dos graus de risco, separando as variáveis do sistema e dos recursos humanos, foi o escolhido, pois possibilita aos gestores visualizar quais pontos são mais

prejudiciais aos projetos executados. Essas informações podem gerar um registro histórico com pontos de melhorias para projetos futuros. Os resultados apresentados de forma separada permitem um acompanhamento mais próximo de cada um dos fatores, com possibilidade de ajustes de pessoal ou nas tecnologias utilizadas.

As figuras a seguir exemplificam a representação gráfica das variáveis linguísticas usadas no modelo:

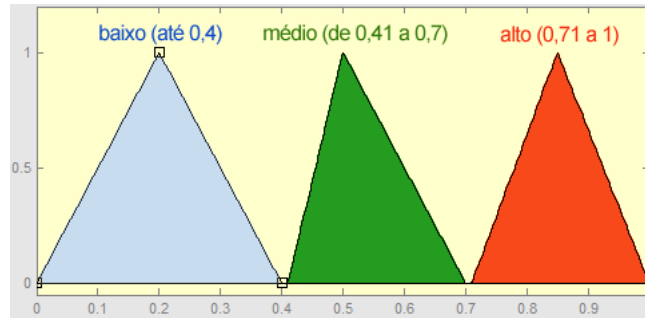


Figura 5.10: Função de pertinência para Grau de Risco das Variáveis do Sistema. Fonte: Autor.

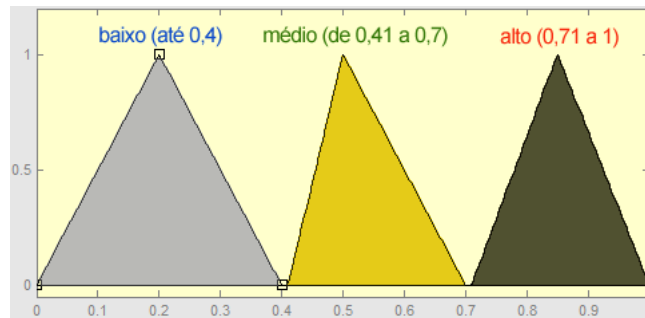


Figura 5.11: Função de pertinência para Grau de Risco das Variáveis dos Recursos Humanos. Fonte: Autor

A função de pertinência que foi usada para associar os conjuntos Fuzzy aos termos linguísticos do modelo é do tipo triangular. Para exemplificar, supondo que V seja um grupo de coordenadas, então $V = (v_1, v_2, v_3, v_4)$, a função de pertinência para esse grupo de coordenadas será:

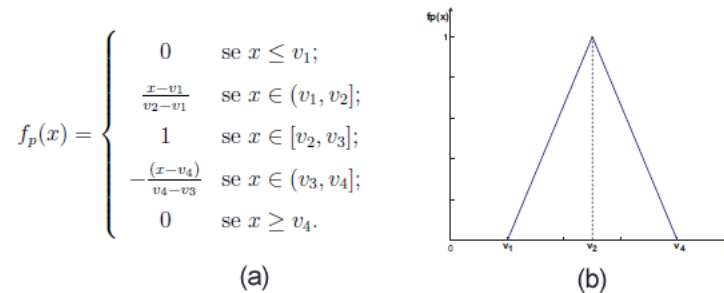


Figura 5.12: Função de pertinência Triangular. Fonte: Autor

5.5 Resultados e Discussão

Esta seção apresenta quatro projetos e informações relativas ao escopo, objetivos e informações adicionais. Após a apresentação do projeto, a modelagem Fuzzy é aplicada utilizando as regras de Fuzzificação e dados das Variáveis do Sistema e dos Recursos Humanos. Na análise das variáveis, utilizando o software MatLAB, o tema mais influente no aumento do risco de cada uma delas, foi escolhido como base para apresentação dos gráficos, comparando assim com um dos dois outros temas. Cada projeto aqui listado tem um resumo com características e informações que foram usadas para formar o cenário ideal, o cenário previsto e o que foi realizado. O cenário previsto é uma previsão elaborada de acordo com os dados iniciais do projeto, ou seja, o projeto ainda não teve início. O cenário realizado trata dos dados que foram coletados ao final do projeto.

5.5.1 Projeto 1 - Rede Social para o Governo do Estado da Bahia

O projeto foi desenvolvido para o Governo do Estado da Bahia. Tratava-se de uma rede social para uso dos professores e alunos de escolas públicas. O objetivo do projeto era compartilhar conhecimento e informações entre professores e alunos. O projeto tinha prazo estimado de seis meses e deveria funcionar via internet.

Um problema identificado foi que os requisitos mudavam a todo instante e haviam desafios na transmissão de vídeo, o que ainda pouco conhecido pela equipe naquele momento. O grupo não havia realizado projetos semelhantes e haviam diversos projetos executados pelas equipes em paralelo. A empresa tinha um índice de rotatividade alto e havia mudanças frequentes no escopo do projeto. Os usuários participantes das entrevistas iniciais tinham opiniões pouco similares sobre o escopo e os objetivos do trabalho. As atividades de animação e ilustração da rede foram terceirizadas e o trabalho foi finalizado no prazo.

CENÁRIO IDEAL PARA O PROJETO

Dentre os problemas que poderiam afetar o risco, foi observado pela equipe que o prazo e a complexidade do projeto exigia um maior número de profissionais atuantes, além de profissionais com experiência em uso de software livre, visto que a aplicação também rodaria nos Infocentros do Governo. Foi verificado também que haveria problemas com a gestão do escopo e dos requisitos e isso seria minimizado com o apoio dos profissionais de T.I. do Governo. A terceirização deveria ser executada por profissionais experientes do mercado para acelerar a entrega do trabalho e garantir mais profissionais da empresa alocados no projeto.

APLICAÇÃO DO MODELO - ANALISANDO VARIÁVEIS DE SISTEMA: CENÁRIO PREVISTO

Os valores de entrada usados para a criação do cenário previsto e para realização do cálculo dos graus de risco, serão aqui informados:

- Valores para Escopo - pouco entendimento do problema (0,84); usuários envolvidos não são os indicados (0,91) e equipe envolvida é inexperiente na gestão do escopo (0,50). O resultado de saída foi 0,85 (risco alto);
- Valores para Complexidade do Software - uso de novas tecnologias (0,95); dificuldades técnicas em implementar a solução (0,90) e assunto é novo ou não é familiar para a equipe (0,91). O resultado de saída foi 0,91 (risco alto);
- Valores para Requisitos Voláteis: mudanças constantes na definição dos requisitos (0,91); equipe é inexperiente em engenharia de requisitos (0,51) e há preferências de usuários ou há políticas governamentais (0,81). Para esses valores de entrada o grau do risco foi 0,94 (risco alto).

Os resultados demonstram a expectativa da equipe com um risco considerado alto. Foram considerados como fatores determinantes: escolha inadequada dos usuários para apoio inicial da equipe na definição do escopo e objetivos, o tema central do projeto era desconhecido pelo grupo e os requisitos sofriam mudanças constantemente.

O cenário previsto para esse projeto, analisando as variáveis do sistema, é mostrado nas três figuras a seguir:

Variável do Sistema - Escopo ou Objetivos Mal Definidos (cenário previsto)

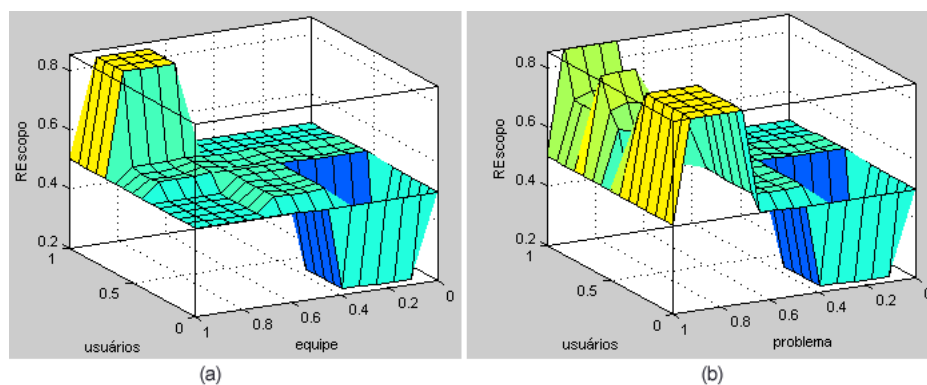


Figura 5.13: Gráfico de Cenário Previsto para Variável do Sistema Escopo ou Objetivos Mal Definidos. Fonte: Autor.

A figura 5.13 (a) mostra o gráfico gerado pelo software Matlab com os dados comparados entre usuários de apoio do projeto não indicados para definição da equipe

e equipe inexperiente na gestão do escopo. A imagem mostra um aumento no risco do escopo quando usuários não indicados estão atuando no apoio de levantamento de dados. O gráfico da figura 5.13 (b) tem semelhanças com a figura 5.13 (a) ao indicar influência no risco quando usuários de apoio do projeto não são os indicados, agora comparando com os dados de pouco entendimento do problema.

Variável do Sistema - Complexidade do Software (cenário previsto)

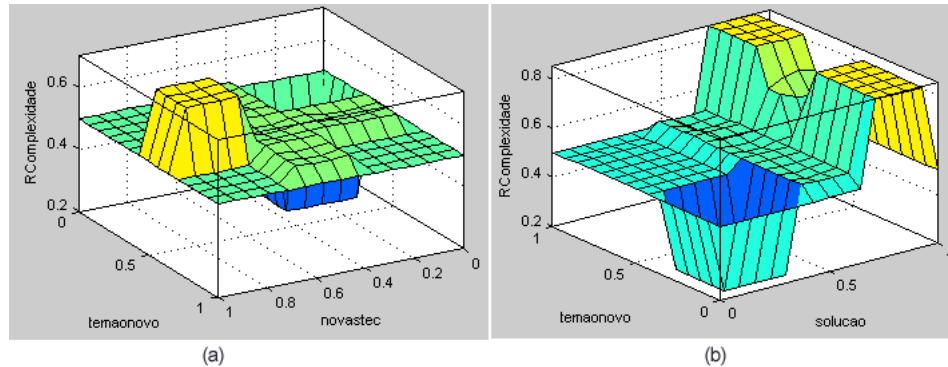


Figura 5.14: Gráfico de Cenário Previsto para Variável do Complexidade do Software. Fonte: Autor.

Nas figuras 5.14 (a) e 5.14 (b) os gráficos tratam da variável linguística Complexidade do Software. A influência de um tema novo ou não familiar para a equipe ocorre nas duas imagens, sendo que na figura 5.14 (b) existe um impacto maior no risco, quando trata da questão de dificuldades técnicas em implementar a solução.

Variável do Sistema - Requisitos Voláteis (cenário previsto)

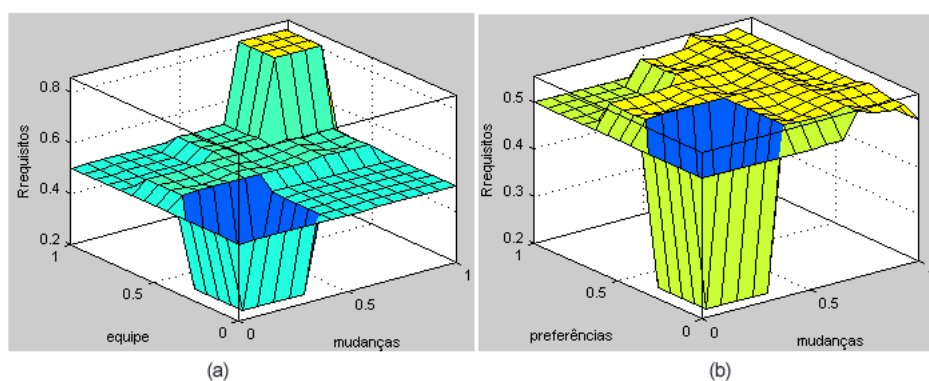


Figura 5.15: Gráfico de Cenário Previsto para Variável Requisitos Voláteis. Fonte: Autor.

Os gráficos das figuras 5.15 (a) e 5.15 (b) mostram o risco para o tema requisitos voláteis, identificando que mudanças frequentes nos requisitos causam um risco maior. Observa-se que o tema preferências de usuários ou políticas governamentais tiveram influencia sobre o risco.

APLICAÇÃO DO MODELO - ANALISANDO VARIÁVEIS DE SISTEMA: CENÁRIO

REALIZADO

As variáveis linguísticas no cenário realizado ficaram com os seguintes valores:

- Escopo: pouco entendimento do problema (0,88); usuários envolvidos não são os indicados (0,98) e equipe envolvida é inexperiente na gestão do escopo (0,62). Saída com valor de 0,89 (risco alto);
- Complexidade do Software: uso de novas tecnologias (0,92); dificuldades técnicas em implementar a solução (0,90) e assunto é novo ou não é familiar para a equipe (0,96). Saída de 0,89 (risco alto);
- Requisitos Voláteis, valores de entrada: mudanças constantes na definição dos requisitos (0,93); equipe é inexperiente em engenharia de requisitos (0,72) e há preferências de usuários ou políticas governamentais (0,73). O resultado de Saída calculado pelo modelo foi 0,89.

O cenário realizado para esse projeto, analisando as variáveis do sistema, é mostrado nas três figuras a seguir:

Variável do Sistema - Escopo ou Objetivos Mal Definidos (cenário realizado)

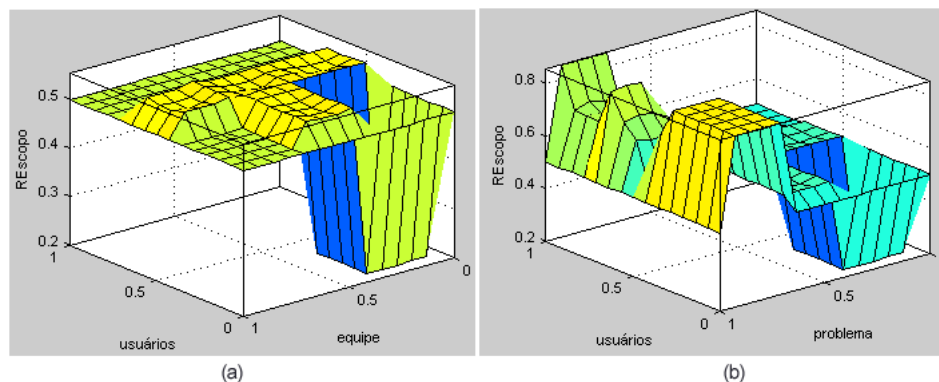


Figura 5.16: Gráfico de Cenário Realizado para Variável Escopo ou Objetivos Mal Definidos. Fonte: Autor.

Nas figuras 5.16 (a) e 5.16 (b), os gráficos tratam da variável linguística Escopo ou Objetivo mal definidos. Foram comparados os dados de usuários envolvidos que não eram os indicados para apoio no projeto e este foi o maior problema.

Variável do Sistema - Complexidade do Software (cenário realizado)

As figuras 5.17 (a) e 5.17 (b) demonstram gráficos da variável linguística Complexidade do Software. Observa-se que, assim como no cenário previsto, o que mais

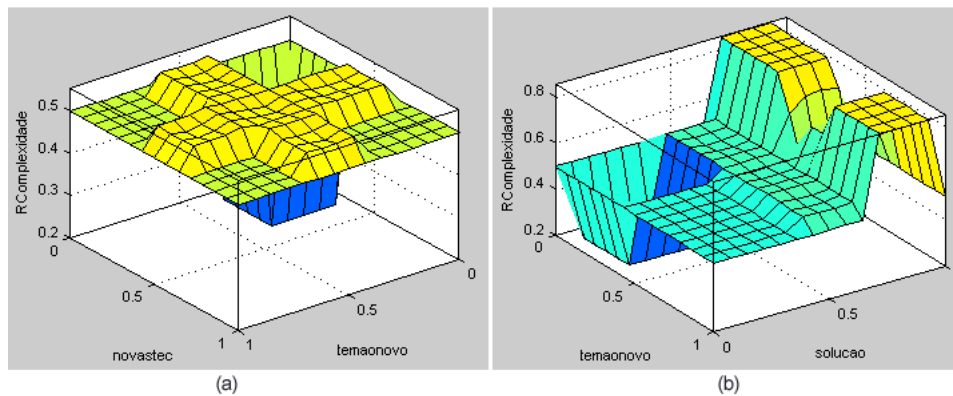


Figura 5.17: Gráfico de Cenário Previsto para Variável do Complexidade do Software. Fonte: Autor.

afetou o risco da variável foi o uso de novas tecnologias, ainda pouco conhecidas pela equipe e a dificuldade técnicas em implementar a solução, pois o problema envolveu questões de streaming de vídeo que demandavam novos softwares e grande largura de banda.

Variável do Sistema - Requisitos Voláteis (cenário realizado)

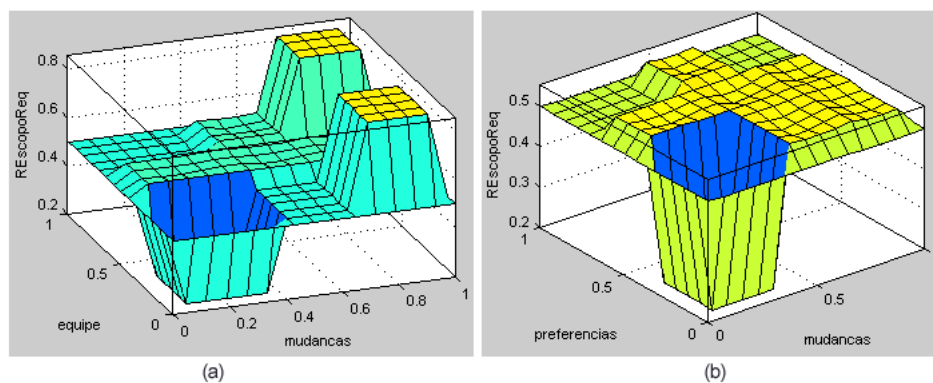


Figura 5.18: Gráfico de Cenário Previsto para Variável Requisitos Voláteis. Fonte: Autor.

Nas figuras 5.18 (a) e 5.18 (b) os gráficos tratam da variável linguística Requisitos Voláteis. Observa-se que, assim como no cenário previsto, o que mais atingiu o risco da variável foi a modificação frequente dos requisitos. Isso gerou dificuldades de entendimento do problema pela equipe, bem como um alto risco.

Assim como na pesquisa de [Leopoldino \(2004\)](#), a análise desse trabalho demonstra que fatores como volatilidade de requisitos e escopo mal definidos prejudicam muito o andamento do projeto. Esses problemas foram descritos como os mais prejudiciais no trabalho, segundo os entrevistados.

O cenário previsto confirma os resultados encontrados, indicando que o modelo atendeu ao propósito estabelecido, por demonstrar que projeto tinha um risco alto. Os resultados confirmam a expectativa inicial, de que o tema central do projeto era de

sconhecido pelo grupo. Isso implicou em atrasos na busca de soluções de vídeo e na gestão dos requisitos, os quais eram constantemente alterados e dificultavam as definições dos objetivos.

APLICAÇÃO DO MODELO - ANALISANDO VARIÁVEIS DOS RECURSOS HUMANOS: CENÁRIO PREVISTO

Os valores de entrada para o sistema Fuzzy ficaram assim definidos:

- Terceirização de Pessoal: até metade da equipe é terceirizada (0,30); terceirizados com boa experiência em projeto de software (0,45) e terceirizados seguem normas de qualidade reconhecidas pelo mercado (0,40). O valor de Saída foi 0,55 (risco médio);
- Para Equipe atuando em projetos paralelos os resultados de entrada foram: atuando em outros projetos sendo finalizados (0,63); equipe alocada em outros projetos e impactando no início de outros (0,46) e equipe alocada em outros projetos e com prazos já vencidos (0,91). Saída foi 0,7 (risco médio);
- Rotatividade de Pessoal: índice de Rotatividade (0,89); perfil com dificuldades de reposição (0,90) e políticas de retenção de pessoal (0,65). Saída foi 0,79 (risco alto).

O cenário previsto para esse projeto, analisando as variáveis dos recursos humanos, é mostrado nas três figuras a seguir:

Variável dos Recursos Humanos - Terceirização de Pessoal (cenário previsto)

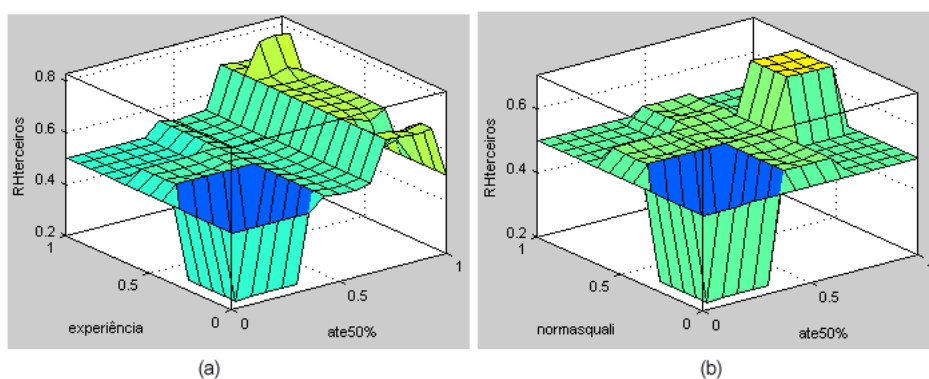


Figura 5.19: Gráfico de Cenário Previsto para Variável dos Recursos Humanos - Terceirização de Pessoal. Fonte: Autor.

As figuras 5.19 (a) e 5.19 (b) demonstram gráficos da variável Terceirização de Pessoal. Como parte da equipe foi terceirizada, houve uma maior preocupação com o andamento dos trabalhos e cumprimento dos prazos. Parte do pessoal terceirizado conhecia normas de qualidade e isso reduziu o risco do projeto.

Variável dos Recursos Humanos - Equipe em Projetos Paralelos (cenário previsto)

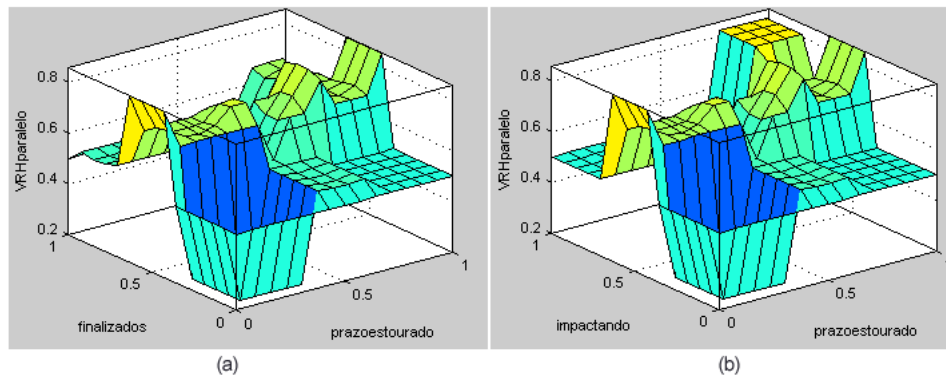


Figura 5.20: Gráfico de Cenário Previsto para Variável dos Recursos Humanos - Equipe em Projetos Paralelos. Fonte: Autor.

As figuras 5.20 (a) e 5.20 (b) tratam de Equipe em Projetos Paralelos, onde a maior dificuldade foi causada por preocupação quanto ao atendimento dos prazos, já que diversos projetos em execução foram atrasados e prejudicaram a alocação de pessoal. Os gráficos apresentam semelhanças, embora o atraso no início de outros trabalhos tenha forte influência no risco, como visto na figura 5.20 (b).

Ao analisar o cenário previsto, os resultados demonstram a expectativa da equipe com um risco considerado médio. Havia uma preocupação com o alto índice de rotatividade, devido ao aquecimento do mercado e ao surgimento de empresas de produção de software na Bahia. Esse fator será usado como base na criação dos gráficos do modelo. Parte do trabalho foi terceirizado, pois a equipe principal tinha muitos projetos em andamento. A empresa terceirizada tinha profissionais que conheciam o modelo de trabalho, pois adquiriram prévia experiência com o grupo e seguiam normas de qualidade.

Variável dos Recursos Humanos - Rotatividade de Pessoal (cenário previsto)

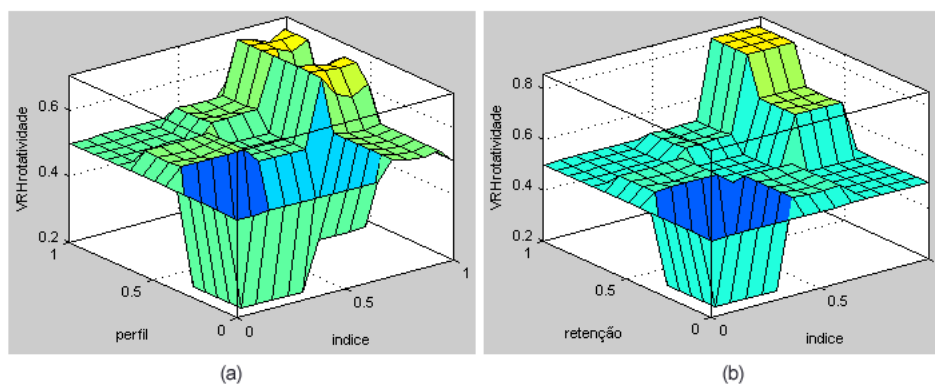


Figura 5.21: Gráfico de Cenário Previsto para Variável dos Recursos Humanos - Rotatividade da Equipe. Fonte: Autor.

As figuras 5.21 (a) e 5.21 (b) tratam de Rotatividade da Equipe e demonstram a grande influência no risco causado pelo índice de rotatividade presente naquele momento.

Apesar da preocupação do gestor com o perfil do pessoal com dificuldades de reposição e possíveis dificuldades com políticas de retenção, os dois fatores tiveram pouca influência no risco deste projeto.

APLICAÇÃO DO MODELO - ANALISANDO VARIÁVEIS DOS RECURSOS HUMANOS: CENÁRIO REALIZADO

- Terceirização de Pessoal: até metade da equipe é terceirizada (0,30); terceirizados com boa experiência em projeto de software (0,45) e terceirizados seguem normas de qualidade reconhecidas pelo mercado (0,40). Saída = 0,55 (risco médio);
- Equipe atuando em projetos paralelos: atuando em outros projetos sendo finalizados (0,63); equipe alocada em diversos projetos impossibilitado o início da execução (0,46) e equipe alocada em outros projetos com prazos finalizados (0,91). Saída de 0,7 (risco médio);
- Rotatividade de Pessoal: índice de rotatividade (0,89); perfil com dificuldades de reposição (0,90) e políticas de retenção de pessoal (0,65). Saída foi 0,87 (risco alto).

O cenário realizado para esse projeto, analisando as variáveis dos recursos humanos, é mostrado nas três figuras a seguir:

Variável dos Recursos Humanos - Terceirização de Pessoal (cenário realizado)

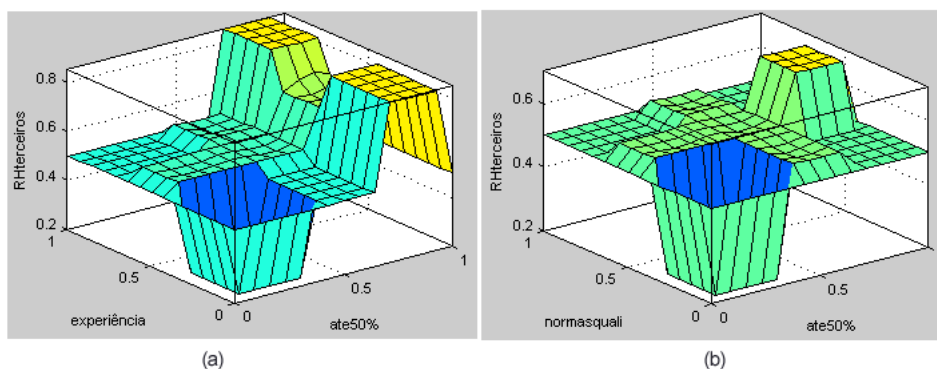


Figura 5.22: Gráfico de Cenário Realizado para Variável dos Recursos Humanos - Terceirização de Pessoal. Fonte: Autor.

As figuras 5.22 (a) e 5.22 (b) tratam da Terceirização de Pessoal. Nos dois gráficos a influência no risco causado pela utilização de pessoal terceirizado ficam evidentes. O conhecimento de normas de qualidade pelo grupo terceirizado evidencia uma diminuição do risco para a variável em análise, conforme visto na Figura 5.22 (a).

Variável dos Recursos Humanos - Equipe em Projetos Paralelos (cenário realizado)

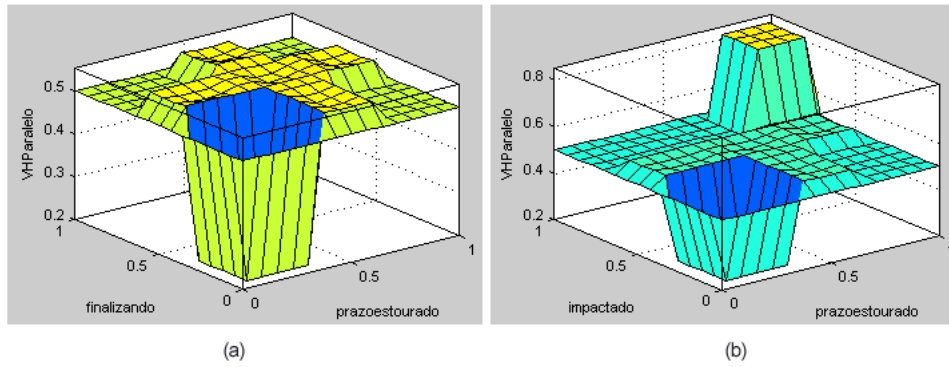


Figura 5.23: Gráfico de Cenário Realizado para Variável dos Recursos Humanos - Equipe em Projetos Paralelos. Fonte: Autor.

Nas figuras 5.23 (a) e 5.23 (b), Equipe em Projetos Paralelos, o vencimento de prazo do projeto reflete na alocação de pessoal.

Variável dos Recursos Humanos - Rotatividade de Pessoal (cenário realizado)

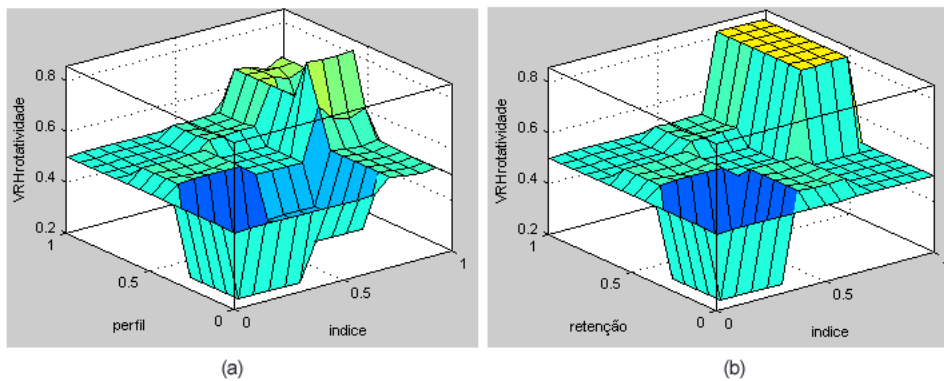


Figura 5.24: Gráfico de Cenário Realizado para Variável dos Recursos Humanos - Rotatividade da Equipe. Fonte: Autor.

As figuras 5.24 (a) e 5.24 (b) mostram dados da Rotatividade da Equipe. Observa-se nas duas imagens que o alto índice de rotatividade de pessoal influenciou o grau de risco do projeto, assim como o cenário previsto. A dificuldade de encontrar pessoal com o perfil necessário aumentou o risco, conforme visto na figura 5.24 (a).

Os resultados do cenário previsto e realizado foram aproximados tanto para variáveis de sistemas quanto para as dos recursos humanos. Algumas expectativas se confirmaram, o que resultou em dificuldades na execução do projeto, tais como saída de pessoal e problemas com o gerenciamento de etapas terceirizadas.

5.5.2 Projeto 2 - Sistema Gestor de Informação

O Sistema Gestor de Informação foi desenvolvido para uma empresa privada atuante no setor de Petroléo e Gás. O sistema deveria funcionar em rede local com sistema Windows, atendendo a aproximadamente trezentos colaboradores. Esse sistema teve como propósito principal gerenciar informações de projetos, pessoas e dados de trabalhos realizados pela empresa. O projeto teve baixa complexidade e a empresa contratante já utilizava um sistema parecido, com requisitos bem próximos, porém menos estáveis. O prazo real de produção foi de sete meses, sendo previsto inicialmente 6 meses.

Somente um profissional foi terceirizado, um ilustrador. Foram implantadas melhorias no processo de retenção de pessoal, o que refletiu em menor saída de profissionais na área. Apenas uma pequena parte da equipe estava alocada em outros projetos. Problemas ocorridos no decorrer do projeto, principalmente em adequação a algumas normas de qualidade do cliente, evidenciaram dificuldades no entendimento do problema.

CENÁRIO IDEAL PARA O PROJETO

Nas reuniões iniciais para análise deste projeto, estudos preliminares indicavam que não haveria dificuldades na execução, principalmente pelo fato de a empresa contratante já ter um sistema parecido, o qual foi estudado previamente pelo grupo. O estudo prévio do antigo sistema permitiu identificar pontos fortes e fracos com propostas de melhoria e isso seria apresentado para aprovação do cliente. O cenário ideal demandava adequação às normas de qualidade exigidas pela contratante e melhorias na performance do novo produto, pois havia diversas reclamações dos usuários quanto ao sistema usado anteriormente. Uma dificuldade frequente nos projetos de software que deveria receber mais atenção da equipe de desenvolvimento, foi identificada neste projeto: trata-se da falta de apoio do cliente na gestão do escopo e dos requisitos. Este problema também foi diagnosticado no trabalho de [Leopoldino \(2004\)](#).

APLICAÇÃO DO MODELO - ANALISANDO VARIÁVEIS DE SISTEMA: CENÁRIO PREVISTO

- Escopo Mal Definido - pouco entendimento do problema (0,30); usuários envolvidos não são os indicados (0,26) e equipe envolvida é inexperiente na gestão do escopo (0,20). Saída de 0,2 (risco baixo);
- Complexidade do Software - uso de novas tecnologias (0,21); dificuldades técnicas em implementar a solução (0,35) e tema é novo ou não é familiar para a equipe (0,17). Saída de 0,21 (risco baixo);
- Requisitos Voláteis: mudanças constantes na definição dos requisitos (0,51); equipe

é inexperiente em engenharia de requisitos (0,45) e há preferências de usuários ou a políticas governamentais (0,81). Para esses valores de entrada, a combinação dos riscos das três variáveis gerou um grau do risco de 0,54 (risco médio).

O cenário previsto para esse projeto, analisando as variáveis do sistema, é mostrado nas três figuras a seguir:

Variável do Sistema - Escopo ou Objetivos Mal Definidos (cenário previsto)

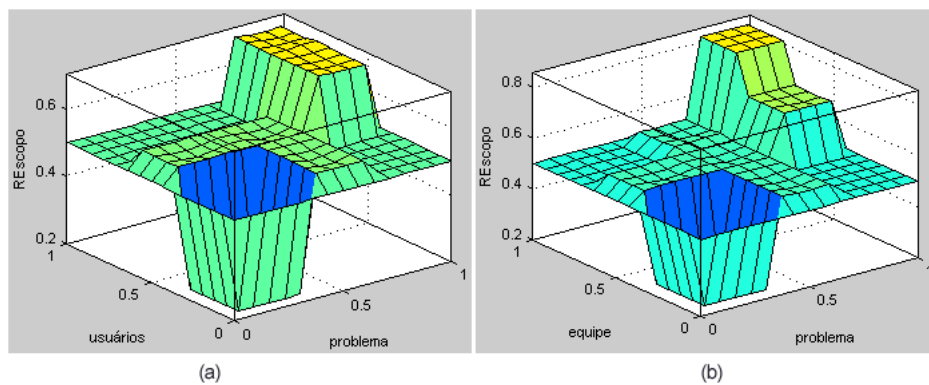


Figura 5.25: Gráfico de Cenário Previsto para Variável do Sistema Escopo ou Objetivos Mal Definidos. Fonte: Autor.

As figura 5.25 (a) e 5.25 (b) mostram os gráficos para os temas de mal definição do escopo e indicam um aumento no risco quando parte da equipe não domina o problema principal do projeto.

Variável do Sistema - Complexidade do Software (cenário previsto)

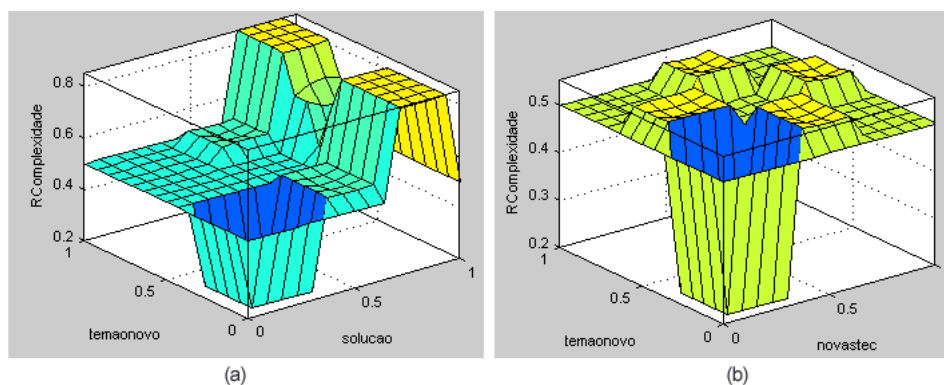


Figura 5.26: Gráfico de Cenário Previsto para Variável do Complexidade do Software. Fonte: Autor.

Nas figuras 5.26 (a) e 5.26 (b) os gráficos tratam de Complexidade do Software e é possível identificar influência nos valores de risco quando do desenvolvimento da nova solução, figura 5.26 (a), e também no uso de novas tecnologias, conforme figura 5.26 (b).

Variável do Sistema - Requisitos Voláteis (cenário previsto)

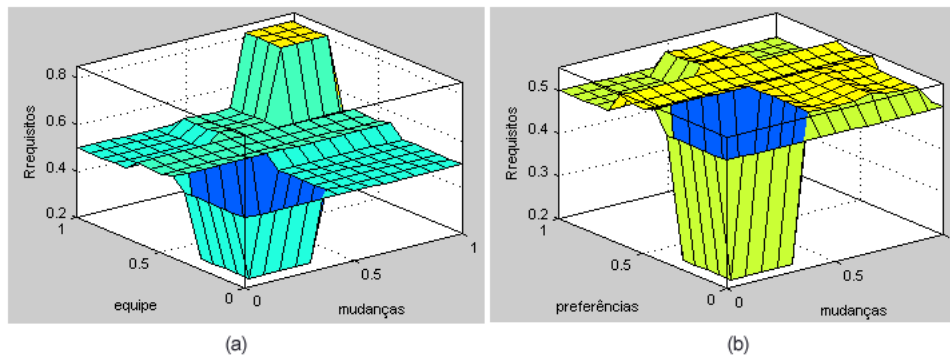


Figura 5.27: Gráfico de Cenário Previsto para Variável Requisitos Voláteis. Fonte: Autor.

As figuras 5.27 (a) e 5.27 (b) mostram o risco para o tema Requisitos Voláteis no projeto. De acordo com as informações iniciais, imaginou-se que as mudanças frequentes nos requisitos poderiam gerar dificuldades na execução do trabalho.

APLICAÇÃO DO MODELO - ANALISANDO VARIÁVEIS DE SISTEMA: CENÁRIO REALIZADO

- Escopo Mal Definido - pouco entendimento do problema (0,36); usuários envolvidos não são os indicados (0,41) e equipe envolvida é inexperiente na gestão do escopo (0,29). Saída de 0,24 (risco baixo);
- Complexidade do Software: uso de novas tecnologias (0,33), dificuldades técnicas em implementar a solução (0,45) e assunto é novo ou não é familiar para a equipe (0,23). Saída de 0,35 (risco baixo);
- Requisitos Voláteis: mudanças constantes na definição dos requisitos (0,71); equipe é inexperiente em engenharia de requisitos (0,33) e há preferências de usuários ou a políticas governamentais (0,60). O resultado de Saída para Requisitos Voláteis foi de 0,71, risco alto.

O cenário realizado para esse projeto, analisando as variáveis do sistema, é mostrado nas três figuras a seguir:

Variável do Sistema - Escopo ou Objetivos Mal Definidos (cenário realizado)

Nos gráficos das figuras 5.28 (a) e 5.28 (b), a variável linguística Escopo ou Objetivo mal definidos teve risco influenciado por problemas em sua definição e no alinhamento do contratante na escolha dos objetivos.

Variável do Sistema - Complexidade do Software (cenário realizado)

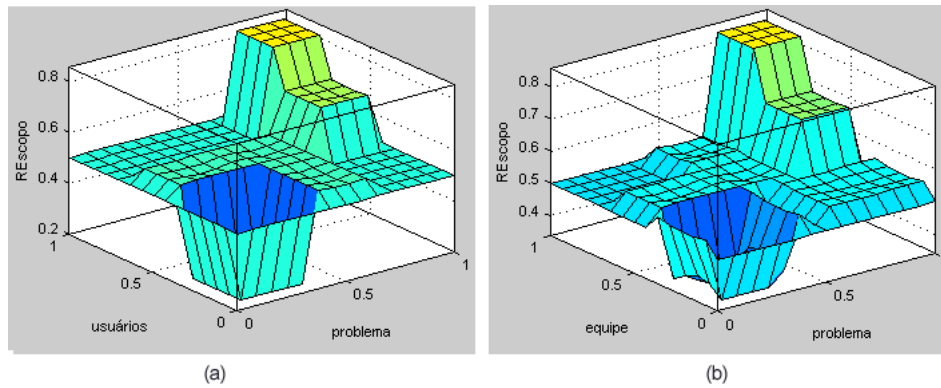


Figura 5.28: Gráfico de Cenário Realizado para Variável Escopo ou Objetivos Mal Definidos. Fonte: Autor.

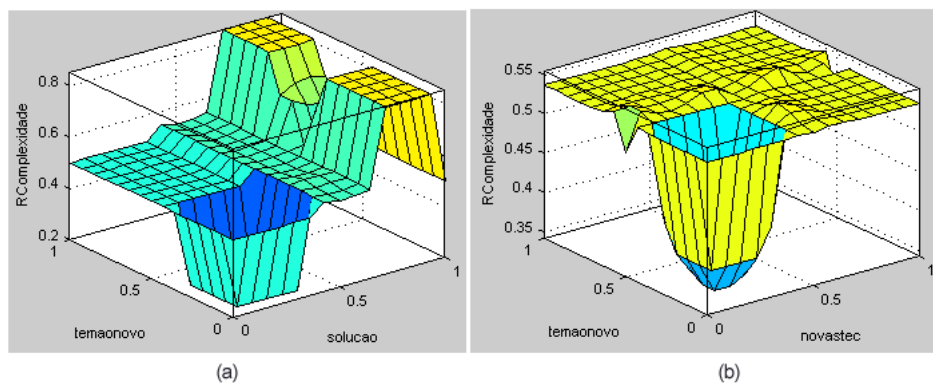


Figura 5.29: Gráfico de Cenário Previsto para Variável do Complexidade do Software. Fonte: Autor.

As figuras 5.29 (a) e 5.29 (b) demonstram gráficos da variável linguística Complexidade do Software. Assim como no cenário previsto, a dificuldade em implementar solução aumentou o grau de risco. Na figura 5.29 (a), o risco foi causado pela dificuldade de uso de novas tecnologias.

Variável do Sistema - Requisitos Voláteis (cenário realizado)

A figuras 5.30 (a) e 5.30 (b) identificam os problemas causados pelas mudanças constantes nos requisitos, o que aumentou significativamente o risco, quando comparado com o cenário previsto. Na figura 5.30 (a), observa-se elevação do risco devido a questões relacionadas a políticas governamentais, problema que fora identificado pelo grupo.

Os cenários realizados e previstos tiveram valores de entrada bem próximos. No tema requisitos voláteis, o cenário realizado demonstrou dificuldades na gestão dos requisitos do projeto com mudanças frequentes. Parte do problema decorreu da mudança de gestão na equipe do cliente

APLICAÇÃO DO MODELO - ANALISANDO VARIÁVEIS DOS RECURSOS HU-

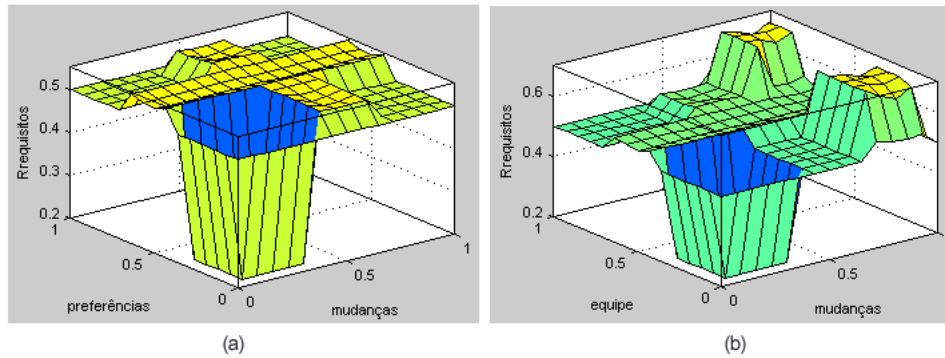


Figura 5.30: Gráfico de Cenário Previsto para Variável Requisitos Voláteis. Fonte: Autor.

MANOS: CENÁRIO PREVISTO

Os valores de entrada para o cenário previsto foram:

- Terceirização de Pessoal: até metade da equipe é terceirizada (0,11); terceirizados com boa experiência em projeto de software (0,59) e terceirizados seguem normas de qualidade reconhecidas pelo mercado (0,45). Saída de 0,54 (risco médio);
- Equipe atua em projetos paralelos: atua em outros projetos em fase de finalização (0,42); equipe alocada em diversos projetos impossibilitado o início da execução (0,21) e equipe alocada em outros projetos e com prazos finalizados (0,31). Resultado de Saída foi 0,34 (risco baixo);
- Rotatividade de Pessoal: índice de Rotatividade (0,31); perfil com dificuldades de reposição (0,44) e políticas de retenção de pessoal (0,22). Saída de 0,2 (risco baixo).

O cenário previsto para esse projeto, analisando as variáveis dos recursos humanos, é mostrado nas três figuras a seguir:

Variável dos Recursos Humanos - Terceirização de Pessoal (cenário previsto)

As figuras 5.31 (a) e 5.31 (b) demonstram gráficos da variável Terceirização de Pessoal, usando como base para comparação o tema terceirizados com boa experiência em projeto de software. O risco presente nas duas imagens indica a preocupação do grupo com o profissional ilustrador terceirizado. Como o projeto demandava grande quantidade de imagens, esse fato gerou preocupação, pois foi o primeiro trabalho desse profissional para a empresa.

Variável dos Recursos Humanos - Equipe em Projetos Paralelos (cenário previsto)

As figuras 5.32 (a) e 5.32 (b) tratam de Equipe em Projetos Paralelos, onde

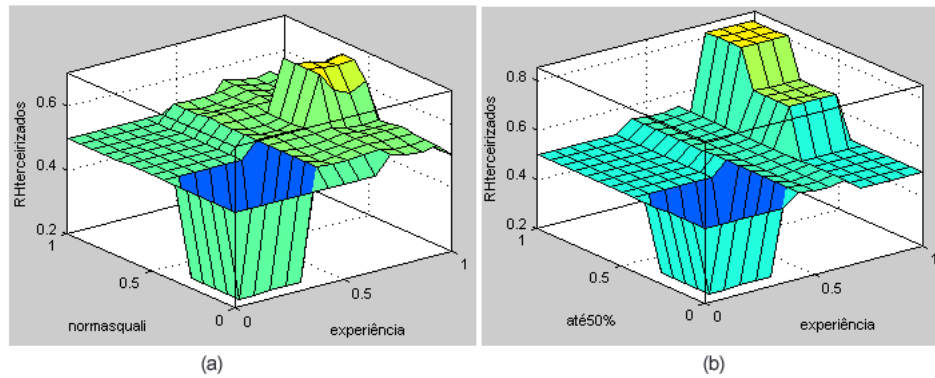


Figura 5.31: Gráfico de Cenário Previsto para Variável dos Recursos Humanos - Terceirização de Pessoal. Fonte: Autor.

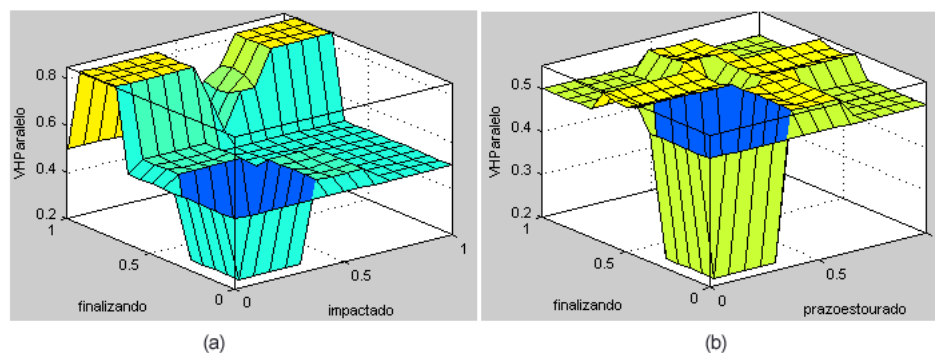


Figura 5.32: Gráfico de Cenário Previsto para Variável dos Recursos Humanos - Equipe em Projetos Paralelos. Fonte: Autor.

o incremento do risco foi causado por preocupação quanto à atuação de profissionais da equipe em outros projetos que ainda estavam sendo finalizados.

Variável dos Recursos Humanos - Rotatividade de Pessoal (cenário previsto)

Nas figuras 5.33 (a) e 5.33 (b), envolvendo a Rotatividade da Equipe, observa-se um aumento do risco devido a preocupação com perfil dos profissionais de saída com dificuldades de reposição.

APLICAÇÃO DO MODELO - ANALISANDO VARIÁVEIS DOS RECURSOS HUMANOS: CENÁRIO REALIZADO

- Terceirização de Pessoal: até metade da equipe é terceirizada (0,17); terceirizados com boa experiência em projeto de software (0,58) e terceirizados seguem normas de qualidade reconhecidas pelo mercado (0,38). Saída de 0,61 (risco médio);
- Equipe atuando em projetos paralelos: atuando em outros projetos e sendo finalizados (0,46), equipe alocada em outros projetos e impactando no início de outros (0,36), equipe alocada em outros projetos e com prazos já estourados em mais de

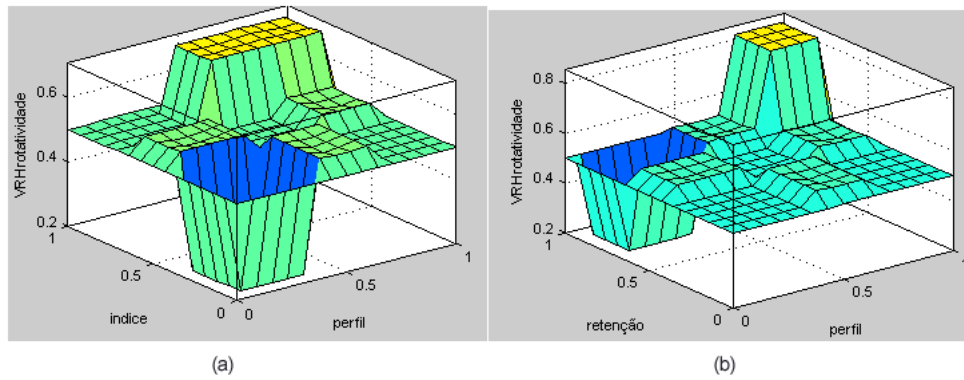


Figura 5.33: Gráfico de Cenário Previsto para Variável dos Recursos Humanos - Rotatividade da Equipe. Fonte: Autor.

um projeto (0,23). Saída de 0,42 (risco baixo);

- Rotatividade de Pessoal: índice de Rotatividade (0,39); perfil com dificuldades de reposição (0,34), políticas de retenção de pessoal (0,47). Saída de 0,2 (risco baixo).

O cenário realizado para esse projeto, analisando as variáveis dos recursos humanos, é mostrado nas três figuras a seguir:

Variável dos Recursos Humanos - Terceirização de Pessoal (cenário realizado)

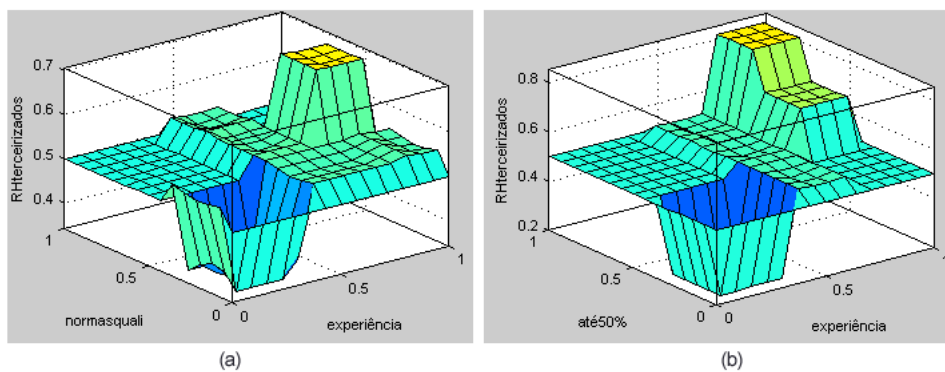


Figura 5.34: Gráfico de Cenário Realizado para Variável dos Recursos Humanos - Terceirização de Pessoal. Fonte: Autor.

As figuras 5.34 (a) e 5.34 (b), mostram os gráficos para Terceirização de Pessoal. Assim como os gráficos gerados no cenário previsto, o realizado demonstra dificuldades na gestão do profissional terceirizado. O projeto teve parte do prazo comprometido devido aos atrasos na entrega do material pelo profissional contratado.

Variável dos Recursos Humanos - Equipe em Projetos Paralelos (cenário realizado)

Nas figuras 5.35 (a) e 5.35 (b), Equipe em Projetos Paralelos, o cenário realizado indicou também as dificuldades de alocação do pessoal, pois parte do grupo,

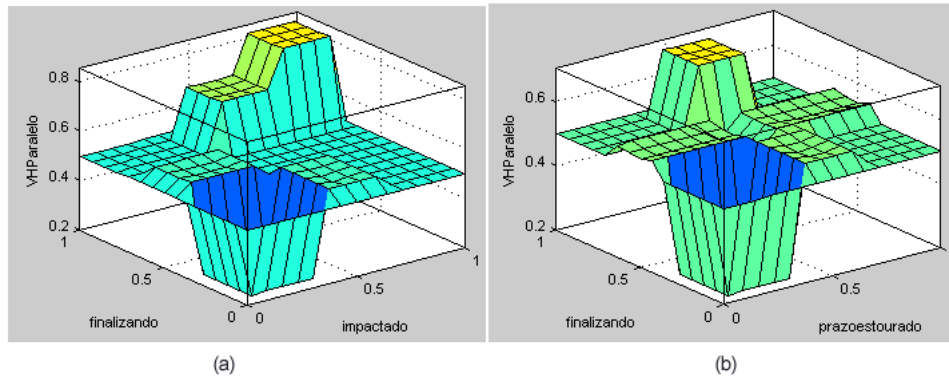


Figura 5.35: Gráfico de Cenário Realizado para Variável dos Recursos Humanos - Equipe em Projetos Paralelos. Fonte: Autor.

inicialmente destacado para atuar no projeto em análise, ainda encontrou dificuldades para finalizar projetos paralelos.

Variável dos Recursos Humanos - Rotatividade de Pessoal (cenário realizado)

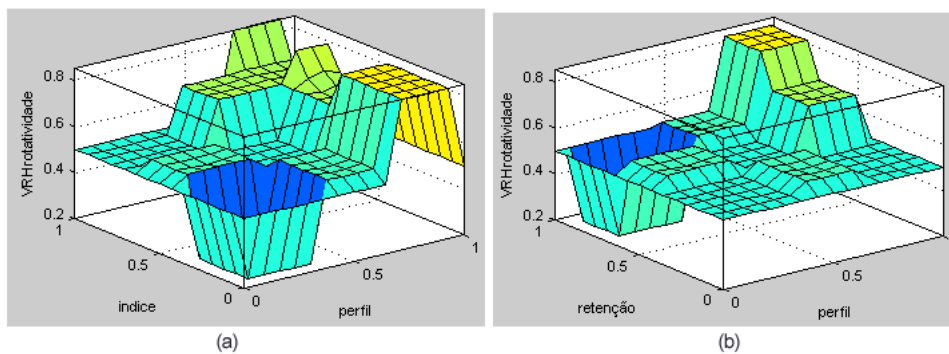


Figura 5.36: Gráfico de Cenário Realizado para Variável dos Recursos Humanos - Rotatividade da Equipe. Fonte: Autor.

As figuras 5.36 (a) e 5.36 (b) tratam de Rotatividade da Equipe e indicam que o índice de rotatividade baixo possibilitou uma redução do risco, conforme visto na figura 5.36 (a). O gráfico da figura 5.36 (b), no entanto, demonstra que o risco teve um aumento considerável ao tratar da retenção de pessoal, mesmo com melhorias realizadas na empresa para diminuir a saída de profissionais.

Os resultados do cenário previsto e do realizado tiveram graus de riscos que podem confirmar a previsão inicial de possíveis problemas para o projeto. Foi possível constatar que no item retenção de pessoal, apesar das políticas implementadas para tanto, o efeito desejado não se concretizou a curto prazo, pois membros da equipe de desenvolvimento deixaram a empresa no decorrer dos trabalhos.

5.5.3 Projeto 3 - Sistema de Capacitação em Educação Ambiental

O objetivo desse projeto era capacitar os colaboradores da empresa na área de Educação Ambiental e conscientizá-los sobre o uso adequado dos recursos. O projeto permitiu substituir aulas e vídeos realizados periodicamente, de forma presencial, por uma aplicação via Internet, com animações, simulação, vídeos e prova online.

Parte da equipe foi terceirizada nas etapas de vídeo, animação e locução, mas desconheciam normas de qualidade aplicadas ao software. A equipe tinha poucos projetos simultâneos e isso possibilitou incluir outros profissionais para agilizar o trabalho. A rotatividade havia diminuído, principalmente após ajustes executados pela direção nas políticas de recursos humanos. O prazo previsto para finalização do sistema foi de quatro meses, mas o trabalho foi entregue antes do prazo.

CENÁRIO IDEAL PARA O PROJETO

Com problemas de retenção de pessoal e índice de rotatividade ainda elevado, o cenário ideal para este projeto deveria concentrar esforços na gestão dos recursos humanos. Havia uma preocupação do grupo quanto ao desenvolvimento de um sistema de transmissão com streaming de áudio e vídeo, indicando a necessidade de aquisição de algo já testado e em uso pelo mercado. Apesar de sistema não ser algo totalmente novo para o grupo, seria importante que uma equipe fosse alocada exclusivamente para a criação de um sistema de media center, permitindo armazenar mídias e transmitir os vídeos do projeto. Não obstante o curto prazo, estudos iniciais indicavam que os terceirizados do projeto deveriam ficar alocados, pelo menos um dia da semana, junto à equipe principal.

APLICAÇÃO DO MODELO - ANALISANDO VARIÁVEIS DE SISTEMA: CENÁRIO PREVISTO

As variáveis linguísticas de entrada tiveram os valores a seguir:

- Valores para Escopo: pouco entendimento do problema (0,76); usuários envolvidos não são os indicados (0,54) e equipe envolvida é inexperiente na gestão do escopo (0,10). Saída de 0,85 (risco alto);
- Complexidade do Software: uso de novas tecnologias (0,80); dificuldades técnicas em implementar a solução (0,42) e tema é novo ou não é familiar para a equipe (0,46). Saída de 0,55 (risco médio);
- Requisitos Voláteis: mudanças constantes na definição dos requisitos (0,26); equipe é inexperiente em engenharia de requisitos (0,15) e há preferências de usuários ou a políticas governamentais (0,30). Saída de 0,21 (risco baixo).

O cenário previsto para esse projeto, analisando as variáveis do sistema, é mostrado nas três figuras a seguir:

Variável do Sistema - Escopo ou Objetivos Mal Definidos (cenário previsto)

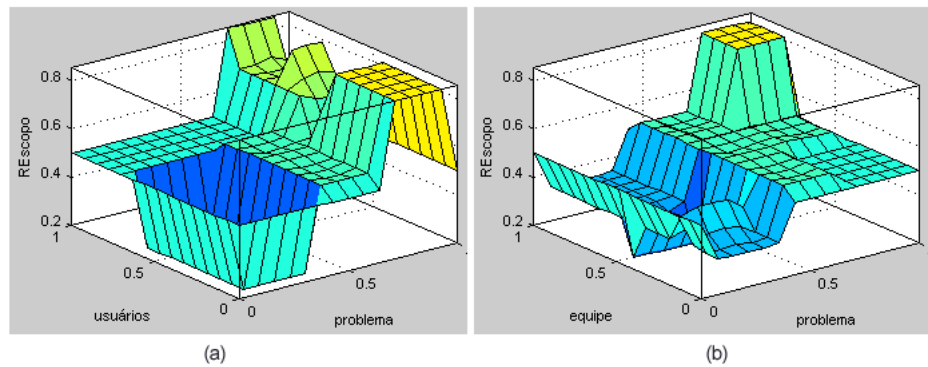


Figura 5.37: Gráfico de Cenário Previsto para Variável do Sistema Escopo ou Objetivos Mal Definidos. Fonte: Autor.

Nas figuras 5.37 (a) e 5.37 (b) os gráficos demonstram mal definição do escopo; identificam que o pouco entendimento do problema teve um impacto maior no grau de risco do projeto. Na figura 5.37 (b) pode-se observar que a equipe envolvida é inexperiente, mas isso teve pouca influência cenário previsto, quando comparado com o gráfico gerado na figura 5.37 (a).

Variável do Sistema - Complexidade do Software (cenário previsto)

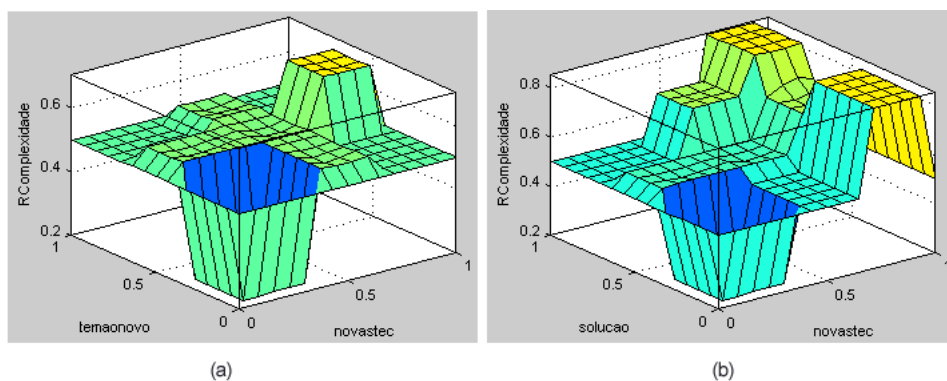


Figura 5.38: Gráfico de Cenário Previsto para Variável do Complexidade do Software. Fonte: Autor.

Nas figuras 5.38 (a) e 5.38 (b), avaliando a variável linguística Complexidade do Software, foi possível identificar que o uso de novas tecnologias no projeto causou preocupação à equipe e gerou grande impacto nos dois gráficos.

Variável do Sistema - Requisitos Voláteis (cenário previsto)

As figuras 5.39 (a) e 5.39 (b) mostram o risco para a avaliação dos Requisitos

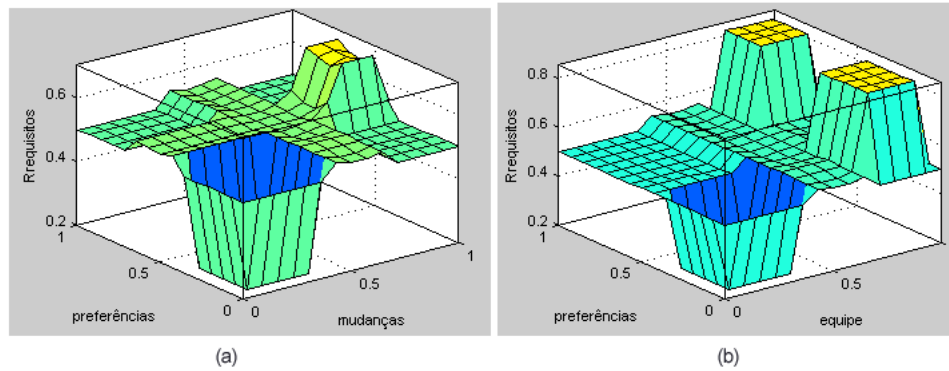


Figura 5.39: Gráfico de Cenário Previsto para Variável Requisitos Voláteis. Fonte: Autor.

Voláteis do projeto. A avaliação inicial da equipe identificou possíveis dificuldades nas preferências de usuários no momento de identificação dos requisitos. A inexperiência da equipe influenciou fortemente o risco, conforme mostrado na figura 5.39 (b).

APLICAÇÃO DO MODELO - ANALISANDO VARIÁVEIS DE SISTEMA: CENÁRIO REALIZADO

Os valores de entrada usados para a criação do cenário realizado foram:

- Valores para Escopo: pouco entendimento do problema (0,83); usuários envolvidos não são os indicados (0,77) e equipe envolvida é inexperiente na gestão do escopo (0,21). Saída de 0,87 (risco alto);
- Complexidade do Software: uso de novas tecnologias (0,77); dificuldades técnicas em implementar a solução (0,59) e assunto é novo ou não é familiar para a equipe (0,80). Saída de 0,68 (risco médio);
- Requisitos Voláteis: mudanças constantes na definição dos requisitos (0,89); equipe é inexperiente em engenharia de requisitos (0,44) e há preferências de usuários ou a políticas governamentais (0,77). Saída de 0,53 (risco médio).

O cenário realizado para esse projeto, analisando as variáveis do sistema, é mostrado nas três figuras a seguir:

Variável do Sistema - Escopo ou Objetivos Mal Definidos (cenário realizado)

Nos gráficos das figuras 5.40 (a) e 5.40 (b) a variável linguística analisada teve risco influenciado por problemas na definição do escopo.

Variável do Sistema - Complexidade do Software (cenário realizado)

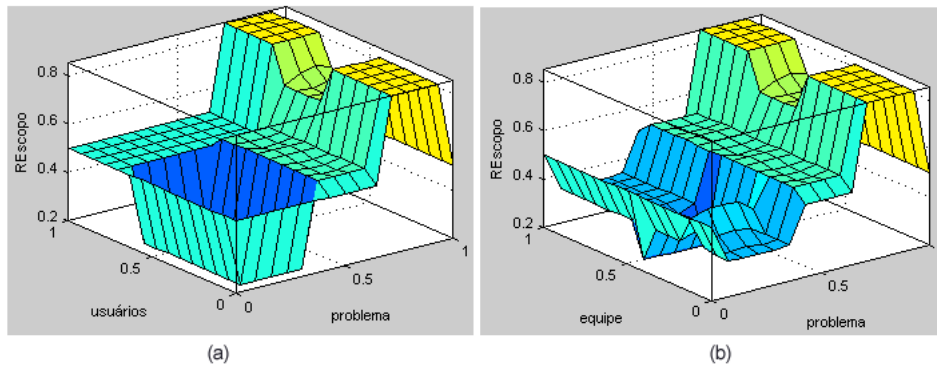


Figura 5.40: Gráfico de Cenário Realizado para Variável Escopo ou Objetivos Mal Definidos. Fonte: Autor.

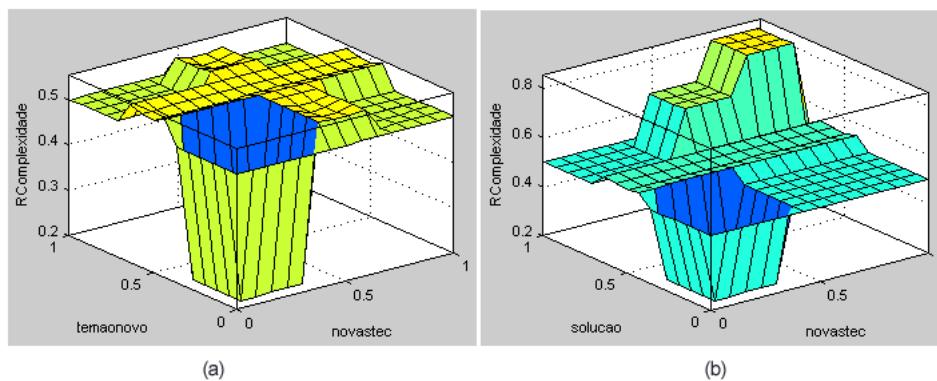


Figura 5.41: Gráfico de Cenário Previsto para Variável do Complexidade do Software. Fonte: Autor.

As figuras 5.41 (a) e 5.41 (b) tratam de Complexidade do Software, onde verifica-se que o aumento do risco é causado por dificuldades de entendimento do problema, o que foi sinalizado no cenário previsto e, conforme citado na pesquisa de [Goncalves \(2006\)](#), é um dos fatores principais para o aumento de risco em um projeto de software.

Variável do Sistema - Requisitos Voláteis (cenário realizado)

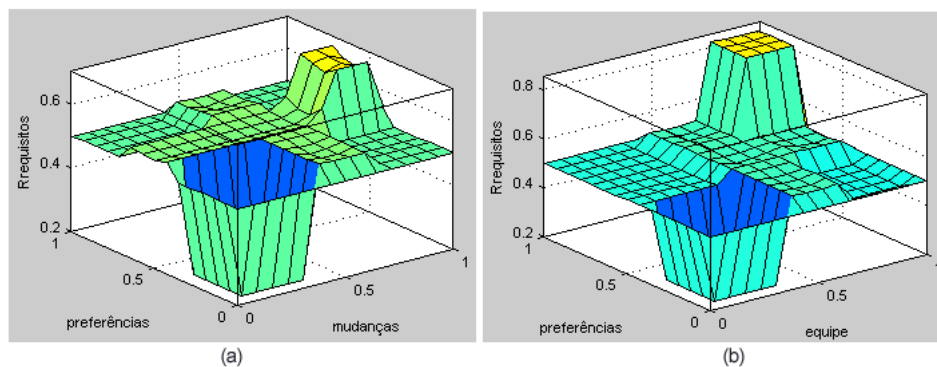


Figura 5.42: Gráfico de Cenário Previsto para Variável Requisitos Voláteis. Fonte: Autor.

A figuras 5.42 (a) e 5.42 (b) identificam os problemas de Requisitos Voláteis. Nota-se que nos dois gráficos houve um risco maior causado por preferências dos usuários

na avaliação dos requisitos, porém no cenário previsto foi possível verificar que o risco aumentou em função da inexperiência da equipe na avaliação dos requisitos. Esses diferentes valores de entrada geraram cenários com riscos diferentes.

Os gráficos gerados nos cenários realizado e previsto tiveram valores não muito distantes. A maior diferença foi notada nos valores do cenário realizado, quando avalia-se o resultado de requisitos voláteis, onde o cenário previsto teve como resultado um risco baixo, valor de 0,21, enquanto no cenário realizado o risco foi alto, valor de 0,53. Quando compara-se os gráficos dos cenários previsto e realizado, avaliando os temas escopo ou objetivo mal definidos e complexidade do software, constata-se que os gráficos e valores resultantes do risco foram mais aproximados.

APLICAÇÃO DO MODELO - ANALISANDO VARIÁVEIS DOS RECURSOS HUMANOS: CENÁRIO PREVISTO

- Terceirização de Pessoal: até metade da equipe é terceirizada (0,56); terceirizados com boa experiência em projeto de software (0,46) e terceirizados seguem normas de qualidade reconhecidas pelo mercado (0,77). Saída de 0,54 (risco médio) ;
- Equipe atuando em projetos paralelos: atuando em outros projetos sendo finalizados (0,11); equipe alocada em diversos projetos impossibilitado o início da execução (0,23) e equipe alocada em outros projetos com prazos finalizados (0,16). Saída de 0,36 (risco baixo);
- Rotatividade de Pessoal: índice de rotatividade (0,21); perfil com dificuldades de reposição (0,34) e políticas de retenção de pessoal (0,15). Saída de 0,36 (risco baixo).

O cenário previsto para esse projeto, analisando as variáveis dos recursos humanos, é mostrado nas três figuras a seguir:

Variável dos Recursos Humanos - Terceirização de Pessoal (cenário previsto)

As figuras 5.43 (a) e 5.43 (b) demonstram gráficos da variável Terceirização de Pessoal. O risco presente nas duas imagens indica a preocupação do grupo com utilização de normas de qualidade pelo grupo terceirizado. A empresa segue o padrão ISO9000 e os trabalhos terceirizados devem atender a esta norma. A experiência dos terceiros em projeto de software teve repercussão sobre o risco, conforme visto na figura 5.43 (a).

Variável dos Recursos Humanos - Equipe em Projetos Paralelos (cenário previsto)

As figuras 5.44 (a) e 5.44 (b) tratam de Equipe em Projetos Paralelos e observa-se que havia diversos profissionais alocados em outros trabalhos ainda em desen-

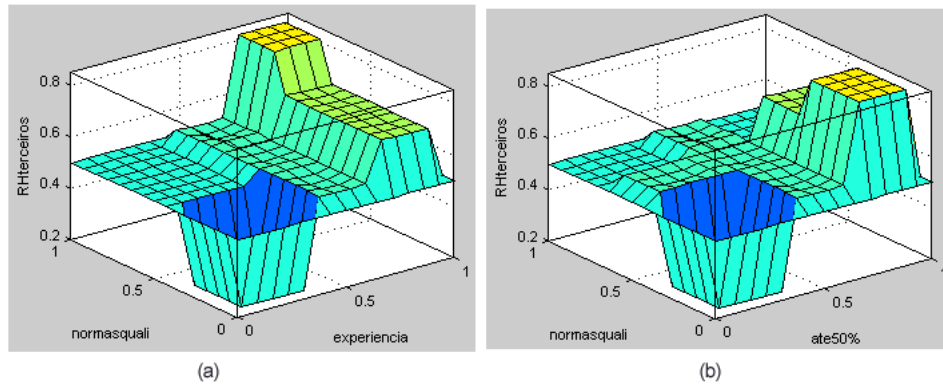


Figura 5.43: Gráfico de Cenário Previsto para Variável dos Recursos Humanos - Terceirização de Pessoal. Fonte: Autor.

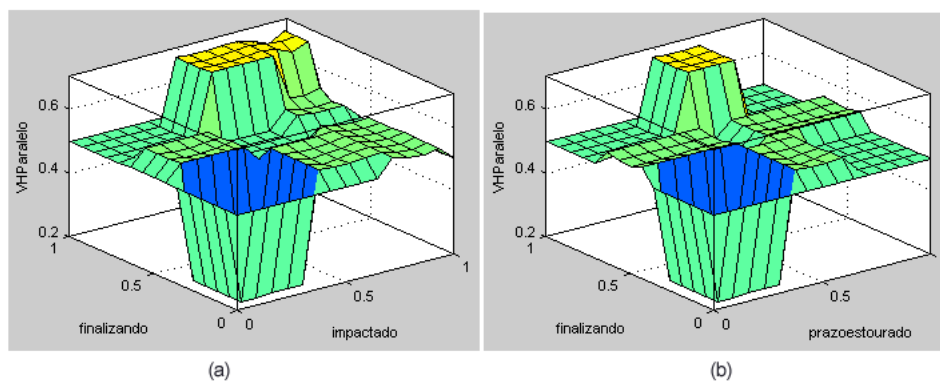


Figura 5.44: Gráfico de Cenário Previsto para Variável dos Recursos Humanos - Equipe em Projetos Paralelos. Fonte: Autor.

volvimento e com dificuldades de finalização. Isso repercutiu no risco, conforme indicado nos dois gráficos.

Variável dos Recursos Humanos - Rotatividade de Pessoal (cenário previsto)

Nas figuras 5.45 (a) e 5.45 (b), envolvendo a Rotatividade da Equipe, observa-se um aumento do risco devido à preocupação com a saída de profissionais da empresa, havia variações no índice de rotatividade. Na figura 5.44 (a) observa-se uma leve diminuição da taxa de risco quando se analisa as políticas de retenção de pessoal.

APLICAÇÃO DO MODELO - ANALISANDO VARIÁVEIS DOS RECURSOS HUMANOS: CENÁRIO REALIZADO

Os valores de entrada para o sistema Fuzzy ficaram assim definidos para o cenário realizado:

- Terceirização de Pessoal: até metade da equipe é terceirizada (0,89); terceirizados com experiência em projeto de software (0,56) e terceirizados seguem normas de

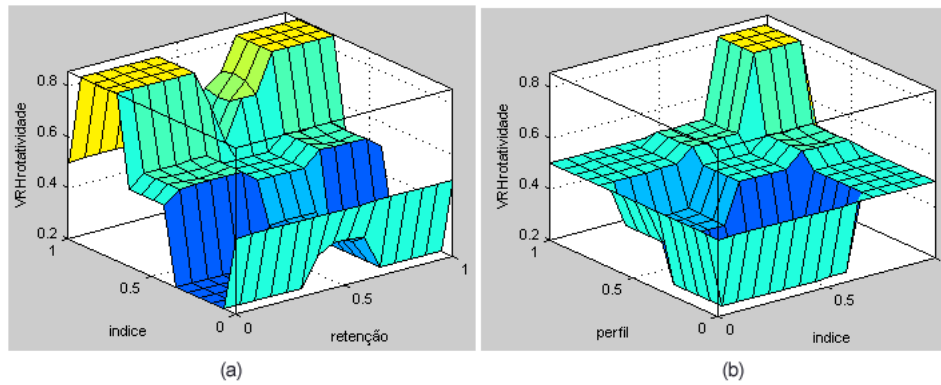


Figura 5.45: Gráfico de Cenário Previsto para Variável dos Recursos Humanos - Rotatividade da Equipe. Fonte: Autor.

qualidade reconhecidas pelo mercado (0,92). Saída de 0,63 (risco médio);

- Equipe atue em projetos paralelos: atua em outros projetos em finalização (0,56); equipe alocada em diversos projetos impossibilitando o início da execução (0,30) e equipe alocada em outros projetos com prazos finalizados (0,33). Saída de 0,32 (risco baixo);
- Rotatividade de Pessoal: índice de rotatividade (0,18); perfil com dificuldades de reposição (0,56) e políticas de retenção de pessoal (0,25). Saída de 0,53(risco médio).

O cenário realizado para esse projeto, analisando as variáveis dos recursos humanos, é mostrado nas três figuras a seguir:

Variável dos Recursos Humanos - Terceirização de Pessoal (cenário realizado)

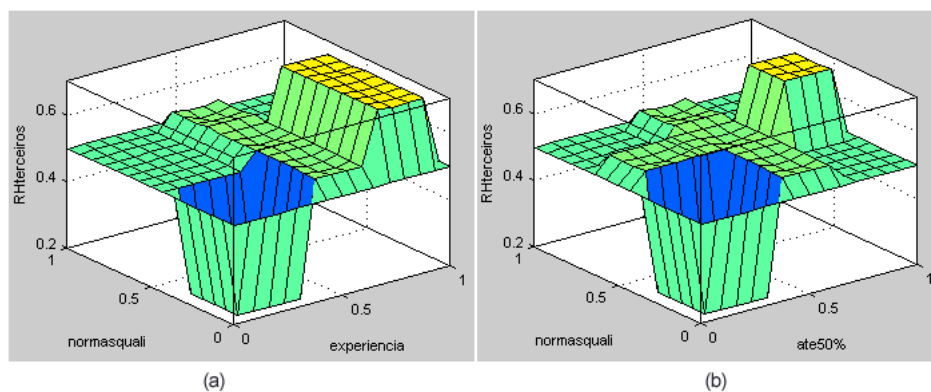


Figura 5.46: Gráfico de Cenário Realizado para Variável dos Recursos Humanos - Terceirização de Pessoal. Fonte: Autor.

As figuras 5.46 (a) e 5.46 (b) mostram os gráficos para Terceirização de Pessoal. Os resultados dos gráficos gerados no cenário previsto e realizado foram próximos para essa variável, inclusive com áreas dos gráficos com imagens bem semelhantes.

Variável dos Recursos Humanos - Equipe em Projetos Paralelos (cenário realizado)

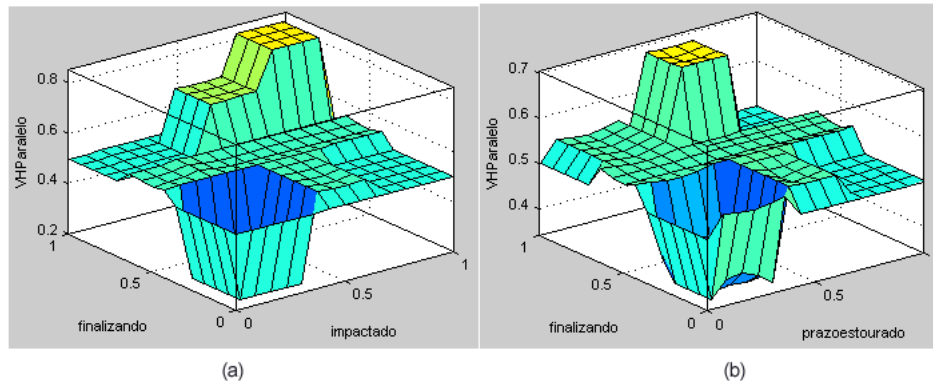


Figura 5.47: Gráfico de Cenário Realizado para Variável dos Recursos Humanos - Equipe em Projetos Paralelos. Fonte: Autor.

Nas figuras 5.47 (a) e 5.47 (b), Equipe em Projetos Paralelos, o cenário realizado e previsto teve valores próximos, o que indica acerto da equipe na construção das previsões iniciais de dificuldades para o projeto. Conforme visto na figura 5.44 (a), o tema profissionais atuando em outros projetos em finalização, indica um risco maior que o gerado no cenário previsto.

Variável dos Recursos Humanos - Rotatividade de Pessoal (cenário realizado)

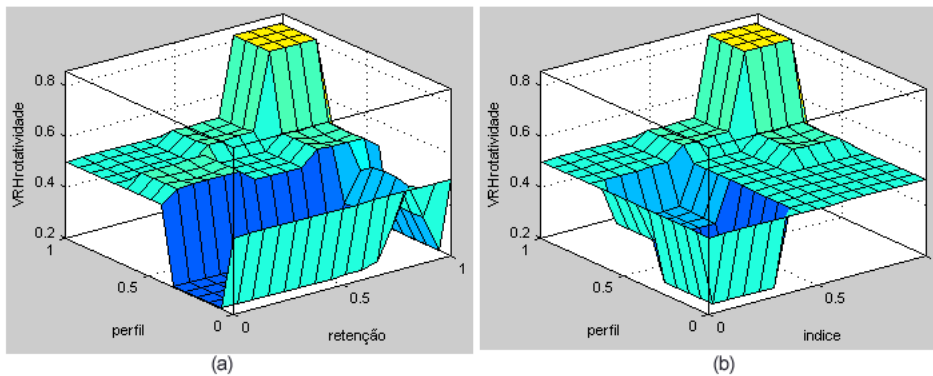


Figura 5.48: Gráfico de Cenário Realizado para Variável dos Recursos Humanos - Rotatividade da Equipe. Fonte: Autor.

As figuras 5.48 (a) e 5.48 (b) tratam dos dados de Rotatividade da Equipe e indicam que o alto índice de rotatividade, previsto no cenário realizado ficou distante do imaginado pela equipe. Assim, o resultado do risco para esse tema foi alto, enquanto que previsões iniciais indicavam que o projeto seria pouco afetado por essa variável.

Os gráficos mostrados para este projeto, analisando-se os dois cenários para as variáveis de sistema e dos recursos humanos, indicam que as previsões iniciais foram concretizadas no cenário realizado. O modelo mostrou-se adequado para aplicação neste projeto. Uma dificuldade indicada no cenário realizado, no que tange aos recursos humanos, indica que os gestores devem ter uma preocupação constante com a retenção de

peçoal. Esse fator pode aumentar de forma considerável o risco de um projeto, conforme demonstrado no trabalho de Leopoldino (2004), no qual a pesquisa com os gestores de projeto indicou grande preocupação com tal fator de risco.

5.5.4 Projeto 4 - *Treinamento Online em Informática Básica Para Deficientes Visuais*

Esse projeto foi desenvolvido para capacitar deficientes visuais em utilização de informática básica e uso de ferramentas de criação internet. Esse público tem reduzidas possibilidades de trabalho, por baixa capacitação e falta de oportunidades no mercado, além disso existem poucas aplicações com recursos de acessibilidade. A idéia principal do trabalho foi criar um sistema que utilizasse recursos de Educação à Distância. O sistema deveria suportar sons e textos via internet ou rodar em redes locais.

Ocorreram dificuldades no levantamento de requisitos e na definição do escopo, além disso tratava-se de um tema novo para a equipe. Não foi necessário contratar muitos profissionais terceirizados, exceto um deficiente visual que ajudou nos testes do sistema e na elaboração do conteúdo além de um consultor com experiência em projetos desses tipo. A rotatividade no grupo era muito baixa. Havia projetos atrasados, o que dificultou o início desse trabalho. Foi necessário o uso de tecnologias ainda novas no mercado, como sistemas de leitura texto, software ainda desconhecido pela equipe. A previsão inicial era de que o projeto teria dificuldades de implementação.

CENÁRIO IDEAL PARA ESTE PROJETO

Uma análise inicial para este projeto indicava vários problemas que poderiam aumentar o risco. Para contornar as dificuldades previstas, como o desconhecimento tema, o desafio de desenvolver algo novo no mercado e a falta de conhecimento em projetos envolvendo acessibilidade, foi considerada a contratação de uma consultoria especializada em projetos para deficientes visuais.

Para questões envolvendo uso de software de audio e leitura de textos, o cenário mais indicado deveria incluir o uso de uma tecnologia já estabelecida no mercado, estável e com custos baixos, já que tratava-se de um software para uso gratuito em instituições de apoio aos cegos. Isso, no entanto, não foi possível nem previsto no orçamento do projeto. As aplicações testadas tinham custo alto e eram muito instáveis, motivo pelo qual foi necessário construir uma nova tecnologia para uso no sistema.

APLICAÇÃO DO MODELO - ANALISANDO VARIÁVEIS DE SISTEMA: CENÁRIO

PREVISTO

As variáveis linguísticas de entrada tiveram os valores:

- Escopo Mal Definido: pouco entendimento do problema (0,91); usuários envolvidos não são os indicados (0,65) e equipe envolvida é inexperiente na gestão do escopo (0,95). Saída de 0,85 (risco alto);
- Complexidade do Software: uso de novas tecnologias (0,95); dificuldades técnicas em implementar a solução (0,98) e tema é novo ou não é familiar para a equipe (0,90). Saída de 0,55 (risco médio);
- Requisitos Voláteis: mudanças constantes na definição dos requisitos (0,87), equipe é inexperiente em engenharia de requisitos (0,76) e há preferências de usuários ou a políticas governamentais (0,45). Saída de 0,21 (risco baixo).

O cenário previsto para esse projeto, analisando as variáveis do sistema, é mostrado nas três figuras a seguir:

Variável do Sistema - Escopo ou Objetivos Mal Definidos (cenário previsto)

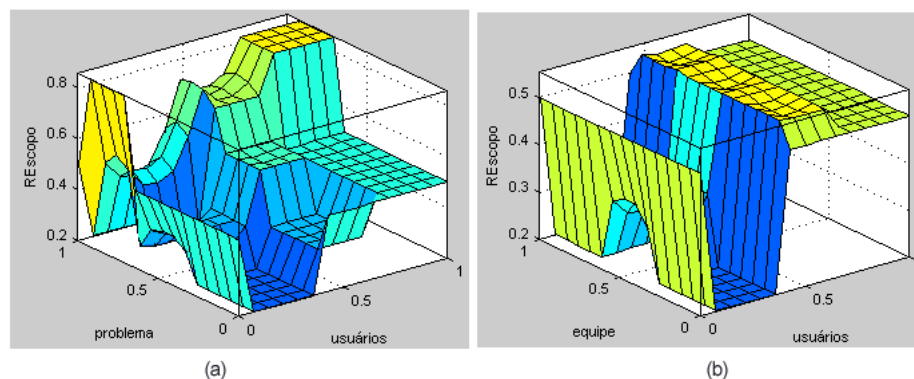


Figura 5.49: Gráfico de Cenário Previsto para Variável do Sistema Escopo ou Objetivos Mal Definidos. Fonte: Autor.

As figura 5.49 (a) e 5.49 (b), com os gráficos tratando de mal definição do escopo, identificam que os usuários escolhidos para apoio no projeto não foram os mais indicados para apoio no escopo e objetivos. O sistema era uma novidade para todos os futuros usuários. Isso teve grande influência no grau de risco do projeto. Na figura 5.49 (b), observa-se que a avaliação inicial para inexperiência da equipe envolvida teve uma influência menor no cenário previsto.

Variável do Sistema - Complexidade do Software (cenário previsto)

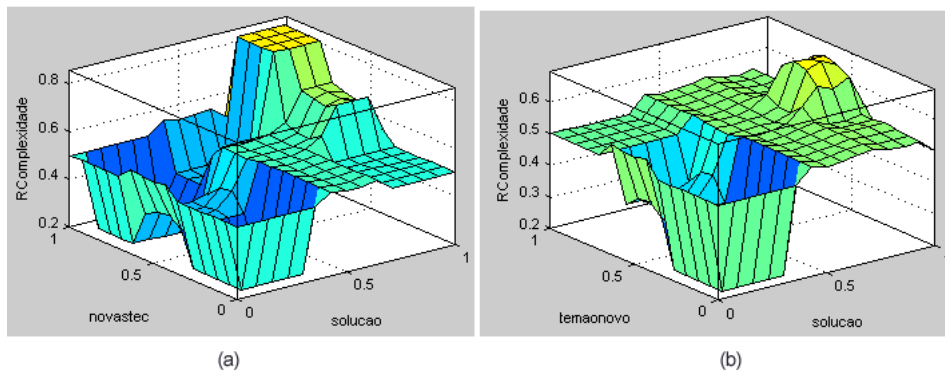


Figura 5.50: Gráfico de Cenário Previsto para Variável do Complexidade do Software. Fonte: Autor.

Nas figuras 5.50 (a) e 5.50 (b), avaliando a variável linguística Complexidade do Software, nota-se que o uso de novas tecnologias tornaria o projeto mais complexo, pois o grupo teria como desafio criar algo novo. Esse fato causou maior preocupação na equipe, logo houve elevação do risco em ambos os gráficos.

Variável do Sistema - Requisitos Voláteis (cenário previsto)

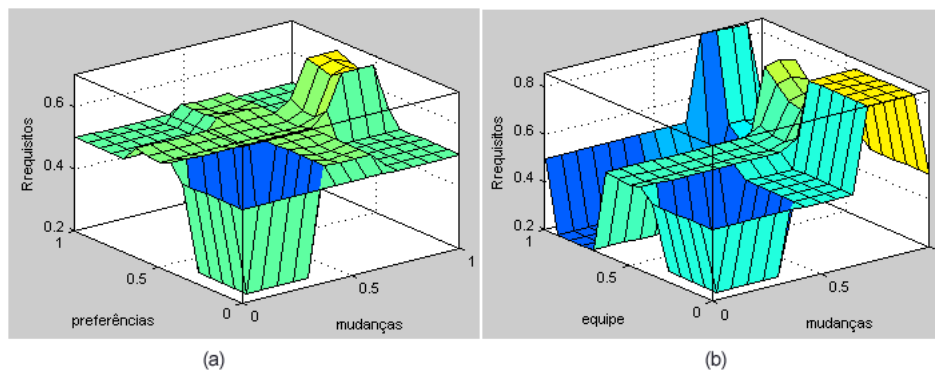


Figura 5.51: Gráfico de Cenário Previsto para Variável Requisitos Voláteis. Fonte: Autor.

As figuras 5.51 (a) e 5.51 (b) mostram o risco para a avaliação dos Requisitos Voláteis do projeto. A avaliação inicial da equipe identificou possíveis dificuldades na mudança dos requisitos, pois os diversos deficientes visuais que apoiaram o projeto tinham opiniões diferentes, principalmente sobre o conteúdo do curso que fora abordado no projeto. Na figura 5.51 (b), as mudanças constantes nos requisitos tornam evidente o aumento do risco.

APLICAÇÃO DO MODELO - ANALISANDO VARIÁVEIS DE SISTEMA: CENÁRIO REALIZADO

Os dados de entrada foram:

- Escopo: pouco entendimento do problema (0,76); usuários envolvidos não são os indicados (0,42) e equipe envolvida é inexperiente na gestão do escopo (0,72). Saída de 0,85 (risco alto);
- Complexidade do Software: uso de novas tecnologias (0,91); dificuldades técnicas em implementar a solução (0,87) e assunto é novo ou não é familiar para a equipe (0,89). Saída de 0,81 (risco alto);
- Requisitos Voláteis: mudanças constantes na definição dos requisitos (0,54); equipe é inexperiente em engenharia de requisitos (0,45) e há preferências de usuários ou há políticas governamentais (0,20). Saída de 0,45 (risco médio).

O cenário realizado para esse projeto, analisando as variáveis do sistema, é mostrado nas três figuras a seguir:

Variável do Sistema - Escopo ou Objetivos Mal Definidos (cenário realizado)

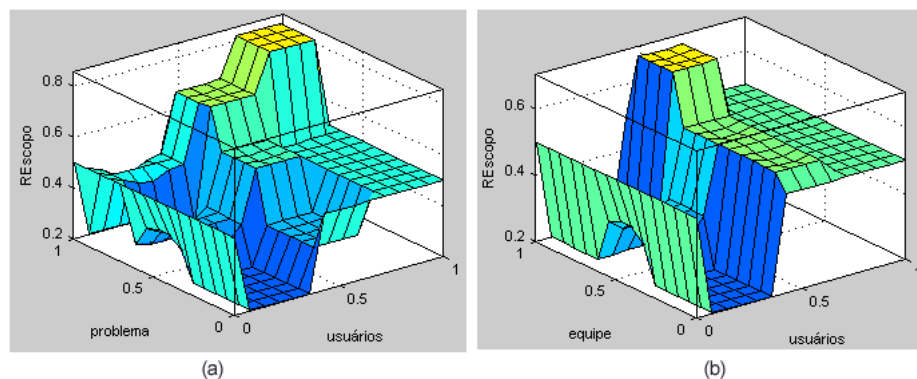


Figura 5.52: Gráfico de Cenário Realizado para Variável Escopo ou Objetivos Mal Definidos. Fonte: Autor.

Nos gráficos das figuras 5.52 (a) e 5.52 (b), a variável linguística Escopo ou Objetivo mal definidos teve risco incrementado por problemas na definição do escopo, confirmando as previsões do cenário realizado. Uma das grandes dificuldades encontradas foi o pouco entendimento do problema pela equipe, evidenciado pelo risco alto do gráfico da figura 5.52 (a).

Variável do Sistema - Complexidade do Software (cenário realizado)

As figuras 5.53 (a) e 5.53 (b) indicam que a Complexidade do Software esteve presente nos dois cenários, previsto e realizado, no entanto, apesar de o valor inserido para dificuldades técnicas no cenário previsto ser alto, no cenário realizado houve uma redução do risco para esse tema, conforme visto na figura 5.53 (a).

Variável do Sistema - Requisitos Voláteis (cenário realizado)

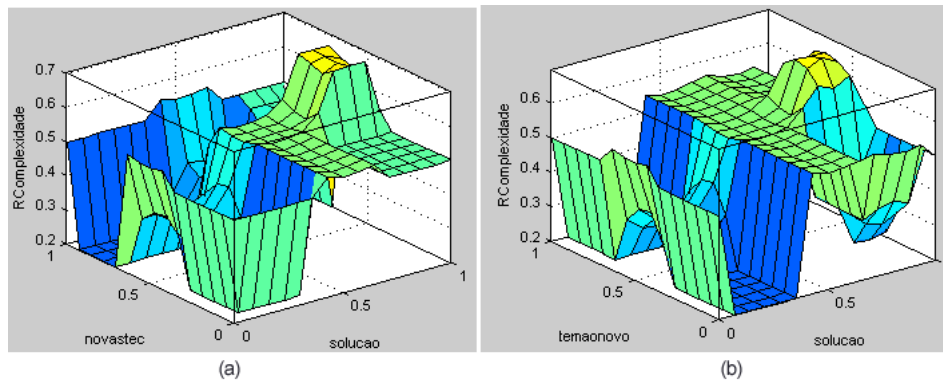


Figura 5.53: Gráfico de Cenário Previsto para Variável do Complexidade do Software. Fonte: Autor.

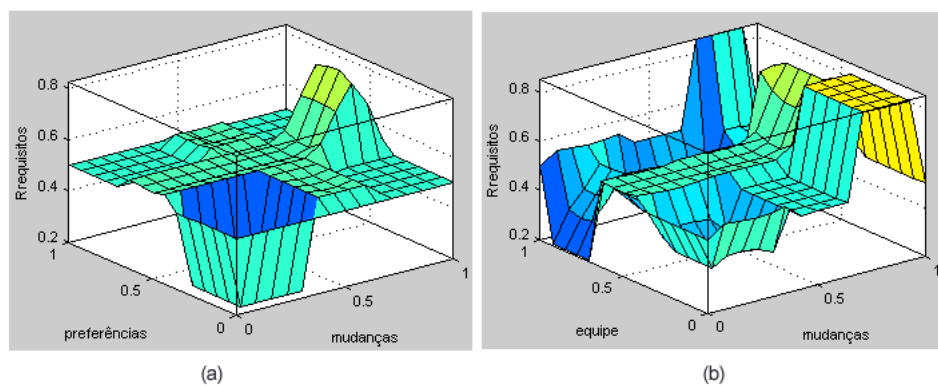


Figura 5.54: Gráfico de Cenário Previsto para Variável Requisitos Voláteis. Fonte: Autor.

A figuras 5.54 (a) e 5.54 (b) identificam os problemas de Requisitos Voláteis. No cenário previsto, os valores de entrada foram maior que os dos colocados no cenário realizado e isso fica evidenciado na figura 5.54 (a). O risco de 0,45 para o cenário realizado foi menor que o previsto, não confirmando as expectativas negativas da equipe, que inseriu dados com valor final de saída de 0,63, ou seja, risco médio. Um dos motivos para que isso tenha acontecido, foi a realização de reuniões para alinhar expectativas dos apoiadores do projeto e a simplificação do trabalho. Segundo [Boehm \(1988\)](#), a clareza dos requisitos deve fazer parte do checklist da avaliação de riscos de um projeto.

APLICAÇÃO DO MODELO - ANALISANDO VARIÁVEIS DOS RECURSOS HUMANOS: CENÁRIO PREVISTO

- Terceirização de Pessoal: até metade da equipe é terceirizada (0,23); terceirizados com boa experiência em projeto de software (0,42) e terceirizados seguem normas de qualidade reconhecidas pelo mercado (0,32). Saída de 0,44 (risco baixo);
- Equipe atua em projetos paralelos: atua em projetos em finalização (0,45); equipe alocada em diversos projetos impossibilitando o início da execução (0,87), equipe alocada em outros projetos e com prazos finalizados (0,76). Saída de 0,66 (risco

médio);

- Rotatividade de Pessoal: índice de Rotatividade (0,32); perfil com dificuldades de reposição (0,43) e políticas de retenção de pessoal (0,17). Saída de 0,36 (risco baixo).

O cenário previsto para o projeto, analisando as variáveis dos recursos humanos, é mostrado nas três figuras a seguir:

Variável dos Recursos Humanos - Terceirização de Pessoal (cenário previsto)

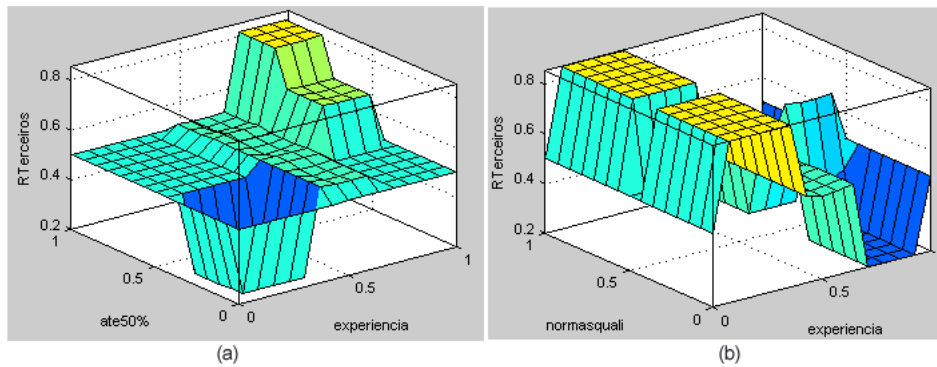


Figura 5.55: Gráfico de Cenário Previsto para Variável dos Recursos Humanos - Terceirização de Pessoal. Fonte: Autor.

As figuras 5.55 (a) e 5.55 (b), envolvendo a Terceirização de Pessoal, indica a preocupação do grupo com a alocação de profissionais terceirizados, que, mesmo em pequeno número, deveriam ter boa experiência em projeto de software. O gráfico mostrado na figura 5.55 (b) indica que o grau de risco teve um aumento considerável devido ao receio do grupo com a possibilidade de os profissionais terceirizados não seguir as normas de qualidade que foram indicadas.

Variável dos Recursos Humanos - Equipe em Projetos Paralelos (cenário previsto)

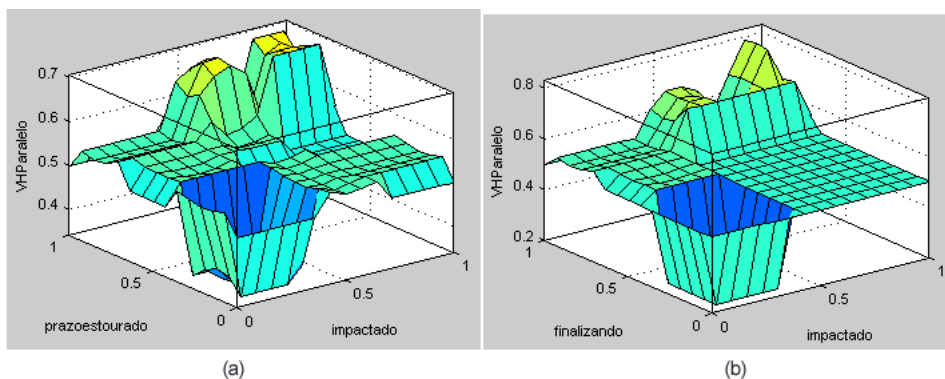


Figura 5.56: Gráfico de Cenário Previsto para Variável dos Recursos Humanos - Equipe em Projetos Paralelos. Fonte: Autor.

Nos gráficos das figuras 5.56 (a) e 5.56 (b), abordando a alocação da Equipe em

Projetos Paralelos, pode-se observar que havia diversos profissionais alocados em outros trabalhos, ou seja, o gestor aguardava a finalização dos trabalhos para que parte do grupo fosse inserido no projeto. O risco previsto pela equipe na construção deste cenário fica mais evidente na figura 5.56 (a), indicando, assim, que outros projetos tiveram seus prazos vencidos e causando atrasos em cadeia.

Variável dos Recursos Humanos - Rotatividade de Pessoal (cenário previsto)

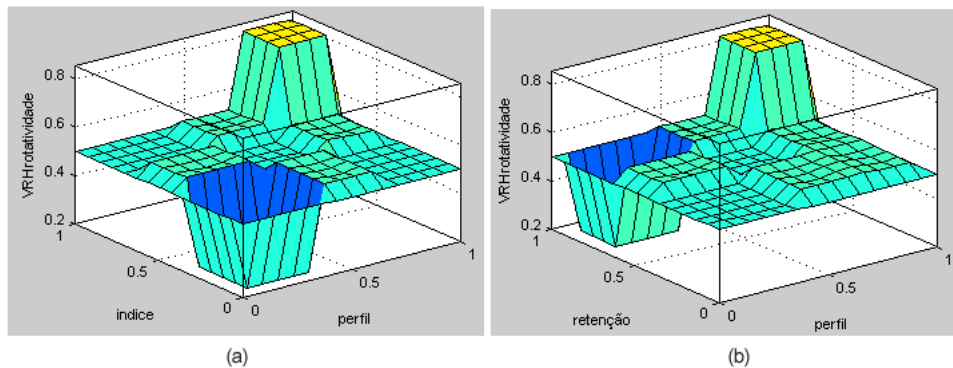


Figura 5.57: Gráfico de Cenário Previsto para Variável dos Recursos Humanos - Rotatividade da Equipe. Fonte: Autor.

As figuras 5.57 (a) e 5.57 (b) tratam da Rotatividade da Equipe. Os dois gráficos tiveram algumas similaridades nas imagens, conforme evidenciado no aumento do grau de risco nas extremidades superiores, indicando que o tema perfil de reposição difícil de ser encontrado foi determinante.

APLICAÇÃO DO MODELO - ANALISANDO VARIÁVEIS DOS RECURSOS HUMANOS: CENÁRIO REALIZADO

- Terceirização de Pessoal: até metade da equipe é terceirizada (0,48); terceirizados com boa experiência em projeto de software (0,27) e terceirizados seguem normas de qualidade reconhecidas pelo mercado (0,22). Saída de 0,33 (risco baixo);
- Equipe atua em projetos paralelos: atua em projetos em finalização (0,56); equipe alocada em diversos projetos impossibilitando o início da execução (0,49), equipe alocada em outros projetos e com prazos finalizados (0,26). Saída de 0,39 (risco baixo);
- Rotatividade de Pessoal: índice de Rotatividade (0,35); perfil com dificuldades de reposição (0,47) e políticas de retenção de pessoal (0,10). Saída de 0,19 (risco baixo).

O cenário realizado para esse projeto, analisando as variáveis dos recursos humanos, é mostrado nas três figuras a seguir:

Variável dos Recursos Humanos - Terceirização de Pessoal (cenário realizado)

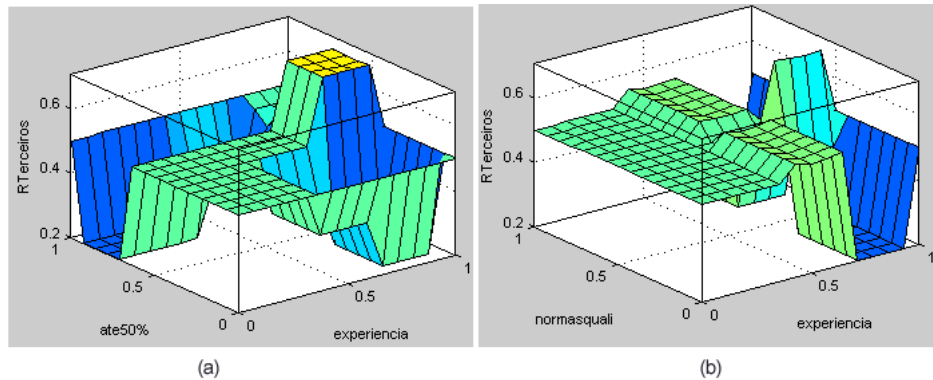


Figura 5.58: Gráfico de Cenário Realizado para Variável dos Recursos Humanos - Terceirização de Pessoal. Fonte: Autor.

As figuras 5.58 (a) e 5.58 (b) mostram os gráficos para Terceirização de Pessoal. Os gráficos gerados evidenciam a diferença de risco para o que foi previsto e realizado. No realizado, para uso de terceirizados no processo, por exemplo, o cenário resultante indicou o projeto com risco baixo, com o valor de 0,33, para as condições de entrada já mencionadas. Isso indicou que a preocupação inicial do grupo com esse tema não foi confirmada ao término do trabalho.

Variável dos Recursos Humanos - Equipe em Projetos Paralelos (cenário realizado)

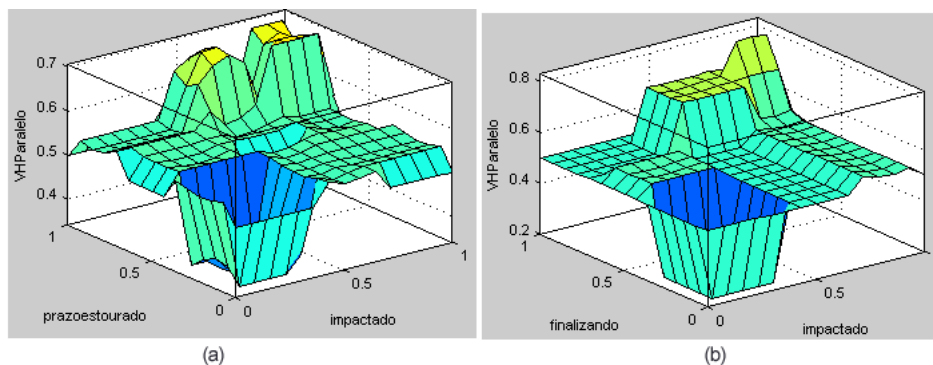


Figura 5.59: Gráfico de Cenário Realizado para Variável dos Recursos Humanos - Equipe em Projetos Paralelos. Fonte: Autor.

Nas figuras 5.59 (a) e 5.59 (b), Equipe em Projetos Paralelos, o cenário realizado e previsto teve valores diferentes, indicando risco maior no previsto com o valor final de 0,62 (risco médio). Já no realizado, os valores foram mais positivos, com indicação de grau de risco no valor de 0,39 (risco baixo).

Variável dos Recursos Humanos - Rotatividade de Pessoal (cenário realizado)

As figuras 5.60 (a) e 5.60 (b) tratam dos dados de Rotatividade da Equipe e indicam que índice de rotatividade alto, sinalizado no cenário realizado, assim como

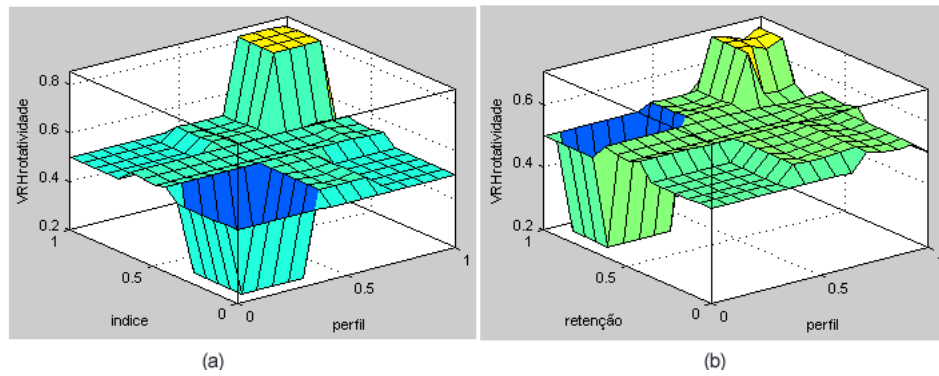


Figura 5.60: Gráfico de Cenário Realizado para Variável dos Recursos Humanos - Rotatividade da Equipe. Fonte: Autor.

no cenário previsto, tiveram avaliações aproximadas. As ações para melhorar o índice de retenção de pessoa foram influentes na redução do risco, conforme mostrado na figura 5.60 (a).

Este projeto, diferente dos relacionados anteriormente, teve índices de acerto para os graus de risco menores que os três primeiros aqui analisados. Isso ocorreu, principalmente, devido à complexidade do problema e desconhecimento do tema pela equipe desenvolvedora. Além disso, o sistema exigiu a utilização de tecnologias de acessibilidade e leitura de textos que eram caras para distribuição no mercado. É importante mencionar que o sistema foi distribuído de forma gratuita para algumas associações de cegos do Estado da Bahia.

Para uma gestão adequada de projetos de software, é recomendado que os riscos sejam identificados, validados e tratados para aumentar a taxa de sucesso dos projetos. Os gestores devem ter especial atenção quanto aos contratados terceirizados e quanto à alocação de profissionais atuantes em outros projetos e, sempre que possível, ter profissionais com formação adequada à função que será desempenhada.

Conforme visto no decorrer deste capítulo, aspectos relacionados ao sistema, como tratamento adequado do risco, gestão dos requisitos, complexidade do software e mudanças frequentes no escopo e objetivos afetam significativamente a produção do software. Essas são dificuldades constantes no processo de produção de um software e podem incrementar sobremaneira os riscos do projeto.

5.5.5 Pesquisa com Profissionais e Índice de Acerto do Modelo

Os dados a seguir demonstram os índices de acertos em cada projeto, comparando o valor previsto e o realizado:

- Projeto 1 (Rede Social): as variáveis do sistema para valores previstos e realizados tiveram índice de acerto de 100%. Já as variáveis dos recursos humanos tiveram índice de 100%;
- Projeto 2 (Sistema Gestor de Informação): as variáveis do sistema para valores previstos e realizados tiveram índice de acerto de 66%, enquanto que as variáveis dos recursos humanos tiveram índice de 100%;
- Projeto 3 (Sistema de Capacitação Online): as variáveis do sistema para valores previstos e realizados tiveram índice de acerto de 66%, ao passo que nas variáveis dos recursos humanos o índice foi de 66%;
- Projeto 4 (Informática Para Deficientes Visuais): as variáveis do sistema para valores previstos e realizados tiveram índice de acerto de 33%. Já as variáveis dos recursos humanos tiveram índice de 33%.

O trabalho fora finalizado com uma pesquisa envolvendo vinte desenvolvedores de software da cidade do Salvador e dez gestores de projetos, que responderam a um questionário no mês de janeiro de 2014 e classificaram a influência dos riscos do seguinte modo:

RISCOS DO SISTEMA		
ESCOPO OU OBJETIVO MAL DEFINIDOS		
Pouco entendimento do problema	Usuários não são os indicados	Equipe envolvida é inexperiente
35%	15%	50%
COMPLEXIDADE DO SOFTWARE		
Uso de novas tecnologias	Dificuldades técnicas	Assunto é novo ou não é familiar
36%	30%	34%
REQUISITOS VOLÁTEIS		
Mudanças constantes	Equipe inexperiente	Preferências de usuários ou a políticas governamentais
46%	30%	24%
RISCOS DOS RECURSOS HUMANOS		
TERCEIRIZAÇÃO DE PESSOAL		
Até 50% da equipe terceirizada	Terceirizados com boa experiência em projeto de software	Terceirizados seguem normas de qualidade reconhecidas pelo mercado
34%	40%	26%
EQUIPE EM PROJETOS PARALELOS		
Atuando em outros projetos em finalização	Em outros projetos e impactando na alocação de pessoal	Em outros projetos com prazos já estourados
35%	25%	40%
ROTATIVIDADE DE PESSOAL		
Índice de Rotatividade	Perfil com dificuldades de reposição	Políticas de retenção de pessoal
45%	25%	30%

Figura 5.61: Pesquisa com Profissionais de T.I. de Salvador. Fonte: Autor.

Considerações finais

O modelo proposto neste trabalho teve como objetivo principal apoiar os gestores de projetos, possibilitando avaliar os riscos que envolve questões relacionadas ao pessoal e à tecnologia. Conforme mostrado no capítulo anterior, diversas situações podem aumentar ou diminuir o risco que um projeto pode ter. O trabalho buscou identificar algumas situações reais que influenciam o incremento do risco, através da criação de um cenário de possibilidades.

Posteriormente, o cenário foi comparado com os resultados reais que foram encontrados após a finalização do projeto e foi possível verificar a eficácia do modelo ao utilizar a Lógica Fuzzy. Diferente do trabalho de [Goncalves \(2006\)](#), esta pesquisa não tratou de riscos envolvendo testes de software, especificação de sistemas ou questões relacionadas a riscos financeiros do projeto. Esses temas, apesar de importantes, podem ser considerados para melhorias do modelo em trabalhos futuros.

Os resultados do trabalho confirmam a necessidade de acompanhamento e administração dos aspectos referentes à gestão de pessoas e às tecnologias utilizadas. O cálculo dos graus de riscos do projeto permite que os gestores analisem pontos para melhoria nos seus processos e possibilita identificar quais dificuldades são mais frequentes no contexto da empresa. Esta pode mapear os riscos, gerar um registro de histórico de eventos e propor ações de resposta para aumentar as chances de êxito em todos os projetos desenvolvidos.

6.1 Conclusões

Após avaliar os resultados obtidos nessa pesquisa, foi possível constatar que o processo de produção do software é uma atividade complexa, influenciada por fatores diversos. Os projetos são dependentes de variáveis que exigem atenção em todas as suas etapas de produção. É necessário realizar a gestão dos riscos de um projeto durante toda a fase do ciclo de vida do software, principalmente no que tange à gestão dos recursos humanos,.

A pesquisa indicou, também, que a avaliação dos fatores que podem prejudicar o andamento dos projetos, em geral, é realizada pelos gestores de acordo com experiência obtida em projetos anteriores, conforme relacionados nos trabalhos de [Goncalves \(2006\)](#)

e [Leopoldino \(2004\)](#). Os dois trabalhos assemelham-se com a presente pesquisa, quando indicam a importância de avaliar e conhecer os riscos do projeto, para com isso reduzir as chances de perda, insatisfação com o cliente e para melhorias no processo de produção do software. Essas informações auxiliam a empresa na definição dos prazos e custos, na alocação de pessoal e na gestão da equipe. É necessário alinhar as estratégias de gestão de pessoal com os fatores relacionados à tecnologia, permitindo que ambos proporcionem eficiência no processo de produção do software.

A escolha da Lógica Fuzzy para aplicação nessa pesquisa, com a definição de uma base de regras e os processos de fuzzificação e defuzzificação atestam as inúmeras possibilidades que o modelo permite. O uso das variáveis e termos linguísticos, como risco baixo, médio e alto, são difíceis de ser evidenciadas em outros modelos matemáticos, além de tornar mais claro o estudo das incertezas.

A análise dos resultados do modelo aqui proposto demonstra que é viável criar ferramentas baseadas em lógica fuzzy para o auxílio à tomada de decisão, [Lima \(2008\)](#). A Lógica Fuzzy neste trabalho permitiu o estudo de um processo de produção complexo, como o vivenciado por profissionais que atuam com software, assim como foi demonstrado pela influência causada por fatores humanos e técnicos do sistema.

6.2 Contribuições

Acreditamos que os objetivos do presente trabalho foram alcançados, na medida em que resultados importantes foram obtidos como fruto da pesquisa. Foi possível constatar a eficiência do modelo ao comparar os cenários previstos e realizados, que tiveram valores finais próximos das expectativas iniciais. Os resultados encontrados e a proposta do modelo podem apoiar os gestores de projetos de software, permitindo implementar ações para minimizar os riscos.

A pesquisa buscou, ainda, uma sensibilização sobre o processo de avaliação de riscos em projetos de software. Essas atividades merecem atenção constante e devem ser tratadas para aumentar as chances de sucesso do projeto. Dessa forma, espera-se que este trabalho amplie a discussão sobre o tema, principalmente por ter um caráter experimental.

6.3 *Atividades Futuras de Pesquisa*

A aplicação do presente estudo pode trazer resultados favoráveis aos gestores de projeto de software. A ampliação do trabalho pode incluir outros temas relevantes no processo de produção, a exemplo de questões como navegabilidade, design de interfaces, segurança de dados, performance da aplicação ou temas relacionados à qualidade do produto.

Atividades futuras dessa pesquisa podem incluir, ainda, a utilização de um grupo maior de base de regras, inserção de outras variáveis linguísticas de entrada ou de saída. O modelo aqui sugerido pode, também, ser adaptado para execução em projetos de aplicações móveis ou sistemas que usam a computação em nuvem.

Referências Bibliográficas

- ACEPROJECT. <http://www.aceproject.com/>. Acessado em maio de 2012.
- AGUIAR, H.; JUNIOR, O. *Lógica Difusa*. Rio de Janeiro: Editora Interciência, 1999.
- ANDRADE, T.; SOUZA, J. Uma linguagem de Padrões de Estimativa de Software para Micro e Pequena Empresas. In: *7ª Conferência Latino-Americana em Linguagens de Padrões para Programação*. 1999.
- ASSESPRO. <http://www.assespro.org.br/>. Acessado em setembro de 2013. [s. d.].
- BARKI, H. An information systems classification scheme: an update. *MIS Quarterly*, p. 209–226, 1993.
- BASILI, V.; ROMBACH, D. The tame project: Towards improvement-oriented software environments. In: *IEEE Transactions on Software Engineering*. [S.l.: s.n.], 1988. p. 758–773.
- BASILI, V.; WEISS, D. A methodology for collecting valid software engineering data. In: *IEEE Transactions on Software Engineering*. [S.l.: s.n.], 1984.
- BASILI, V. R. Software development: A paradigm for the future. In: *Proceedings of the 13th Annual International Computer Software and Applications Conference*. Orlando: [s.n.], 1999. p. 471–485.
- BFPUG. <http://www.bfpug.com.br/>. Acessado em novembro de 2010. [s. d.].
- BOEHM, B.W.A. Spiral Model of Software Development and Enhancement. In: *IEEE Computer*. Vol. 21, n. 5, 1988.
- CARLETON, A. D. et al. *Software Measurement for DoD Systems: Recommendation for Initial Core Measures*. Setembro 1992. Software Engineering Institute, Carnegie Mellon University.
- CHARETTE, R. N. Why software fails. *IEEE Spectrum*, v. 42, n. 9, p. 42–49, 2005.
- CMMI. <http://www.cmmi.org.br/>. Acessado em setembro de 2013. [s. d.].
- DEMARCO, T. *Controle de Projetos de Software*. 9 ed. Rio de Janeiro: Editora Campus, 1991.
- FENTON, N. E.; KAPOSÍ, A. A. Metrics and software structure. *J. Information and Software Tech*, p. 301–320, 1987.
- FILHO, W. d. P. P. *Engenharia de Software*. [S.l.]: LCT, 2009.

- GROUP, G. *Quantitative Software Management*. Setembro 2011. URL: www.gartner.com.
- GONCALVES, Encarnação de Lourdes B. Andreo. *Gerenciamento de Risco de Software* Dissertação de Mestrado. UNIVERSIDADE METODISTA DE PIRACICABA. 2006
- HALL, E. *Managing Risk - Methods for Software Systems Development*. [S.l.]: Addison-Wesley, 1998.
- IDC BRASIL <http://br.idclatin.com/>. Acessado em novembro de 2012.
- IFPUG. *Counting Practices Manual*. Version 4.3, January, 2010
- ISO. <http://www.iso.org.br/>. Acessado em setembro de 2013. [s. d.].
- ITIL <http://www.itil-officialsite.com/>. Acessado em novembro de 2013.
- JAN, J. S. R. *Neuro and Soft Computing*. [S.l.]: Prentice Hall, 2004.
- KEMERER, C. F. An empirical validation of software cost estimation. *Communications of the ACM*, 1987.
- KEMERER, C. F. Reability of function points measurement: A field experiment. *Communications of the ACM*, 1993.
- KLIR, G. J.; YUAN, B. *Fuzzy Sets, Fuzzy Logic ans Fuzzy Systems*. [S.l.]: World Scientific, 1996.
- PEREIRA, H.B.; BONESS, W.J.S. Modelagem Fuzzy da Violência e Criminalidade *MODELAGEM FUZZY DA VIOLÊNCIA E CRIMINALIDADE*, XLIISBPO.
- PMI - Project Management Institute. www.pmi.org/. Acessado em maio de 2012. [s. d.]
- KM PROJECT. www.kmproject.be/. Acessado em maio de 2012.
- BARKI, H. An information systems classification scheme: an update. *MIS Quarterly*, p. 209–226, 1993.
- KRIESERS, P. Junho 2011. URL: baguete.com.br/colunistas/colunas/51/paulo-krieser.
- KUNCHEVA, L. I.; STEIMANN, F. Fuzzy diagnosis: editorial. *Artificial Intelligence in Medicine*, n. 16, p. 121–128, 1999.
- LEOPOLDINO, C. B. de. *Avaliação de Riscos em Desenvolvimento de Software*. Dissertação de Mestrado. Universidade Federal do Rio Grande do Sul. 2004.
- Lima, Marcio B. *A Lógica Fuzzy tipo 2 e um estudo de caso aplicado no Controle de Tráfego Aéreo*. Dissertação de Mestrado. Universidade Federal da Bahia. 2008.
- LÓPEZ, P. A. P.; SILVA, C. F. C. I. *Um modelo para estimativas de custo auxiliando na Gerência de Projetos de Software*. São Leopoldo: [s.n.], 2004. Artigo Técnico, UNISINOS.

- LUCA,, C. D. *Em seis meses, Brasil Mais TI forma 70 mil jovens*. Acessado em março de 2013. 2013.
- MACHADO, C. Filipak. de. *A-Risk. Método para Indentificar e Quantificar Riscos de Prazos em Projetos de Desenvolvimento de Software*. Dissertação de Mestrado. PUCPR - Pontifícia Universidade Católica do Paraná. 2002.
- MALDONADO, J. C.; ROCHA, A. R. C.; WEBER, K. C. *Qualidade de Software: Teoria e Prática*. São Paulo: Prentice Hall, 2001.
- MARCONI, F. V. *Gerenciamento de Projetos de Tecnologia da Informação*. [S.l.]: Campus, 2008.
- MCNEILL, F. M.; THRO, E. *Fuzzy Logic: a practical approach*. London: AP Professional, 1994.
- MENDEL, J. M. Fuzzy logic systems for engineering: a tutorial. *Proc. IEEE*, v. 83, n. 3, p. 345–377, 1995.
- MENEZES, L. C. M. *Gestão de projetos*. São Paulo: Atlas, 2001.
- MS PROJECT. <http://office.microsoft.com/>. Acessado em maio de 2012.
- NORVIG, P.; RUSSEL, S. *Inteligencia Artificial*. [S.l.]: Campus, 2006.
- OMAN P.; PFLEEGER S. L. *Applying Software Metrics*. *IEEE Computer Society Press*, 1997.
- PASSINO, K. M.; YURKOVICH, S. *Fuzzy Logic: a practical approach*. California: Addison Wesley, 1998.
- PETERS, James F, PEDRYCS, Witold. *Software Engineering: An Engineering Approach..* John Wiley and Sons, 2001.
- PRESSMAN, R. *Engenharia de Software: um enfoque prático*. São Paulo: McGraw-Hill, 2005.
- PMBOK, Guia PMBOK. *PMBOK*, 2000.
- RIOS, E.; FILHO, T. R. M. *Testes de Software*. [S.l.]: Alta Books, 2006.
- RIQTEK. <http://www.riqtek.com/>. Acessado em maio de 2012.
- ROBERTS, F. S. *Measuremet Theory with Applications to Decision Making, Utility, and the Social Sciences*. Boston: Addison-Wesley, 1979.
- ROCHA, A. R. C.; MALDONADO, J. C. *Qualidade de Software*. [S.l.]: Prentice Hall, 2004.
- ROCHA, A. R. C.; MALDONADO, J. C.; WEBER, K. C. *Qualidade de Software: Teoria e Prática*. São Paulo: Prentice Hall, 2001.

- ROCHE, J. M. Software metrics and measurement principles. In: *SIGSOFT Softw. Eng. Notes*. New York, NY, USA: ACM, 1994. v. 19, p. 77–85.
- RUDOLPH, E. E. *Productivity in computer application development*. Dissertação (New Zealand Working Paper 9) — Univ. Auckland, Dept. of Management Studies, Auckland, 1983.
- SHAW, I. S.; SIMÕES, M. G. S. *Controle e Modelagem Fuzzy*. São Paulo: Editora Edgard Blücher, 1999.
- Sommerville, I. *Engenharia de Software*. Pearson, 2003.
- SOUSA, A. G. de. *Análise de Pontos de Função Estendida: Métrica de Software Baseada na Abordagem das Dimensões Tecnológica e Ambiental/Contextual*. Dissertação de Mestrado. CIMATEC. 2006.
- TAJIMA, D.; MATSUBARA, T. The computer software industry in japan. *IEEE Computer*, v. 15, n. 5, 1984.
- VAZQUEZ, C. E.; SIMÕES, G. S.; ALBERT, R. M. *Análise de Pontos de Função: Medição, Estimativas e Gerenciamento de Projetos de Software*. 3ª ed. São Paulo: Editora Erica, 2005.
- VIEIRA, F. M. *Gerenciamento de Projetos de TI*. [S.l.]: Elsevier, 2005.
- WARD, J. L. *Project Management Terms: A Working Glossary*. [S.l.]: ESI International, 2000.
- ZADEH, L. Fuzzy sets. *Information Control* 8, p. 338–353, 1965.

MODELO PARA AVALIAÇÃO DE RISCOS EM PROJETOS DE SOFTWARE

Alex de Oliveira Coelho

Salvador, Fevereiro/2014.