

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL  
FACULDADE DE INFORMÁTICA  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO**

**UM MODELO PARA ESTIMATIVA  
DE ESFORÇO EM PROJETOS  
DE REENGENHARIA DE  
SOFTWARE**

**SILVIA CRISTINA NUNES DAS DÔRES**

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre em Ciência da Computação na Pontifícia Universidade Católica do Rio Grande do Sul.

Orientador: Prof. Dr. Duncan Dubugras Alcoba Ruiz

**Porto Alegre  
2015**



# **FICHA CATALOGRÁFICA**



# **TERMO DE APRESENTAÇÃO**



## **DEDICATÓRIA**

Ao meu afilhado Luiz Felipe e ao meu namorado Leandro.



## AGRADECIMENTOS

À Deus, que em sua infinita bondade me permitiu concluir mais esta etapa em minha formação.

À minha grande e amada família, pelo apoio em todos os momentos de minha vida, em especial da minha trajetória acadêmica. Meus tios, tias, primos e primas, pai e irmã, a saudade é imensa, mas espero que os resultados valham a pena. Tudo sempre será por vocês.

Ao meu namorado, companheiro e maior incentivador, Leandro. Obrigada por toda a paciência, carinho e incentivo, fostes a luz em todos os momentos de escuridão deste período.

Ao meu orientador Prof. Duncan Ruiz, por acreditar no meu trabalho desde o início, pela paciência em me manter no foco e pelas críticas sem as quais eu não teria conseguido construir este trabalho. Além disso, agradeço pelo esforço em conseguir recursos que financiassem integralmente a minha pesquisa.

À Profa. Sabrina Marczak, por todas as valorosas sugestões e revisões ao trabalho.

Aos colegas da PUCRS por dividirem experiências durante estes dois anos de aprendizado. Agradeço especialmente ao Bernardo, que me incentivou a vir para esta Universidade excelente. Também agradeço aos colegas do GPIN, pela companhia diária, sugestões importantes (as famosas prévias) e momentos de descontração.

À todos os demais amigos que fiz em Porto Alegre, especialmente os da CG da UFRGS, por tornarem este período longe da família mais leve e divertido.

À HP Brasil, por financiar a pesquisa e especialmente ao André Kalsing, que tornou possível grande parte da realização da pesquisa.

Às empresas e aos participantes do estudo empírico, por compartilharem suas experiências e ajudarem a construir este trabalho.

À todos os que torceram por mim, muito obrigada.



# UM MODELO PARA ESTIMATIVA DE ESFORÇO EM PROJETOS DE REENGENHARIA DE SOFTWARE

## RESUMO

A estimativa de esforço é um dos cerne de um projeto de desenvolvimento de software uma vez que é usada para muitas finalidades, tais como, estimativa de custo, cronograma, alocação de recursos, orçamento e investimentos em software. Dada a relevância da área, existem diversas pesquisas que se preocupam principalmente em propor novos modelos e utilizar novas técnicas para melhorar a precisão das estimativas ou avaliar o melhor modelo a ser aplicado em um dado contexto. No contexto específico de projetos de reengenharia de software, há carência de trabalhos relacionados e, ao contrário do que ocorre para o desenvolvimento de um projeto novo, pouco se sabe sobre como ocorre a estimativa de esforço em projetos deste tipo na prática. Neste contexto, este trabalho teve como objetivo propor um modelo para estimativa de esforço em reengenharia de software, que inclui as etapas a serem realizadas para planejamento, aplicação, monitoramento e aprendizagem desta estimativa. Tal modelo foi proposto com base na literatura relacionada, em desafios, boas práticas e lições aprendidas identificados na indústria a partir da realização de dois estudos empíricos. Tais estudos envolveram um estudo de campo exploratório e estudo de caso, em organizações que realizam reengenharia de software.

**Palavras-Chave:** Estimativa de Esforço, Estudo Empírico, Modelo, Reengenharia de Software.



# A MODEL FOR EFFORT ESTIMATION IN SOFTWARE REENGINEERING PROJECTS

## ABSTRACT

Effort estimation is in the core of a software development project since it is useful for cost estimation, resource allocation planning, and follow-up of software investment and budget. Given the importance of the area, there are several studies that are mainly concerned on proposing new models and on the use of new techniques to improve the accuracy of estimates or to evaluate the best model to be applied in a given context. In the specific context of software reengineering projects, there is a lack of related work. Indeed, on the contrary of what occurs for the development of new projects, there is very few knowledge on how the effort estimation is done in that type of projects. In this context, this work aims to propose a model to estimate effort in software reengineering, which includes the steps to carry-out for planning, implementation, monitoring and learning of this estimation. This model was proposed based on the related literature, challenges, good practices and lessons learned identified in two empirical studies in the industry. Such studies involved an exploratory field study and a case study in organizations that carry out software reengineering projects.

**Keywords:** effort estimation, empirical study, model, software reengineering.



## LISTA DE FIGURAS

Figura 2.1 – Processo Reengenharia de Software (Adaptado de [PP01]) . . . . .	35
Figura 2.2 – Processo de Estimativa de Esforço (Adaptado de [MEN14]) . . . . .	39
Figura 3.1 – Desenho da Pesquisa . . . . .	56
Figura 3.2 – Metodologia do Estudo de Campo Inicial . . . . .	60
Figura 3.3 – Metodologia do Estudo de Caso . . . . .	66
Figura 4.1 – Grafo da Categoria Sistema Legado . . . . .	81
Figura 4.2 – Grafo da Categoria Cliente . . . . .	82
Figura 4.3 – Grafo da Categoria Equipe . . . . .	83
Figura 4.4 – Grafo da Categoria Projeto . . . . .	85
Figura 5.1 – Grafo da Categoria Sistema Legado no Estudo de Caso . . . . .	95
Figura 5.2 – Grafo da Categoria Cliente no Estudo de Caso . . . . .	97
Figura 5.3 – Grafo da Categoria Equipe no Estudo de Caso . . . . .	99
Figura 5.4 – Grafo da Categoria Organização . . . . .	100
Figura 5.5 – Grafo da Categoria Projeto no Estudo de Caso . . . . .	101
Figura 6.1 – Consolidação dos Desafios . . . . .	111
Figura 6.2 – Consolidação das Boas Práticas . . . . .	114
Figura 7.1 – Modelo Proposto . . . . .	120
Figura 7.2 – Planejamento Estratégico . . . . .	124
Figura 7.3 – Planejamento da Estimativa . . . . .	128
Figura 7.4 – Estimativa de Esforço . . . . .	132
Figura 7.5 – Monitoramento e Calibragem . . . . .	136
Figura 7.6 – Aprendizagem . . . . .	138
Figura 7.7 – Modelo de Estimativa de Esforço em Diferentes Contextos de Pro- cessos de Reengenharia . . . . .	140
Figura A.1 – Interface de Upload de Arquivos StArt . . . . .	174
Figura A.2 – Informações Gerais dos Artigos . . . . .	174
Figura A.3 – Aplicação dos Critérios de Inclusão e Exclusão na Fase I . . . . .	175
Figura A.4 – Extração dos Dados na Ferramenta StArt . . . . .	178
Figura A.5 – Evolução do número de artigos ao longo dos anos. . . . .	179
Figura A.6 – Comparativo entre os tipos de publicação e o número de artigos. . . . .	180
Figura A.7 – Código para gerar SNA dos autores, dentro do ambiente R. . . . .	184

Figura A.8 – Grafo da rede social dos autores que pesquisam sobre Soluções de Apoio a Estimativa de Esforço em Projetos de Desenvolvimento de Software.	185
Figura A.9 – Distribuição das Técnicas de Estimativa 1978-2013. . . . .	192
Figura A.10 – Abordagens de Aprendizagem de Máquinas Aplicadas para Estimar Esforço em Desenvolvimento de Software. . . . .	193
Figura A.11 – Técnicas Combinadas . . . . .	194
Figura A.12 – Abordagens de OAM e Baseadas em Modelo mais Utilizadas . . . . .	194
Figura A.13 – Domínios Explorados em Soluções de Estimativa de Esforço. . . . .	195
Figura A.14 – Técnicas de Estimativa de Esforço mais Utilizadas de Acordo com o Domínio de Desenvolvimento . . . . .	195
Figura A.15 – Datasets públicos utilizados para validação de soluções . . . . .	196

## LISTA DE TABELAS

Tabela 2.1 – Requisitos básicos para soluções de estimativa de esforço . . . . .	50
Tabela 3.1 – Questões Aplicadas nas Entrevistas - Primeira Fase . . . . .	62
Tabela 3.2 – Questões Aplicadas nas Entrevistas - Segunda Fase . . . . .	63
Tabela 3.3 – Questões Aplicadas nas Entrevistas - Estudo de Caso . . . . .	69
Tabela 4.1 – Caracterização dos Entrevistados por Organização - Estudo de Campo Inicial . . . . .	75
Tabela 5.1 – Papel dos Entrevistados nos Projetos . . . . .	92
Tabela 6.1 – Atividades de Estimativa . . . . .	107
Tabela 6.2 – Síntese dos Desafios Encontrados . . . . .	109
Tabela 6.3 – Síntese das Boas Práticas Aplicadas . . . . .	112
Tabela 7.1 – Mapeamento das Boas Práticas para o Modelo de Estimativa. . . . .	121
Tabela 7.2 – Mapeamento das Lições Aprendidas para o Modelo de Estimativa . .	122
Tabela 7.3 – Mapeamento dos Desafios para o Modelo de Estimativa . . . . .	123
Tabela A.1 – Anais de Conferências Seleccionados a Partir da Listagem de Qualis da CAPES . . . . .	166
Tabela A.2 – Periódicos Seleccionados a Partir do Trabalho de [JØR07] . . . . .	166
Tabela A.3 – Cronograma da Revisão Sistemática . . . . .	170
Tabela A.4 – Número de artigos originalmente extraídos dos motores de busca . .	172
Tabela A.5 – Número de Artigos Encontrados Manualmente . . . . .	173
Tabela A.6 – Número de Artigos Encontrados em Motores de Busca e na Busca Manual . . . . .	179
Tabela A.7 – Número de artigos separados por nome de conferência . . . . .	180
Tabela A.8 – Número de artigos organizados por periódicos. . . . .	181
Tabela A.9 – Departamentos, faculdades e institutos interessados pelo tema. . . . .	182
Tabela A.10 – Autores com mais de duas publicações sobre o tema . . . . .	183
Tabela A.11 – Matriz de Autores para Geração da Rede Social . . . . .	184
Tabela A.12 – Visão geral dos resultados - Parte 1 (Artigo 1-39) . . . . .	187
Tabela A.13 – Visão geral dos resultados - Parte 2 (Artigo 40-79) . . . . .	188
Tabela A.14 – Visão geral dos resultados - Parte 3 (Artigo 80-109) . . . . .	189
Tabela A.15 – Distribuição de Frequência (absoluta e relativa) do aspecto “Método de Pesquisa” . . . . .	189
Tabela A.16 – Distribuição de Frequência (absoluta e relativa) do aspecto “Foco da Contribuição” . . . . .	190

Tabela A.17 – Distribuição de Frequência (absoluta e relativa) do aspecto “Tipo de Solução” .....	190
Tabela A.18 – Distribuição de Frequência (absoluta e relativa) do aspecto “Valor de Representação” .....	190
Tabela A.19 – Distribuição de Frequência (absoluta e relativa) do aspecto “Etapa do Desenvolvimento” .....	190
Tabela A.20 – Técnicas de Estimativa no Decorrer dos Anos .....	191
Tabela A.21 – Validação das Soluções - Parte 1 .....	197
Tabela A.22 – Validação das Soluções - Parte 2 .....	198
Tabela A.23 – Validação das Soluções - Parte 3 .....	199
Tabela A.24 – Critérios de Organização dos Artigos .....	210
Tabela A.25 – Critérios de Qualidade .....	211
Tabela A.26 – Lista de Periódicos .....	212
Tabela A.27 – Lista de Conferências - Parte 1 .....	213
Tabela A.28 – Lista de Conferências - Parte 2 .....	214
Tabela A.29 – Formulário de Extração de Dados .....	215
Tabela A.30 – Aplicação dos critérios de qualidade nos 109 artigos selecionados - Parte 1 .....	216
Tabela A.31 – Aplicação dos critérios de qualidade nos 109 artigos selecionados - Parte 2 .....	217
Tabela A.32 – Aplicação dos critérios de qualidade nos 109 artigos selecionados - Parte 3 .....	218
Tabela B.1 – Levantamento das questões e Estruturação do Guia para Entrevista	227
Tabela B.2 – Reuniões para Revisão do Guia para a Entrevista .....	227
Tabela B.3 – Validação de Face e Conteúdo do Guia para Entrevista .....	228
Tabela B.4 – Pré-Teste (Entrevista Piloto) .....	228
Tabela B.5 – Autorização das Empresas Participantes (via e-mail) .....	228
Tabela B.6 – Aplicação das Entrevistas .....	228
Tabela C.1 – Levantamento das questões e Estruturação do Guia para Entrevista	233
Tabela C.2 – Reuniões para Revisão do Guia para a Entrevista .....	233
Tabela C.3 – Validação de Face e Conteúdo do Guia para Entrevista .....	234
Tabela C.4 – Pré-Teste (Entrevista Piloto) .....	234
Tabela C.5 – Autorização da Empresa Participante .....	234
Tabela C.6 – Imersão na Organização das Entrevistas .....	234
Tabela C.7 – Aplicação das Entrevistas .....	235





## LISTA DE SIGLAS

- AG – Algoritmos Genéticos
- AM – Aprendizagem de Máquina
- APF – Análise de Pontos de Função
- CMMI – *Capability Maturity Model Integration*
- COCOMO – *Constructive Cost Model*
- EDGE – *Enabling Delivery and Global Excellence*
- LOC – *Lines of Code*
- RBC – Raciocínio Baseado em Casos
- RNA – Rede Neural Artificial
- SLIM – *Software Lifecycle Management*
- UCP – *Use Case Points*
- UML – *Unified Modeling Language*



# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b> .....	<b>29</b>
1.1	MOTIVAÇÃO .....	30
1.2	OBJETIVOS .....	31
1.3	ORGANIZAÇÃO DO VOLUME .....	32
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b> .....	<b>33</b>
2.1	REENGENHARIA DE SOFTWARE .....	33
2.1.1	ABORDAGENS DE REENGENHARIA DE SOFTWARE .....	34
2.1.2	PROCESSO DE REENGENHARIA DE SOFTWARE .....	35
2.2	ESTIMATIVA DE ESFORÇO .....	37
2.2.1	PROCESSO DE ESTIMATIVA DE ESFORÇO .....	38
2.2.2	TÉCNICAS DE ESTIMATIVA DE ESFORÇO .....	39
2.3	ESTIMATIVA DE ESFORÇO EM REENGENHARIA DE SOFTWARE: TRABALHOS RELACIONADOS .....	46
2.3.1	ABORDAGEM DE SNEED 2005 .....	47
2.3.2	ABORDAGEM DE BALDASSARRE <i>ET AL.</i> [2003], CAIVANO <i>ET AL.</i> [2006] E BALDASSARRE <i>ET AL.</i> [2006] .....	50
2.3.3	ABORDAGEM DE BOEHM ET AL. [1995] .....	52
2.4	ANÁLISE CRÍTICA .....	53
2.5	CONCLUSÕES DO CAPÍTULO .....	54
<b>3</b>	<b>METODOLOGIA DE PESQUISA</b> .....	<b>55</b>
3.1	DESENHO E ETAPAS DA PESQUISA .....	55
3.2	ASPECTOS METODOLÓGICOS .....	57
3.3	METODOLOGIA DO ESTUDO DE CAMPO INICIAL .....	59
3.3.1	PLANEJAMENTO .....	59
3.3.2	COLETA DOS DADOS .....	61
3.3.3	ANÁLISE DOS DADOS .....	63
3.4	METODOLOGIA DO ESTUDO DE CASO .....	65
3.4.1	PLANEJAMENTO .....	66
3.4.2	ELABORAÇÃO DO ROTEIRO DE ENTREVISTAS .....	68
3.4.3	COLETA DE DADOS .....	69

3.4.4	ANÁLISE DOS DADOS .....	70
3.5	CONCLUSÕES DO CAPÍTULO .....	70
<b>4</b>	<b>APLICAÇÃO E RESULTADOS DO ESTUDO DE CAMPO INICIAL .....</b>	<b>73</b>
4.1	CARACTERIZAÇÃO DAS ORGANIZAÇÕES E DOS ENTREVISTADOS .....	73
4.2	RESULTADOS .....	74
4.2.1	REENGENHARIA DE SOFTWARE DO PONTO DE VISTA DAS ORGANIZAÇÕES .....	75
4.2.2	PROCESSO DE ESTIMATIVA DE ESFORÇO EM REENGENHARIA .....	77
4.2.3	FATORES QUE IMPACTAM NA ESTIMATIVA DE ESFORÇO EM PROJETOS DE REENGENHARIA DE SOFTWARE .....	79
4.3	CONCLUSÕES DO CAPÍTULO .....	88
<b>5</b>	<b>APLICAÇÃO E RESULTADOS DO ESTUDO DE CASO .....</b>	<b>89</b>
5.1	CONFIGURAÇÃO DO ESTUDO E MÉTODO DE PESQUISA .....	89
5.2	PROCESSO DE REENGENHARIA .....	89
5.3	CARACTERIZAÇÃO DOS PROJETOS ANALISADOS .....	91
5.4	CARACTERIZAÇÃO DOS RESPONDENTES E SUA PARTICIPAÇÃO .....	92
5.5	ELEMENTOS DE ANÁLISE .....	92
5.5.1	PROCESSO DE ESTIMATIVA DE ESFORÇO .....	92
5.5.2	FATORES QUE IMPACTAM NA ESTIMATIVA DE ESFORÇO EM PROJETOS DE REENGENHARIA DE SOFTWARE .....	94
5.6	CONCLUSÕES DO CAPÍTULO .....	105
<b>6</b>	<b>CONSOLIDAÇÃO DOS RESULTADOS DOS ESTUDOS EMPÍRICOS .....</b>	<b>107</b>
6.1	ATIVIDADES DE ESTIMATIVA .....	107
6.2	DESAFIOS .....	108
6.3	BOAS PRÁTICAS .....	110
6.4	LIÇÕES APRENDIDAS .....	113
6.5	CONCLUSÕES DO CAPÍTULO .....	117
<b>7</b>	<b>MODELO DE ESTIMATIVA DE ESFORÇO PARA PROJETOS DE REENGENHARIA DE SOFTWARE .....</b>	<b>119</b>
7.1	ESTRUTURA DO MODELO .....	119
7.1.1	MAPEAMENTO DOS DESAFIOS, BOAS PRÁTICAS E LIÇÕES APRENDIDAS DE ESTIMATIVA EM PROJETOS DE REENGENHARIA PARA O MODELO PROPOSTO .....	120

7.2	PLANEJAMENTO ESTRATÉGICO .....	122
7.2.1	DEFINIR/ATUALIZAR ESTRATÉGIA DE ARMAZENAMENTO DE DADOS HISTÓRICOS .....	124
7.2.2	DEFINIR/ATUALIZAR ESTRATÉGIA DE GERENCIAMENTO DE CONHECIMENTO .....	125
7.2.3	DEFINIR/ATUALIZAR A ESTRATÉGIA DE PREPARAÇÃO DA EQUIPE DE ESTIMATIVA .....	125
7.2.4	DEFINIR/ATUALIZAR PLANO DE TRATAMENTO E MITIGAÇÃO DE RISCOS ..	126
7.2.5	CONSOLIDAR PLANEJAMENTO ESTRATÉGICO DE ESTIMATIVA DE ESFORÇO .....	127
7.3	PLANEJAMENTO DA ESTIMATIVA .....	127
7.3.1	ANALISAR DEMANDA .....	129
7.3.2	CONSULTAR BASE DE CONHECIMENTO .....	129
7.3.3	SELECIONAR TÉCNICAS DE ESTIMATIVA .....	129
7.3.4	DEFINIR OS DADOS A SEREM COLETADOS .....	129
7.3.5	DEFINIR AS TÉCNICAS DE COLETA DOS DADOS .....	130
7.3.6	GERAR PLANEJAMENTO DAS ESTIMATIVAS .....	130
7.4	ESTIMATIVA .....	131
7.4.1	LEVANTAMENTO DE DADOS E ANÁLISE DE RISCOS .....	131
7.4.2	ESTIMATIVA .....	134
7.4.3	COMPARAÇÃO E VALIDAÇÃO .....	134
7.5	MONITORAMENTO E CALIBRAGEM .....	135
7.5.1	MONITORAMENTO .....	136
7.5.2	CALIBRAGEM .....	137
7.5.3	ATUALIZAÇÃO DA BASE .....	137
7.6	APRENDIZAGEM .....	137
7.6.1	AVALIAR AS ESTRATÉGIAS ADOTADAS E AVALIAR AS MÉTRICAS DO PROJETO .....	139
7.6.2	CONSOLIDAR AS LIÇÕES APRENDIDAS E INCLUIR LIÇÕES APRENDIDAS NA BASE CONHECIMENTO .....	139
7.7	MODELO DE ESTIMATIVA DE ESFORÇO EM DIFERENTES CONTEXTOS DE PROCESSOS DE REENGENHARIA .....	139
7.7.1	PROJETOS DE ESCOPO FECHADO .....	141
7.7.2	PROJETOS DE ESCOPO ABERTO .....	141
7.8	CONCLUSÕES DO CAPÍTULO .....	142

<b>8</b>	<b>CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS</b>	<b>143</b>
8.1	DISCUSSÃO DOS RESULTADOS	143
8.2	CONTRIBUIÇÕES	144
8.3	LIMITAÇÕES E AMEAÇAS A VALIDADE	144
8.4	TRABALHOS FUTUROS	145
	<b>REFERÊNCIAS</b>	<b>147</b>
	<b>APÊNDICE A – REVISÃO SISTEMÁTICA: SOLUÇÕES PARA ESTIMATIVA DE ESFORÇO EM PROJETOS DE DESENVOLVIMENTO DE SOFTWARE</b>	<b>153</b>
A.1	INTRODUÇÃO	153
A.2	ESTIMATIVA DE ESFORÇO EM DESENVOLVIMENTO DE SOFTWARE	155
A.2.1	TÉCNICAS DE ESTIMATIVA DE ESFORÇO	155
A.2.2	TRABALHOS RELACIONADOS	162
A.2.3	PROTOCOLO	163
A.2.4	AVALIAÇÃO DO PLANEJAMENTO	169
A.2.5	CRONOGRAMA	169
A.3	EXECUÇÃO DA FASE I	170
A.3.1	BUSCA POR STRING	170
A.3.2	BUSCA MANUAL	172
A.3.3	ARTIGO DE CONTROLE	172
A.3.4	ARTIGOS RELACIONADOS	173
A.4	EXECUÇÃO DA FASE II	176
A.4.1	AVALIAÇÃO DE QUALIDADE	176
A.4.2	EXTRAÇÃO DOS DADOS	177
A.4.3	MOTOR DE BUSCA E BUSCA MANUAL	177
A.4.4	ANO	178
A.4.5	TIPO DE PUBLICAÇÃO	179
A.4.6	DEPARTAMENTOS, INSTITUTOS E FACULDADES	181
A.4.7	AUTORES	183
A.5	RESULTADOS	186
A.5.1	VISÃO GERAL DOS ESTUDOS	186
A.5.2	TÉCNICAS DE ESTIMATIVA DE ESFORÇO	191
A.5.3	DOMÍNIO DE DESENVOLVIMENTO	193
A.5.4	VALIDAÇÃO DE SOLUÇÕES	195

A.6	CONSIDERAÇÕES FINAIS .....	200
A.6.1	LIMITAÇÕES .....	201
A.6.2	CONTRIBUIÇÃO .....	201
A.7	REFERÊNCIAS BIBLIOGRÁFICAS .....	202
A.8	CRITÉRIOS USADOS PARA ORGANIZAR OS ARTIGOS SELECIONADOS PELAS STRINGS DE BUSCA E PELA BUSCA MANUAL, FASE I .....	210
A.9	CRITÉRIOS DE QUALIDADE USADOS NOS ARTIGOS SELECIONADOS PARA A FASE II .....	211
A.10	LISTAGEM DAS CONFERÊNCIAS E PERIÓDICOS .....	212
A.11	FORMULÁRIO DE EXTRAÇÃO DOS DADOS .....	215
A.12	APLICAÇÃO DOS CRITÉRIOS DE QUALIDADE NOS 109 ARTIGOS SELECI- ONADOS .....	216
A.13	LISTAGEM DOS ARTIGOS SELECIONADOS E LIDOS NA FASE II .....	219

**APÊNDICE B – PROTOCOLO PARA ESTUDO DE CAMPO: ESTIMATIVA DE  
ESFORÇO EM PROJETOS DE REENGENHARIA DE SOFTWARE .....** 227

B.1	OBJETIVO DO ESTUDO DE CAMPO .....	227
B.2	ORGANIZAÇÃO DO PROTOCOLO .....	227
B.2.1	PROCEDIMENTOS .....	227
B.2.2	ESCOLHA DAS PESSOAS ENTREVISTADAS .....	227
B.2.3	RECURSOS UTILIZADOS .....	228
B.2.4	QUESTÕES DE PESQUISA .....	229
B.2.5	METODOLOGIA .....	229
B.2.6	VALIDAÇÃO DO ROTEIRO .....	229
B.2.7	SELEÇÃO DOS PARTICIPANTES DA PESQUISA .....	230
B.2.8	APLICAÇÃO DA ENTREVISTA .....	230
B.2.9	ANÁLISE .....	230
B.2.10	ROTEIRO DE ENTREVISTA: FASE I .....	230
B.2.11	ROTEIRO DE ENTREVISTA: FASE II .....	232

**APÊNDICE C – PROTOCOLO PARA ESTUDO DE CASO: ESTIMATIVA DE  
ESFORÇO EM PROJETOS DE REENGENHARIA DE SOFTWARE .....** 233

C.1	VISÃO GERAL DO ESTUDO DE CASO .....	233
C.2	ORGANIZAÇÃO DO PROTOCOLO .....	233
C.2.1	PROCEDIMENTOS .....	233
C.2.2	ESCOLHA DAS PESSOAS ENTREVISTADAS .....	234

C.2.3	RECURSOS UTILIZADOS .....	234
C.2.4	DIMENSÕES, QUESTÕES E PROVÁVEL FONTE DE COLETA DE DADOS DO GUIA PARA ENTREVISTA SEMI-ESTRUTURADA.....	235

## 1. INTRODUÇÃO

Estimativa precisa dos custos do projeto é um pré-requisito essencial para fazer um projeto de reengenharia de software [SNE95]. Entende-se reengenharia como sendo a renovação ou reconstrução de software, e envolve a análise e mudança de um sistema de software para reconstituí-lo em uma nova forma [CC90]. O objetivo é entender o software existente (especificação, projeto, implementação) e reimplementá-lo para melhorar a funcionalidade, o desempenho ou a implementação do sistema [RH96]. Os sistemas existentes geralmente passam por reengenharia por ser mais barato do que a manutenção ou substituição dos mesmos [PRE11] [SOM07]. No entanto, para tomar esta decisão, a organização deve saber o quanto a reengenharia vai custar e quando esse investimento será visível [SNE05].

No início da década de 90 surgiram os primeiros estudos sobre economia de projetos de reengenharia de software [SNE91] e, desde então, um grande número de pesquisas tem sido feitas e várias experiências práticas obtidas. Antigamente, o esforço em projetos de reengenharia era calculado apenas com base no tamanho do sistema, sem levar em conta a complexidade e qualidade do mesmo [VIS01]. Com o crescimento e amadurecimento da área tornou-se importante levar em conta os fatores de complexidade e qualidade nos custos de reengenharia de software. Reengenharia é uma alternativa para a manutenção, ou compra de um pacote padrão ou a não fazer nada, portanto, é importante para a gestão da organização saber qual o retorno sobre o investimento para cada alternativa [SNE05].

Um estudo realizado por Bergey et al. [BST+05] concluiu que mais de 50% dos projetos de reengenharia falham. As razões para o fracasso em projetos de reengenharia incluem: o uso de uma estratégia inadequada, requisitos mal coletados, planejamento inadequado e fracasso para dar conta da complexidade do sistema a ser reconstruído.

Estimar adequadamente o esforço para a realização da reengenharia é uma atividade crucial durante o planejamento da mesma, pois conecta medições para custo e cronograma do projeto. A determinação do esforço necessário para a realização do projeto permite planejar adequadamente quaisquer futuras atividades e é realizada por uma variedade de razões como: a seleção de projetos, planejamento de recursos, programação, monitoramento de *status*, avaliação de desempenho da equipe, dentre outras [PRE11], [SOM07].

Apesar de existirem diversas soluções bem consolidadas para apoio à realização de estimativas de esforço em projetos de desenvolvimento de software, poucas dessas soluções são apropriadas especificamente para o contexto de reengenharia de software. E mesmo as soluções existentes não são voltadas para formalização de um processo bem consolidado, execução e manutenção deste processo, e sim para cálculo da estimativa.

No que diz respeito a literatura na área, foram desenvolvidos estudos no contexto de estimativa de esforço em projetos de reengenharia, como [BCV03], [BBCV06], [CLV06], [BCH+95] e [SNE05]. Estes estudos, porém, tratam de apresentar soluções para realização de estimativa de esforço, de modo que não foram identificados estudos que apliquem uma abordagem qualitativa para entendimento do fenômeno de estimativa de esforço em reengenharia.

## 1.1 Motivação

Em um estudo realizado pelo *The Standish Group* em 1995, denominado de *Chaos Report*, cujo foco foi a indústria de software comercial, foi detectado que 31,1% dos projetos de software foram cancelados antes mesmo de serem concluídos e 52,7% custaram 189% a mais do que o previsto inicialmente, enquanto que apenas 16,2% dos projetos foram completados no tempo e custo estimados. Porém, muitos destes projetos entregues não refletem o que foi especificado, uma vez que apenas 42% dos requisitos originais foram desenvolvidos.

Embora essa pesquisa tenha sido realizada na década de 90, ainda hoje estes números são reais na indústria de software. O *Chaos Manifesto* de 2013 identifica que apesar de 39% dos projetos tenham sido entregues no prazo, orçamentos e tido os recursos necessários para as funções, 43% dos projetos enfrentaram situações opostas a esta, pois estão atrasados, acima do orçamento e/ou sem os recursos necessários para sua conclusão e 18% foram cancelados antes do término ou entregues e nunca utilizados [STA13].

Dentre os fatores identificados pelo *The Standish Group* [STA13] como sendo determinantes do insucesso do projeto está predominantemente o estouro do tempo e/ou do orçamento. Os dados mostram que em 2012 cerca de 74% dos projetos ultrapassaram o tempo previsto, enquanto 59% ultrapassaram o orçamento.

Estes dados de projetos de software reais evidenciam o que a literatura acerca de Gerenciamento de Projetos já afirma amplamente: que a estimativa de esforço de desenvolvimento de software representa uma importante área de pesquisa e é uma das tarefas mais relevantes no planejamento e gerenciamento de processo de desenvolvimento de software [SOM07] [PRE11] [PFL04].

Dada a relevância da realização de estimativas de esforço, muitas soluções foram propostas no intuito de auxiliar este processo. Estas soluções envolvem modelos algorítmicos como, por exemplo, COCOMO [BOE84] [BCH+95] e SLIM [PUT78], Julgamento de Especialistas [TMKiM12] [JS07] [JØR05], Regressão Estatística [NSB08] [SYB08] [TRR08], Modelos Dinâmicos e soluções baseadas em aprendizagem de máquina, que incluem, Árvores de Regressão e Classificação [BBdSdC13] [BBR12], Lógica Fuzzy [KJZ+11][ZN13],

Redes Neurais Artificiais [AO10] [KTB08], Algoritmos Genéticos [SMLE02], Estimativa por Analogia [SS97] [WLT09], dentre outras.

Com o avanço da tecnologia, muitas mudanças estão acontecendo na indústria de software e, portanto, as aplicações de software têm que estar em sintonia com as mais recentes tecnologias e tendências para permanecer competitivas no mercado. A exigência de manter o ritmo do mercado e cumprir requisitos adicionais demonstra a necessidade de reengenharia de sistemas, onde o software é transformado de um estado para outro estado.

Um processo de reengenharia software tem características que implicam em novas exigências do modelo de estimativa, como componentes de software mais detalhados (ou seja, código-fonte, documentação de projeto) do que aqueles normalmente utilizados nos processos de desenvolvimento (requisitos de sistema e documentos de análise) [PRE11]. Além disso, inclui atividades que dependem de parâmetros de qualidade e complexidade do sistema legado [VIS01]. Nas comunidades científica e industrial os processos de reengenharia têm uma história mais recente em comparação ao desenvolvimento e manutenção ordinária. Por estas razões, os projetos de reengenharia têm maiores riscos em termos de pessoas, programação e custos [BCV03]. Assim, o monitoramento intenso durante a execução do projeto deve ser realizado a fim de gerenciar os riscos de subestimação e superestimação [VIS01].

No contexto de apoio a realização de estimativa de esforço em projetos de reengenharia, nota-se a escassez de trabalhos que relacionem propostas de apoio a realização de estimativas de esforço em projetos de reengenharia de software concebidas a partir da realização deste processo na prática, de maneira a aproximar a solução proposta da realidade organizacional.

Assim, este trabalho atuou junto a esta lacuna, propondo um modelo para apoiar a realização de estimativas de esforço em projetos de reengenharia de software, baseado na literatura e em boas práticas e lições aprendidas acerca de estimativa aplicadas em projetos reais.

## **1.2 Objetivos**

Este trabalho teve como objetivo geral propor um modelo para apoiar a realização de estimativas de esforço em projetos de reengenharia de software, que leve em consideração o processo de estimativa como um todo e não apenas o cálculo da estimativa. Os objetivos específicos relacionados foram:

1. Levantamento de base teórica em Reengenharia de Software e Estimativa de Esforço;
2. Levantamento de dados sobre a realização de estimativa de esforço em reengenharia de software, a partir da realização de um estudo empírico;

3. Elaboração de um modelo para realização de estimativa de esforço em projetos de reengenharia de software, com base na consolidação dos dados obtidos no estudo empírico.

### **1.3 Organização do Volume**

Este trabalho está dividido em 8 capítulos, organizados da seguinte maneira:

- Capítulo 1: apresenta a introdução, motivação e objetivos do trabalho.
- Capítulo 2: apresenta a fundamentação teórica da pesquisa.
- Capítulo 3: apresenta o desenho da pesquisa e a metodologia aplicada para realização do estudo empírico.
- Capítulo 4: apresenta a realização e os resultados do estudo de campo inicial.
- Capítulo 5: apresenta a realização e os resultados do estudo de caso.
- Capítulo 6: apresenta a consolidação dos resultados dos estudos.
- Capítulo 7: apresenta a proposta do modelo de estimativa de esforço para projetos de reengenharia de software.
- Capítulo 8: apresenta as considerações finais do trabalho.

## 2. FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta a base teórica referente aos principais conceitos relacionados a esta pesquisa. Na Seção 2.1 são abordados os conceitos referentes ao processo de Reengenharia de Software: definição, abordagens e modelo de processo. Na Seção 2.2 a estimativa de esforço é apresentada, juntamente com as principais técnicas adotadas. Por fim, trabalhos relacionados são apresentados na Seção 2.3.

### 2.1 Reengenharia de Software

Software evolui independente do domínio da aplicação, tamanho ou complexidade. As mudanças dirigem esse processo. No âmbito de software ocorrem alterações quando são corrigidos erros, quando há adaptação a um novo ambiente, quando o cliente solicita novas características ou funções e quando a aplicação passa por um processo de reengenharia para proporcionar benefício em um contexto moderno [PRE11].

Quando o sistema não é fácil de ser mantido sendo, porém, de grande utilidade, ele deve ser reconstruído. Partindo-se do sistema existente (via código-fonte, interface ou ambiente), são abstraídas as suas funcionalidades e são construídos o modelo de análise e projeto do software. Este processo é denominado reengenharia de software [PQ00].

A reengenharia de software está relacionada à reimplantação de sistemas legados para torná-los mais fácil de manter. A reengenharia pode envolver uma nova documentação, organização e reestruturação do sistema, conversão do sistema em uma linguagem mais moderna, modificação e atualização da estrutura e dos valores dos dados de sistema [SOM07].

Chikofsky *et al.* [CC90] definem a reengenharia como sendo a renovação ou reconstrução de software, e envolve a análise e mudança de sistema de software para reconstruí-lo de uma nova forma. O objetivo é entender o software existente (especificação, projeto, implementação) e reimplantá-lo para melhorar a funcionalidade, o desempenho ou a implementação do sistema [RH96].

A distinção principal entre reengenharia e desenvolvimento de um novo software é o ponto de partida para o desenvolvimento. Em vez de começar com uma especificação escrita, o sistema antigo funciona como uma especificação para o sistema novo [PFL04].

## 2.1.1 Abordagens de Reengenharia de Software

### 1. Big Bang

Apresentada por [BIG89] a abordagem “big bang”, substitui todo o sistema de uma só vez. Esta abordagem é frequentemente utilizada por projetos que precisam resolver um problema imediatamente, como a migração para uma arquitetura de sistema diferente.

A vantagem desta abordagem é que o sistema é levado a um novo ambiente de uma só vez. Nenhuma interface entre os componentes antigos e novos deve ser desenvolvida. A desvantagem desta abordagem é que o resultado tende a ser de projetos monolíticos que podem não ser sempre adequados. Para grandes sistemas, esta abordagem pode consumir muitos recursos ou exigir grande quantidade de tempo até que o sistema alvo seja produzido. O risco com esta abordagem é alto, o sistema “novo” tem que ter seu funcionalmente intacto e funcionar em paralelo com o sistema antigo para garantir a funcionalidade. Esta operação em paralelo pode ser difícil e cara de se fazer. Uma das maiores dificuldades é o controle de mudança, entre o tempo que o novo sistema é iniciado e terminado, muitas mudanças poderão ser feitas no sistema antigo, e isto tem de ser refletido no novo sistema [BIG89] [BRO95].

### 2. Abordagem Incremental

Na abordagem incremental [BRO95] [BCV03], ou iterativa, os módulos do sistema passam pela reengenharia e são adicionados gradualmente à medida que novas versões do sistema são geradas. O projeto é dividido em módulos de reengenharia com base em seções do sistema existente.

As vantagens desta abordagem são que os componentes do sistema são produzidos mais rapidamente e é mais fácil de detectar erros uma vez que os novos componentes estão claramente identificados. Como as versões intermediárias são liberadas, o cliente pode ver o progresso e identificar rapidamente uma funcionalidade perdida. Outra vantagem é que a mudança para o sistema antigo pode ser mais fácil de tratar, uma vez que alterações nos componentes que não estão sob a reengenharia não têm impacto sobre o componente atual.

Uma desvantagem para a abordagem incremental é que o sistema demora mais tempo para ser concluído com a versões provisórias múltiplas que requerem controle de configuração cuidadoso. Outra desvantagem é que a estrutura inteira do sistema não pode ser alterada, apenas a estrutura dentro dos módulos específicos. Isso requer cuidadosa identificação de componentes no sistema existente e extenso planejamento da estrutura do sistema de destino. Esta abordagem tem um risco menor do que o *Big*

*Bang* porque, como cada componente passa pela reengenharia, os riscos na parte de código podem ser identificados e monitorados.

### 2.1.2 Processo de Reengenharia de Software

Diversos autores apresentam seus modelos de processo de reengenharia de sistemas [PRE11] [SOM07] [PFL04] [RH96] [SNE95] [PP01]. Tais processos se diferenciam por denominações das etapas principais e/ou supressão e contração de etapas. Porém, o cerne o processo é o mesmo e pode ser sintetizado nas seguintes etapas, apresentadas na Figura 2.1:

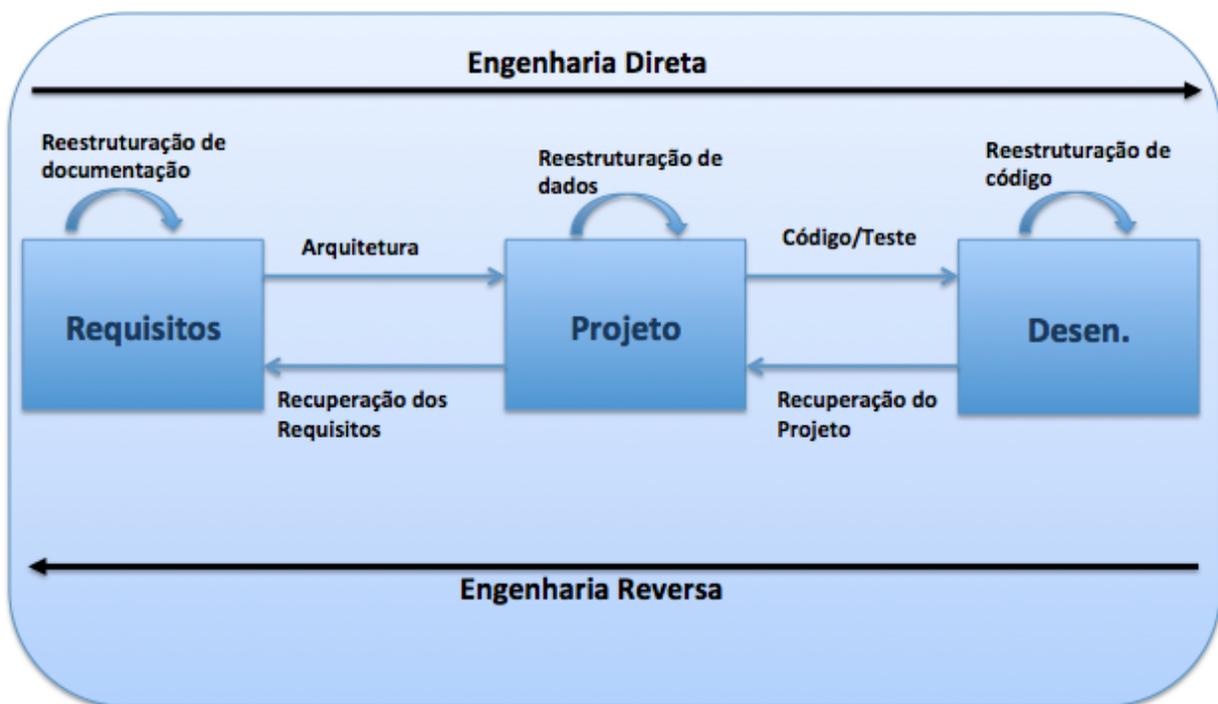


Figura 2.1 – Processo Reengenharia de Software (Adaptado de [PP01])

**Engenharia Reversa:** é o processo de análise de um sistema para identificação dos componentes deste sistema e suas inter-relações, com objetivo de criar representações do sistema em outra forma ou em um nível mais alto de abstração. Na engenharia reversa, os requisitos, projeto essencial, a estrutura e o conteúdo do sistema legado devem ser recapturados. Além de capturar as relações técnicas e interações, informações sobre a aplicação de regras de negócio e processos que provaram ser úteis na gestão do negócio também deve ser recuperados. Os principais objetivos da engenharia reversa são: gerar visões alternativas, recuperar informações perdidas, detectar efeitos colaterais e facilitar a reutilização. A eficácia deste processo irá afetar o sucesso do projeto de reengenharia. A

engenharia reversa não envolve mudanças no sistema ou a criação de um novo sistema, ela é um processo de exame, sem alterar a sua funcionalidade geral.

A Engenharia Reversa envolve os sub-processos de Reestruturação dos Documentos (Redocumentação), Recuperação do Projeto e Reestruturação do Código:

1. Reestruturação dos Documentos: documentação pobre é a marca registrada de muitos sistemas legados [PRE11]. Neste caso, pode aplicar uma das seguintes alternativas, dependendo do contexto da organização:
  - (a) Criar documentação consome muito tempo: se o sistema funciona, pode-se optar por conviver com o que se tem. Em alguns casos, essa atitude é correta. Não é possível recriar documentação para centenas de programas de computador. Se um programa for relativamente estático, está chegando ao fim de sua vida útil, e provavelmente não terá uma modificação significativa.
  - (b) A documentação tem que ser atualizada, mas a organização apresenta recursos limitados: deve-se empregar uma abordagem “documentar quando usar”. Pode não ser necessário redocumentar totalmente o aplicativo. Em vez disso, partes do sistema que estão passando por alteração são completamente documentadas.
  - (c) O sistema é crítico para o negócio e deve ser totalmente documentado: mesmo neste caso, uma abordagem inteligente é limitar a documentação a um mínimo essencial.
2. Reestruturação do Código: alguns sistemas legados tem uma arquitetura de programa razoavelmente sólida, mas os módulos individuais foram codificados de um modo que se torna difícil de entendê-los, testá-los e mantê-los. Nesses casos, o código dentro dos módulos suspeitos pode ser reestruturado.

Para executar essa atividade, o código fonte é analisado por meio de uma ferramenta de reestruturação. As violações das construções de programação estruturada são registradas, e o código é então reestruturado ou mesmo reescrito em uma linguagem de programação mais moderna. O código reestruturado resultante é revisado e testado para garantir que não tenham sido introduzidas anomalias. A documentação interna do código é atualizada.
3. Reestruturação de Dados: diferentemente da reestruturação de código, que ocorre em um nível relativamente baixo de abstração, a reestruturação de dados é uma atividade de reengenharia em escala completa. A arquitetura de dados atual é dissecada, e os modelos de dados necessários são definidos. Identificam-se os objetos de dados e atributos, e as estruturas de dados existentes são revisadas quanto à qualidade.

**Engenharia Direta:** também chamada de engenharia progressiva, ocorre após o nível de abstração desejado ser atingido (durante a engenharia reversa). Engenharia direta

corresponde exatamente ao processo normal de desenvolvimento de software, a partir do nível de abstração alcançado no processo de engenharia reversa. Ou seja, se o software está sendo redesenhado para atender a uma nova arquitetura global do sistema, o processo de engenharia reversa deve extrair os requisitos de software, e o processo de engenharia direta iria começar com o desenvolvimento de um novo *design*. Durante esta fase, garantia de qualidade e disciplinas de gerenciamento de configuração devem ser aplicadas. Técnicas de medição também devem ser aplicadas, para avaliar a melhoria do software e para identificar potenciais áreas de risco.

Apesar de não estar claramente definida no processo da Figura 2.1, a Reengenharia envolve ainda a etapa de Planejamento. Segundo [SNE95], o planejamento é a etapa que mais impacta no sucesso do projeto de reengenharia, devido às decisões cruciais que são tomadas nesta fase. O planejamento inclui análise de viabilidade do projeto (determinar o retorno do investimento, avaliar as necessidades e objetivos que o sistema existente atende, dentre outras), análise do portfólio (priorizar as aplicações a serem submetidas ao processo de reengenharia de acordo com a qualidade técnica e o valor que agregam ao negócio), estimativa de custo/esforço (estimar o esforço e o orçamento que serão empregados para a realização da reengenharia do sistema), análise de custo benefício (comparação das estimativas com o custo esperado) e contratação (estabelecer tarefas e distribuir o esforço de desenvolvimento).

Apesar de reengenharia ser muitas vezes usada como um meio de reduzir os riscos e reduzir os custos de operação e manutenção do software legado, reengenharia não é um processo sem riscos [RH96]. O planejamento auxilia os gerentes de projeto na identificação precoce de riscos e na preparação para a estimativa e avaliação de riscos de reengenharia, além de fornecer um quadro realista de expectativas. A identificação dos riscos é essencial para a avaliação de riscos, análise de riscos eficaz e gestão de riscos.

## 2.2 Estimativa de Esforço

A gestão bem sucedida de projetos de software começa com uma estimativa precisa do esforço de desenvolvimento [PRE11] [SOM07]. Imprecisão nas estimativas continua a ser um dos fatores-chave que contribuem para falhas de projeto de software [STA13]. Subestimar o esforço a ser gasto gera pressões que comprometem a qualidade do desenvolvimento do projeto, enquanto que, superestimar pode elevar o custo e diminuir a competitividade.

Técnicas de estimativa de esforço para projeto de software são projetadas para converter uma estimativa de tamanho (como um número estimado de linhas de código-fonte) para uma estimativa de esforço (pessoas por hora, dia ou mês).

A previsão exata tem sido difícil, já que muitos desses relacionamentos não são bem compreendidos. Melhorar as técnicas de estimativa disponíveis para gerentes de projeto facilitaria o controle mais eficaz do tempo e os orçamentos no desenvolvimento de software.

### 2.2.1 Processo de Estimativa de Esforço

O propósito da estimativa de esforço como parte do gerenciamento do projeto de software é prever a quantidade de esforço requerido para completar o conjunto de tarefas que compõe o ciclo de vida do projeto. Esta predição é baseada em um conjunto de entradas tais como, conhecimento/dados de projetos anteriores similares e características que os estimadores acreditam ser relacionadas com o esforço [MEN14].

A realização da estimativa de esforço é representada pelo processo da Figura 2.2, adaptada de [MEN14]. Tal processo foi proposto no contexto de Desenvolvimento de Aplicações Web, mas representa as atividades de estimativa de esforço para desenvolvimento de software no geral, tal como descrito em [PRE11], [SOM07], dentre outros.

1. **Requisitos do Sistema:** representam qualquer conjunto de requisitos (funcionais e não-funcionais) que são obtidos a partir da Elicitação realizada com o cliente. Eles podem ser detalhados usando linguagem natural ou outra notação, como UML (*Unified Modeling Language*).
2. **Estimativa de Tamanho:** representa a estimativa relacionada ao tamanho do “problema” a ser resolvido, que uma vez implementado representará um sistema de software entregável. Essa estimativa pode ser representada, por exemplo, em contagem de Linhas de Código (LoC - *Lines of Code*) ou Análise de Pontos de Função (APF).
3. **Fatores de Custo:** representada fatores não relacionados ao tamanho, mas que acredita-se estarem associados com o esforço, uma vez que eles afetam a quantidade total de esforço necessário para desenvolver o sistema. Exemplos de fatores de custo são: tamanho do time de desenvolvimento, experiência do time no negócio, uso ou não de ferramentas de apoio.
4. **Dados e/ou Conhecimento de Projetos Passados Finalizados:** representam (a) dados concretos sobre os últimos projetos concluídos, recolhidos pela empresa; (b) conhecimento de especialistas (gerentes de projeto e desenvolvedores) obtido em projetos que são de alguma forma similares ao projeto que necessita ter o esforço estimado.
5. **Esforço Estimado:** o total de esforço (geralmente medido em pessoas/hora) que será necessário para completar o projeto e entregar o sistema.

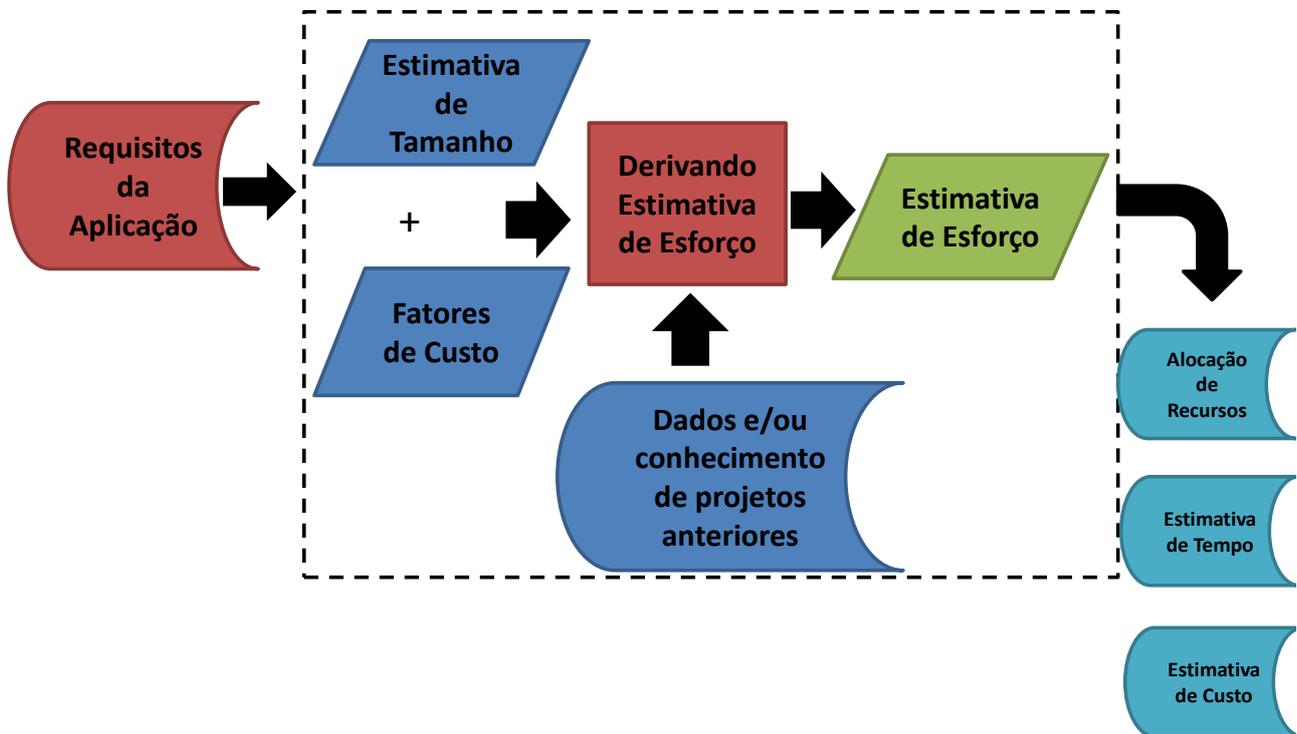


Figura 2.2 – Processo de Estimativa de Esforço (Adaptado de [MEN14])

6. Recursos Alocados: representa o processo de decisão acerca dos recursos (humanos, ambientais, ferramentas) que precisarão ser alocados para o projeto, como resultado do esforço estimado.
7. Duração Estimada: representa a estimativa do tempo necessário para que o projeto seja concluído.
8. Estimativa de Custo: representa o custo do projeto, que é baseado no esforço estimado, mais custos de contingência e de lucros.

### 2.2.2 Técnicas de Estimativa de Esforço

Dada a relevância da estimativa de esforço para o desenvolvimento de software, diversas técnicas foram propostas no decorrer dos anos com o objetivo de apoiar este processo. Com isto, também foram propostas diversas formas de classificação para estas técnicas, como por exemplo [MEN14]. Tendo em vistas as diversas opções existentes, este trabalho adota a taxonomia sugerida por [BAC00] por entender que esta, além de

ser a primeira proposta de taxonomia na área, é a mais adequada ao estado da arte das estimativas classificando, se não todas, grande parte das técnicas existentes na literatura. Assim, as técnicas de estimativa são classificadas como se segue:

1. Baseadas em Modelo;
2. Dinâmicas;
3. Baseadas em Expertise;
4. Baseadas em Regressão;
5. Orientadas a Aprendizagem de Máquina;
6. Compostas.

### Baseadas em Modelo

O final dos anos 70 produziu uma grande variedade de modelos robustos de estimativa como o SLIM [PUT78], Checkpoint [JON91], SEER [JEN83] e COCOMO [BOE84].

Muitos deles são modelos proprietários e, portanto, não podem ser comparados e contrastados, em termos de estrutura do modelo. Teoria ou experimentação determinam a forma funcional destes modelos. Esta seção discute SLIM e COCOMO81, dois dos mais populares modelos conhecidos.

#### 1. SLIM

Larry Putnam desenvolveu o modelo de ciclo de vida de software (*Software Life-Cycle Model* - SLIM) no final de 1970 [PUT78]. O SLIM é baseado na análise do ciclo de vida de Putnam em termos de uma chamada distribuição Rayleigh da equipe do projeto versus tempo. Ele suporta a maioria dos métodos de estimativa de tamanho populares, incluindo linhas de código, pontos de função, etc. Ele faz uso de uma curva de Rayleigh para estimar esforço do projeto, cronograma e taxa de defeito. SLIM pode gravar e analisar dados de projetos previamente preenchidos que são então utilizados para calibrar o modelo, ou se os dados não estão disponíveis, um conjunto de perguntas podem ser respondidas para obter estes valores a partir do banco de dados existente.

#### 2. COCOMO

O modelo de estimativa de custo e cronograma (Constructive Cost Model - COCOMO) foi originalmente publicado em 1984 [BOE84]. Tornou-se um dos mais populares modelos paramétricos de estimativa de custo da década de 1980. Porém, devido à evolução nos processos de desenvolvimento de software, o COCOMO passou a ter dificuldades em estimar os custos de software desenvolvidos para novos processos de ciclo

de vida e competências. O esforço de pesquisa para o COCOMO II foi iniciado em 1994 na Universidade do Sul da Califórnia para abordar as questões sobre modelos não sequenciais e processos rápidos de desenvolvimento, reengenharia, reutilização, abordagens orientadas a objeto, etc. COCOMO II foi publicado em 1995 [BCH<sup>+</sup>95] e tem três submodelos: Composição de Aplicação – mais conveniente para a fase de prototipação em um ciclo de vida espiral - , *Design* precoce – quando os requisitos não são bem conhecidos e a arquitetura do software não foi bem explorada - e Pós-arquitetura de modelos – onde se trabalha com o real desenvolvimento e manutenção do produto de software.

## Dinâmicas

Técnicas dinâmicas reconhecem explicitamente que esforço e fatores de custo mudam ao longo do desenvolvimento do sistema, isto é, eles são dinâmicos (não estáticos) ao longo do tempo.

Esta é uma diferença significativa em relação às outras técnicas, que tendem a confiar em modelos estáticos e previsões com base em “quadros” de uma situação de desenvolvimento em um determinado momento no tempo.

No entanto, fatores como prazos, níveis de pessoal, requisitos de *design*, necessidades de treinamento, orçamento, dentre outros, flutuam ao longo do desenvolvimento e causam flutuações correspondentes na produtividade da equipe do projeto. Isto, por sua vez, tem consequências no prazo e orçamento do projeto.

As técnicas dinâmicas mais importantes são baseadas na abordagem dinâmica de sistemas cujo modelo foi proposto por Jay Forrester em 1968 [FOR68].

## Baseadas em *Expertise*

Técnicas baseadas em *expertise* são úteis na ausência de dados empíricos e quantificados. Elas capturam o conhecimento e a experiência de profissionais dentro de um domínio de interesse, fornecendo estimativas com base em uma síntese dos resultados conhecidos de todos os últimos projetos em que o especialista esteja a par ou em que ele participou. A desvantagem óbvia deste método é que uma estimativa é tão boa quanto o parecer do perito, e não há nenhuma maneira geral para testar essa opinião até que seja tarde demais para corrigir o dano se que a opinião se provar errada. Anos de experiência não necessariamente se traduzem em elevados níveis de competência. Além disso, mesmo a mais competente das pessoas, às vezes, simplesmente estima errado. Duas técnicas foram desenvolvidas para capturar o parecer dos peritos, além de tomar medidas para mitigar a possibilidade de que o julgamento de um especialista seja perdido, estas técnicas são *Delphi* e *Planning Poker*.

1. **Delphi:** a técnica baseada em expertise *Delphi* foi desenvolvido pela *Rand Corporation*, no final da década de 40, como forma de realizar predições de eventos futuros. Com o passar do tempo, tal abordagem passou a ser utilizada como forma de guiar um grupo determinado de indivíduos a um consenso sobre determinado assunto [BAC00]. O *Delphi* mostra-se útil quando há necessidade de estimar valores sem o apoio de grandes volumes de dados empíricos [BAC00]. Embora limitada, a técnica pode servir de apoio a outras técnicas, como foi o caso de sua utilização na calibragem Bayesiana dos dados do COCOMO 2 e na especificação de informações prioritárias e necessárias à calibragem.
2. **Planning Poker:** outra técnica baseada em *expertise* é o *Planning Poker*, largamente utilizado e divulgado pela metodologia ágil Scrum [SB01]. O termo *Planning Poker* foi criado por James Grenning e popularizado por Mike Cohn. O Scrum prega que a atividade de estimar deve ser realizada por todos os membros envolvidos no processo, durante uma reunião de planejamento, que define um conjunto de tarefas prioritárias a serem realizadas em um período de duas a quatro semanas após a reunião: este período de realização das tarefas é denominado *Sprint*. A granularidade utilizada na realização das estimativas para cada tarefa no *Planning Poker*, ao contrário do *Delphi*, baseia-se em uma série numérica que visa limitar o número de escolhas possíveis dos participantes buscando agilidade e diminuição no tempo gasto durante as estimativas. As estimativas se baseiam na série numérica de *Fibonacci*.

Assim como no *Delphi*, o *Planning Poker* onera as equipes de desenvolvimento, pois estas necessitam deslocar seus membros para reuniões onde são estimados os valores empíricos das demandas [TEN10]. Realizar as estimativas de forma empírica pode trazer, dentre outros, dois visíveis problemas: i) falta de precisão nas estimativas de prazos, onde estes quando mal mensurados podem causar aumento da pressão na equipe e, conseqüentemente, desgaste de seus membros por trabalhar em demasia para cumpri-lo; e ii) dependência do conhecimento humano, o que se torna prejudicial para as estimativas de esforço, se a rotatividade de pessoal for alta.

## Baseadas em Regressão

Modelos de regressão estatística estimam o esforço de desenvolvimento de software como uma variável dependente. Tamanho de software (em métricas como linhas de código ou pontos de função) é usado como uma variável independente.

Em alguns modelos, outros parâmetros, como a linguagem de programação de desenvolvimento ou sistema operacional pode ser usado como variáveis independentes adicionais para um modelo de regressão múltipla. Os modelos de regressão tem a vantagem de possuir uma base matemática sólida, bem como medidas de qualidade de ajuste (*goodness fit*), ou seja, quão bem a curva corresponde ao conjunto de dados especificado.

Porém, modelos de regressão estatística são muito suscetíveis ao efeito de *outliers* (itens de dados que podem estar completamente fora de sintonia com o resto do conjunto de dados). Além disso, um modelo de regressão também precisa de um conjunto de dados relativamente grande, o que pode ser um problema no campo de estimativa de software [FWD97].

## Orientadas a Aprendizagem de Máquina

As técnicas para as estimativas de software baseadas em Aprendizagem de Máquina (AM) representam uma alternativa às técnicas tradicionais, as quais são baseadas em regressão ou na *expertise* humana [TEN10].

O método para a criação de modelos de aprendizagem de máquina é a exploração dos dados históricos do domínio, feita por algoritmos específicos, na tentativa de formular ou inferir um conjunto de regras que permitam a dedução de valores futuros [SF95].

Dentre as abordagens de aprendizagem de máquina para o desenvolvimento de modelos preditivos de software destacam-se: Redes Neurais Artificiais, Sistemas *Fuzzy*, Raciocínio Baseado em Casos, Árvores de Classificação e Regressão e Algoritmos Genéticos.

### 1. Redes Neurais Artificiais

De acordo com Gray e McDonell [GM97] redes neurais artificiais é a técnica de construção de modelos de estimativa mais comumente utilizada como uma alternativa à regressão de mínimos quadrados.

As redes neurais artificiais (RNAs) adotam uma abordagem de aprendizagem derivada de um modelo preditivo. A rede é projetada para um conjunto específico de entrada, por exemplo, pontos por função, linguagem de programação, etc, bem como a saída (s), por exemplo, o esforço de desenvolvimento. A rede recebe um conjunto de processos conhecidos (o conjunto de treino) que é utilizado para “treinar” a rede, ou seja, determinar os pesos associados a cada entrada na rede.

Uma vez que a rede está treinada e estável, o esforço de desenvolvimento de um novo caso pode ser previsto, substituindo os valores de entrada relevantes para o caso específico. RNAs são reconhecidas por sua capacidade de fornecer bons resultados quando se trata de problemas onde existem relações complexas entre entradas e saídas, e onde os dados de entrada são distorcidos por altos níveis de ruído [GM97].

Apesar da robustez da técnica, algumas desvantagens são observadas como a falta de imunidade a problemas comuns em técnicas estatísticas, como existência de *outliers*, valores incompletos ou perdidos. Além disso, tem-se o fato de a rede apresentar um funcionamento dito “caixa-preta”, ou seja, não detalham as informações de pro-

cessamento até chegar ao resultado final, o que pode ser crítico no momento de se realizar uma análise causal [TEN10].

## 2. Sistemas *Fuzzy*

Um sistema *fuzzy* é um mapeamento de valores em termos linguísticos, por exemplo, “muito baixo”, “baixo”, “alto” e “muito alto” para um conjunto de valores de variáveis correspondentes. Tanto a entrada quanto a saída do sistema *fuzzy* pode ser numérica ou linguística.

A principal vantagem da utilização de sistemas *fuzzy* para estimativas de software é a sua fácil compreensão, devido ao uso de termos linguísticos, fazendo com que o mesmo possa ser analisado e criticado por pessoas sem conhecimento ou treinamento prévio. Como desvantagem, tem-se a dificuldade de especificação de um sistema que permita uma alta precisão de resultados mantendo uma interface interpretável, onde geralmente, sistemas mais complexos precisam de mais regras, levando a um aumento de complexidade e decréscimo de poder de interpretação [GM97] [KAJH99].

## 3. Raciocínio Baseado em Casos (Estimativa por Analogia)

O Raciocínio Baseado em Casos (RBC), também conhecido como Estimativa por Analogia (no contexto de estimativa de esforço), [WM94] é uma técnica de resolução de problemas que resolve novos problemas adaptando soluções que foram usadas para resolver problemas antigos. RBC recupera um ou mais casos semelhantes ao problema atual e tenta modificar estes casos para ajustar aos parâmetros do problema atual. Na estimativa de esforço de desenvolvimento de software, cada caso pode ser um desenvolvimento de software anterior, enquanto o problema atual é extrair uma estimativa adequada para o projeto atual. Como resultado, o desenvolvimento deve ser mais rápido e a estimativa de tempo ajustada em conformidade.

A vantagem da utilização dessa abordagem é que pode-se justificar decisões com base em casos anteriores utilizados na resolução de um problema. Além disso, a abordagem RBC é intuitivamente semelhante ao julgamento de especialistas adotado em muitas organizações que dependem de uso de desenvolvedores experientes para estimar esforço do projeto. Estes indivíduos estimam adaptando o julgamento de desenvolvimentos de sistemas anteriores.

O problema dessa técnica é que os projetos têm que ter a mesma característica para que as estimativas tenham uma boa proximidade do real, o que nem sempre é possível, principalmente no contexto de reengenharia de sistemas e de manutenção.

## 4. Árvores de Classificação e Regressão

Árvores de classificação e de regressão são conceitos similares, mas diferem na representação do atributo que se deseja prever. Ambas se utilizam de um conjunto

de dados previamente conhecidos, induzindo as regras necessárias para classificarem esses dados. O método mais comumente utilizado para construção da árvore é o *top-down*. Essa estratégia consiste da análise de qual atributo dos registros de dados melhor divide o conjunto de dados em subpopulações disjuntas. Dentre as abordagens utilizadas para a realização dessa divisão estão: o cálculo do erro médio quadrado, o cálculo de entropia, dentre outros [SF95] [GM97].

As árvores de regressão são mais utilizadas em métricas de software do que as de decisão. Isso é dado pela própria necessidade que o domínio impõe de obtenção de resultados numéricos. Além de serem utilizadas para estimar esforço, as árvores de regressão também são utilizadas para estimar defeitos [GM97].

A vantagem em se utilizar as árvores de regressão é que elas são simples de serem aplicadas sobre um conjunto de dados, já que existem várias ferramentas que as implementam, como o WEKA por exemplo [WFT<sup>+</sup>15]. Como desvantagens tem-se a dificuldade de compreensão, pois quanto mais níveis e nodos a árvore tiver, menos compreensível ela se torna. Além disso, por depender de dados do projeto, a árvore é sensível à *outliers*, isso quer dizer que a qualidade dos dados é importante para que se obtenham bons resultados com essa técnica.

## 5. Algoritmos Genéticos

Algoritmo Genético (AG) utiliza conceitos da Teoria da Evolução e da Genética na resolução de problemas. Em relação à Teoria da Evolução, os AGs se baseiam na tese de que os organismos que melhor se adaptam a seu ambiente têm maiores chances de ter suas características reproduzidas em uma nova geração, já em relação à Genética o ponto principal utilizado é o fato dos filhos herdarem as características dos pais [FLGdC11].

Inicialmente é gerada uma população inicial de possíveis soluções (também chamadas indivíduos ou cromossomos) para o problema. Então, busca-se, em um processo iterativo, gerar uma boa solução por meio da evolução das melhores soluções da população atual, de acordo com uma função de aptidão, que mede a qualidade de cada solução. Para a definição de cada nova população a partir das soluções escolhidas, três operadores genéticos são geralmente aplicados: elitismo (cópias simples das melhores soluções), cruzamento (combinação de pares de soluções) e mutação (altera a composição de algumas soluções, permitindo a criação de soluções ainda não observadas) [FLGdC11]. A sequência de operações seleção-reprodução se repete até que seja atingido um critério de parada tal como, número máximo de repetições do ciclo (geração), obtenção de pelo menos uma solução satisfatória ou a falta de melhoria, por um dado número de gerações, do valor de aptidão do melhor indivíduo [FLGdC11].

Dentre as vantagens de utilizar esta abordagem está o fato de ela realizar buscas simultâneas em várias regiões do espaço de soluções, permitindo que se encontrem

diversas soluções, tornando a abordagem um método de busca global. Em relação às desvantagens, a principal seria o custo computacional, que pode ser contornado com a utilização de tecnologias distribuídas, já que os AGs são facilmente paralelizáveis.

Em estimativa de esforço, os AGs são usados em combinação com outras técnicas como Redes Neurais Artificiais, para a otimização dos parâmetros de cada *dataset*, como em [BDSM12], ou com Grey Relational Analysis, para encontrar os melhores pesos para cada parâmetro de estimativa [HCC08].

## Compostas

Por fim, [BAC00] afirma que nenhuma técnica em especial é adequada para todas as situações. Portanto, na maioria das vezes faz-se uso de uma combinação de técnicas para se obter melhor acurácia nos resultados. Com base nisso, diversas soluções emergiram já propondo composição de técnicas de maneira a tornar a solução proposta mais generalizável aos diversos contextos de desenvolvimento. Assim, técnicas compostas nada mais são do que modelos, metodologias, métodos, dentre outros que combinam duas ou mais técnicas para realização de estimativas de esforço.

### **2.3 Estimativa de Esforço em Reengenharia de Software: Trabalhos Relacionados**

Estimativa precisa dos custos do projeto é um pré-requisito essencial para fazer um projeto de reengenharia [SNE95]. Os sistemas existentes geralmente passam por reengenharia por ser mais barato do que a manutenção ou substituição dos mesmos [PRE11] [SOM07]. No entanto, para tomar esta decisão, a organização deve saber o quanto a reengenharia vai custar e quando esse investimento será visível.

No início da década de 90 surgiram os primeiros estudos sobre a economia de projetos de reengenharia de software [SNE91]. Antes disso, o esforço em projetos de reengenharia era calculado apenas com base no tamanho do sistema, sem levar em conta a complexidade e qualidade inerentes do sistema legado. Com o crescimento e amadurecimento da área tornou-se importante levar em conta os fatores de complexidade e qualidade nos custos de reengenharia de software. Reengenharia é uma alternativa para manutenção, compra de um pacote padrão ou a não fazer nada. Portanto, é importante para a gestão da organização saber qual o retorno sobre o investimento para cada alternativa.

Em relação aos trabalhos que tratam de estimativa de esforço em reengenharia, ainda existe uma escassez de pesquisas neste assunto. Uma revisão sistemática sobre soluções de apoio à estimativa de esforço em desenvolvimento de software (apresentada

no Apêndice A) identificou apenas três abordagens neste sentido. Tais abordagens são apresentadas a seguir.

### 2.3.1 Abordagem de Sneed 2005

O trabalho de Sneed [SNE05] aborda a estimativa de custo e tempo em reengenharia a partir de um processo de oito passos, apoiados por ferramentas. Tal processo foi derivado com base na experiência do autor e validado em estudos de casos.

O autor recomenda que, antes de começar a reengenharia, seja identificado o objetivo da mesma: se é uma migração de tecnologia, uma renovação, etc. Em todos os casos o objeto de análise será o mesmo (o sistema legado existente), mas os custos, riscos e benefícios variam de acordo com cada um. Uma vez definido o objetivo, aplica-se a técnica de estimativa. Os passos para a estimativa de custo e tempo de acordo com método proposto são os seguintes:

1. Análise dos requisitos de reengenharia
2. Análise do sistema existente
3. Transformação das métricas de código-fonte
4. Computação do tamanho do sistema ajustado
5. Definição dos objetivos de qualidade
6. Calibragem dos dados de produtividade
7. Análise de risco e computação dos fatores de risco
8. Cálculo do esforço e do tempo requerido

1. Análise dos requisitos de reengenharia: esta atividade consiste na aplicação de um questionário para coletar os requisitos de usuário. São coletados requisitos de três categorias: requisitos de tamanho, requisitos de complexidade e requisitos de qualidade. Os requisitos de tamanho são todos os componentes de sistema que afetam o projeto de reengenharia tais como, número de arquivos de código, número de transações online, número de tabelas na base de dados, etc. Os requisitos de complexidade são uma avaliação, feita pelos usuários, da complexidade do sistema existente em comparação ao sistema proposto, o usuário avalia a complexidade do ambiente, das estruturas de dados, da arquitetura, do código e das interfaces, usando uma escala nominal que vai de “muito alta” à “muito baixa”. Em relação a qualidade, os usuários avaliam as características de qualidade que podem ser afetadas pela reengenharia,

tais como desempenho, usabilidade, manutenibilidade, etc. O resultado da avaliação de complexidade e qualidade, juntamente com o tamanho do sistema alvo (sistema após a reengenharia) em total de artefatos de software, fornece uma primeira impressão do quanto custará o projeto de reengenharia. Estes dados podem ser analisados junto a projetos de reengenharia finalizados e devem servir como base para uma análise de custo/benefício do projeto de reengenharia.

2. **Análise do sistema existente:** caso decida-se por continuar o projeto de reengenharia, deve-se realizar a análise do sistema legado existente. É recomendável que esta atividade seja realizada automaticamente e em poucos dias, do contrário se tornará muito cara. As ferramentas para realizar essa tarefa são analisadores estatísticos. Estas ferramentas devem prover um relatório de todas as deficiências encontradas no código, bem como um relatório de métricas com a contagem de todas as características do software. O relatório de métricas deve conter as métricas de complexidade e qualidade extraídas do código, tais como: acoplamento, coesão, complexidade ciclomática, complexidade de classe, modularidade, testabilidade, reusabilidade, manutenibilidade e portabilidade, além da métrica de “conversibilidade”, que é baseada na proporção de 1:1, 1:m e m:m trechos de código convertíveis. A partir dessas métricas é derivada uma complexidade e uma qualidade média. O tamanho do sistema pode ser derivado em diferentes unidades como linhas de código, pontos por função e pontos por objeto. O autor propõe o uso da ferramenta *SoftAnal* que analisa os dados de sistemas escritos em Assembler, Java e C#.
3. **Transformação das métricas de código-fonte:** nesta etapa, os resultados da etapa 2 são transferidos para uma ferramenta denominada *SoftCalc* para realizar a estimativa de custo final. Esta ferramenta constrói um modelo de dados com três tabelas: uma tabela de métricas de tamanho, uma de métricas de complexidade e uma de métricas de qualidade. O usuário poderá atribuir pesos de 0.5 a 2 para métricas de complexidade e de qualidade de acordo com o impacto da métrica na estimativa.
4. **Computação do tamanho do sistema ajustado:** o tamanho do sistema ajustado é o resultado da multiplicação do tamanho do sistema bruto (em linhas de código, pontos de função, etc), multiplicados por um fator de complexidade e um fator de qualidade. Estes fatores são determinados a partir da complexidade do sistema e qualidade do sistema. Os coeficientes para estes cálculos foram baseados em experiências prévias do autor em projetos de reengenharia. Para estes cálculos também é utilizada a ferramenta *SoftCalc*.
5. **Definição dos objetivos de qualidade:** nesta etapa, o analista define os objetivos de qualidade e os fatores de influência. Para isto, são criadas outras duas tabelas na ferramenta *SoftCalc*: tabela das características de qualidade desejadas e tabela dos fatores de influência, a primeira inclui todas as características de qualidade que são

mensuráveis no código. Os analistas também devem atribuir a taxa que eles gostariam que cada característica atingisse no novo sistema. A ferramenta calcula a diferença entre a taxa de qualidade existente e a desejada. Também é calculado o fator de qualidade ajustado (QA) que é a soma da diferença entre as taxas de qualidade individuais mais 1.

6. Calibragem dos dados de produtividade: os dados de produtividade são derivados da literatura ou de projetos anteriores. A produtividade é definida como o relacionamento entre o tamanho do sistema ajustado e pessoas/dia (esforço) requeridas para realizar a reengenharia de um sistema desse tamanho. Os dados de esforço podem ser importados para a ferramenta *SoftCalc*, uma vez que ela já possui o valor do tamanho do sistema ajustado, ela calcula a produtividade. A produtividade pode ser influenciada por vários fatores de projeto, tais como grau de apoio ferramental, quantidade de viagens requeridas, facilidade de comunicação, maturidade do processo e condições de trabalho. Fica a cargo do usuário definir quais fatores utilizar e o peso de cada fator sobre a produtividade. O autor considera que a seleção e predição do impacto de cada fator de influência nos custos do projeto é o maior desafio da estimativa de custo.
  
7. Análise de risco e computação dos fatores de risco: todos os projetos envolvem algum grau de risco, mas em um projeto de reengenharia os riscos são muito menores do que no desenvolvimento de um novo projeto. Os principais riscos são: perda de performance, incompatibilidade do sistema e falta de aceitação por parte da equipe de programação (equipe do cliente). A ferramenta *SoftCalc* possui uma tabela com a probabilidade de cada risco ocorrer, o impacto do risco, a possibilidade de redução e o fator de risco. Os dados desta tabela foram determinados com base na experiência do autor em projetos passados.
  
8. Cálculo do esforço e do tempo requerido: O esforço (E) é calculado multiplicando o tamanho do sistema ajustado (TA) pelo fator de qualidade ajustado (QA), e dividindo este resultado pelo fator de produtividade ajustado (PA). Por fim, é multiplicado o fator de risco ajustado (RA) a este total, o esforço final é derivado em pessoas/mês:  $E = [(TA \times QA) / PA] \times RA$ . Já para o cálculo do tempo, foi modificado o algoritmo do COCOMO que define que a duração do desenvolvimento de um produto de software é  $2,5 \times$  Esforço, para este caso, assumiu-se que um projeto de reengenharia permite um maior grau de paralelismo nas tarefas, então o multiplicador foi reduzido para 1,5.

### 2.3.2 Abordagem de Baldassarre *et al.* [2003], Caivano *et al.* [2006] e Baldassarre *et al.* [2006]

A proposta destes autores é um processo de calibração dinâmica para realização de estimativas projetos em reengenharia de software. A elaboração da proposta partiu de requisitos que os autores consideraram essenciais em um modelo de estimativa de modo geral, mais alguns requisitos particularmente voltados para projetos de reengenharia.

A Tabela 2.1 apresenta esses requisitos. A sigla G (geral) indica as características consideradas determinantes para um modelo de estimativa de esforço/custo para qualquer tipo de projeto de desenvolvimento de software, e a sigla R (reengenharia) indica as características que devem ser atendidas para um projeto de reengenharia.

Tabela 2.1 – Requisitos básicos para soluções de estimativa de esforço

ID	Requisito	Foco
1	O modelo de estimativa deve usar as métricas disponíveis na organização e deve ser calibrável aos dados locais, com o objetivo de se obter previsões adequadas as características do processo.	G
2	Se ocorrer uma mudança no processo de desenvolvimento ocasionada pelo aumento da maturidade dos desenvolvedores, o modelo de estimativa deve indicar onde ocorreu a mudança, de modo que ela possa ser adequadamente controlada em termos de pessoas, orçamento e cronograma.	G
3	Se ocorrer uma mudança no processo de desenvolvimento ocasionada por melhorias contínuas no processo, o modelo de estimativa deve ser sensível às mudanças e deve permitir avaliar o quão benéficas elas são.	G
4	O modelo deve ser customizável, em tempo de execução, para mudanças de parâmetros e de fatores de impacto, que podem ocorrer devido à mudanças no processo.	G
5	O modelo deve ser baseado em métricas de qualidade do código fonte do sistema legado, cujos valores devem ser conhecidos no início do projeto.	R
6	O modelo de estimativa deve ser executado em uma pequena parte do projeto de cada vez. A dimensão dessa parte depende do processo de reengenharia em uso.	R

Tomando como base esse conjunto de características desejáveis a um modelo de estimativa para projetos de reengenharia, foi proposto um modelo com 8 passos:

Um projeto (PR) é definido como uma execução de um processo P, onde:

PR é um conjunto  $PR_1, PR_2 \dots PR_k$  de  $k$  subprojetos, ou componentes de projeto. Ou seja, a execução do projeto vai ser de forma modular e cada  $PR_i$  deve ser um componente do projeto onde se possa realizar reengenharia de forma independente dos demais.

P é um conjunto  $P_1, P_2 \dots P_h$  de  $h$  subprocessos, ou componentes de processo.

Para cada componente de projeto  $PR_i$  será executado o subprocesso  $P_i$  correspondente.

1. Início: no início do projeto o Modelo de Estimativa (MD) do Esforço Esperado (EE) é usado, o EE é a aplicação do MD para todo  $P_i$ . Este modelo consiste em duas funções: a função de estimativa (FE), obrigatória, função principal para estimar o esforço e a função corretiva (FC), opcional, função para estimar e ajustar o erro obtido pela função principal. Na primeira execução deste modelo a estimativa pode ser obtida a partir de dados de projetos anteriormente desenvolvidos, julgamento de especialistas ou outra técnica presente na literatura. Deve também ser definido um conjunto de aprendizagem (L), que contém as métricas utilizadas para derivar a estimativa.
2. Cálculo do Erro: no final da execução de  $P_i$ , o Esforço Real (ER) requerido para sua execução deve ser conhecido, então o erro (ERRO) é calculado como sendo:  $ERRO = (EE - ER)/ER$ .
3. Registrando Pontos de Dados: no final de cada repetição de  $P_i$  são armazenados os dados de EE e ER. O conjunto de dados funciona como uma série histórica de dados.
4. Conjunto de Aprendizagem Móvel: Quando o erro ER atingir o limiar estabelecido pelo gerente de projetos, o modelo atual de estimativa MD deve ser recalibrado, usando um novo conjunto de aprendizagem L.
5. Calibragem do Modelo de Estimativa: usando o novo conjunto de aprendizagem L o modelo é recalculado.
  - 5.1 Análise dos principais componentes: se o conjunto M for muito numeroso será necessário reduzir extraindo os principais componentes. Uma forma de redução é substituir os elementos por uma combinação linear das métricas existentes.
  - 5.2 Adaptação da Função de Estimativa: dentre todas as métricas presentes em M, aquelas melhor correlacionadas com o ER são selecionadas. Usando essas métricas como parâmetro é derivada uma nova função FE. Se nenhuma métrica de M tiver correlação o ER, então tais métricas não caracterizam o esforço requerido para realização do processo PR e devem ser substituídas.
6. Atualizando os pontos de dados no conjunto de aprendizagem: todos os pontos de dados registrados devem ser atualizados com o cálculo do novo valor de EE.

7. Definindo a função corretiva: dentre as métricas que foram identificadas como não correlacionadas no passo 5.2, são selecionadas aquelas que melhor contabilizam os erros gerados pela nova FE. Essas métricas são os fatores de correção (FC) e o EE é ajustado a partir delas. Se todas as métricas são correlacionadas no passo 5.2, então não há FCs e, conseqüentemente, não há ajuste na estimativa.
8. Continuação do Projeto: após a calibragem do modelo, volta-se ao passo 2 até o fim do projeto.

Este modelo foi proposto e validado inicialmente em [BCV03]. A validação do modelo foi feita a partir da aplicação do mesmo na reengenharia de um sistema bancário. O projeto foi dividido nos subprojetos de Engenharia Reversa e Renovação. Não foi utilizado nenhum modelo de estimativa como referência para o ponto de partida, foi realizado um projeto piloto que derivou valores de desempenho em número de linhas de código por hora (NLOC/h), para cada subprocesso. Esse valor foi utilizado para derivar a FE. Ao final da aplicação do modelo a taxa de erro obtida foi de 16% no subprocesso de Engenharia Reversa e 8% no subprocesso de renovação. Uma comparação formal com outro modelo de estimativa não foi realizada, diante da justificativa de que não havia um modelo compatível.

No trabalho de [CLV06] é proposta uma evolução deste modelo e uma ferramenta para apoiar o processo de estimativa. Nesta versão, os dados para a início do projeto são derivados de projetos anteriores e calculados por Estimativa por Analogia. Durante a execução do projeto é criado um repositório de projeto, particionado em repositório de produto e repositório de processo, o primeiro contém os identificadores dos componentes de projeto executados e as medidas de seus atributos de entrada e saída, já o repositório de processo contém para cada componente de processo o número de pessoas envolvidas e as medidas dos atributos de processo (esforço estimado, esforço real, data de início, data de fim, etc). Estes dados servem para alimentar a calibragem das estimativas no decorrer do processo e como base histórica no final. Para validação foi utilizado o mesmo sistema bancário utilizado em [BCV03]. Por fim, [BBCV06] apresenta os componentes da arquitetura da ferramenta apresentada em [CLV06].

### 2.3.3 Abordagem de Boehm et al. [1995]

Este trabalho não trata de uma solução exclusivamente voltada para estimativa de esforço em reengenharia, mas sim da evolução do tradicional modelo de estimativa de esforço COCOMO81 [BOE84], que em sua primeira versão apoiava mais efetivamente a estimativa para o desenvolvimento de projetos novos, e na COCOMO 2.0 possui adaptações para apoiar reuso e reengenharia.

O modelo COCOMO 2.0 [BCH<sup>+</sup>95], assim como o COCOMO81, propõe um processo para cálculo da estimativa de esforço de projetos de software com base no tamanho do sistema e em parâmetros denominados Fatores de Custo, esses fatores foram identificados com base na análise dos dados de diversos projetos de desenvolvimento de software e categorizados em: fatores de produto (por exemplo, tamanho da base de dados, complexidade do produto), fatores de plataforma (por exemplo, restrição de tempo de execução, restrição de armazenamento, volatilidade da plataforma), fatores de pessoais (por exemplo, capacidade do analista, capacidade do programador) e fatores de projeto (por exemplo, uso de práticas modernas de programação, uso de ferramentas, desenvolvimento distribuído). São ainda definidas fórmulas para escalar e multiplicar cada um dos fatores afim de calcular a estimativa de esforço, todas as fórmulas e constantes definidas foram baseadas em análise projetos de desenvolvimento e em boas práticas relatadas na literatura.

A proposta do modelo para reengenharia é usar o modelo geral COCOMO 2.0 (fatores de produto, plataforma, pessoais e de projeto), levando em consideração o fator de aceleração de desenvolvimento. Esse fator está relacionado com o uso de ferramentas para tradução de código, supondo-se que o código traduzido de uma linguagem de programação para outra pode ser imediatamente utilizado, isto é, sem despende esforço de modificação/melhorias. Assim, no cálculo do esforço deve ser adicionado o parâmetro AT (porcentagem de código que onde a reengenharia pode ser realizada por tradução automática), e quanto maior for o valor de AT, maior a redução no esforço para realização da reengenharia. A proporção dessa redução de esforço foi derivada a partir de um estudo de caso citado pelo autor.

## 2.4 Análise Crítica

As abordagens descritas destacam a estimativa de esforço em reengenharia de software sob duas perspectivas. Enquanto[SNE05] e [BCH<sup>+</sup>95] tratam a realização da estimativa como uma atividade que ocorre apenas no início do projeto, [BCV03], [BBCV06] e [CLV06] propõe um processo que prevê uma calibragem contínua das estimativas, levando em conta que mudanças podem ocorrer no decorrer do processo.

Percebe-se através da descrição das abordagens, que as soluções propostas, até então, para apoio à estimativa de esforço em projetos de reengenharia de software tratam a estimativa como sendo uma tarefa apenas técnica, ou seja, o cálculo do esforço necessário para realização do projeto. Sendo assim, todas as soluções possuem entradas e saídas para este cálculo e passos que devem ser efetuados para a realização da estimativa, com apenas na análise do sistema legado e de dados técnicos do projeto.

Porém, a estimativa de esforço é a base para definição de cronograma, alocação de recursos e orçamento do projeto, e existem vários fatores não quantificáveis envolvidos

no projeto que podem influenciar a estimativa, tais como fatores organizacionais, pessoais, etc. Assim, identificar e integrar estes fatores ao processo de estimativa de esforço em reengenharia é uma possibilidade de pesquisa em aberto.

## **2.5 Conclusões do Capítulo**

Neste capítulo foi apresentado o referencial teórico da pesquisa, que tem como base Reengenharia de Software e Estimativa de Esforço. Em relação à Reengenharia, foi apresentado o processo básico e os tipos de reengenharia. Já em relação à Estimativa de Esforço, também foi apresentado o processo básico, além das principais técnicas existentes.

Por fim, foram apresentadas os trabalhos voltados para estimativa de esforço no contexto de reengenharia de software, cujo foco é o cálculo desta estimativa.

Nos próximos capítulos serão apresentados os esforços realizados nesta pesquisa com o intuito de explorar a estimativa de esforço em reengenharia sobre a perspectiva da indústria e utilizar este conhecimento para expandir o processo de estimativa para além do cálculo.

### 3. METODOLOGIA DE PESQUISA

Neste capítulo é apresentada a metodologia de pesquisa utilizada. A Seção 3.1 apresenta o desenho da pesquisa e suas etapas. A Seção 3.2. apresenta uma discussão sobre os aspectos metodológicos. A Seção 3.3 apresenta a metodologia do estudo de campo inicial, enquanto a Seção 3.4 apresenta a metodologia do estudo de caso.

Embora tenha sido realizada uma ampla revisão teórica, não se tem conhecimento de que o problema apresentado tenha sido abordado sob a mesma perspectiva, ou seja, uma abordagem de processo de estimativa de esforço em reengenharia não apenas na realização da estimativa, mas no seu acompanhamento durante todo o projeto. Assim, essa pesquisa caracteriza-se como um estudo predominantemente exploratório, sendo a metodologia de pesquisa principal a realização de estudos empíricos.

Neste estudo de natureza exploratória, foram usados métodos qualitativos pelo fato de envolver o estudo de estimativa de esforço em projetos de reengenharia no seu contexto real, com a compreensão do estado da arte em situações em que a prática se antecipa à teoria [HOP96]. Com relação à natureza exploratória do estudo, a pesquisa exploratória tem como principal finalidade desenvolver, esclarecer e modificar conceitos e ideias, com vista a formulação de novas teorias, modelos e hipóteses pesquisáveis em estudos posteriores [GIL08]. Gil [GIL08] ainda define que a pesquisa do tipo exploratória muitas vezes constitui-se na primeira etapa de uma investigação mais ampla.

Em relação ao método utilizado foram desenvolvidos dois estudos de campo: um inicial e um estudo de caso, que são apresentados nas próximas seções. O objetivo de ambos os estudos foi identificar o processo de estimativa de esforço em projetos de reengenharia, as boas práticas e desafios envolvidos neste tipo de projeto.

Por tratar-se de uma pesquisa qualitativa, deve-se ter claras as limitações deste tipo de pesquisa, principalmente no que se refere ao número de empresas estudadas, restringindo a generalização dos resultados obtidos. No capítulo 8 deste estudo aborda-se com profundidade a questão dos limites da pesquisa.

#### 3.1 Desenho e Etapas da Pesquisa

O desenho de pesquisa contempla as atividades realizadas com o intuito de atingir o objetivo principal da pesquisa. Tal objetivo principal, conforme mencionado na seção 1.2 foi a criação de um modelo para estimativa de esforço em projetos de reengenharia de software, respondendo a seguinte questão de pesquisa: *Como direta e indiretamente impactantes na estimativa de esforço em reengenharia de software podem ser relacionados em um modelo específico para este processo?*

O desenho é apresentado na Figura 3.1. Em seguida são descritas as fases da pesquisa.

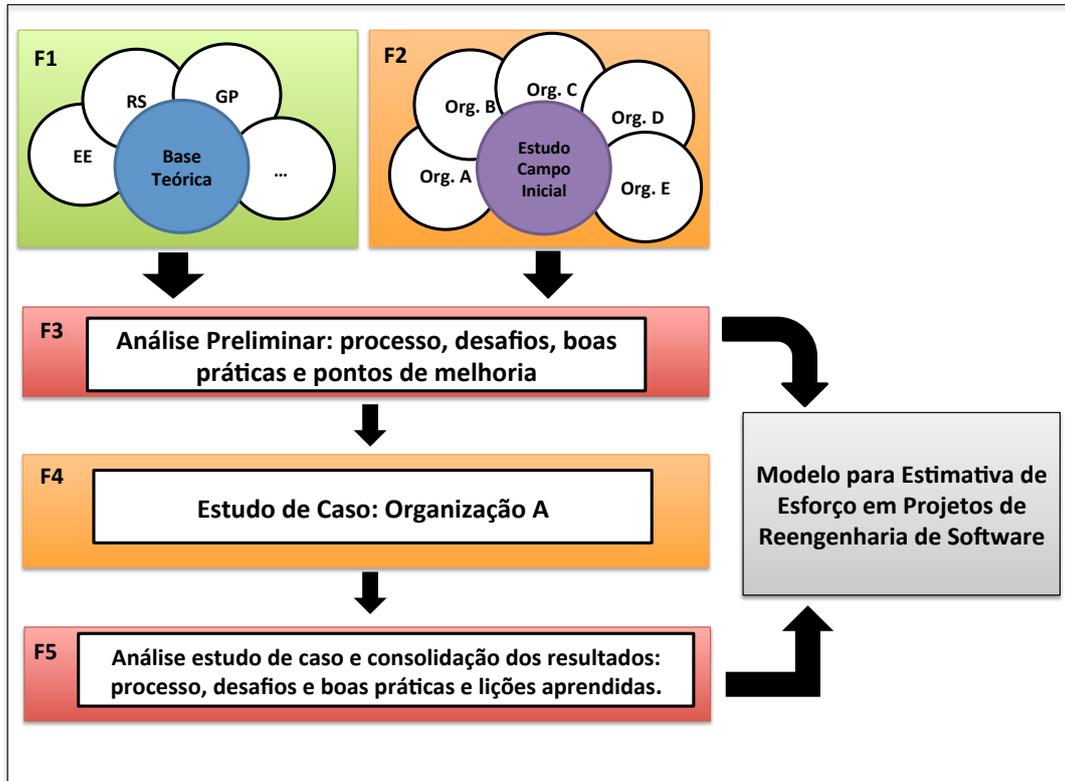


Figura 3.1 – Desenho da Pesquisa

**Fase 1:** nesta fase foi realizado o estudo de base teórica da área. Fazem parte da base teórica o estudo de Gerência de Projetos de Software, com ênfase em estimativa de esforço, Reengenharia de Software e a ligação entre as áreas. Durante esta fase foi realizada uma revisão sistemática da literatura, de acordo com o método proposto por [KC07], o objetivo desta revisão foi identificar soluções de apoio à estimativa de esforço em desenvolvimento de software, de modo geral. Esta revisão foi realizada, pois, em um primeiro momento, esta pesquisa visava propor uma solução mais técnica (algoritmo, ferramenta) para apoio a estimativa de esforço em reengenharia. Porém, as soluções existentes voltadas para esta área (apresentadas como trabalhos relacionados na seção 2.3) apontaram para uma carência de estudos com caráter mais empírico, que investigassem de maneira mais ampla o processo de estimativa em reengenharia, não só do ponto de vista técnico. Uma síntese dos resultados desta revisão sistemática é apresentada no Apêndice A deste trabalho.

**Fase 2:** foi realizado um estudo de campo inicial em 5 (cinco) empresas de desenvolvimento de software. Este estudo visou levantar dados gerais sobre o processo de estimativa de esforço em reengenharia, sob o ponto de vista de diversas organizações.

**Fase 3:** nesta fase foi realizada a análise crítica preliminar dos dados obtidos no estudo de campo inicial. Esta análise permitiu o levantamento de categorias de fatores impactantes na estimativa de esforço em reengenharia, além de pontos a serem melhorados e detalhados na segunda parte do estudo de campo.

**Fase 4:** foi realizado um estudo de caso em uma organização de desenvolvimento de software. Este estudo foi complementar ao primeiro e visou analisar de forma mais detalhada o processo de estimativa de esforço de reengenharia, a partir da análise de 3 (três) projetos com diferentes características. Nesta fase foram incorporados os pontos de melhoria identificados na fase 3.

**Fase 5:** foi realizada a análise dos dados do estudo de caso, que, juntamente com os dados do estudo de campo e da base teórica, serviram como base para construção do modelo de estimativa para estimativa de esforço em reengenharia. A criação modelo agrega as atividades, boas práticas e lições aprendidas durante as pesquisas empírica e teórica.

### 3.2 Aspectos Metodológicos

Conforme apresentado no desenho de pesquisa, a primeira etapa da investigação realizada neste trabalho envolveu a realização de um chamado "Estudo de Campo Inicial" e a segunda etapa foi a realização de um Estudo de Caso.

De acordo com a literatura, um estudo de campo se caracteriza como o estudo dos praticantes reais de uma determinada atividade e de como eles resolvem problemas reais [SS08]. Uma vez que a Engenharia de Software é uma área intensamente orientada à pessoas (pessoas criam o software, pessoas mantêm o software, pessoas evoluem o software), há um grande interesse na aplicação deste tipo de estudo nessa área. O tipo de pesquisa a ser empregado pode ser de dois tipos: qualitativa e quantitativa.

A pesquisa qualitativa é uma abordagem para explorar e compreender o significado que indivíduos ou grupos atribuem a um problema social ou humano. O processo de pesquisa envolve questões e procedimentos emergentes, os dados são normalmente recolhidos no ambiente de trabalho dos participante, a análise de dados constrói indutivamente indicações para temas gerais e o pesquisador faz interpretações sobre o significado dos dados, e a estrutura do relatório final é flexível. Em suma, este tipo de pesquisa tem uma abordagem indutiva e foco no significado individual, bem como a importância de representar a complexidade de uma situação.

Já pesquisa quantitativa possui uma abordagem para testar teorias objetivas examinando relações verdadeiras entre variáveis. Estas variáveis, por sua vez, podem ser medidas em instrumentos de coleta (questionários), de modo que os dados numerados po-

dem ser analisados utilizando procedimentos estatísticos. O relatório final deste tipo de estudo consiste em introdução, literatura e teoria, métodos, resultados e discussão. Este tipo de pesquisa objetiva testar teorias dedutivamente e ser capaz de generalizar e replicar as descobertas.

Assim, esta pesquisa é de natureza qualitativa, uma vez que pretende investigar como é realizada a estimativa de esforço em projetos de reengenharia de software, no contexto real onde esta atividade ocorre, levantando as práticas e desafios envolvidos neste processo. A principal vantagem deste tipo de pesquisa, segundo [SEA08], é o fato de os métodos qualitativos forçarem o pesquisador a se aprofundar na complexidade do problema ao invés de abstraí-la. Desta forma, os resultados são mais ricos e mais informativos, ajudando a responder questões que envolvem variáveis que são difíceis de quantificar, mas essenciais quando trata-se de atividades com forte influência de fatores humanos.

No que diz respeito à métodos para a realização de estudos de campo qualitativos, a literatura na área [MCG95], [TGdA02] aponta para a existência de três: pesquisa-ação, etnografia e estudo de caso. Com exceção do estudo de caso, que será tratado em breve, a aplicabilidade dos demais métodos não será discutida neste trabalho. O fato é que cada um desses métodos tem uma finalidade, um procedimento bem definido e um conjunto de resultados. Porém, o que se vê atualmente nas pesquisas na área é que tais métodos, por terem um foco bem definido, não atendem as necessidades de todas as pesquisa ditas qualitativas.

Neste contexto, pesquisadores como [AAMPR13] [MFCM13] tem optado por uma abordagem onde são utilizadas técnicas de coleta e análise de dados qualitativas, porém, que não se classificam diretamente como um dos métodos supracitados. Essas pesquisas são tipicamente denominadas como "pesquisas qualitativas" ou "estudos de campo qualitativos" e visam explorar a realização fenômeno (processo, atividade, etc) na área de engenharia de software, sob o ponto de vista das pessoas que atuam neste fenômeno. Geralmente a coleta de dados é feita através entrevistas semi-estruturadas realizadas em múltiplas organizações de desenvolvimento de software e a análise adota uma metodologia fundamentada em dados como *Grounded Theory* [SC90] ou Método de Comparação Constante [SEA08]. Esta abordagem é tipicamente empregada quando se deseja obter informações de diversas organizações.

Assim, esta pesquisa em sua primeira fase utilizou-se desta abordagem, uma vez que se desejava obter uma visão geral do processo de estimativa de esforço nos projetos de reengenharia, e este objetivo seria melhor atingido investigando o maior número de organizações possível, pois a diversidade no ambiente, negócio, etc permitiria uma maior riqueza de dados. Os detalhes deste estudo, aqui denominado de "estudo de campo inicial" são apresentados na próxima seção.

Uma vez finalizada a primeira parte do estudo, identificou-se a necessidade de uma investigação mais profunda e controlada do processo de estimativa de esforço em pro-

jetos de reengenharia: não apenas impressões gerais sobre o processo de estimativa da organização, mas uma análise detalhada de projetos onde este processo tenha sido aplicado, afim de se obter uma caracterização mais detalhada das nuances deste processo e validar os dados obtidos no estudo de campo inicial. Assim, como complemento a exploração realizada no estudo de campo inicial foi realizado um estudo de caso.

Segundo [HOP96], o estudo de caso é particularmente adequado ao exame exploratório dos fenômenos ainda pouco estudados e que precisam ser investigados em seu ambiente de ocorrência. A aplicação deste método é indicada quando se tem necessidade de aprender sobre o estado da arte e gerar novas teorias apoiadas na prática, entender a natureza e a complexidade do processo, enquanto este acontece, e trazer novos fatores e informações, evidenciados durante a execução do processo estudado [YIN10]. A metodologia utilizada no estudo de caso desenvolvido neste trabalho será apresentada na Seção 3.4.

### **3.3 Metodologia do Estudo de Campo Inicial**

O objetivo deste estudo de campo foi compreender como é realizado o processo de estimativa de esforço em projetos de reengenharia de software na contexto real: qual o processo adotado, quem são os envolvidos, quais as ferramentas e técnicas utilizadas, quais os desafios enfrentados e as boas práticas aplicadas. Assim, era de interesse da pesquisa que o maior número possível de organizações de desenvolvimento de software fossem estudadas, de maneira a se obter maior diversidade nas informações obtidas.

A Figura 3.2 apresenta os passos adotados para realização desta fase da pesquisa. O Apêndice B apresenta o Protocolo e Roteiros de Entrevista do estudo de campo. O Capítulo 4 apresenta os resultados obtidos neste estudo.

#### **3.3.1 Planejamento**

Na etapa de planejamento foram definidos os objetivos e selecionadas as técnicas de coleta e análise de dados que seriam utilizadas na pesquisa. Além disso, durante o planejamento, foram selecionados e convidados os profissionais com experiência em estimativa de esforço em projetos de reengenharia.

Para identificar os profissionais candidatos à entrevista buscou-se por organizações que realizam reengenharia de software na região metropolitana de Porto Alegre. Esta busca foi realizada no LinkedIn e em *websites* de diversas organizações. Além disso, foram contatados diretamente profissionais de organizações indicadas por colaboradores da pesquisa.

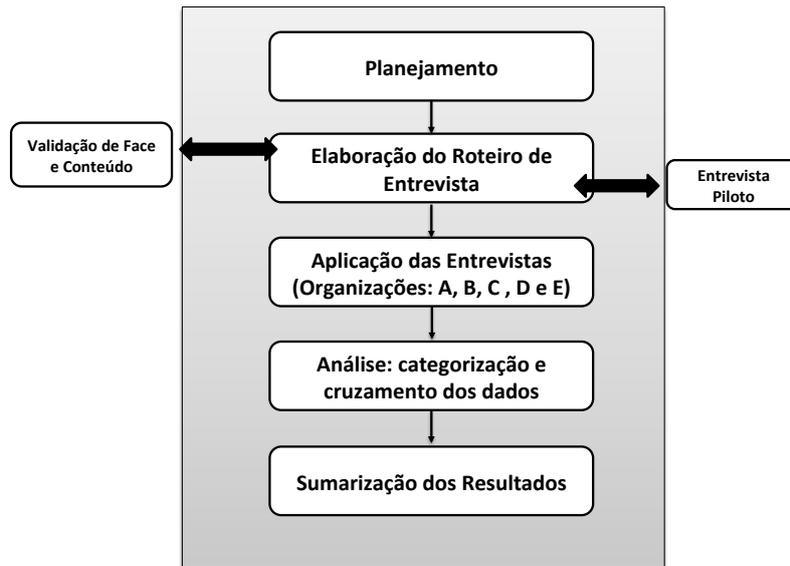


Figura 3.2 – Metodologia do Estudo de Campo Inicial

A partir de uma lista preliminar de organizações candidatas, foram enviados e-mails para representantes destas organizações, apresentando a pesquisa e pedindo o contato dos responsáveis pelas atividades de estimativa de esforço em projetos de reengenharia. Foi mencionado no e-mail que as pessoas indicadas estivessem trabalhando neste tipo de projeto atualmente, ou que tivessem trabalho no último ano, pelo menos. Esta medida foi adotada para selecionar apenas participantes que realizam reengenharia de software constantemente ou que realizaram recentemente, pois uma vez que o método de coleta seria a realização de entrevistas, a qualidade dos dados seria extremamente afetada pela capacidade do participante de recordar as atividades realizadas com o maior grau de detalhamento possível.

De um total de 14 convites enviados, foram recebidas respostas de sete profissionais (taxa de respondentes = 50%), que voluntariamente aceitaram participar no estudo. Os participantes são provenientes de 5 diferentes organizações de TI. Uma organização é uma companhia de telecomunicação, outra desenvolve sistemas para cartões de convênio e gestão de frotas, e três outras são organizações de desenvolvimento de software de médio e grande porte. Uma caracterização mais detalhada das organizações e dos participantes é fornecida no capítulo 4.

Uma vez definidos os objetivos e os participantes, foi elaborado um roteiro de entrevistas que elencava as questões a serem abordadas. O tipo de entrevista aplicado foi a semiestruturada com questões do tipo abertas, uma vez que a flexibilidade desta configuração permite obter maior detalhamento sobre o contexto em que os entrevistados estão imersos, já que novas questões podem ser desenvolvidas no decorrer da entrevista.

Para validação deste roteiro de entrevistas, foi realizada validação de face-conteúdo com quatro profissionais com experiência no tema: 3 pesquisadores de Engenharia de Software, com experiência em Gerenciamento de Projetos, e 1 profissional com experiência em estimativa de esforço. A fase de validação contribuiu para o refinamento das questões que compuseram a versão final do roteiro. As questões originais foram reordenadas e reescritas para permitir a inclusão de uma terminologia mais próxima à adotada pelos profissionais no seu dia-a-dia. Além disso, foi realizada uma entrevista-piloto com um profissional com 5 anos de experiência na realização de estimativa de esforço em projetos reengenharia. Esta entrevista piloto auxiliou na avaliação da confiabilidade e consistência do questionário, além de fornecer uma estimativa de duração da entrevista. A entrevista piloto teve caráter avaliativo do roteiro e não foi utilizada na fase de análise. Juntamente com o roteiro de entrevistas foi elaborado um termo de confidencialidade para notificar aos participantes de que nenhum dado obtido seria utilizado sem autorização prévia.

As entrevistas foram conduzidas com base no roteiro previamente elaborado e todas elas foram gravadas. Durante a entrevista, os participantes foram estimulados a falar livremente, enquanto respondiam as questões. Todas as entrevistas foram realizadas no local de trabalho dos participantes.

Após a realização de cada entrevista os dados gravados foram transcritos e codificados de maneira a: 1) identificar códigos relevantes no contexto de estimativa de esforço em projetos de reengenharia, 2) analisar a relação entre esses códigos e 3) identificar as categorias para agrupar esses códigos previamente encontrados. A síntese dessas categorias é uma lista de elementos de processo reengenharia e de fatores que afetam a estimativa de esforço em projetos deste tipo, detalhados por cada categoria e ilustrados a partir da seleção de trechos significativos das entrevistas que os mencionem.

### 3.3.2 Coleta dos Dados

A coleta de dados procedeu a partir de entrevistas semiestruturadas com profissionais responsáveis pela realização de estimativa de esforço em projetos de reengenharia. Tais entrevistas foram empregadas como forma de obter desses profissionais seu conhecimento tácito relacionado a como é realizada a estimativa de esforço em projetos de reengenharia, focando no processo, desafios e práticas adotadas, e demais informações que consideram importante na estimativa.

Dada a natureza cíclica do método de análise empregado (Método de Comparação Constante, a ser tratado na próxima seção), que prevê que diversas etapas de coleta e análise de dados devem ser realizadas até se obter um resultado satisfatório, foram realizadas duas etapas de coleta de dados nesta pesquisa.

No total, 7 profissionais de quatro diferentes organizações participaram da pesquisa. Sendo 6 profissionais na primeira etapa e 2 na segunda, uma vez que um dos profissionais participou de ambas as etapas.

### Primeira Etapa da Coleta de Dados

Nesta etapa das entrevistas, os profissionais definiram o que é reengenharia de software para a organização, bem como descreveram o processo de estimativa de esforço utilizado neste tipo de projeto, atividades, técnicas, ferramentas, pessoas envolvidas, os desafios encontrados e boas práticas aplicadas.

A duração das entrevistas variou entre 28 e 45 minutos. Essa variação está relacionada com as características individuais de cada entrevistado (por exemplo, grau de detalhamento na resposta, tempo médio para resposta, etc).

A Tabela 3.1 apresenta as questões empregadas nesta primeira fase das entrevistas. Tais questões podiam levar a outras dependendo da resposta obtida.

Tabela 3.1 – Questões Aplicadas nas Entrevistas - Primeira Fase

<b>Questões</b>
1. O que são projetos de reengenharia para você? No que eles diferem dos projetos de manutenção?
2. Quantos projetos de reengenharia de sistemas estão sendo gerenciados por você neste momento?
3. Quais as características principais dos projetos de reengenharia desenvolvidos aqui? (tipos de sistemas, nível de documentação, linguagem do sistema legado, linguagem do sistema novo, porte dos projetos (conceito de porte no contexto da organização em questão)?
4. Qual (ais) o (s) modelo (s) de processo de reengenharia de sistemas utilizado (s)? (cascata, incremental/modular)? Se houver mais de um, em que contexto cada um é utilizado?
5. Em que (quais) momento (s) do ciclo de desenvolvimento é realizada a estimativa de esforço em projetos de reengenharia?
6. Como é realizada (envolvidos, técnica aplicada, apoio ferramental utilizado, parâmetros, documentos consumidos e gerados)?
7. Qual são as principais vantagens do processo atualmente aplicado (quão eficiente é)?
8. E as desvantagens? Na sua opinião, como estas desvantagens podem ser contornadas?
9. Gostaria de adotar uma outra abordagem para realização de estimativa de esforço em projetos de reengenharia? Caso sim, qual abordagem e por que não adota?

## Segunda Etapa da Coleta de Dados

Esta etapa foi necessária para aprofundar as informações sobre como é realizada a estimativa de esforço no levantamento de requisitos de um projeto de reengenharia. Uma vez que a Engenharia Reversa é caracterizada como uma etapa essencial do processo de reengenharia, não ficou claro na primeira etapa de aplicação do estudo se esta atividade era realizada e de que maneira a estimativa de esforço era impactada. Sendo assim, elaborou-se uma nova etapa do estudo (respeitando-se todos os passos da metodologia apresentados anteriormente) para coletar essas informações.

Nesta etapa, porém, foram entrevistados profissionais de apenas duas das quatro organizações que participaram da primeira fase, devido a não disponibilidade dos profissionais em participarem da pesquisa. A duração das entrevistas variou entre 30 e 64 minutos.

A Tabela 3.2 apresenta as questões empregadas na segunda fase das entrevistas.

Tabela 3.2 – Questões Aplicadas nas Entrevistas - Segunda Fase

<b>Questões</b>
1. Como é feita a especificação (análise) de requisitos para projetos de reengenharia de software (etapas, envolvidos, técnicas aplicadas, ferramentas utilizadas)? Caso não seja mencionado especificamente a Engenharia Reversa como sendo utilizada, questionar o por que.
2. Como é estimado o esforço nesta fase? (envolvidos, técnicas aplicadas, métricas/parâmetros, documentos consumidos e gerados, ferramentas utilizadas).
3. Caso não seja estimado esforço para esta fase: Por quê não é feito? Como justificam/negociam o esforço gasto nesta fase junto as partes interessadas pelo sistema?
4. Quanto o esforço aplicado nesta fase impacta no esforço total do projeto?
5. Existe alguma dificuldade na realização de estimativa de esforço nesta fase? Qual (is)?
6. Caso exista, o quais as ações tomadas para contornar estas dificuldades?

### 3.3.3 Análise dos Dados

Por se tratar de uma pesquisa qualitativa a análise dos dados é dita interpretativa, e deve-se utilizar métodos de análise qualitativos. Uma análise qualitativa é um processo não matemático de interpretação, que objetiva descobrir conceitos e relações nos dados brutos e organizar tais conceitos e relações em um esquema explanatório teórico [SC90], isto é, um esquema que explique/explique os dados coletados.

Geralmente, pesquisas qualitativas geram uma grande quantidade de dados que devem ser analisados. Para isso, o conjunto de dados deve ser reduzido para um formato compreensível, o que é tradicionalmente feito através do processo de codificação [SS08]. Tal processo consiste em usar os objetivos da pesquisa como um guia e, assim, desenvolver um esquema para categorizar os dados. Depois dessa categorização, os dados podem ser analisados para prover uma caracterização baseada nos esquemas de codificação dos pesquisadores [SS08].

O método de análise utilizado nesse trabalho foi o Método de Comparação Constante (*constant comparison method*) [SEA08]. Este método consiste em três etapas: Codificação Aberta, Codificação Axial e Codificação Seletiva.

**Codificação Aberta:** consiste em marcar, com códigos ou rótulos, partes do texto que são relevantes para um tema em particular ou uma ideia interessante no estudo.

Tais códigos podem ser elaborados antes (códigos pré-elaborados) ou depois (códigos-pós elaborados) da aplicação do estudo. O primeiro caso dá-se quando os objetivos do estudo estão claros e bem definidos antes do início deste. Já o segundo caso é utilizado quando os objetivos do estudo são iniciais e abertos. Em ambos os casos o conjunto de códigos desenvolve uma estrutura, com sub-códigos e categorias, que emergem da análise dos dados.

**Codificação Axial:** em seguida, para cada código as passagens marcadas nas entrevistas são agrupadas em padrões, de acordo com os códigos e sub-códigos com que foram assinaladas. Nesta etapa são reagrupados os dados que foram quebrados em códigos na etapa anterior.

O contexto do qual cada passagem foi destacada é importante e deve ser incluído na consideração de cada grupo de passagens. Este é momento que intensivas ou constantes comparações ocorrem, uma vez que os dados são revisados várias vezes para que sejam identificados relacionamentos entre as categorias e os códigos.

O foco da etapa é uniformizar explicações de um fenômeno em destaque, em particular os “comos” e os “porquês”.

**Codificação Seletiva:** por fim, na codificação seletiva são geradas “memórias de campo” que articulam uma proposição (ou hipótese preliminar a ser considerada) ou uma observação sintetizada dos dados codificados. No caso de geração de hipóteses, estas devem ser avaliadas em uma próxima etapa do estudo, que envolve novamente a coleta e análise de dados, até que seja formulada uma proposição que se mostre representativa.

Neste trabalho optou-se por não selecionar uma categoria principal, pois o método determina que haja uma circularidade entre a coleta e análise de dados, até que a saturação da teoria seja alcançada, ou seja, que nenhuma nova descoberta seja realizada.

Após transcritas, as entrevistas da primeira fase foram analisadas com o auxílio ferramenta MaxQDA [Sof14]. Na codificação aberta, os dados das entrevistas foram

analisados para proporcionar a criação de códigos, associados com trechos de texto. Em seguida, na codificação axial, foram criados relacionamentos entre os códigos. Foram identificadas 4 categorias que representam os principais fatores que influenciam na estimativa de esforço em projetos de reengenharia: Clientes, Sistema Legado, Projeto e Equipe de Desenvolvimento. Além disso, foram identificadas as atividades de estimativa de esforço que são realizadas durante os projetos de reengenharia. A partir dessa análise foi possível obter um conjunto inicial de desafios e práticas do processo de estimativa esforço em projetos de reengenharia, tais dados foram associados as categorias fatores de influência.

No entanto, pode-se observar que as informações obtidas diziam respeito a estimativa de esforço realizada apenas após a etapa de análise e projeto do sistema, ou seja, quando já se tinham os requisitos especificados. Esta informação pode ser observada não só através da descrição das atividades e dos parâmetros e técnicas empregadas pelos profissionais na realização da estimativa (por exemplo: tamanho, complexidade), como, em alguns casos, pelo próprio papel deste profissional no projeto que indicava que este atuava apenas a partir da implementação, não sendo, portanto, responsável pela estimativa antes dessa fase.

A análise da segunda etapa das entrevistas transcorreu de forma semelhante ao realizado anteriormente, sendo o diferencial o fato de que, como já existiam categorias definidas, buscou-se relacionar os novos dados com essas categorias. Ao final, não foram identificadas novas categorias nessa fase, pois a já existentes suportaram as novas descobertas em termos de desafios e práticas.

### **3.4 Metodologia do Estudo de Caso**

Conforme dito anteriormente, a realização deste estudo de caso foi necessária para aprofundar e detalhar os dados obtidos no estudo de campo inicial.

Estudo de caso é definido como uma investigação empírica acerca de um fenômeno contemporâneo dentro de seu contexto de ocorrência real, especialmente quando os limites entre o fenômeno e o contexto não são claramente evidentes [YIN10]. Projetos de estudo de caso pode ser de caso único ou de vários estudos de caso, e podem envolver uma única unidade (holística) ou várias unidades (integrado) de análise. Os estudos de caso são tipicamente exploratórios ou confirmatórios. Estudos de caso exploratórios são usados como investigação inicial de algum fenômeno, para derivar novas hipóteses e construir teorias. Já estudos de caso de confirmação são usados para testar teorias existentes.

Há uma variedade de diferentes fontes de dados utilizadas em um estudo de caso. Os dados qualitativos, incluindo entrevistas e observação, desempenham um papel central, uma vez que oferecem informações valiosas sobre o caso. Os dados quantitativos envolvem números e classes. A coleta destes dados é executada sobre uma bem definida unidade

de análise, ou "caso". Em Engenharia de Software, a unidade de análise pode ser uma empresa, um projeto, uma equipe, um desenvolvedor individual, um episódio ou evento especial, um produto de trabalho específico, etc. [YIN10].

Este estudo de caso seguiu a metodologia proposta por [YIN10], por ser a mais amplamente utilizada em trabalhos do tipo. A Figura 3.3 apresenta os passos realizados.

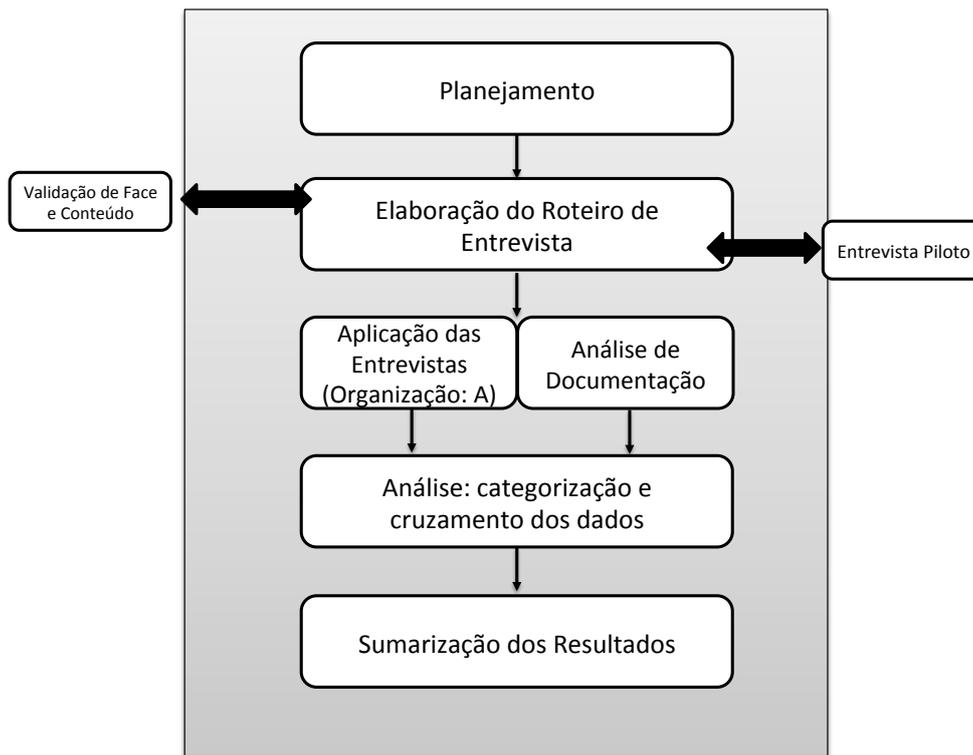


Figura 3.3 – Metodologia do Estudo de Caso

### 3.4.1 Planejamento

O ponto de partida para elaboração do estudo de caso foi a elaboração do Protocolo do Estudo de Caso, este protocolo consiste de um planejamento dos procedimentos e regras gerais que serão aplicados no estudo de caso. Elaborar um protocolo é uma das táticas principais para aumentar a confiabilidade da pesquisa e destina-se a orientar o pesquisador ao realizar a coleta de dados do estudo de caso [YIN10].

Os componentes básicos do estudo de caso de acordo com [YIN10] são:

1. as questões do estudo;

2. suas proposições; e
3. sua(s) unidade (s) de análise.

No caso do estudo de caso, as questões geralmente são do tipo qualitativas e assumem as formas de "por quê" e "como". Nesse caso, como este estudo tratava-se de um complemento do estudo de campo inicial, a objetivo principal, que levou a questão de pesquisa, foi identificar "como é realizada a estimativa de esforço em projetos de reengenharia de software?".

Em relação às proposições, o estudo de campo inicial deu indícios de que não há diferenças significantes entre o processo de estimativa de esforço em projetos de reengenharia em relação ao desenvolvimento de software novo. Portanto, este estudo visava, a partir da análise de 3 projetos de reengenharia, obter um melhor esclarecimento sobre esta questão.

Por fim, as unidades de análise, ou "casos", são projetos de reengenharia de software. Apesar de haver mais de uma unidade de análise (3, no caso), este projeto é classificado como estudo de caso único, uma vez que apenas um contexto organizacional foi investigado.

### Seleção da Organização e Unidades de Análise

O estudo foi desenvolvido na unidade de Transformação e Modernização de Software de uma organização multinacional com filiais em vários países da Europa, nos Estados Unidos e no Brasil. A organização atua em diversos setores de TI, sendo a unidade visitada responsável pela prestação de serviços de software, incluindo projetos desenvolvimento, consultoria e treinamento. Além desta unidade, localizada em Porto Alegre, há outras com a mesma finalidade no Brasil, responsáveis por atender demandas de clientes externos. É uma organização de grande porte para seu segmento e sua matriz está localizada nos Estados Unidos.

Esta organização, doravante chamada de Organização A, foi uma das investigadas no estudo de campo inicial. A seleção desta organização, em detrimento das demais, deu-se por ser a única com uma área de desenvolvimento exclusivamente voltada para projetos de reengenharia de software, com parte da equipe dedicada a este tipo de projeto.

Os projetos de reengenharia escolhidos para serem analisados na pesquisa foram selecionados de acordo com as características dos mesmos em relação à natureza da reengenharia realizada e às técnicas de estimativa aplicada.

A organização disponibilizou acesso irrestrito aos procedimentos deste estudo, tanto do acompanhamento do processo, quanto à documentação do projeto. No capítulo 5 são apresentados os detalhes dos projetos analisados.

## Operacionalização do Estudo de Caso

A realização do estudo de caso na Organização A foi possível graças a parceria estabelecida entre o grupo de pesquisa em que os pesquisadores envolvidos neste estudo atuam e a organização.

O contato inicial foi estabelecido com o Consultor de Tecnologia, responsável pela prática de Transformação e Modernização, que já conhecia a pesquisa, pois havia participado do estudo de campo inicial e demonstrou grande interesse nos resultados da mesma. Este contato viabilizou o agendamento de uma reunião com todos os responsáveis pela área de transformação e modernização de software, além de gestores de projetos da unidade como um todo.

Nesta reunião foi realizada a apresentação da pesquisa e foi concedida a aprovação para realização do estudo de caso. Nesta mesma reunião foram selecionados os projetos que iriam constituir a unidade de análise do estudo, bem como os primeiros respondentes das entrevistas.

### 3.4.2 Elaboração do Roteiro de Entrevistas

A preparação do roteiro de entrevista deu-se com a mesma formalidade apresentada no estudo de campo (validação de face-conteúdo e entrevista piloto). A validação de face-conteúdo foi realizada por um pesquisador e por um profissional de gerenciamento de projetos, ambos com experiência nas áreas de estimativa de esforço e reengenharia. Além disso, foi realizada uma entrevista-piloto com um profissional com 5 anos de experiência na realização de estimativa de esforço em projetos reengenharia. Esta entrevista piloto auxiliou na avaliação da confiabilidade e consistência do questionário, além de fornecer uma estimativa de duração da entrevista. A entrevista-piloto teve caráter avaliativo do roteiro e não foi utilizada na fase de análise.

O diferencial neste caso em relação ao estudo de campo inicial foi a inclusão de informações obtidas no estudo de campo na formulação das questões, ou seja, as categorias identificadas no primeiro estudo serviram como base para formulação de novas questões. A Tabela 3.3 apresenta as questões do roteiro de entrevistas.

O Apêndice C contém o protocolo do estudo de caso, bem como o roteiro de entrevistas.

Tabela 3.3 – Questões Aplicadas nas Entrevistas - Estudo de Caso

<b>Questões</b>
1. Quais as características principais deste projeto de reengenharia? (tipo de sistema, nível de documentação, porte do projeto)?
2. Qual (ais) atividade(s) de estimativa de esforço você realizou no projeto? (estimar, calibrar, monitorar, etc)?
3. Em que momento do projeto esta (s) atividade (s) foi (ram) realizada (s)?
4. Como foi (ram) realizada (s)? (envolvidos, técnica aplicada, métricas/parâmetros, documentos consumidos e gerados, ferramentas utilizadas).
5. O fato de existir um sistema legado como ponto de partida para o projeto influenciou as estimativas? Como?
6. Existiu algum fator relacionado a equipe do projeto que afetou a estimativa de esforço? Qual (ais)?
7. Existiu algum fator relacionado ao cliente que afetou a estimativa de esforço? Qual (ais)?
8. Do seu ponto de vista, quais são as principais dificuldades enfrentadas acerca de estimativa de esforço?
9. Em quais atividades no processo de reengenharia cada dificuldade foi encontrada?
10. Para cada dificuldade encontrada, qual foi a solução adotada?
11. Como esta solução foi identificada?
12. Você considera que a estimativa de esforço para este projeto foi realizada foi satisfatória? Se sim, quais os fatores que levaram a isto? Senão, por quê? O que deu errado?
13. Além das informações fornecidas acima, você gostaria de acrescentar algum outro fator relevante a realização de estimativa de esforço em projeto de reengenharia?

### 3.4.3 Coleta de Dados

A coleta de dados foi constituída por fontes primárias e secundárias. A fonte primária foi a realização de entrevistas. Ao todo foram realizadas 6 entrevistas semiestruturadas com os responsáveis diretos e indiretos pelas estimativas de esforço nos projetos de reengenharia investigados.

As fontes secundárias consultadas foram os documentos dos projetos, tais como proposta técnica e planilhas de estimativa. Além disso, foram disponibilizados os documentos de cunho geral como descrição do processo de desenvolvimento e organogramas.

Foi realizada uma imersão de duas semanas na organização, durante 4 horas por dia. Neste período foi realizada a análise da documentação, principalmente da técnica de

estimativa de esforço aplicada, paralela a análise eram esclarecidas dúvidas e levantadas questões a serem abordadas nas entrevistas. Também foram identificados e contatados os profissionais a serem entrevistados. As entrevistas ocorreram no decorrer das quatro semanas seguintes, de acordo com a disponibilidade dos entrevistados. Todas as entrevistas foram gravadas, para posterior transcrição.

O critério inicial para definição dos entrevistados centrou-se na unidade de análise de nos objetivos do estudo de caso. A Organização A trabalha com projetos de preço fixo, ou seja, o custo do projeto é definido em tempos de pré-venda da solução, e não deve ser alterado no decorrer do projeto. Como o custo deriva diretamente do esforço (ver figura 2.2, Capítulo 2), a estimativa de esforço é realizada nesta fase também. Assim, os participantes inicialmente selecionados para serem entrevistados eram os responsáveis por realizar estimativa na fase de pré-venda. Porém, durante o período de imersão, conforme foi-se ampliando a compreensão sobre o processo de reengenharia, foram identificadas novas pessoas que atuavam de alguma forma na estimativa de esforço deste projeto e estas pessoas também foram convidadas a participar da pesquisa.

#### 3.4.4 Análise dos Dados

A análise das entrevistas foi realizada de acordo com o Método de Comparação Constante [SEA08], já apresentado neste capítulo. Nesta fase, além das categorias identificadas no estudo de campo inicial (Sistema, Cliente, Projeto e Equipe), foi identificada uma nova categoria, denominada Organização, que agrupa as características organizacionais que impactam na estimativa de esforço nos projetos de reengenharia.

Além disso, foi realizada a análise dos documentos dos projetos, relacionados à estimativa de esforço. Os documentos analisados foram: proposta técnica do projeto, planilhas de estimativa e plano do projeto.

A utilização de mais de uma fonte de informação possibilitou ampliar a descrição, explicação e compreensão do objeto de estudo, de uma maneira que não foi possível no estudo de campo inicial, tendo-se, com isso, o principal motivador para aplicação destes dois métodos.

### 3.5 Conclusões do Capítulo

Este capítulo apresentou a base metodológica utilizada para condução deste trabalho. Foram discutidos o desenho da pesquisa, incluindo os métodos e como esses métodos foram aplicados em cada estágio da pesquisa. Esta pesquisa é de natureza exploratória e adota como principal método de pesquisa estudos de campo.

Os próximos capítulos abordarão como os métodos de pesquisa aqui descritos foram usados para construir um modelo de processo para estimativa de esforço em projetos de reengenharia.



## **4. APLICAÇÃO E RESULTADOS DO ESTUDO DE CAMPO INICIAL**

Neste capítulo serão relatados os resultados obtidos no estudo de campo inicial. A Seção 4.1 apresenta a caracterização geral das organizações e dos participantes do estudo. A Seção 4.2 apresenta os resultados da pesquisa e a Seção 4.3 as conclusões do capítulo.

### **4.1 Caracterização das Organizações e dos Entrevistados**

Como este trabalho visa compreender como dá-se a realização de estimativa de esforço em projetos de reengenharia de software reais, 5 organizações de desenvolvimento de software foram visitadas e a pesquisa foi conduzida com os responsáveis pela tarefa de estimar esforço. Estas empresas foram selecionadas de maneira a se obter uma maior diversidade nas informações capturadas.

Na Seção 3.3.1 foi relatada a seleção das organizações e dos participantes. A seguir são brevemente descritas estas 5 organizações. Todos os dados referentes a elas são anônimos devido à questão de confidencialidade. As informações aqui apresentadas foram obtidas com base no roteiro de entrevistas, nos sites das empresas e informações repassadas via e-mail pelos responsáveis. Infelizmente nem todas possuíam sites ou concederam as informações, logo não foi possível obter o mesmo conjunto de informações para todas as empresas.

1. Organização A: é a divisão de serviços empresariais e tecnológicos globais de uma grande empresa multinacional, com sedes na Europa, Estados Unidos, Índia e América do Sul. Esta divisão é formada pela combinação de consultoria em Transformação e Modernização de Software com Desenvolvimento e Manutenção de Sistemas. A unidade visitada nesta pesquisa foi a de Porto Alegre, onde a área de Transformação e Modernização de Software atua realizando reengenharia de software para empresas de diversos setores como saúde, previdência e bancário, não só do Rio Grande do Sul, mas em todo país. Como atua no desenvolvimento de software para clientes externos, as estimativas são realizadas durante a pré-venda do projeto, e não devem ser modificadas no decorrer deste, principalmente no que diz respeito ao custo do projeto (diretamente derivado do esforço). Assim, o foco maior é na estimativa realizada nesta etapa de pré-venda, embora haja monitoramento durante o projeto.
2. Organização B: multinacional de desenvolvimento e consultoria de software com sedes na América do Sul, Europa e Ásia. Realiza desenvolvimento e reengenharia de software para empresas de diversos ramos da economia, tais como bancário, saúde,

mineração e siderurgia. Embora seus clientes seja predominantemente externos, eventualmente atua no desenvolvimento de software para uso interno. Porém, o maior foco nas estimativas é na realização de estimativas em fases iniciais do projeto, que assim como na Organização A, também possuem preço fixo.

3. Organização C: é um dos maiores grupos empresariais multimídia do país. Nas mídias tradicionais, suas emissoras de televisão e de rádio e seus jornais, presentes em todas as plataformas, são líderes de mercado no Rio Grande do Sul e em Santa Catarina. Em relação ao desenvolvimento de software, esta organização possui uma unidade responsável por desenvolver e realizar reengenharia tanto dos sistemas que são usados internamente (como sistemas de recursos humanos), quanto dos sistemas disponibilizados para o público em geral. Por desenvolver software para um cliente interno, as estimativas dos projetos podem ser alteradas e refinadas no decorrer do projeto.
4. Organização D: empresa especializada em processamento e serviços para a indústria de meios de pagamento. Atua no desenvolvimento de sistemas para cartões convênio e gestão de frotas. Hoje atua como processadora *full service*, de âmbito nacional, focada na operacionalização de cartões de crédito para Emissores, Adquirentes e Bandeiras. Assim como a Organização C, esta empresa desenvolve e realiza reengenharia de sistemas para uso interno e produtos para empresas contratantes. A flexibilidade de prazos permite que a estimativa seja realizada em tempo de execução do projeto.
5. Organização E: esta empresa é especializada na comercialização de software de gestão para diversos setores da economia, tais como agroindústria, bancário e manufatura. Por possuir produtos prontos para comercialização, a área dita de reengenharia é responsável por realizar customizações destes produtos de acordo com a necessidade dos clientes. A estimativa é realizada no momento da venda e não é alterada posteriormente.

A Tabela 4.1 apresenta uma caracterização geral dos participantes da entrevista, de acordo com a organização em que atuam. A tabela apresenta o cargo e o tempo de experiência no mesmo de cada participante, e a coluna "Fase" indica a fase da Coleta de Dados que este participante atuou, de acordo com o apresentado na seção 3.3.2.

## 4.2 Resultados

Nessa seção são detalhados os resultados desta pesquisa qualitativa. Foram identificadas as principais atividades relacionadas a estimativa de esforço em projeto de reengenharia, além de quatro categorias que representam os principais fatores relacionados

Tabela 4.1 – Caracterização dos Entrevistados por Organização - Estudo de Campo Inicial

<b>Org.</b>	<b>Participante</b>	<b>Cargo</b>	<b>Tempo de Exp. no Cargo</b>	<b>Fase</b>
A	P1	Consultor de Tecnologia	7	1
	P2	Gerente de Projetos	10	1
B	P3	Líder de Métricas	10	1 e 2
C	P4	Gerente de Projetos	2	1 e 2
D	P5	Coordenador da Fábrica de SW	5	1
	P6	Coordenador da Fábrica de SW	4	1
	P7	Gerente de Projetos	10	2
E	P8	Gerente de Projetos	5	1

ao projeto de reengenharia, que influenciam direta ou indiretamente na estimativa de esforço. As categorias relacionadas a fatores de influência agrupam desafios e práticas de estimativa de esforço. Os desafios são ocorrências que dificultam a estimativa de esforço, podendo levar a uma menor acurácia desta estimativa. As boas práticas são ações realizadas para minimizar os problemas acarretados pelos desafios ou simplesmente para melhorar o processo de estimativa, buscando obter uma melhor acurácia.

Os desafios e práticas representam aquilo que os entrevistados acreditam que afeta positivamente ou negativamente a estimativa de esforço em projetos de reengenharia e que, portanto, devem ser levados em consideração. A identificação dessas informações é relevante para o contexto de reengenharia de software, no sentido de que identificar desafios e boas práticas para contorná-los apoia a entrega do projeto no tempo e orçamento determinados.

Uma descrição detalhada de cada fator de influência é realizada a seguir. Com o objetivo de fundamentar as informações são apresentados trechos das entrevistas.

#### 4.2.1 Reengenharia de Software do Ponto de Vista das Organizações

Conforme apresentado na Tabela 3.1 (Questões Aplicadas nas Entrevistas - Primeira Fase), a primeira pergunta feita aos entrevistados era o que eles entendiam por projetos de reengenharia. Esta pergunta foi feita com o objetivo de verificar se o conceito de reengenharia das empresas estava alinhado com a literatura na área.

Este questionamento é interessante uma vez que trabalhos como [KG11], tratam do problema que existe na prática em definir os limites entre um projeto de reengenharia e um projeto de manutenção. Embora ambos sejam considerados Evolução de Software, cada um tem um escopo e um processo relacionado, que deve ser tratado separadamente. Porém, na prática vemos trabalhos como [DLPS02], que abordam a reengenharia como sendo um tipo de manutenção "massiva", por exemplo. Este tipo de confusão prejudica a pesquisa na área, uma vez que dificulta o estudo dos processos separadamente, como deve

ser feito. Assim, era importante para este trabalho saber se o que estava sendo analisado era realmente um processo de reengenharia.

Conforme definido no Capítulo 2, Reengenharia de Software envolve uma reestruturação completa em algum, ou vários aspectos de um sistema de software (linguagem de programação, arquitetura, dados, etc.), com o objetivo de adequar este sistemas às necessidades de negócio vigentes para a organização. O sistema após a reengenharia, também chamado de sistema alvo, pode manter exatamente as mesmas funcionalidades que tinha anteriormente ou sofrer acréscimos, também de acordo com as necessidades do negócio.

Neste estudo, de todas as organizações investigadas, apenas a Organização E não realizava reengenharia de fato. Esta organização, como informado anteriormente, trabalha com um catálogo de sistemas de gestão, voltados para diversos ramos de negócio. Tais sistemas, uma vez vendidos para os clientes, podem sofrer pequenas alterações, como exclusão, inclusão ou edição de algum relatório, para se adequar as necessidades do cliente. Assim, embora haja modificação do sistema, esta não é suficiente para se caracterizar como reengenharia, estando mais próxima de ser uma manutenção evolutiva. O trecho de entrevista abaixo apresenta as afirmações do entrevistado sobre o processo de "reengenharia" da organização.

*"A Organização E trabalha com software de gestão, então existe uma matriz em São Paulo que desenvolve esses produtos. Nós, como filial, desenvolvemos personalizações para os clientes que compram esses produtos maiores. Então, o que é reengenharia para nós: de acordo as necessidades do mercado chega para nós a necessidade de personalizar este produto, que nós chamamos de 'produto-pai', então existe um sistema já bem definido e de acordo com as necessidades deste cliente nós redesenhamos algumas funcionalidades".[P8 - Organização E]*

As demais organizações possuem divergência na nomenclatura que adoram para projetos de reengenharia. Esta nomenclatura pode variar de uma organização para outra ou na mesma organização, dependendo do que será realizado no projeto. Por exemplo, a Organização A trabalha com os conceitos de modernização e transformação, sendo a primeira uma reengenharia sem acréscimo ou mudanças de funcionalidades e a segunda envolvendo esta mudança. Abaixo são apresentados alguns desses conceitos.

*"[...] as vezes nós temos soluções de um cliente que estão em determinadas tecnologias mais antigas e a nós temos que modernizar para tecnologias mais novas, sem mudar nada do escopo que foi pré-definido, aí realmente eu estou apenas modernizando uma aplicação. Claro que tu podes levar daqui funcionalidades novas para, digamos, nessa modernização tu agregares um escopo de coisas novas, aí seria uma transformação".[P2 - Organização A]*

*"Aqui a gente tem vários projetos que normalmente fazem migração, as vezes de uma plataforma para outra ou mesmo de tecnologias ou versões diferentes para uma versão mais atualizada de uma mesma plataforma".[P3 - Organização B]*

#### 4.2.2 Processo de Estimativa de Esforço em Reengenharia

Um dos pontos de interesse deste estudo era entender como é realizada a estimativa de esforço em projetos de reengenharia na prática e com base no relato das experiências coletadas, elaborar um modelo que auxilie na realização desta atividade. Para isto, os entrevistados foram questionados sobre como e quando realizavam estimativas em projetos de reengenharia, e também sobre os detalhes desta estimativa (técnicas utilizadas, pessoas envolvidas, ferramentas de apoio, documentos gerados e consumidos, parâmetros).

Em relação aos dados capturados neste estudo, foram identificadas as seguintes atividades de estimativa de esforço:

1. Realização da Estimativa
2. Comparação e Validação da Estimativa
3. Monitoramento e Calibragem da Estimativa
4. Atualização da Base Histórica e Aprendizagem

A seguir, estas atividades são melhor detalhadas.

##### Realização da Estimativa

Trata-se da atividade principal de estimativa de esforço, uma vez que consiste na aplicação de uma ou mais técnicas para derivar o esforço para um projeto.

Uma das principais motivações para realização desta pesquisa era o de verificar se havia alguma diferença nesta atividade para projetos de reengenharia, em relação a mesma atividade realizada para outros tipos de desenvolvimento de software, como projeto de desenvolvimento de software novo (conforme descrito no Capítulo 2, Seção 2.2.1).

Conforme apresentado na caracterização das organizações, algumas delas (Organização A e B), trabalham com projeto de escopo fechado, ou seja, no início do projeto é elaborado o escopo do mesmo, onde estão descritas as funcionalidades e requisitos necessários para ter o sistema no ar. Com o escopo definido, é realizada a estimativa de esforço, que posteriormente deriva as estimativas de custo e de tempo. Com isso, prazo e orçamento são então negociados e aprovados.

As técnicas aplicadas para realização da estimativa variam de acordo com a empresa, podendo ser usada uma ou mais. No Capítulo 2 foram apresentadas as principais técnicas de estimativa de esforço existentes, e uma lista de trabalhos que aplicam essas técnicas pode ser encontrada no Apêndice A.

O principal diferencial é que enquanto empresas, como a Organização A, selecionam a técnica de acordo com o projeto a ser realizado, as demais (B, C e D) trabalham com uma ou mais técnicas fixas, que são aplicadas para todos os projetos de reengenharia, impreterivelmente.

Apesar da estimativa de esforço ser um ponto crucial para o projeto, toda a responsabilidade envolvida nesta atividade fica a cargo da pessoa que está realizando a estimativa. Essa pessoa seleciona e aplica as técnicas, valida os resultados e entrega para o cliente (geralmente não o esforço estimado, e sim o custo e o prazo). A exceção neste estudo é a Organização A, onde existe uma equipe externa à organização, responsável por verificar e validar as estimativas antes de elas serem repassadas para o cliente.

### Comparação e Validação da Estimativa

Esta atividade foi identificada das Organizações A e D, onde mais de uma técnica é utilizada para realização das estimativas. Assim, várias técnicas são aplicadas, gerando cada uma um valor para estimativa, estes valores são comparados e avaliados para se derivar um único valor. Esta atividade, quando realizada, ocorre imediatamente após a Realização da Estimativa.

No caso da Organização A, o responsável pela estimativa decide empiricamente qual o valor de estimativa utilizar. Enquanto que na organização D, é utilizada a técnica PERT (*Program Evaluation and Review Technique*). Esta técnica consiste em descobrir a duração de uma atividade baseando-se em três estimativas possíveis para a atividade: estimativa Otimista (O), Pessimista (P) e Mais Provável (MP). A combinação dessas três possibilidades é o grande diferencial da técnica PERT, pois ela pondera as incertezas e riscos envolvidos na atividade [PP01].

A utilização de uma técnica formal garante, segundo os responsáveis por estimar na Organização D, mais segurança e assertividade nos valores.

### Monitoramento e Calibragem da Estimativa

Esta atividade foi relatada pelos participantes das Organizações A e B, que são as organizações que trabalham com projeto de escopo fechado. Nestes projetos, conforme relatado anteriormente, a estimativa de esforço (e conseqüentemente a de prazo, custo e recursos) é realizada no início do projeto (no caso da Organização A é feita em tempo de pré-venda do projeto e no caso da Organização B após a realização de um projeto preliminar).

Assim, por mais já tenham sido fixados valores junto ao cliente, é importante monitorar se o que foi estimado em cada etapa está sendo cumprido e caso não esteja o porquê disso, podendo, com isso, identificar e tratar desvios no processo. Esta atividade, quando

realizada, ocorre em marcos do projeto definidos de acordo com o processo, geralmente no início ou final de cada módulo de desenvolvimento.

A Organização C não realiza calibragem, pois ela efetivamente realiza a estimativa a cada etapa do projeto.

A Organização D, por trabalhar com escopo aberto, permite que durante o projeto, os desenvolvedores solicitem alteração de prazo, mediante justificativa. Neste momento, a atividade em questão é avaliada e um novo prazo é estabelecido com base em um acordo entre gerente e desenvolvedor. Não é utilizada nenhuma técnica formal para definir esta alteração de prazo, geralmente o próprio desenvolvedor "estima" as horas adicionais e informa para o gerente.

### Atualização da Base Histórica e Aprendizagem

Todas as organizações analisadas possuem uma base histórica de projetos de reengenharia previamente realizados. Essa base consiste em planilhas Excel com os dados dos projetos anteriores, tais como (esforço estimado, esforço realizado, produtividade da equipe, número de defeitos, número de *bugs*, etc). Os projetos prévios são geralmente categorizados de acordo com suas características como tipo de negócio, tecnologias utilizadas, porte, etc.

Apenas a Organização D aplica algum tipo de automatização neste processo, uma vez que esta organização desenvolveu uma ferramenta de apoio à estimativa, onde são cadastrados os parâmetros para cálculo da mesma para cada atividade do projeto, exemplos de parâmetros cadastrados são: tipo de unidade do sistema a ser desenvolvida (cadastro, tela, relatório, etc), experiência mínima do desenvolvedor (júnior, pleno ou sênior), se envolve cálculo de valores financeiros, de faz *download* ou *upload* de arquivos, etc . Todos esses parâmetros são multiplicados por pesos, que variam de acordo com as características do projeto. Tais pesos são derivados dos projetos prévios e revisados anualmente pela equipe de gerência. Assim, além de manter atualizada a base histórica de projetos, ainda é realizada uma espécie de aprendizagem sobre os projetos mantidos nesta base.

#### 4.2.3 Fatores que Impactam na Estimativa de Esforço em Projetos de Reengenharia de Software

Esta seção apresenta os fatores identificados como impactantes na estimativa de esforço em projetos de reengenharia. Estes fatores foram agrupados em 4 principais categoria: sistema legado, equipe, cliente e projeto.

Dentro de cada categoria os fatores são basicamente divididos em desafios e boas práticas (soluções para lidar com os desafios, ou simplesmente uma prática para melhorar o

processo). A identificação destes fatores é complementar ao processo de estimativa identificado acima, uma vez que, conforme foi apresentado, em termos de atividades não há diferença entre o que é realizado em estimativa de esforço para um projeto de desenvolvimento de software novo, em relação a um projeto de reengenharia. A diferença efetiva esta justamente nos fatores particulares a projetos de reengenharia, que impactam em como as atividades são realizadas.

## Sistema Legado

Uma vez que a reengenharia trata da análise e reconstrução de um sistema existente, o estado em que se encontra este sistema antes da realização da reengenharia impacta diretamente no esforço que deverá ser empregado no projeto. Neste contexto, um dos principais desafios enfrentados é a **falta de documentação do sistema**, isto é, não existem documentos que permitam o entendimento do sistema, tais como especificações, documentos de arquitetura, de *design*, dentre outros. Também é caracterizado como falta de documentação a existência de documentos defasados que, por mais que existam, não refletem o estado atual do sistema. A falta de documentação impacta de várias maneiras a estimativa de esforço, pois retira a vantagem do responsável pela estimativa em ter todas as características do sistema identificadas, sem que se tenha que realizar elicitação de requisitos, além de elevar o esforço para realização do projeto com tarefas de engenharia reversa. Todas as organizações analisadas vivenciam essa situação, abaixo são apresentados alguns trechos de entrevista que ilustram isso.

*"Nenhum cliente...quase ninguém tem nada. Dificilmente um cliente que tem as coisas documentadas". [P2 - Organização A]*

*"A maioria não tem documentação e quando tem, as vezes, ela é desatualizada, o que acaba mais atrapalhando do que ajudando. Então, é praticamente como se não tivesse documentação". [P3 - Organização B]*

Uma vez iniciada a análise do sistema legado a **qualidade em que se encontra o código fonte** também irá afetar a estimativa. Características como legibilidade, precisão nas regras de negócio, estrutura de dados, complexidade, podem aumentar ou diminuir o esforço.

*"[...]em alguns momentos eu tenho o código escrito, nós até estamos com uma situação dessas, o cliente tem um legado, onde tem serviços que na visão dele podem ser utilizados por outros sistemas, mas na prática, se tu fores olhar tá cheio de if lá dentro, se for tal coisa faz isso...não é um serviço que de fato eu possa reutilizar". [P3 - Organização B]*

A Figura 4.1 apresenta graficamente os fatores que afetam a estimativa de esforço, relacionadas ao sistema legado. Note que não são identificadas práticas neste caso, pois as práticas para minimizar esses desafios são elencadas em Projetos (Seção 4.4).

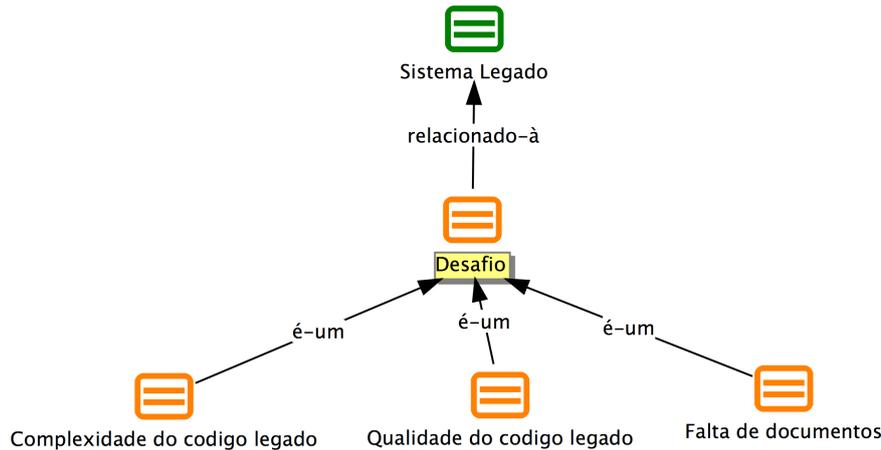


Figura 4.1 – Grafo da Categoria Sistema Legado

## Cliente

Estimativa de esforço geralmente é uma tarefa realizada pela equipe de desenvolvimento de software e chega indiretamente para o cliente, através da estimativa de custos do projeto (estimativa de esforço é a métrica com maior influência no custo de projetos, em função disto, muitas vezes são tratadas como sinônimos [JS07]). Porém, esta pesquisa identificou que há um crescente **interesse do cliente em saber como a estimativa é realizada**, em termos de qual técnica foi empregada e a se esta técnica tem algum tipo de "embasamento teórico".

*"[...] tem alguns momentos que antes de tu colocares uma proposta par ao cliente existe um momento de aprovação. Essa aprovação...claro, isso depende muito da alçada, do tipo do projeto, de quanto nós estamos falando, digamos assim, certamente se questiona 'como é que tu chegaste nessa estimativa' [...]". [P2 - Organização B]*

Outro desafio relacionado ao cliente é quando este **subestima o esforço a ser empregado no projeto**, em função das características do sistema, mas sem conhecer a qualidade do código fonte e sua influência na estimativa de esforço (Seção 4.1). Este problema é comum a diversos tipos de projetos de software, mas é particularmente agravado em projetos de reengenharia, uma vez que o cliente já "conhece"o sistema e tende a avaliar o tamanho do projeto em relação às funcionalidades que este sistema (o legado) apresenta, mas desconhecendo a qualidade por trás do mesmo.

*"estimar também está associado a custo, e passar isso para o cliente. As vezes ele não entende quando o analista diz "mas isso tem uma série de riscos envolvidos", ele fala "não, mas esse projeto é pequenininho, deve ser barato". Não, tem uma série de riscos envolvidos ali que podem comprometer uma estimativa que tu dás em termos de custo". [P3 - Organização B]*

"o cliente acha que tu só tens que mudar a tecnologia, mas não, tu tens que olhar com mais detalhes, em algum momento mostrar para ele que existe essas questões no projeto que nós podemos tratar e as vezes tem que negociar muitas vezes, até uma mudança de escopo ou algo do tipo, porque ele imaginou que era só mudar a tecnologia, mas nós também não vamos pegar essa tua coisa ruim e jogar em uma tecnologia moderna, nós queremos implementar uma coisa melhor para ele né". [P2 - Organização A]

Para tentar minimizar os desafios relacionados com o cliente, busca-se **integrar o cliente a este processo de estimativa de esforço**, não no sentido de fazê-lo realizar esta estimativa, mas em deixar transparente para ele o processo adotado e estar disponível para questionamentos acerca deste processo.

"nós não costumamos abrir as estimativas para o cliente, mas tem cliente que quer saber os valores para se sentir mais confortável com o projeto. Nestes casos, a gente abre os valores para ele". [P1 - Organização A]

A Figura 4.2 apresenta graficamente as associações da categoria Cliente.

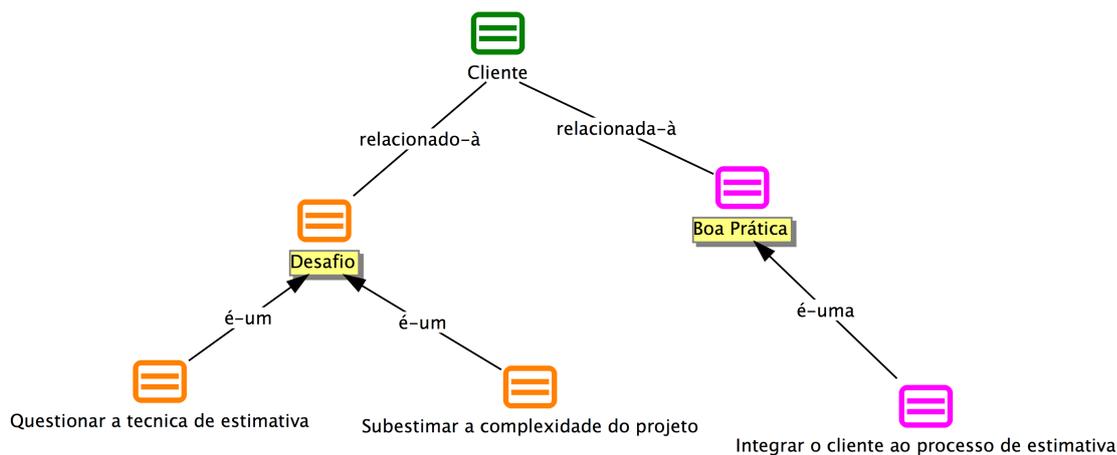


Figura 4.2 – Grafo da Categoria Cliente

## Equipe

Características da equipe de desenvolvimento de software também afetam diretamente a estimativa de esforço, não só de maneira direta, isto é, fornecendo parâmetros usados no cálculo dessa estimativa como a **experiência na arquitetura, na linguagem de programação, no tipo de negócio a ser tratado no projeto**, além da **produtividade**. Tais parâmetros são semelhantes aos utilizados em outros tipos de projeto de desenvolvimento de software, como manutenção, desenvolvimento de software novo e aplicações web. Além desses parâmetros, existem diversos outros relacionados aos recursos huma-

nos como conhecimento, perfil, etc. Mas nesse trabalho são apresentados apenas os que foram mencionados nas entrevistas.

O principal desafio relacionado a equipe de desenvolvimento está diretamente relacionado a técnica de estimativa de esforço utilizada, sendo que a mais problemática é também a mais utilizada, que é a estimativa de esforço por especialista. O fato é que o maior desafio ao utilizar essa técnica é quando o **especialista não tem experiência suficiente para estimar o esforço** para um dado projeto. Isso pode ser minimizado se houver mais de um especialista, que auxilie o inexperiente com seu conhecimento, a partir de atividades de **compartilhamento de conhecimento sobre estimativa**.

*"hoje é bem isso que a gente vem trabalhando, tentar trabalhar com o know how dos especialistas, aliando quem é mais experiente, até para prepará-los e ali chegar em um consenso. É legal tu ouvires a insegurança do menos experiente, até pra tu teres uma margem, isso evita que tu superestimes o trabalho, o esforço". [P4 - Organização C]*

Outra boa prática em relação a equipe de desenvolvimento é **investir em treinamento ou em contratação de profissionais com conhecimento em mais de uma técnica de estimativa**.

*"é uma área que existe um pessoal de métrica envolvido, então tem pessoas certificadas em PF, por exemplo, e outras técnicas também que apoiam esse processo tanto de pré venda quanto para execução". [P1 - Organização A]*

*"Nós estamos estudando agora para aprender uma nova técnica, que já é uma norma ISO, que é a COSMIC, ela é um pouco diferente, mas ela tem alguma coisa a ver com PF, que a gente já usa, mas ela tem uma aplicação melhor". [P2 - Organização A]*

A Figura 4.3 apresenta as associações identificadas na categoria Equipe de desenvolvimento.

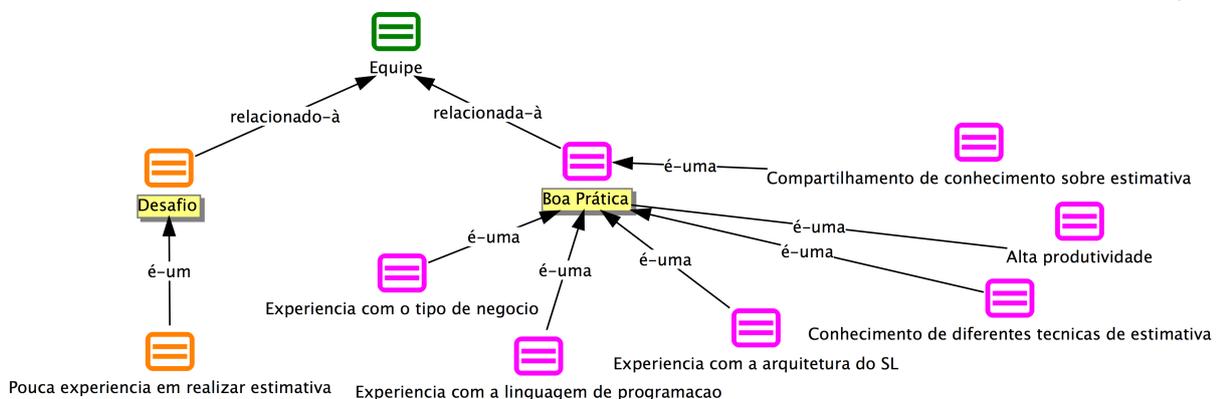


Figura 4.3 – Grafo da Categoria Equipe

## Projeto

Por fim, mas não menos importante estão os fatores relacionados ao projeto de reengenharia em si, apresentados graficamente na Figura 4.4.

Dentre os desafios identificados temos inicialmente o **porte do projeto**, que geralmente é médio ou grande. Tal fator, combinado com a necessidade que algumas organizações possuem de realizar as **estimativas antes do fechamento do contrato** (estimativa preliminar), afeta as estimativas no sentido de obrigar o profissional responsável a calculá-la **sem a possibilidade de ter um entendimento completo do escopo do projeto**. Uma vez os projetos de reengenharia são geralmente de médio ou grande porte, uma análise minuciosa do sistema legado para extrair as características necessárias para realizar uma estimativa mais assertiva **não é viável em termos de tempo**.

*"São dois projetos bem grandes, um projeto tem dois anos de duração e o outro tem quase 5 anos. Mas assim, esse de 5 anos a gente está falando da migração não de uma aplicação, são mais de 100 aplicações que tem que ser migradas dentro desse período, onde deve haver todo um aprendizado dessa estrutura do cliente e isso que vai embasar justamente as estimativas para as outras ondas de migração que a gente vai ter."* [P2 - Organização A]

*"A estimativa é feita antes do projeto estar fechado. Como aqui a gente trabalha com uma solução... a gente vende uma solução, eu tenho que saber botar preço na coisa, então o botar preço vem antes do início".* [P1 - Organização A]

*"as vezes tu tem que adiantar algumas questões em tempo de pré-venda, antes de tu venderes e o cliente não te dá tempo suficiente para tu fazeres um bom levantamento de requisitos, um entendimento da solução, para tu ires no detalhe."* [P2 - Organização A]

Para contornar esse desafio as empresas negociam junto ao cliente a **realização de um projeto piloto, ou projeto preliminar**, onde um módulo do sistema é analisado, com o objetivo de se conhecer de maneira geral o sistema e usar essas informações para embasar a estimativa de esforço para o restante do projeto. Além disso, também procura-se ter um **processo bem definido**, não necessariamente exclusivo para projetos de reengenharia, mas que funcione adequadamente para este tipo de projeto e no qual a equipe esteja apta a trabalhar. Outro fator importante é **calibrar constantemente as estimativas**, na medida em que o projeto vai sendo desenvolvido.

*"Eu não posso lá na frente dizer: eu vou te estimar isso aqui e aquilo ali é o que vai embasar todo meu trabalho lá na frente. Eu tenho que rodar um primeiro ciclo e fazer uma análise do que aquilo ali tá sendo...do resultado daquele processo, porque justamente a gente não tem documentação para se basear".* [P2 - Organização A]

*"Pra reengenharia exclusivamente não, mas para processo de desenvolvimento tem uma metodologia própria, um processo de desenvolvimento já da empresa".* [P3 - Organização B]

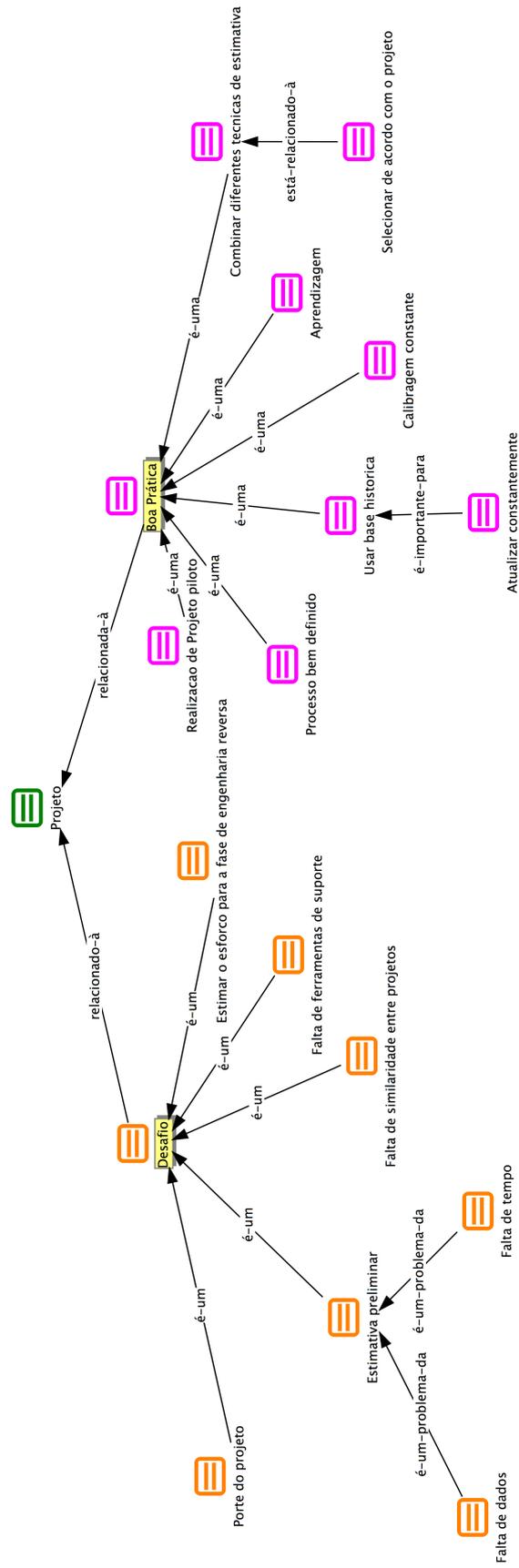


Figura 4.4 – Grafo da Categoria Projeto

"A Organização A tem um processo escrito que é o EAD que é baseado em práticas que são utilizadas fora, na Organização A como um todo. É uma prática que a gente já implementa faz um tempo". [P1 - Organização A]

"Aquele performance que eu tenho naquele primeiro processo, naquele primeiro ciclo, eu tenho que olhar a minha estimativa lá na frente, reestimar o que tá na frente, vê se eu tenho desvios do que eu planejei lá no início e isso ir se retroalimentando e calibrando a estimativa que inicialmente a gente fez." [P2 - Organização B]

Outra boa prática adotada pelas organizações é a de **utilizar uma base histórica com as informações de projetos passados** como insumo para estimar dados de projetos novos. Essa prática é recomendada pela literatura vigente como uma boa prática para realização de estimativa de esforço de maneira geral, ou seja, em todos os tipos de desenvolvimento do projeto. No contexto de reengenharia, ela esbarra do desafio de encontrar projetos similares para basear as estimativas, **já que os projetos apresentam características bastante distintas entre si.**

"Como eu tenho a base de dados históricos, então eu tenho uma história que me diz que o meu tempo de análise de negócio leva em média tanto, mas a gente sabe que isso varia de projeto pra projeto, tem projeto que é dois meses, tem projeto que é um ano. Mas a gente consegue olhar e tirar uma média". [P5 - Organização D]

"O que geralmente a gente faz é: faz um trabalho onde, baseado em histórico, o quanto a gente leva de tempo conhecendo as características básicas dessas soluções que tem que ser modernizadas, monta uma solução, colocando para o cliente que vai existir um período onde vão ser avaliadas essas aplicações e ao final desse período a gente consegue calibrar um pouco mais essas estimativas". [P3 - Organização B]

"[...] como esses projetos não são corriqueiros aqui dentro da empresa, ele tem menor índice para a base histórica, tem menor números e projetos para poder formar uma base histórica consistente. E cada vez que tem eles são diferentes, porque eu preciso classificar esses projetos em vários setores pra então eu fazer esse levantamento, tenho que caracterizar eles em cada tipo de diferente projeto, pra depois poder compará-los, pra poder ter uma base histórica. E como são poucos eu acabo tendo uma base histórica de projetos diferentes, que eu não consigo comparar eles, eles são bem diferenciados, a produtividade deles, tu não consegue comparar, tu não consegue usar uma medida padrão, a não ser que você tenha um número muito grande de projetos pra poder obter uma estatística confiável, ter um numero grande de amostra pra poder utilizar esse histórico como uma métrica futura, com base numa melhor estimativa depois". [P2 - Organização A]

A Organização D, além de usar a base histórica para embasar as estimativas, ainda utiliza esta base para aperfeiçoar a própria técnica de estimativa, em um processo de **aprendizado** contínuo.

*"Então a gente desenvolveu em 2006 uma estimativa que era mais ou menos um chutômetro no início e com o tempo a gente vai atualizando aquela base de dados e melhorando a técnica". [P5 - Organização D].*

*"O sistema calcula, ele faz a multiplicação da quantidade de entradas pelo peso da tecnologia no momento atual, com o passar dos anos a gente faz uma reestimativa daquilo e ajusta os pesos pro ano seguinte, então uma base histórica desses pesos, baseada na base histórica do projeto, e todo ano é reajustado de maneira que aquilo ali fique o mais enxuto possível". [P6 - Organização D]*

Além de utilizar base histórica, outra prática apontada na literatura, e adotada pelas organizações A e D, é a **utilização de mais de uma técnica para estimar esforço**. Neste caso, duas ou mais técnicas são executadas em paralelo e o resultado obtido é comparado e analisado para gerar uma estimativa de esforço mais assertiva. As organizações que aplicam essa prática ressaltam, ainda, que **selecionar as técnicas de acordo com as características do projeto** a ser estimado melhora ainda mais a qualidade das estimativas.

*"Assim, depende muito do tipo de projeto. Se a gente consegue ter acesso e tem subsídios de ter informações, conhecer o sistema que já existe, consegue enxergar telas, consegue enxergar campos que tem que ser utilizados, consegue enxergar os dados ali por trás, a gente pode utilizar PF, é uma técnica que a gente utiliza. Daqui a pouco a gente entra numa situação que a gente não tem uma visão, pois são muito sistemas e tu não tem as vezes até a possibilidade de ir tela a tela e entender esses dados, então você pode utilizar, por exemplo, contagem por linhas de código. Trabalhar com UCP também é algo comum, então isso depende muito da situação, do que a gente tem em mãos pra trabalhar e com base nisso, do que a gente receber do cliente de informação o pessoal vai...bom, pra esse caso vamos utilizar tal métrica, pra esse caso vamos usar esse tipo de estimativa, etc." [P2 - Organização A]*

O grande desafio de usar várias técnicas, ou até mesmo somente uma, é a **falta de apoio ferramental para realização de estimativa de esforço**, que em grande parte é feita em planilhas Excel, configuradas pela própria equipe. Essa escassez de ferramentas também se estende para a consulta de dados em base histórica.

*"Nós usamos uma planilha de estimativa, uma planilha de contagem de PF, desenvolvida por nós mesmos para adequar a técnica de PF na nossa planilha, então alguns requisitos de contagem...depende do cliente, o cliente as vezes tem processos diferentes de contagem".[P2 - Organização A]*

*"S – E você tem alguma ferramenta que te auxilie nessa estimativa de esforço? Porque quem dá a estimativa é o especialista, mas aí pra gerenciar essas estimativas, esses dados, tem alguma ferramenta que te auxilie? J – Não." [P4 - Organização C]*

Por fim, a segunda rodada do estudo, que tratou de **estimativa de esforço na fase de especificação de requisitos em projetos de reengenharia** constatou que este

é um dos grandes desafios ainda a serem tratados efetivamente pelas empresas e que atualmente tais empresas não adotaram uma prática padrão para tratar disso.

*"Sabe que essa é uma das fases mais difíceis que a gente tem, porque pra fazer essa estimativa de quanto eu vou gastar, definir os requisitos, levantar os requisitos, escrever os requisitos, validar os requisitos e depois ter eles prontos pra desenvolver, eu dependendo do que eu recebi, e o que eu pode ter n variáveis e não estar consolidado, que é o que acontece na maioria das vezes [...]. Essa estimativa é uma das mais complicadas que eu vejo, porque ela é muito aberta ainda". [P7 - Organização D]*

### **4.3 Conclusões do Capítulo**

No estudo apresentado neste capítulo, foram identificados, via pesquisa qualitativa, atividades, desafios e práticas relacionados a realização de estimativa de esforço em projetos de reengenharia de software. Tais informações foram coletadas junto a especialistas de quatro diferentes organizações de desenvolvimento de software através de entrevistas semi-estruturadas contendo questões abertas, que forneceram flexibilidade para captação de maiores detalhes sobre o contexto em que cada entrevistado estava inserido. Para análise desses dados foi empregado o Método de Comparação Constante, que possui um conjunto de procedimentos de coleta e análise de dados a fim de desenvolver conceitos e comparando continuamente incidentes específicos nos dados [SEA08]. O pesquisador refina esses conceitos, identifica suas propriedades, explora suas relações uns com os outros e integra-os em um modelo explicativo coerente.

Apesar da relevância das informações identificadas neste estudo, percebeu-se uma escassez de detalhes de como o processo, os desafios e as práticas podiam ser relacionados a um projeto em particular de reengenharia.

Assim o próximo capítulo apresenta um estudo de caso de três projetos de reengenharia de software, onde podem ser ilustradas muitas das situações identificadas neste estudo, que permitiram um maior entendimento do processo analisado.

## 5. APLICAÇÃO E RESULTADOS DO ESTUDO DE CASO

Neste capítulo são apresentados os resultados referentes ao estudo de caso.

A realização deste estudo permitiu a confirmação dos resultados obtidos no estudo de campo inicial no contexto de projetos específicos de reengenharia de software. Na Seção 5.1 são apresentadas as configurações e o método aplicado no estudo de caso. Na Seção 5.2 é apresentada uma visão geral do processo de reengenharia da organização. Na Seção 5.3 é apresentada a caracterização dos projetos analisados. Na Seção 5.4 é apresentada a caracterização dos respondentes da pesquisa. Por fim, na Seção 5.5 são apresentados os resultados do estudo de caso.

### 5.1 Configuração do Estudo e Método de Pesquisa

Este estudo foi desenvolvido na unidade de desenvolvimento de software apresentada no Capítulo 4 como Organização A. Esta unidade é uma divisão de serviços empresariais e tecnológicos globais de uma grande empresa multinacional com sedes na Europa, Estados Unidos, Índia e América do Sul. Esta divisão é formada pela combinação de consultoria em Transformação e Modernização de Software com Desenvolvimento e Manutenção de Sistemas.

O estudo de caso foi realizado na filial de Porto Alegre, onde a área de Transformação e Modernização de Software atua realizando reengenharia de software para empresas de diversos setores como: saúde, previdência e bancário, não só do Rio Grande do Sul, mas de todo país. O estudo não considerou as demais unidades de desenvolvimento da organização.

Atualmente esta unidade não possui avaliação de maturidade, mas esta se preparando para ser avaliada no nível 3 do CMMI (*Capability Maturity Model Integration*).

### 5.2 Processo de Reengenharia

A Organização A utiliza na execução dos seus serviços de modernização e desenvolvimento de aplicações uma metodologia bem definida de desenvolvimento de software.

Essa abordagem para prestação dos serviços de desenvolvimento, modernização, gerenciamento e manutenção de aplicações é embasada em uma metodologia própria que compila toda experiência e resultados positivos adquiridos ao longo dos anos pela orga-

nização, em diversos países. Ela é adotada globalmente por todas as equipes nos mais variados clientes o que permite aprimorá-la às diversas necessidades de projetos.

Tal metodologia é operacionalizada através de um *framework* integrado e mantido globalmente chamado EDGE (*Enabling Delivery and Global Excellence*), onde estão definidos os processos necessários para gerenciar todas as atividades envolvidas no desenvolvimento de sistemas.

Este *framework* é baseado em ferramentas e métodos corporativos, melhores práticas e é totalmente alinhado com os requisitos dos modelos ISO-9001:2008 [ABdNT08], SEI-CMMI Nível 5 [CMM10], PMBOK [PMI14] e Prince2™ [BEN09]. Ele contém processos, formulários, modelos, procedimentos e exemplos para suportar diversos tipos de trabalho.

Segundo informações apuradas durante a imersão na organização, o processo é igual para todas as outras unidades de desenvolvimento e é baseado no processo da unidade que possui o maior nível de maturidade de acordo com o modelo CMMI.

Por trabalhar somente com clientes externos à organização, todo o planejamento dos projetos é feito em tempo de pré-venda da solução. Ou seja, a análise preliminar do sistema legado e o levantamento de novos requisitos para o projeto (se for o caso), que servem como base para a estimativa de esforço, são realizadas neste momento. Além da estimativa de esforço, são derivadas as estimativas de custo e de prazo, no momento da pré-venda. As fases do projeto são: planejamento, análise, construção, testes, homologação e implantação. Como a estimativa ocorre antes do início do projeto, ela não é considerada como sendo uma atividade de projeto. Porém, como é a atividade de maior interesse para esta pesquisa, foi adicionada uma fase extra ao processo, chamada de Negociação do Projeto. Como o objetivo não é relatar o processo de reengenharia existente, são apenas explicadas brevemente cada uma das fases:

1. Negociação do Projeto: nesta fase é realizada a análise do problema a ser solucionado, esta análise pode ser feita com base no sistema legado, em documentos de projeto, em entrevistas com as partes interessadas no sistema (*stakeholders*) ou em uma combinação de todas essas técnicas. A partir desta análise é proposta e estimada a solução a ser desenvolvida. Caso a solução seja aceita pelo cliente, um acordo é estabelecido e dá-se início ao projeto. Um maior detalhamento do processo de estimativa da organização é apresentado na seção 5.5.1.
2. Planejamento: são definidos junto ao cliente os padrões a serem seguidos no projeto, a alocação de recursos é validada, são estabelecidos pontos de controle e o código fonte do sistema legado é repassado para a organização onde é configurado um ambiente para sua execução e análise.
3. Análise: é realizada a engenharia reversa do sistema legado. Além disso, caso a reengenharia envolva o acréscimo de novas funcionalidades, estas são levantadas junto aos *stakeholders* com técnicas de entrevista.

4. Construção: o sistema alvo é desenvolvido conforme determinado na análise. Nesta fase pode ocorrer migração de dados, caso o projeto de reengenharia envolva reestruturação de dados.
5. Teste: o sistema alvo é testado, sendo que um dos testes realizados refere-se a comparação com o sistema legado.
6. Homologação: definição e execução de testes junto aos usuários do sistema.
7. Implantação: o sistema é entregue e implantado no ambiente do cliente. São entregues os manuais, além dos demais documentos do projeto gerados.

### 5.3 Caracterização dos Projetos Analisados

Os projetos foram selecionados pela pesquisadora, juntamente com integrantes da organização. Como critérios de seleção, optou-se por projetos com diferentes graus de reengenharia, conforme é apresentado a seguir. Além disso, foi solicitado junto a organização a análise de um projeto em andamento e de um projeto finalizado. A equipe da organização sugeriu, então, que fossem analisados dois projetos finalizados, pois eles tinham dois casos opostos do ponto de vista de estimativa, que acharam interessantes serem estudados.

O Projeto 1 é o projeto em andamento que tem como objetivo realizar a reengenharia de um sistema para uma empresa fornecedora de soluções de TI para instituições financeiras. O projeto consiste em reformular o canal de *Internet Banking*, além de estruturar o módulo de administrador de canais. A reengenharia deste sistema envolve a reestruturação total da camada de arquitetura, além do acréscimo de pequenas funcionalidades.

O Projeto 2 é um dos projetos já finalizados e enfrentou grandes problemas na realização de suas estimativas. O objetivo deste projeto foi realizar a reengenharia de um sistema de gestão de previdência privada para uma empresa de seguros. A reengenharia deste sistema envolvia a reestruturação de dados e de código do sistema legado, além do acréscimo de novas funcionalidades.

O Projeto 3 é o segundo projeto já finalizado e foi considerado um caso de sucesso em termos de estimativa. O objetivo deste projeto era a reengenharia de um sistema de gestão de seguros de vida, para a mesma empresa cliente do Projeto 2. A reengenharia deste sistema envolvia apenas a reestruturação de código do sistema legado.

## 5.4 Caracterização dos Respondentes e sua Participação

A pesquisa foi desenvolvida de acordo com a metodologia apresentada no Capítulo 3. Foram realizadas entrevistas com 6 profissionais, selecionados de acordo com seu envolvimento nas atividades de reengenharia dos projetos analisados. Foram entrevistados 2 consultores de tecnologia, 2 gerentes de projeto, 1 arquiteto e 1 analista de sistemas. Todos os participantes entrevistados possuem pelo menos 7 anos de experiência na área de Tecnologia da Informação, sendo o tempo médio de experiência de 15 anos. Os entrevistados possuem um tempo de atuação na organização entre 8 meses e 9 anos. Dentre estes participantes, apenas o P1 participou também do estudo de campo (apresentado no Capítulo 4).

As entrevistas tiveram uma duração média de 44 minutos (entre um mínimo de 30 minutos e um máximo de 1 hora e 10 minutos) e contaram com total disponibilidade e atenção dos participantes. Foram fornecidas todas as informações solicitadas, sempre respeitando a política de privacidade e confidencialidade da organização.

A distribuição dos participantes em relação aos projetos é apresentada na Tabela 5.1. A tabela também apresenta as atividades de estimativa de esforço realizada por cada entrevista no (s) projeto (s) em que participou (aram).

Tabela 5.1 – Papel dos Entrevistados nos Projetos

<b>Entrevistado</b>	<b>Papel</b>	<b>Projeto</b>	<b>Atividade</b>
P1	Consultor de Tecnologia	1, 2 e 3	estimar esforço em tempos de pré-venda
P2	Consultor de Tecnologia	2	estimar esforço em tempos de pré-venda
P3	Gerente de Projetos	1	monitorar as estimativas
P4	Gerente de Projetos	2 e 3	monitorar e calibrar as estimativas
P5	Arquiteto	2	calibrar as estimativas
P6	Analista de Sistemas	2	calibrar as estimativas

## 5.5 Elementos de Análise

A seguir apresentam-se os elementos analisados e as categorias obtidas com a análise dos dados do estudo de caso.

### 5.5.1 Processo de Estimativa de Esforço

De acordo com a Seção 5.2, o processo de estimativa de esforço se inicia na fase de pré-venda do projeto e a estimativa gerada embasa o orçamento e o cronograma

propostos. Sendo assim, esta é uma atividade de grande influência no fechamento do contrato de realização do projeto.

Nesta fase de negociação o cliente apresenta o problema e fornece os insumos para realização da estimativa, estes insumos podem ser o código-fonte do sistema legado, a documentação do sistema, uma lista de requisitos ou uma combinação destes elementos. A equipe de consultoria realiza, então, a análise do problema e dos artefatos fornecidos pelo cliente, com o objetivo de projetar uma solução e estimar o esforço necessário para desenvolvê-la.

A técnica utilizada para estimar varia de acordo com a natureza do projeto e com os artefatos fornecidos pelo cliente. Não há uma formalização desta relação técnica-projeto-artefatos e a escolha cabe diretamente ao responsável pela estimativa.

De acordo com os entrevistados P1 e P2, responsáveis por realizar as estimativa na fase de pré-venda, as técnicas são escolhidas dentre COCOMOII, *Use Case Points* e *Bottom-up*. As duas primeiras são técnicas já consagradas na literatura e, no caso da Organização A, são selecionadas de acordo com insumo que o cliente fornece para realização do projeto: a primeira é utilizada quando se tem acesso ao código-fonte do sistema legado e a segunda quando se tem acesso a uma lista de requisitos, de onde podem ser derivados casos de uso. Já a técnica *Bottom-up* foi desenvolvida pela própria organização e consiste em uma planilha que deriva a estimativa para projetos cujas características da solução proposta já são bem conhecidas pela organização. Assim, caso um projeto a ser estimado possua uma solução proposta semelhante a de um projeto já realizado, a planilha de estimativa *Bottom-up* é utilizada. Esta planilha deriva a estimativa para cada componente do sistema (por exemplo Java, Java Script, JSP, XML, etc), de acordo com os módulo do sistema, e para cada uma das etapas do processo de reengenharia. A soma das estimativas por etapa resulta na estimativa total do projeto. Por isso a técnica foi denominada *Bottom-up*.

Uma vez realizada a estimativa, ela é validada por uma equipe que envolve o time de métricas co-localizado no Rio de Janeiro (e que atente demandas de várias unidades da organização), gestores da organização e equipe de venda. Se aprovada, esta estimativa é apresentada para o cliente. O cliente pode, em tempo de pré-venda, solicitar alterações nestas estimativas.

Uma vez finalizada a fase de pré-venda e iniciado o projeto, a equipe de consultoria repassa as estimativas geradas (esforço, custo, prazo) para o gerente responsável pelo projeto. O papel do gerente em relação às estimativas é monitorar os valores estimados para cada fase de maneira a manter o planejamento inicial. Caso seja necessário, este gerente pode realizar ajustes nas estimativas, desde que acordados com o cliente. O ideal é que não sejam realizadas recalibrações de grande porte nas estimativas do projeto, uma vez que o acordo estabelecido com o cliente prevê escopo e orçamento fechados.

Estas são as etapas de estimativa comuns a todos os projetos da organização, não só os de reengenharia. Os projetos analisados neste estudo, seguiram em sua maioria

o previsto neste processo e, na medida que surgiram desafios que impediam o bom andamento deste processo, foram empregadas soluções com o intuito de contorná-los. Estas experiências são relatadas as próximas seções.

### 5.5.2 Fatores que impactam na estimativa de esforço em projetos de reengenharia de software

Esta seção apresenta os fatores identificados como impactantes na estimativa de esforço em projetos de reengenharia. Além das 4 categorias identificadas no estudo de campo inicial (sistema, cliente, equipe e projeto), este estudo identificou ainda a categoria "Organização", onde foram agrupadas as características relacionadas a organização, que impactam direta ou indiretamente na estimativa de esforço.

Assim como no estudo de campo inicial, nesta análise cada categoria é dividida em desafios e boas práticas identificadas nos projetos estudados.

#### Sistema Legado

Em todos os projetos analisados o sistema legado foi um fator de grande influência para a estimativa de esforço.

Os Projetos 2 e 3 enfrentaram o desafio clássico de projetos de reengenharia que é a **falta de documentação do sistema legado** e/ou **documentação de má qualidade**. Este fator dificulta a realização das estimativas, pois obriga os analistas a despenderem mais tempo na análise do sistema legado e em entrevistas com os clientes.

*"Ele basicamente era fazer uma mudança de tecnologia, ele era um projeto em VB e foi transcrito pra .NET e o nível de documentação dele era nada, absolutamente nada, no PROJETO 3 não tinha nada."*[P4 - Projeto 3]

*"em relação a estimativa, os principais problemas eram escassez de documentação e falta de conhecimento daquele domínio de negócio que se tá fazendo."*[P1 - Projetos 1 e 3].

Já o Projeto 1 teve como ponto forte o fato de haver **documentação disponível com alto grau de assertividade em relação ao sistema existente**, o que facilitou o entendimento do sistema e agilizou o processo de estimativa.

*"A documentação, quando o analista começa a fazer a análise, ele requisita toda a documentação que o cliente tem em termos de regra de negócio e dá para ser dizer que a documentação do cliente é uma documentação boa, então até hoje eu acho que não teve nenhuma atividade que o time tenha precisado para especificar que o cliente não*

tenha nenhuma documentação, em alguns casos foi detectado que a documentação estava desatualizada, mas poucos casos.”[P3 - Projeto 1]

**A qualidade e a complexidade do código legado** também foram fatores relevantes. No geral, a qualidade dos sistemas legados sob análise foi considerada ruim pelos especialistas, ao passo que a complexidade foi considerada alta, especialmente no Projeto 2, onde a equipe não tinha domínio da tecnologia legada. Esses fatores influenciaram no processo de estimativa, pois aumentaram o tempo gasto para analisar o sistema e extrair os dados necessários para aplicar as técnicas de estimativa.

*"Sem dúvida teve um peso muito grande, porque tudo que envolvia o legado...o sistema legado era muito mal feito, era muito enjambado, se é que eu posso usar essa palavra assim, porque ele tinha um time de pessoas que trabalhava dando manutenção nele e o pessoal fazia de qualquer jeito, então a gente nunca sabia exatamente o que tava lá, e como era tecnologia que a gente não conhecia, dataflex, a interpretação desse código legado era muito difícil, então muitas vezes a gente recorria ao time que dava sustentação para esse legado para nos apoiar nas decisões e muitas vezes a gente via que não estava correto".* [P4 - Projeto 2]

*"Então se eu for listar os fatores que impactaram para essa diferença de estimativa, seria: a qualidade do código legado, porque nós só avaliamos o tamanho e não consideramos a qualidade e complexidade dele".* [P2 - Projeto 2]

A Figura 5.1 apresenta os fatores relacionados ao sistema legado identificados no estudo de caso.

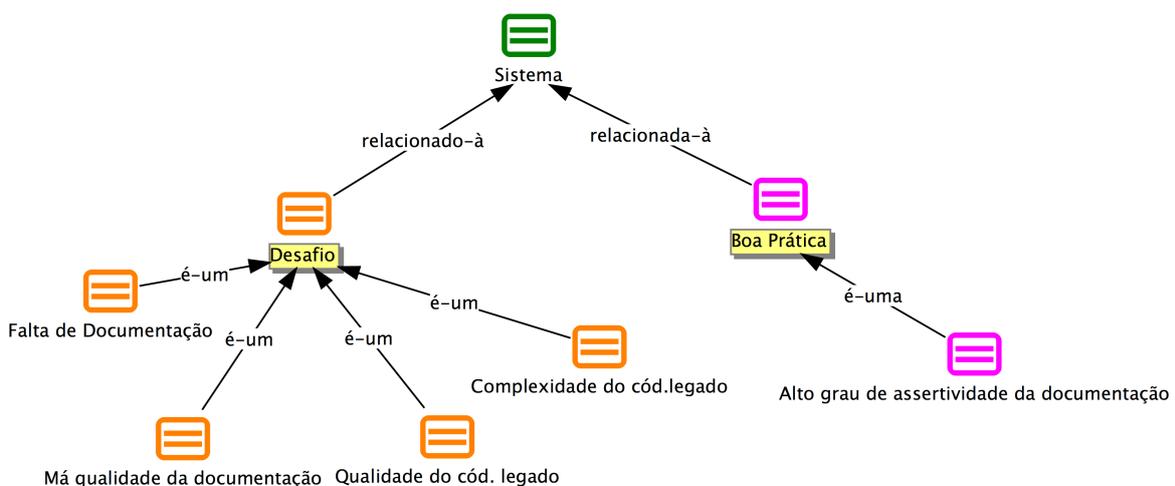


Figura 5.1 – Grafo da Categoria Sistema Legado no Estudo de Caso

## Cliente

O cliente foi considerado como um dos fatores chaves de influência na estimativa de projeto. Apesar de nenhuma das técnicas de estimativa aplicadas levar em consideração fatores relacionados ao cliente no cálculo da estimativa, estes fatores influenciaram diretamente no processo de predição.

O primeiro ponto é a participação do cliente: um **cliente participativo**, isto é, disponível para esclarecer dúvidas (principalmente na fase inicial do projeto, onde ocorrem as estimativas) é considerado um fator de sucesso. Já a **indisponibilidade do cliente** acarreta em um grande desafio para os estimadores, principalmente se o sistema legado for complexo de se analisar e não possuir documentação (ver seção "Sistema Legado"), como foi o caso do Projeto 2.

*"E com certeza também foi a parceria muito forte com o cliente, com a equipe de TI do cliente, então esse relacionamento TI com TI, talvez se fosse TI com negócio as coisas tivessem sido diferentes, mas é TI com TI e ambas estão muito próximas, então a gente consegue ter um alinhamento muito forte com relação à controlar o escopo, a realmente trabalhar de forma colaborativa" [P3 - Projeto 1]*

*"o feedback do cliente era muito ruim, eles não tinham muita gente para se envolver e não tinha documentação, não tinha nada. E tinham muitas demandas legais, tinham muitas questões de cálculo envolvidas que os usuários não sabiam". [P4 - Projeto 2]*

Outro desafio relacionado ao cliente é quando este **questiona a estimativa**, tanto as técnicas aplicadas quanto os valores obtidos. Este desafio está relacionado ao fato de que o **cliente geralmente tende a subestimar o custo do projeto**. O agravante desta situação nos projetos de reengenharia é o fato de o cliente já possuir um certo conhecimento sobre o sistema e usar este conhecimento para subestimar o trabalho a ser realizado. Esta situação foi particularmente vivenciada no Projeto 2.

*"o padrão é não abrir o número de horas, é passar um preço. Mas tem clientes que querem saber quantas horas isso significa, quantas pessoas estão alocadas. Nesse caso a estimativa é aberta completamente, ele quer saber quantas horas vai levar a análise do projeto, a construção, os testes. Tem alguns clientes que pedem até o currículo do profissional que vai trabalhar, então cada caso é um caso, mas o cliente interferem sim". [P1]*

*"o cliente questiona, mas a gente consegue mostrar...dependendo do cliente a gente trabalha em conjunto para que ele...quando a gente estima a gente assume algumas premissas e ele pode olhar e dizer "isso não é verdadeiro, isso é", a gente tenta trabalhar junto com ele, a gente abre aqui a quantidade de esforço para que ele se sinta confiável no trabalho que a gente tá fazendo".[P2 - Projeto 2]*

*"Acho que essa habilidade de controlar o cliente para se manter dentro do escopo é a habilidade do gerente de projetos e eu acho que esse é o principal ponto que afeta*

as estimativas, porque a maioria das vezes que a gente faz uma estimativa a gente faz do 'caminho feliz' do projeto, porque a gente pode até estimar o caminho não tão feliz, só que na negociação do preço acaba virando o caminho feliz, porque o cliente pede desconto, mas a gente sabe que 99% das vezes não é o caminho feliz". [P4 - Projeto 2 e 3]

Apesar de ter sido considerado um caso de sucesso, o Projeto 3 também enfrentou desafios, e um deles foi o fato de **o cliente interferir diretamente nos valores**, não concordando com o que foi estimado.

"muitas vezes o cliente solicita um cronograma mais agressivo, então muitas vezes a estimativa que foi gerada pode não ser seguida, pode ser reduzida. Então ele afeta assim, no final, em alguns casos ele que define pra quando ele quer o projeto [...] no PROJETO 3 ele deu um prazo e um preço e a gente teve que fazer caber ali dentro do esforço estimado." [P1 - Projeto 3]

A Figura 5.3 apresenta os fatores relacionados aos clientes, identificados no estudo de caso.

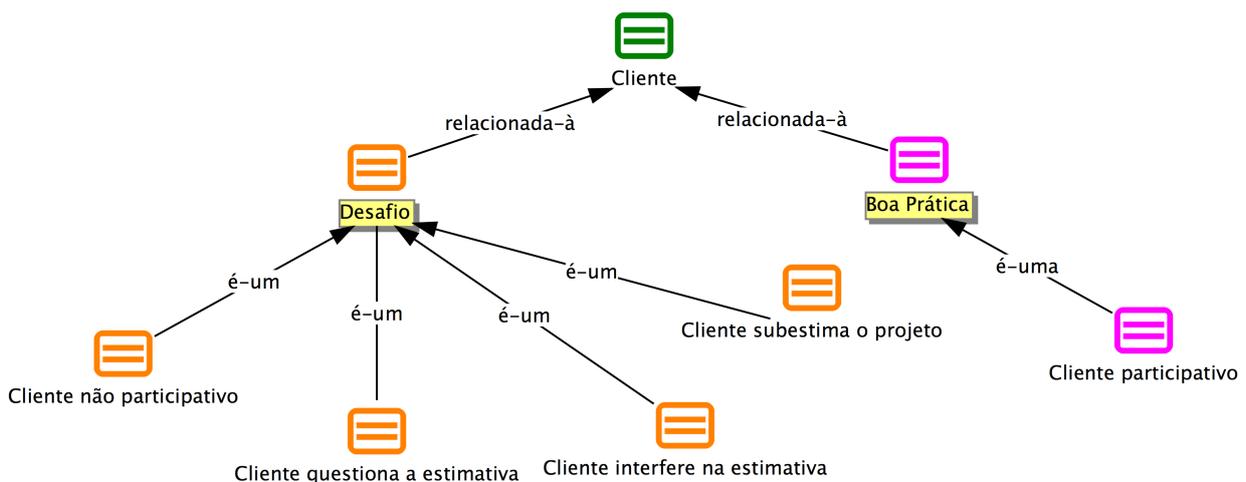


Figura 5.2 – Grafo da Categoria Cliente no Estudo de Caso

## Equipe

A equipe é primordial: tanto a equipe que realiza a estimativa, quanto a equipe que irá desenvolver o projeto.

Este fato ficou bem claro neste estudo de caso, pois quando a equipe tinha **conhecimento do domínio de negócio**, como foi o caso do Projeto 1 e do Projeto 3, conseguiu-se contornar em grande parte os desafios relacionados ao sistema legado (falta de documentação, qualidade do código legado, etc).

*"fator de sucesso é realmente a senioridade da equipe no negócio, acho que isso daí foi fundamental, porque a equipe que está trabalhando hoje no PROJETO 1 é a mesma equipe que já trabalhou em projetos similares e já trouxe essa experiência da área financeira, então tem analistas com mais de 10 anos de experiência só na área financeira e isso para mim foi fator chave". [P3 - Projeto 1]*

Já no caso em que **equipe não possuía este conhecimento de domínio**, Projeto 2, o processo de estimativa deve lidar com os desafios relacionados ao sistema.

*"era um domínio de negócio que nós não tínhamos, que era o negócio de previdência, e nós precisávamos estimar sem conhecer o negócio de previdência"[P1 - Projeto 2].*

*"não sei, no começo a gente olhava para aquilo e não fazia sentido, até porque além de ter a barreira de conhecimento do negócio, ainda tinha a barreira de conhecer uma linguagem diferente, com estrutura diferente."[P6 - Projeto 2]*

*"o que afetou a estimativa inicial, primeiro foi a falta de conhecimento do negócio, esta estimativa partiu de uma RFP onde tinham dezenas de requisitos em muito alto nível. Então eu acho que o que afetou a estimativa no começo foi a falta de experiência no negócio e a falta de uma estratégia para cumprir esse gap". [P5 - Projeto 2]*

**Uma vez que os sistemas legados envolvem os mais diversos tipos de tecnologias, frequentemente a equipe não tem experiência em lidar com elas** e necessita de uma curva de aprendizado e/ou da contratação de alguém com maior experiência. Todos os projetos analisados tiveram essa dificuldade. No Projeto 1, em especial, a equipe também enfrentou **dificuldade no entendimento das tecnologias do sistema alvo**.

*"tecnologia era muito antiga e a gente não tinha ninguém que conhecesse o legado, acho que se tivesse uma pessoa com conhecimento de Dataflex no projeto isso teria sido um fator muito positivo e essa pessoa podia centralizar a leitura do código" [P4 - Projeto 2].*

*"Os principais riscos, na verdade, quando iniciei a gestão foi identificar que era uma tecnologia nova a ser usada na solução, sem domínio nenhum da equipe, então existia esse risco, devido essa tecnologia ser nova e equipe não ter nenhuma experiência." [P3 - Projeto 1].*

*"Desconhecimento da tecnologia alvo e do paradigma alvo, ou seja, to be, não se conhecia também o as is, não se tinham pessoas com conhecimento prático em desenvolvimento web nem em Dataflex." [P5 - Projeto 2]*

Por fim, saber lidar com todos os fatores e desafios, prezando sempre pela maior qualidade do processo de estimativa, é tarefa da **equipe que realiza a estimativa**. No caso da Organização A, esta equipe foi responsável pela implantação de várias boas práticas, como uso de múltiplas técnicas de estimativa e seleção da técnica de acordo com o projeto.

Essas pessoas também são as responsáveis por aplicar as lições aprendidas dos projetos anteriores.

A Figura 5.3 apresenta os fatores relacionados com a Equipe, identificados no estudo de caso.

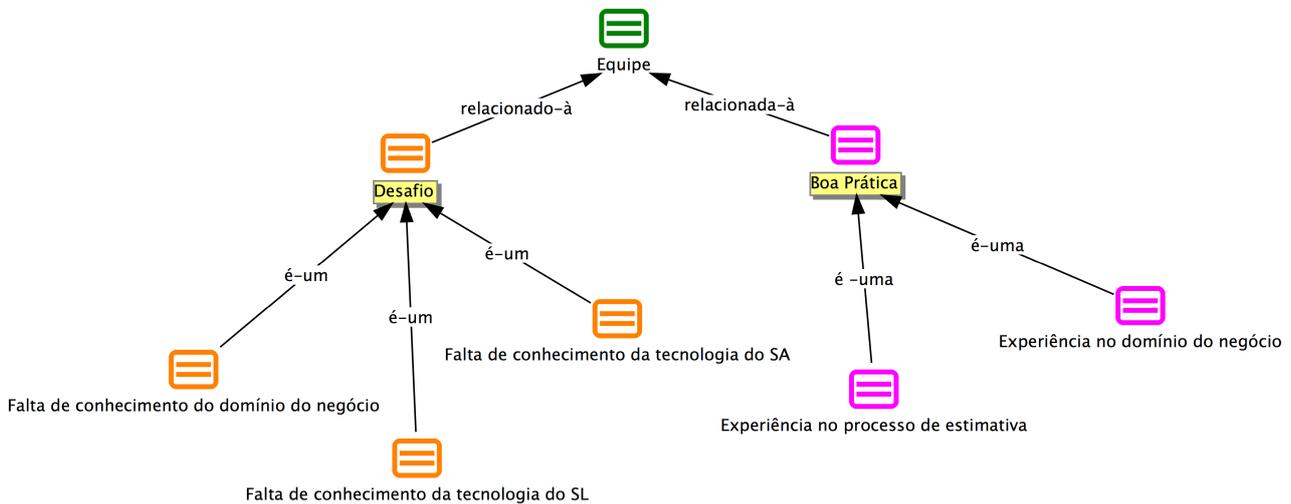


Figura 5.3 – Grafo da Categoria Equipe no Estudo de Caso

## Organização

Dentre os fatores organizacionais que afetaram as estimativas o principal é o processo adotado (processo de reengenharia, no caso) que apesar de bem definido foi considerado como sendo um **processo burocrático** e muitas das tarefas realizadas não foram contabilizadas nas estimativas para não deixá-las mais caras.

*"[...] o processo organizacional afetou bastante, porque é um processo extremamente robusto, muito grande, então ele acabou de certa forma exigindo um esforço maior e esse esforço foi absorvido em tempo de execução."* [P3 - PROJETO 1]

Em relação às tarefas não contabilizadas, a que mais afetou as estimativas do Projeto 1, foi a **implantação de um modelo de qualidade** (CMMi, nível 3). Neste caso, o esforço despendido para realização de atividades específicas da implantação do modelo, como reuniões de gestão, elaboração de documentação, dentre outras, não foram levadas em conta no momento da estimativa.

*"Uma coisa que impactou bastante, que foi colocado como risco foi o próprio CMMi, então a organização determinou que em 7 meses, 8 meses, ela queria atingir o nível 3 do CMMi e na estimativa inicial não foi previsto o esforço CMMi, e isso consumiu muito, impactou bastante, dá pra se dizer até que no cenário atual que a gente se encontra, aonde a gente tem que buscar uma ou outra negociação do cliente, de fatar um pouco essa entrega, eu com certeza posso afirmar que o CMMi também impactou muito nisso, de a*

*gente não está tão confortável agora, faltando uma semana e pouco pra entrega". [P3 - Projeto 1].*

Outro fator organizacional é referente a **pressão de outros setores da organização**, que estão interessados diretamente no fechamento do contrato do projeto e abstraem certos riscos de estimativa para que o acordo ocorra, este caso foi identificado no Projeto 2.

*Ah, sim, diversos. Vendas, diretor da área, depende muito do projeto e da situação da empresa, da situação daquela área, as vezes vai pegar o projeto, mesmo assumindo riscos, porque não tem outro projeto para pegar, ou vai ser desalocado um time, etc. Então tem muita interferência, a gente tenta conduzir, mostrar por um modelo matemático, mesmo que seja fraco, por n documentos. [P2 - Projeto 2].*

*"Não é só uma estimativa, tu é afetado por muitos fatores que são externos as estimativas, uma pressão pra fechar o negócio." [P6 - Projeto 2]*

A Figura 5.4 apresenta estes fatores organizacionais.

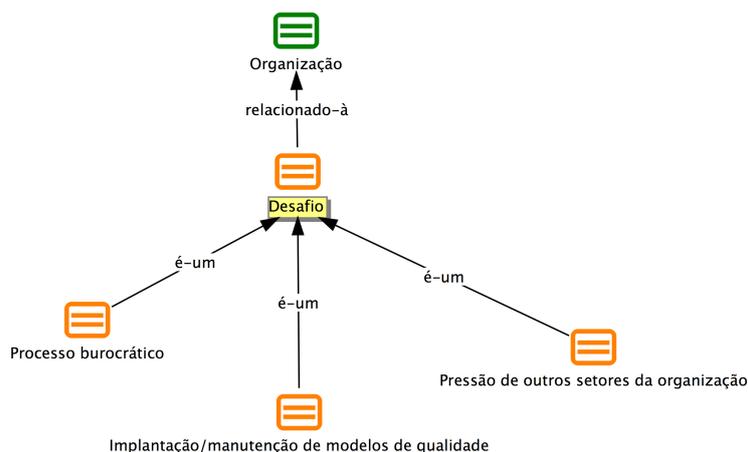


Figura 5.4 – Grafo da Categoria Organização

## Projeto

Por fim, a Figura 5.5 apresenta os fatores relacionados ao projeto que influenciaram na estimativa de esforço.

O Projeto 2 foi o que apresentou os problemas mais graves relacionados à estimativa. O principal deles foi a **má definição do escopo** do projeto, ou seja, no momento da pré-venda a equipe do projeto não conseguiu definir junto ao cliente qual era o real escopo do projeto de reengenharia, até que ponto o sistema legado deveria ser migrado e a extensão das novas funcionalidades a serem implementadas.

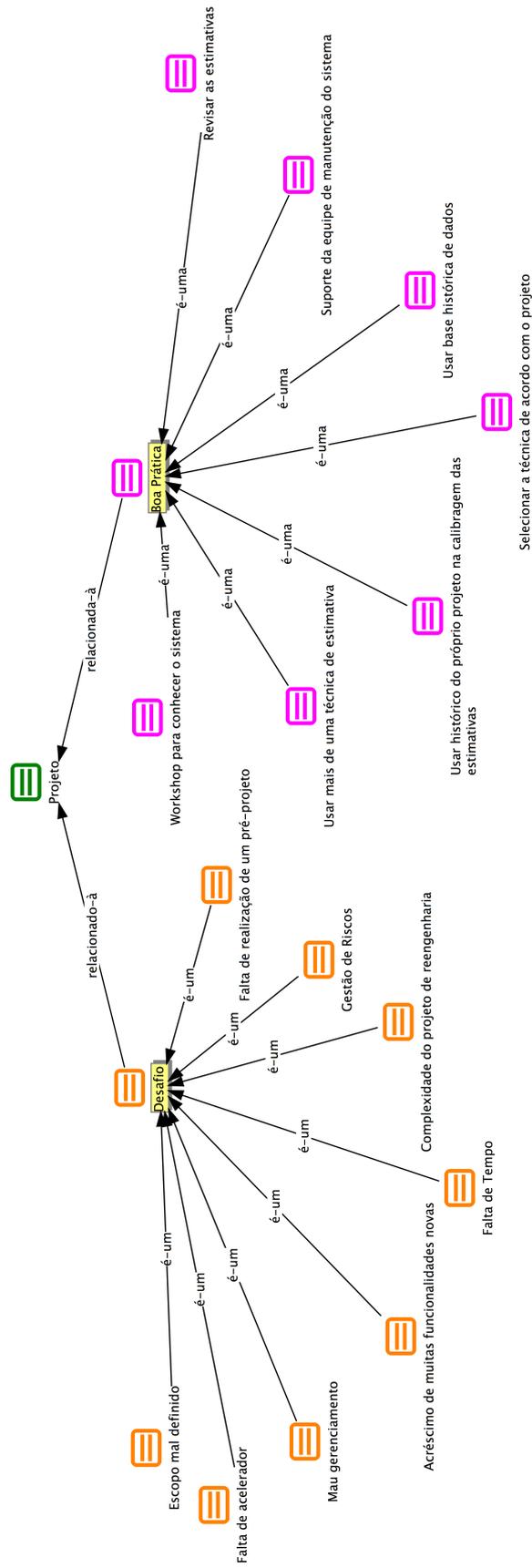


Figura 5.5 – Grafo da Categoria Projeto no Estudo de Caso

*"só que o documento de casos de uso dizia que era pra fazer de um jeito e eles diziam que não tava batendo com o legado e eu dizia "não, cara, mas tu assinou esse caso de uso", então esse caso de uso assinado era o escopo do projeto, e eles diziam "não, mas o legado tem que bater também, e o projeto é o legado + RFP, então tem que bater". [P2 - Projeto 2]*

*"do PROJETO 2 teve uma grande variação do previsto para o realizado, o realizado foi duas vezes quase que maior, por um monte de fatores, o primeiro deles era que o escopo não tava bem fechado"[P4 - Projeto 2]*

O Projeto 2 apresentou graves desvios da estimativa durante a execução do projeto. Estes desvios foram influenciados, em grande parte, pelo fato de que foi previsto durante a estimativa que seria utilizada uma ferramenta que aceleraria a engenharia reversa do sistema, uma vez que a ferramenta traduziria o código do sistema legado para uma linguagem intermediária, que poderia ser mais facilmente adaptada para o sistema novo. Esta ferramenta, porém, se mostrou ineficiente e seu uso foi descontinuado, **retirando com isso o fator de aceleração de esforço previsto na estimativa.**

*"Se a gente tivesse uma ferramenta, que era o intuito do projeto, pra extrair as regras de negócio do sistema legado e a gente simplesmente fazer uma modernização naquele sistema as is, eu concordo. Se a gente fosse fazer uma modernização as is e implantar novas funcionalidades, eu também concordo. O problema é que o projeto foi feito o que: foi feita uma modernização e foi embutida complexidade nas funcionalidades que estavam sendo modernizadas e novas funcionalidades que a gente teve que mudar completamente a estrutura do novo projeto, em termos de arquitetura e de base de dados, então muitos conceitos do legado já não eram mais aplicáveis no novo."[P4 - Projeto 2]*

Ainda em relação ao Projeto 2, houve um **mau gerenciamento** por parte do responsável pelo projeto, que não tratou adequadamente os desvios, permitindo com que o projeto entrasse em uma situação crítica de estouro de estimativa. O mau gerenciamento também foi responsável por permitir que uma **grande taxa de alterações no escopo do projeto**, que por si só já seria um problema, fosse inserida sem o devido gerenciamento de mudança, agravando ainda mais a situação.

*"na entrega da fase 2 ele me reportou um atraso, um problema de custo, quando a gente foi fazer o replanejamento a gente viu que tinha um desvio muito grande, a gente viu que o PM tinha feito um escopo? muito grande no projeto, tava aceitando muita mudança de escopo do cliente sem reclamar e sem fazer o controle de mudança correto"[P4 - Projeto 2].*

Outro desafio, este enfrentado em todos os projetos analisados, foi que os **dados necessários para realizar a estimativa tiveram de ser levantados durante a fase de pré-venda**, o curto prazo para realização desta atividade, agravado pela falta de documentação relacionada ao sistema legado, dificultou o entendimento mais profundo do problema a ser solucionado.

*"Eu acho que uma pré-venda bem feita , com tempo para levantar os dados e fazer uma estimativa correta, influencia diretamente no resultado das estimativas. Então, como não temos esse tempo, a gente sabe que corre um risco muito maior". [P4 - Projetos 2 e 3]*

De modo geral, para atender as solicitações do cliente de modificação das estimativas, a estratégia adotada é minimizar alguns riscos no projeto, de maneira a não passar o custo disso para o cliente. Porém, torna-se um desafio posterior a **gestão destes riscos** que foram minimizados.

*"Se a gente for embarcar na estimativa todos os fatores de risco o projeto fica muito caro, então a gente não passa esse risco pro cliente, senão o projeto não vende. Deveria-se estimar os riscos, mas por uma questão de custo a gente acaba assumindo esses riscos e tentando gerenciar durante o projeto". [P1 - Projetos 1, 2 e 3]*

Por fim, em relação aos desafios, mesmo com o sistema legado disponível, nem sempre a estimativa consegue ser assertiva. Um dos problemas relacionados a isto é que o sistema pode envolver um tipo de negócio e/ou tecnologia cuja equipe não tem experiência. Assim, é importante que seja realizado um projeto preliminar, ou projeto piloto, onde a equipe terá oportunidade de analisar minuciosamente uma parte do sistema e, com isso, ser capaz de estimar a complexidade para realização do restante do projeto. Nos projetos analisados sentiu-se **falta da realização deste projeto piloto** no Projeto 1 e principalmente no 2, diante de todas as dificuldades enfrentadas.

*"Acredito que todas essas questões que eu citei deveriam ter sido levadas em conta na pré-venda, cabia ter elaborado um pré-projeto, porque esse é um projeto estratégico, é um projeto grande, de mais de 2 anos." [P3 - Projeto 1]*

*"Tinham riscos muito altos nesse projeto, tinha que ter sido feita um POC para provar o que funcionava e o que não funcionava" [P6 - Projeto 2]*

Diante dos inúmeros desafios enfrentados durante o Projeto 2, algumas boas práticas foram adotadas. A primeira delas foi empregada com o intuito de ampliar o conhecimento sobre o escopo do projeto a ser estimado e consistiu na realização de um *workshop* envolvendo a equipe que iria atuar no projeto e a equipe do cliente. Este *workshop* deveria servir tanto para a equipe do projeto obter conhecimento sobre o domínio de negócio, quanto sobre o sistema legado (tecnologia envolvida e funcionalidades). Porém este tipo de atividade não se mostrou adequada para o caso em que a equipe não tinha familiaridade com o domínio de negócio, pois são necessários tempo e envolvimento maiores para se obter este conhecimento. Sendo assim, a prática do *workshop* só serviu para conhecer as funcionalidades e um pouco da tecnologia do sistema.

*"Quando a gente fez o primeiro plano, que era pra fazer o kick off, a gente fez um workshop de uma semana, dentro do cliente, pra entender os conceitos de previdência, uma espécie de imersão no ambiente do cliente com boa parte do time: a gente levou desenvol-*

*vedores, testers, todos os analistas que tavam desenvolvidos e eu, que era o gerente". [P4 - Projeto 2]*

Quando o Projeto 2 atingiu o ponto mais crítico, onde a estimativa teve que ser refeita, optou-se por utilizar uma abordagem que usa o **conhecimento obtido no próprio projeto para calibrar a estimativa para as atividades ainda não realizadas**. Esta prática de calibragem dinâmica não era prevista no processo, mas vem sendo adotada pela organização deste então.

*"nós começamos a criar uma base histórica ao longo do projeto , usando ferramentas que eu mesmo criei, que eram planilhas, eram históricos do que a gente vinha fazendo, a gente foi usando o real estimado e o real realizado, e foi assim que a gente foi fazendo estimativa, porque o nosso conhecimento de negócio foi aumentando, a nossa arquitetura, ao longo do tempo foi ficando mais madura, mais estável, a gente já tinha uma forma de fazer as coisas e tinha referencias, quando entrava um cara novo ele já tinha pra onde olhar. O fato da arquitetura técnica, do código já existir referência, isso também deixou nossa estimativa muito mais assertiva e a medida que o conhecimento de negócio foi aumentando, a gente também conseguiu ser mais assertivo". [P6 - Projeto 2]*

Em relação aos Projeto 1 e 3, que não tiveram problemas na realização das estimativas, uma boa prática foi que, apesar da complexidade do projeto de reengenharia a ser realizado, **a empresa já era familiarizada com o tipo de negócio envolvido no projeto**, e em sua base histórica já possuía soluções similares a que seria desenvolvida. Assim, a **escolha da técnica de estimativa adequada ao projeto** levou a utilização de *botton-up*, técnica própria da empresa, cujos estimadores tem grande facilidade em trabalhar.

*"uso botton-up é quando eu conheço o sistema destino, quando eu sei como vai ser. A gente também usa quando é alteração em cima de um sistema que a gente já fez. Basicamente é assim, ou quando é alteração de algum sistema que a gente já fez ou quando é um sistema que a gente conhece previamente, que foi o caso do PROJETO 1 e do PROJETO3". [P1 - Projetos 1 e 3]*

*"Além de botton-up, usamos outras técnicas: UCP é sempre quando a gente tem uma RFP em alto nível, como é esse caso aqui. COCOMO é quando a gente tem o número de linhas de código do sistema. UCP é o nosso preferencial porque a gente sempre recebeu a RFP em muito alto nível. A gente manda também pro pessoal de métricas o mesmo documento que a gente recebe e eles contam em PF" [P1 - Projetos 1, 2 e 3]*

O Projeto 3, em particular, foi considerado um caso de sucesso nas estimativas, um dos fatores de sucesso que contribuíram para isso foi o fato da equipe **responsável pela manutenção do sistema estar disponível para esclarecimento de dúvidas**.

*"o único conhecimento que a gente tinha era que a gente assumiu o suporte desse projeto, o AMS que existe aqui, que é fazer toda a manutenção do projeto, então a gente assumiu esse legado e tem um time dentro da ORGANIZAÇÃO que suporte esse legado,*

*correção de bugs e outras melhorias, então a gente assumiu esse segundo projeto de modernização e a gente já tinha praticamente 1 ano de sustentação, e aí a documentação que a gente tinha foi o que agente conversou com o pessoal" [P4 - Projeto 3].*

Por fim, uma boa prática geral é a **revisão das estimativas** por uma equipe de especialistas do projeto e outras partes interessadas da organização.

*"O ciclo de pré-venda, até o cliente assinar o contrato, ele tem todo um ciclo de etapas dentro de uma área que a gente chama de solution, que faz toda a governança da oportunidade e depois que a gente faz toda a estimativa, monta a solução, ela é revisada por várias pessoas, isso sempre acontece, não tem exceção. E essas pessoas revisam a estimativa, revisam a solução, fazem várias reuniões, as vezes tem até 10 pessoas participando disso, e questionam todos os aspectos da solução, "tu já fez esse tipo de projeto, porque tu estimou assim, que técnicas tu utilizou", e quem é que participa?! É o líder da prática de vendas, o líder da prática de delivery, o líder de gestão, o líder de testes, o líder de análise, uma pessoa da área de métricas da HP, então isso sempre acontece." [P1 - Projetos 1, 2 e 3]*

## 5.6 Conclusões do Capítulo

Este estudo de caso teve como objetivo complementar as informações obtidas do estudo de campo inicial, a partir da análise de 3 projetos de reengenharia de software com diferentes características.

A realização deste estudo foi particularmente interessante, pois se pode observar todo o ciclo de vida da estimativa de esforço em diferentes contextos de projeto e como o processo padrão se adaptou para comportar as particularidades dos projetos.

Este estudo de caso também permitiu a identificação de uma nova categoria em relação ao estudo de campo inicial. Isto se deveu ao fato de terem sido analisados projetos específicos, o que permitiu ao pesquisador e aos participantes da pesquisa uma reflexão geral sobre tudo que afetou estes projetos. Assim, não foram só identificados os fatores ditos principais, mas todos aqueles que tiveram alguma influencia na estimativa de esforço.

No próximo capítulo serão consolidados os resultados de ambos os estudos realizados em forma de lições aprendidas, que servirão de apoio à concepção do modelo que é o objetivo deste trabalho.



## 6. CONSOLIDAÇÃO DOS RESULTADOS DOS ESTUDOS EMPÍRICOS

Tanto o estudo de campo inicial quanto o estudo de caso, evidenciaram diversos aspectos de estimativa de esforço em projetos de reengenharia de software, que são consolidados neste capítulo.

No estudo de campo inicial foram identificadas quatro categorias para agrupar os desafios e boas práticas de estimativa de esforço em projetos de reengenharia de software: *cliente, sistema, projeto e equipe*. A realização do estudo de caso possibilitou a identificação de uma nova categoria, a *organização*, que agrupa os fatores organizacionais que influenciam no processo de estimativa de esforço. Os desafios representam situações que dificultam a estimativa de esforço, podendo levar a uma menor acurácia desta estimativa. As boas práticas são ações realizadas para minimizar os problemas acarretados pelos desafios ou simplesmente para melhorar o processo de estimativa, buscando obter uma melhor acurácia. Além disso, ambos os estudos identificaram um conjunto de atividades realizadas no processo de estimativa de esforço em reengenharia.

As informações e a experiência obtidas com a realização dos estudos empíricos, permitiu a elaboração de um conjunto de lições aprendidas, que sintetizam aspectos importantes relacionados a estimativa de esforço em reengenharia, que devem ser levados em consideração na elaboração de um modelo que vise apoiar este processo.

Na Seção 6.1 as atividades de estimativa de esforço são apresentadas, a Seção 6.2 apresenta a consolidação dos desafios, a Seção 6.3 apresenta a consolidação das boas práticas. Por fim, a Seção 6.4 apresenta as lições aprendidas que servirão como base para concepção do modelo, juntamente com as boas práticas e as atividades identificadas.

### 6.1 Atividades de Estimativa

A Tabela 6.1 apresenta as atividades identificadas no processo de estimativa realizado nas organizações estudadas no estudo de campo (E1) e no estudo de caso (E2).

Tabela 6.1 – Atividades de Estimativa

<b>Atividade</b>	<b>Fonte</b>
Realização da Estimativa	E1 e E2
Comparação e Validação das Estimativas	E1 e E2
Monitoramento e Calibragem da Estimativa	E1 e E2
Atualização da Base Histórica e Aprendizagem	E1

1. Realização da Estimativa: atividade onde a estimativa de esforço é efetivamente calculada, com base no tamanho do projeto e demais parâmetros que possam impactar nas horas a serem gastas. Esta atividade inclui a seleção da(s) técnica(s) de estimativa e o levantamento dos dados necessários para a aplicação da(s) técnica(s) empregada(s) pela organização.
2. Comparação e Validação: esta atividade é realizada por organizações onde mais de uma técnica de estimativa é aplicada. Neste momento, os valores estimados são revisados por especialistas e/ou combinados em fórmulas estatísticas, para gerar o valor de estimativa de esforço do projeto ou do módulo do projeto a ser estimado.
3. Monitoramento e Calibragem das Estimativas: consiste no acompanhamento do projeto para identificação e tratamento de desvios e/ou riscos de desvios em relação ao que foi estimado, de maneira que o projeto se mantenha dentro do planejado ou, caso seja inevitável replanejar, que isto seja feito de maneira formal.
4. Atualização da Base Histórica e Aprendizagem: todas as organizações estudadas mantêm registro dos dados históricos de estimativa e acompanhamento do projeto, embora apenas a organização D mantenha esta base de maneira estruturada e utilize este conhecimento para melhorar o processo de estimativa.

## 6.2 Desafios

Os desafios foram classificados de acordo com as categorias identificadas na análise dos dados do estudo de campo inicial e do estudo de caso. A Tabela 6.2 apresenta a síntese dos desafios encontrados.

Em relação ao sistema legado, os desafios estão relacionados com a qualidade e complexidade do código-fonte legado (D01) (D02). Isto prejudica a realização das estimativas, pois quanto menor a qualidade e maior a complexidade, mais difícil a análise do código-legado para a extração dos dados necessários para a aplicação das técnicas de estimativa, além de afetar o valor das mesmas. Ademais, tem-se como desafio a falta de documentação ou a má documentação do sistema (D03)(D04), esta situação retira a vantagem do responsável pela estimativa em ter todas as (ou a maioria das) características do sistema identificadas, sem que se tenha que realizar elicitação de requisitos ou análise do sistema legado, além de elevar o esforço para realização do projeto com tarefas de engenharia reversa.

Os clientes são fatores chave para o desenvolvimento do projeto, quando eles não se envolvem ou não tem clareza da solução que almejam, tornam mais difícil para a equipe do projeto desenvolver esta solução, a começar pela realização das estimativas

Tabela 6.2 – Síntese dos Desafios Encontrados

<b>Categoria</b>	<b>ID</b>	<b>Desafio</b>	<b>Fonte</b>
Sistema	D01	Qualidade do código-fonte legado	E1 e E2
	D02	Complexidade do código-fonte legado	E1 e E2
	D03	Falta de documentação	E1 e E2
	D04	Má qualidade da documentação	E2
Cliente	D05	Cliente não participativo	E2
	D06	Interfere nas estimativas	E2
	D07	Questionamento da técnica de estimativa	E1 e E2
	D08	Subestimação da complexidade do projeto	E1 e E2
Equipe	D09	Falta de conhecimento nas tecnologias do sistema alvo	E2
	D10	Falta de conhecimento no negócio	E2
	D11	Falta de conhecimento nas tecnologias do sistema legado	E2
	D12	Falta de conhecimento do sistema	E2
Projeto	D13	Falta de experiência na realização de estimativas	E1
	D14	Escopo mal definido	E2
	D15	Acréscimo de muitas funcionalidades novas em relação ao sistema legado	E2
	D16	Falta de tempo para realizar a estimativa	E1 e E2
	D17	Estimar esforço para a fase de engenharia reversa	E1
	D18	Falta de dados para realizar a estimativa	E1
	D19	Tamanho do projeto	E1
	D20	Complexidade do projeto de Reengenharia	E2
	D21	Falta de ferramentas de apoio à estimativa	E1 e E2
	D22	Falta de similaridade entre projetos	E1
	D23	Fator de aceleração não disponível em tempos de execução	E2
Organização	D24	Gestão de riscos	E2
	D25	Processo burocrático	E2
	D26	Implantação/manutenção de modelo de qualidade	E2
	D27	Pressão de outros setores da organização	E2

(D05). O cliente pode ainda desejar interferir sobre as técnicas utilizadas para derivar a estimativa, querendo obter detalhes de seu funcionamento, para justificar os valores estimados (D06)(D07). Também em relação à realização da estimativa, o cliente pode tentar interferir nos valores estimados, tentando diminuí-los, por achar que o projeto é mais simples (D08). Essa subestimação acontece em todos os tipos de projeto de desenvolvimento, mas em reengenharia tem o agravante de o cliente já conhecer o sistema legado e achar que o fato de ele estar sendo "somente"renovado, não acarreta em tanto esforço.

Desenvolvimento de software é uma atividade orientada à pessoas, logo, a equipe é um fator determinante para o sucesso do mesmo. Assim, problemas ou dificuldades enfrentadas na equipe podem ter grande impacto no esforço a ser gasto em um projeto. Falta de conhecimento do negócio, da tecnologia envolvidas (tanto do sistema legado quanto do

sistema alvo), ou até mesmo a falta de experiência na utilização das técnicas de estimativas empregadas, podem onerar significativamente este processo (D09) à (D13).

Em relação aos fatores do projeto de reengenharia que impactam nas estimativas, o maior problema é causado por escopo mal definido (D14) ou que sofre grandes alterações, como acréscimos de muitas funcionalidades novas em relação as existentes no sistema legado (D15). Este desafio é agravado pela falta de tempo para realizar as estimativas (D16), principalmente se elas forem realizadas na fase inicial do projeto, onde tenha que se estimar esforço para as atividades de análise, como a engenharia reversa (D17). Essa pressa na realização das estimativas faz com que a coleta de dados não seja feita adequadamente (o que gera a falta de dados) (D18), principalmente se o tamanho do projeto for grande (D19) e a complexidade da reengenharia a ser realizada for elevada (D20). Fatores que contornariam esse problema como a existência de ferramentas para realização da estimativa ou de projetos similares anteriores para basear as previsões, podem não estar disponíveis no projeto (D21)(D22). Já em tempo de acompanhamento das estimativas, os valores estimados podem sofrer grandes desvios se tiver sido considerado algum fator de aceleração (como uma ferramenta para automatizar a engenharia reversa) que se prove ineficiente ou não esteja disponível (D23). Finalmente, em relação ao projeto, tem-se como desafio a gestão dos riscos relacionados as estimativas. Estes riscos podem ser negligenciados em tempo de realização da estimativa com o objetivo de atender uma demanda do cliente, por exemplo (geralmente para diminuir os custos), mas podem vir a causar problemas durante o andamento do projeto, como desvios nos valores estimados (D24).

Por fim, os desafios relacionados a organização de modo geral estão ligados, principalmente, com a burocracia do processo (de reengenharia), que muitas vezes não é levada em consideração no momento da realização das estimativas, mas posteriormente pode vir a desencadear desvios nos valores estimados (D25). Dentre as atividades que são estimadas podem estar aquelas relacionadas a implantação ou manutenção de um modelo de qualidade, como reuniões, elaboração de documentos, dentre outras (D26). Além disso, os responsáveis pela realização da estimativa devem lidar com a pressão de outras áreas da organização, que interferem na estimativa afim de atingir objetivos de negócios, como o fechamento do contrato do projeto, por exemplo (D27).

A Figura 6.1 apresenta a consolidação dos desafios encontrados e suas relações com as categorias identificadas no estudo.

### **6.3 Boas Práticas**

As boas práticas relacionadas a estimativa de esforço em reengenharia são tipicamente soluções propostas com o objetivo de tratar algum desafio existente neste processo.

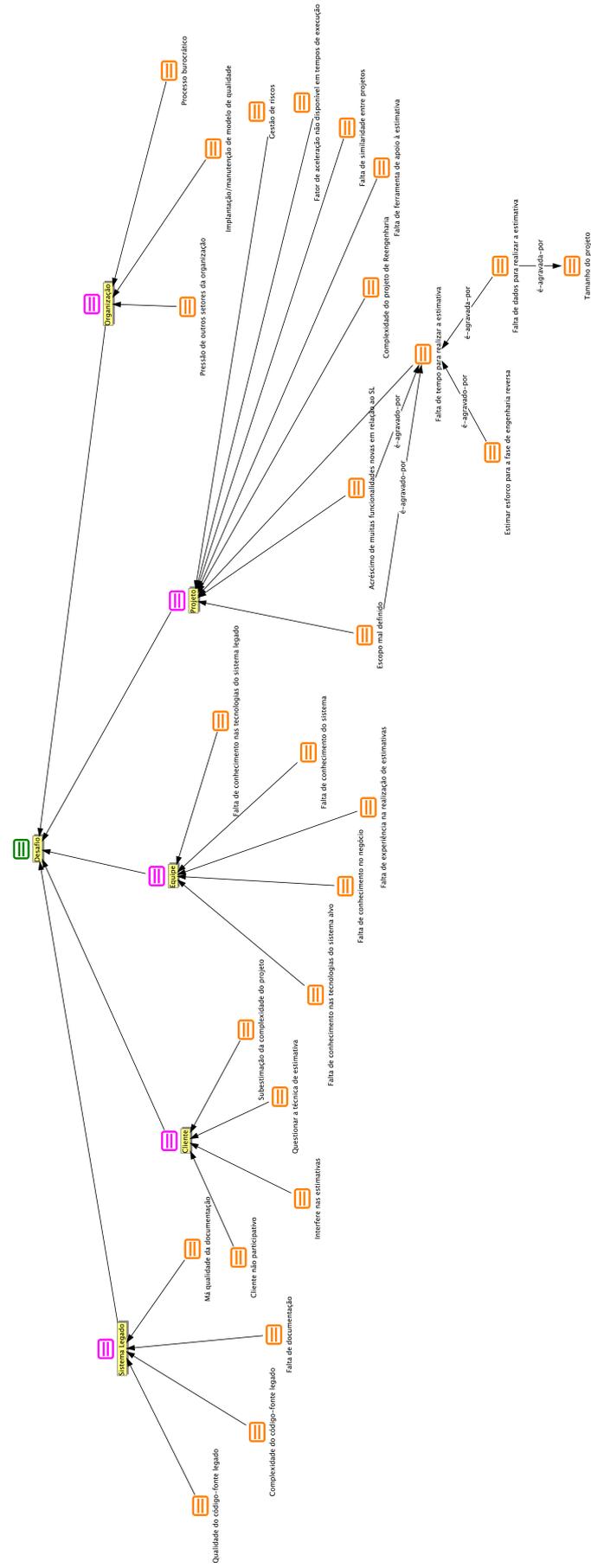


Figura 6.1 – Consolidação dos Desafios

A Tabela 6.3 apresenta as boas práticas identificadas no estudo de campo inicial e no estudo de caso.

Tabela 6.3 – Síntese das Boas Práticas Aplicadas

<b>Categoria</b>	<b>ID</b>	<b>Prática</b>	<b>Fonte</b>
Sistema	BP01	Sistema possuir documentação, e ela estar atualizada	E2
Cliente	BP02	Integrar o cliente ao processo de estimativa	E1
Equipe	BP03	Conhecimento do processo de estimativa	E2
	BP04	Conhecimento de diferentes técnicas de estimativa	E1 e E2
	BP05	Conhecimento do domínio do negócio	E1 e E2
	BP06	Conhecimento da tecnologia do sistema legado	E1 e E2
	BP07	Conhecimento das tecnologias a serem empregadas no sistema alvo	E2
Projeto	BP08	Realizar um pré-projeto para apoiar o levantamento de dados para estimativa	E2
	BP09	Realizar <i>workshop</i> para entendimento do sistema a ser migrado	E2
	BP10	Ter suporte da equipe de manutenção do sistema	E2
	BP11	Estabelecer, manter e utilizar base histórica de projetos	E1 e E2
	BP12	Utilizar mais de uma técnica de estimativa	E1 e E2
	BP13	Selecionar a técnica de estimativa de acordo com as características do projeto	E1 e E2
	BP14	Revisão das estimativas	E1 e E2
	BP15	Boa gerência	E2
	BP16	Monitoramento constante do projeto	E2
	BP17	Calibragem das estimativas apoiada nos dados do próprio projeto	E2
	BP18	Gestão de conhecimento de estimativa	E1
Organização	BP19	Processo bem definido	E1

Em relação ao sistema não é possível controlar a complexidade ou a qualidade do código-fonte legado. Porém, sempre que existir documentação do sistema e ela estiver atualizada (BP01), o processo de análise deste sistema para levantamento de dados necessários na estimativa se torna mais fácil.

Não é possível, também, obrigar o cliente a ser participativo ou evitar que ele questione o projeto. Mas pode-se integrar ele ao processo de estimativa, apresentando o processo aplicado, de modo a justificar os valores obtidos (BP02).

No que diz respeito a equipe, esta deve estar preparada para lidar com as mais diferentes situações: seja em termos técnicos de realização da estimativa em si (conhecimento do processo de estimativa (BP03) e conhecimento de diferentes técnicas de estimativa (BP04)), ou em termos do tipo de projeto a ser realizado (conhecimento do tipo de negócio (BP05), conhecimento das tecnologias do sistema legado (BP06) e conhecimento das tecnologias a serem empregadas no sistema alvo (BP07)).

Em termos de projeto, no que diz respeito a realização das estimativas, pode-se realizar um pré-projeto para entendimento do sistema e mitigação dos problemas de conhecimento do domínio de negócio e do sistema legado (BP08) (para entendimento do sistema legado pode-se realizar também um *workshop* de imersão no cliente (BP09), caso não haja tempo para o pré-projeto). Tanto durante o pré-projeto quanto durante o *workshop* podem-se coletar os dados necessários para realização das estimativas. Pode-se ainda, sempre que possível, obter auxílio da equipe que realizava manutenção do sistema, que tem conhecimento profundo do código legado (BP10). Estes dados também podem ser obtidos da base histórica de projetos, mantida e atualizada pela organização (BP11). Uma vez obtidos os dados necessários para realização da estimativa três boas práticas a serem aplicadas são: (1) utilizar mais de uma técnica de estimativa (BP12), (2) selecionar tais técnicas de acordo com as características do projeto (BP13) e (3) revisar as estimativas realizadas (BP14), as duas primeiras visam tornar estimativa mais específica para o contexto do projeto em questão, enquanto que a revisão garante que os valores estimados serão aprovados por outros especialistas, além daquele (s) que realizou (aram) a estimativa.

Uma boa gerência é essencial para manter os valores dentro do estimado, lidando com os desafios e problemas que surgirem (BP15). Para isso, os responsáveis pela gerência do projeto devem realizar um monitoramento constante, capaz de identificar e tratar os desvios para que o projeto, de maneira geral, se mantenha dentro do previsto (BP16). Caso seja possível ou necessário realizar calibragem nos valores da estimativa, essa calibragem deve ser feita com base no conhecimento obtido com o próprio projeto, no que já foi previamente realizado (BP17). Por fim, em relação à gerência do processo de estimativa, é uma boa prática utilizar as lições aprendidas nos projetos para melhorar o processo, o que pode ser feito a partir de uma estratégia de gerência de conhecimento (BP18).

Em relação a organização, uma boa prática é a existência de um processo de desenvolvimento bem definido (BP19). Embora muitas vezes este processo possa ser considerado burocrático, o fato de haver um processo bem definido sobre o qual estimar é melhor do que a inexistência desse processo.

A Figura 6.2 apresenta a consolidação das boas práticas encontradas e suas relações com as categorias identificadas no estudo.

## 6.4 Lições Aprendidas

Os dois estudos realizados ilustraram diversos aspectos relacionados a estimativa de esforço em projetos de reengenharia de software. A seguir destacam-se os principais aspectos que, comparados com a teoria na área de estimativa de esforço e de reengenharia, formam uma série de lições aprendidas, que servirão como base de sustentação para o modelo proposto neste estudo.

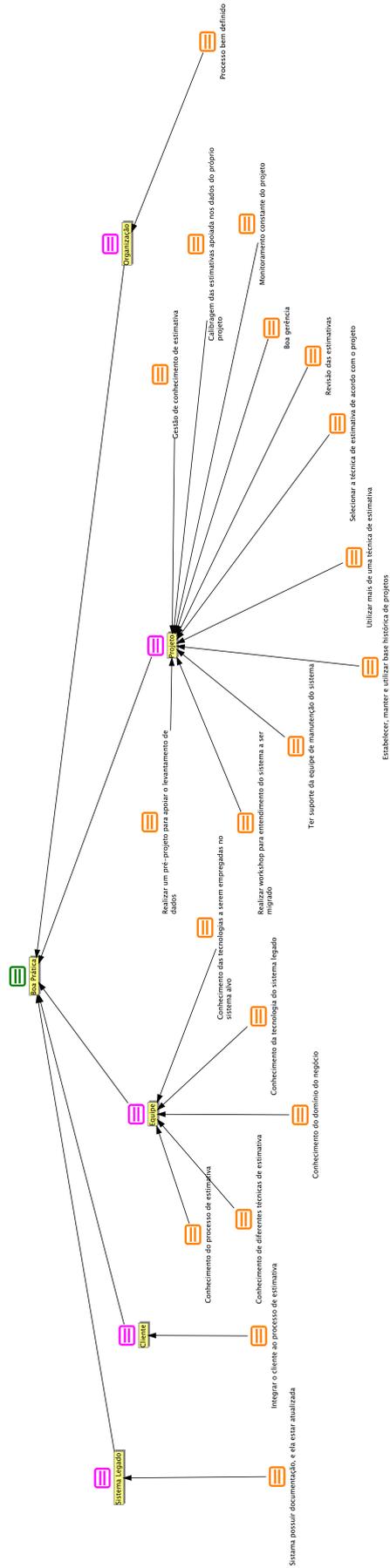


Figura 6.2 – Consolidação das Boas Práticas

1. Deve haver uma definição clara do escopo do projeto de reengenharia (L01)

Conforme apresentado na definição do processo de reengenharia (Capítulo 2) e enfatizado no relato do estudo empírico, um dos grandes desafios para realização da estimativa é estabelecer o escopo do projeto de reengenharia e quais os esforços necessários para realizar a estimativa para este tipo de projeto. Esta necessidade é mencionada por [SNE05] que diz que, dependendo do tipo de projeto de reengenharia realizado, este terá um impacto maior ou menor no valor e no processo de estimativa, em termos de riscos, benefícios e custos para a organização e para o cliente.

2. A realização de um projeto preliminar é essencial para o sucesso da estimativa, principalmente quando não houver familiaridade com o tipo de projeto (L02)

Dentre os principais desafios da estimativa estão a falta de experiência e/ou conhecimento nas tecnologias envolvidas no projeto de estimativa (tecnologia do sistema legado e/ou do sistema alvo), no domínio do projeto, etc. Os especialistas entrevistados relatam que se houvesse a realização de um pré-projeto, muitos desses problemas poderiam ser tratados antes da realização das estimativas.

Porém, dentre as empresas analisadas nesta pesquisa, apenas a Organização B aplica regularmente esta prática, as demais, ou não sofrem pressão para estimar preliminarmente o projeto ou não se sentem confortáveis de propor esta alternativa junto ao cliente.

Sendo assim, com base na (BP08 - Realizar pré-projeto) aplicada na Organização B e no relato dos especialistas, percebeu-se a necessidade de haver a realização de um pré-projeto para entendimento do projeto a ser desenvolvido e, com isso, embasar melhor as estimativas a serem realizadas.

3. A estimativa calculada deve ser revisada antes de aprovada para o planejamento do projeto (L03)

Um exame cuidadoso das estimativas calculadas foi identificado por [BS14] como um dos fatores que afetam a acurácia da estimativa de esforço, o trabalho cita o estudo realizado por [LP95] que mostrou que a precisão das estimativas estava correlacionada positivamente com a análise dos dados estimados. Nos estudos empíricos, esta prática foi identificada na Organização A e na Organização D (BP14 - Revisão das estimativas), que utilizam mais de uma técnica para estimar e necessitam chegar a uma decisão sobre qual o valor final de estimativa a ser considerado.

4. Deve haver monitoramento constante das estimativas do projeto (L04)

O monitoramento do projeto é uma atividade padrão da gerência do projeto e sua frequência varia de acordo com o modelo de processo adotado na organização [PRE11]. No contexto da estimativa de esforço é importante que este acompanhamento ocorra

de maneira constante, para que sejam percebidos e tratados os possíveis desvios ou riscos de ocorrências de desvios.

No contexto dos estudos empíricos esta foi uma prática percebida principalmente na organizações A e D (BP16 - Monitoramento constante do projeto).

5. Caso haja necessidade de recalibrar as estimativas, deve-se usar dados do próprio projeto (L05)

A experiência nos estudos de campo mostrou que sempre que havia necessidade de se realizar a calibragem nos dados do projeto optava-se por utilizar os dados do próprio projeto (obtidos até o momento) para realização dessas estimativas (BP17 - Calibragem das estimativas apoiada nos dados do próprio projeto). Esta boa prática aplicada na indústria é a base conceitual do processo proposto por [BCV03], que utiliza o conceito de estimativa dinâmica para realização das estimativas do projeto durante o andamento do mesmo.

O estudo de [SB95] relata que a alteração das estimativas no decorrer do projeto tem um impacto significativamente positivo na precisão dessas estimativas, pois a medida em que vai se obtendo mais conhecimento sobre o projeto, melhor a precisão das estimativas geradas.

6. Deve haver aprendizagem sobre o processo de estimativa de esforço em reengenharia (L06)

Como todo processo que envolve grande fluxo conhecimento, é importante que haja um controle destas informações, pois do contrário poderão ser perdidas a medida que se perde as pessoas responsáveis pela realização das atividades. O efeito da aprendizagem nas atividades de estimativa de esforço foi estudado por [JS04], que mostrou que esta atividade é útil para mostrar o que deu errado em estimativas anteriores, de maneira a melhorar as habilidades de estimativa dos responsáveis por essa tarefa. Das organizações analisadas, apenas a Organização D possui a prática de usar explicitamente o conhecimento dos projetos finalizados para melhorar a técnica de estimativa (mas não o processo) (BP18 - Gestão do Conhecimento da Estimativa)

7. Deve haver gestão de riscos sobre o processo de estimativa (L07)

A maioria dos desafios identificados em relação a estimativa de esforço não podem ser tratados em tempo de realização da estimativa, pois são relacionados com fatores organizacionais e pessoais (do cliente, da equipe), sendo, portanto, riscos a serem analisados para o projeto. Jorgensen *et al.* [JH10] realizaram um estudo sobre como a identificação de riscos influencia a estimativa de esforço. Os resultados mostraram que há determinados contextos em que a estimativa pode se tornar mais otimista se mais esforços forem gastos no processo de análise de riscos. Este estudo também

mostrou que os riscos devem ser identificados antes da realização da estimativa, para que se avalie o quando eles devem afetar esta estimativa.

## **6.5 Conclusões do Capítulo**

Este capítulo apresenta a síntese dos resultados obtidos a partir da realização do estudo de campo inicial e do estudo de caso.

Os primeiros resultados dizem respeito a atividades de estimativa de esforço identificadas, que vão além daquelas tradicionalmente relacionadas a estimativa de esforço, por autores como [MEN14] [PRE11] e [SOM07], indicando assim que a estimativa de esforço é mais do que apenas realizar a estimativa, mas acompanhar o projeto de maneira a se manter dentro do estimado e também aprender com as experiências de estimativas passadas.

Em seguida foram agrupados os desafios e as boas práticas obtidas em ambos os estudos. Esse conjunto de dados culminou na identificação de um conjunto de lições aprendidas que, juntamente com as atividades e boas práticas servirão como base para concepção do modelo proposto.

Em relação aos desafios, estes são uma importante contribuição para o modelo, uma vez que podem ser mapeados em fatores de risco para o processo de estimativa e o modelo proposto visa identificar e tratar estes riscos. Além disso, os desafios são os fatores que desencadeiam as boas práticas. Assim, são importantes para identificar o contexto do processo em que estas práticas devem ser aplicadas, possibilitando, com isso, o mapeamento das práticas para as atividades do processo onde elas serão úteis.

O próximo capítulo apresenta a proposta do modelo de estimativa de esforço para projetos de reengenharia de software.



## **7. MODELO DE ESTIMATIVA DE ESFORÇO PARA PROJETOS DE REENGENHARIA DE SOFTWARE**

Neste capítulo, é apresentado o modelo proposto para estimativa de esforço em projetos de reengenharia de software. Este modelo é baseado nos resultados obtidos a partir da realização de um estudo de campo inicial (Capítulo 4) e de um estudo de caso (Capítulo 5), sobre a realização do processo de estimativa de esforço em reengenharia na prática.

O modelo é composto de subprocessos e dos relacionamentos entre eles. Além disso, oferece uma base para a condução de estimativa de esforço em projetos de reengenharia de software visando: (1) minimizar possíveis desafios, (2) garantir a sua viabilidade para diferentes processos de reengenharia, e (3) aprimorar a capacidade do processo de reengenharia como um todo.

A Seção 7.1 apresenta a estrutura do modelo, a Seção 7.2 descreve o subprocesso de Planejamento Estratégico, a Seção 7.3 apresenta o subprocesso de Planejamento da Estimativa, a Seção 7.4 apresenta o subprocesso de Estimativa, a Seção 7.5 apresenta o subprocesso de Monitoramento e Calibragem, a Seção 7.6 apresenta o subprocesso de Aprendizagem e a Seção 7.7 apresenta exemplos de aplicação do modelo para diferentes contextos de projetos de reengenharia.

### **7.1 Estrutura do Modelo**

O modelo foi elaborado para atuar como facilitador do processo de estimativa de esforço em projetos de reengenharia de software. A forma como ele foi concebido permite a identificação de fraquezas e oportunidades de melhorias nos projetos. Para isso, o modelo sugere a existência de duas dimensões: organizacional (onde está inserido o subprocesso de Planejamento Estratégico) e de projetos (onde estão inseridos os demais subprocessos). Este modelo difere das abordagens anteriormente propostas para estimativa de esforço em reengenharia (ver Capítulo 2), por não tratar de técnicas para cálculo da estimativa de esforço, mas sim de quais as atividades são relacionadas com a estimativa dentro do projeto de reengenharia. A Figura 7.1 apresenta a visão geral do modelo.

O primeiro subprocesso é o Planejamento Estratégico da estimativa. Este subprocesso envolve a definição das estratégias da organização em relação às estimativas para projetos de reengenharia, que deverão ser aplicadas em todos os projetos deste tipo ao longo do tempo. Esta fase deve ser realizada antes da realização dos projetos, pois utiliza o conhecimento obtido com os projetos finalizados para otimizar as estratégias de estimativa para novos projetos.

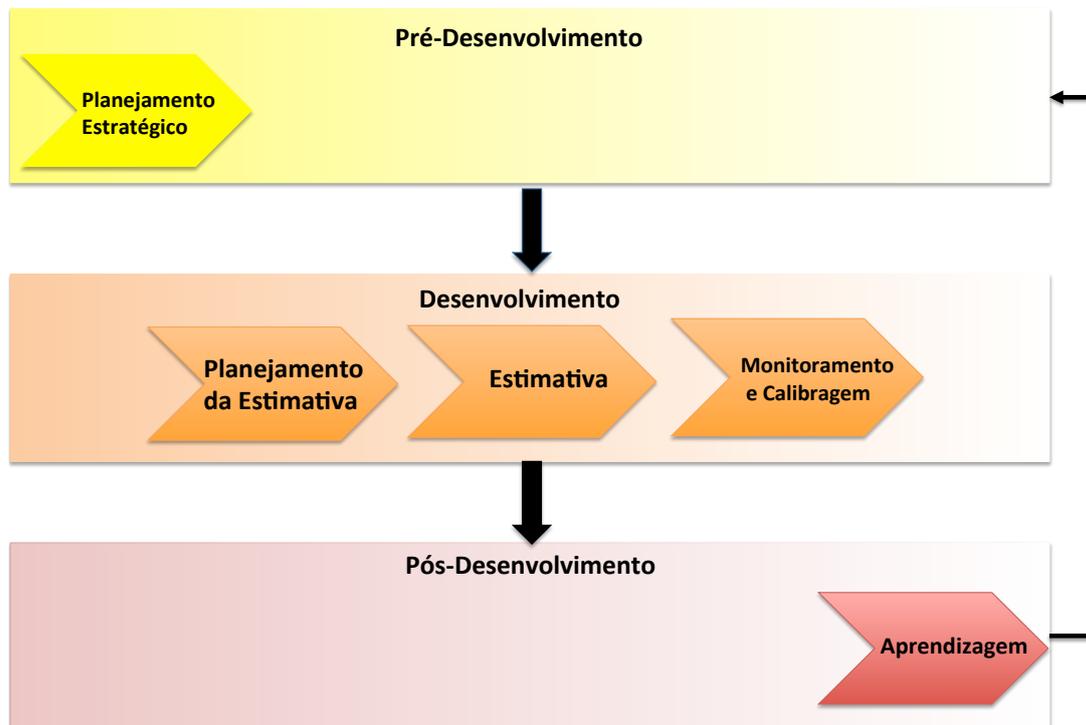


Figura 7.1 – Modelo Proposto

Em tempo de projeto, ou seja, quando a estimativa é efetivamente realizada, são propostos os subprocessos de Planejamento da Estimativa, Estimativa e Monitoramento e Calibragem. Aqui são definidas e aplicadas as técnicas de estimativa para o projeto, é realizado o acompanhamento e calibragem da estimativa e são apurados os dados para atualização da base histórica de projetos.

Por fim, após a realização de cada projeto é proposto o subprocesso de Aprendizagem, onde as lições aprendidas acerca do processo de estimativa vão ser identificadas e avaliadas, e servirão como base para realimentar o modelo, em uma nova etapa de Planejamento Estratégico.

#### 7.1.1 Mapeamento dos Desafios, Boas Práticas e Lições Aprendidas de Estimativa em Projetos de Reengenharia para o Modelo Proposto

Os subprocessos foram desenvolvidos com base na experiência obtida a partir da realização dos estudos empíricos apresentados nos Capítulos 4 e 5, e estão associados às boas práticas e lições aprendidas consolidadas no Capítulo 6, com o intuito de prevenir ou resolver possíveis desafios. Logo, estes desafios não estão diretamente inseridos no

modelo, mas serviram como motivação para criação de atividades, com o intuito prevenir sua ocorrência ou resolve-los caso ocorram.

A Tabela 7.1 apresenta o mapeamento boas práticas para o modelo de estimativa. Conforme apresentado na seção 6.4, as boas práticas BP08, BP14, BP16, BP17 e BP18 serviram de apoio a concepção de lições aprendidas, portanto, elas não foram mencionadas na formulação do modelo. As boas práticas B11, B12 e B13 estão relacionadas a mais de um subprocesso.

Tabela 7.1 – Mapeamento das Boas Práticas para o Modelo de Estimativa

<b>Subprocesso</b>	<b>Boa Prática</b>	<b>Título da Boa Prática</b>
Planejamento Estratégico	BP03	Conhecimento do processo de estimativa
	BP04	Conhecimento de diferentes técnicas de estimativa
	BP11	Estabelecer, manter e utilizar uma base histórica de projetos
Planejamento da Estimativa	BP12	Utilizar mais de uma técnica de estimativa
	BP13	Selecionar a técnica de acordo com as características do projeto
Estimativa	BP01	O sistema possuir documentação e ela estar atualizada
	BP05	Conhecimento do domínio do negócio
	BP06	Conhecimento nas tecnologias do sistema legado
	BP07	Conhecimento nas tecnologias do sistema alvo
	BP09	Realizar <i>workshop</i> para entendimento do sistema a ser migrado
	BP10	Ter suporte da equipe de manutenção do sistema
	BP11	Estabelecer, manter e utilizar uma base histórica de projetos
	BP12	Utilizar mais de uma técnica de estimativa
	BP13	Selecionar a técnica de acordo com as características do projeto

A Tabela 7.2 apresenta o mapeamento das lições aprendidas para os subprocessos do Modelo de Estimativa. A lição aprendida L06 está relacionada a mais de um subprocesso.

A Tabela 7.3 apresenta o mapeamento dos desafios para os subprocessos do Modelo de Estimativa, esse mapeamento indica o subprocesso onde os desafios foram tratados. Vale ressaltar que os desafios relacionados ao Cliente e a Organização não foram explicitamente tratados neste modelo, pois são externos ao projeto de reengenharia de software. Assim como no caso das boas práticas e lições aprendidas, há desafios que estão relacionados a mais de um subprocesso.

Tabela 7.2 – Mapeamento das Lições Aprendidas para o Modelo de Estimativa

<b>Subprocesso</b>	<b>Lição Aprendida</b>	<b>Título da Lição Aprendida</b>
Planejamento Estratégico	L06	Deve haver aprendizagem sobre o processo de estimativa de esforço de reengenharia
	L07	Deve haver gestão de riscos sobre o processo de estimativa
Planejamento da Estimativa	L01	Deve haver uma definição clara do escopo do projeto de reengenharia
Estimativa	L03	A estimativa calculada deve ser revisada antes de aprovada para o planejamento do projeto
	L02	A realização de um projeto preliminar é essencial para o sucesso da estimativa, principalmente quando não houver familiaridade com o tipo de projeto
Monitoramento e Calibragem	L04	Deve haver monitoramento constante das estimativas do projeto
	L05	Caso haja necessidade de recalibrar a estimativa, deve-se usar os dados do projeto
Aprendizagem	L06	Deve haver aprendizagem sobre o processo de estimativa de esforço de reengenharia

As próximas seções apresentam o detalhamento dos subprocessos e a descrição de como as boas práticas e lições aprendidas foram aplicadas no modelo.

## 7.2 Planejamento Estratégico

Estimativa de esforço é um dos principais requisitos para um gerenciamento efetivo de projeto de desenvolvimento de software, pois serve como base para derivação de custos, alocação de recursos e estimativa de prazos para o projeto [PRE11]. Isto serve para todos os tipos de desenvolvimento de software, e reengenharia não é uma exceção.

Porém, mesmo diante dessa importância da estimativa para o projeto, a gestão desta atividade é, na maioria das vezes, negligenciada pela alta gestão da organização, ficando a cargo apenas dos responsáveis pelo gerenciamento do projeto. Essas pessoas recebem crédito tanto pelo sucesso quanto pelo insucesso das estimativas, pois cabe à elas tomar todas as decisões acerca de como a estimativa será conduzida.

Assim, um dos grandes desafios percebidos durante os estudos empíricos conduzidos neste trabalho foi: a falta da definição de como o processo de estimativa deveria ser realizado e/ou de como proceder diante dos diferentes riscos a este processo.

Tabela 7.3 – Mapeamento dos Desafios para o Modelo de Estimativa

<b>Subprocesso</b>	<b>Desafio</b>	<b>Título do Desafio</b>
Planejamento Estratégico	D03	Falta de documentação
	D04	Má qualidade da documentação
	D10	Falta de conhecimento no negócio
	D11	Falta de conhecimento na tecnologia do sistema legado
	D13	Falta de experiência na realização das estimativas
	D23	Fator de aceleração não disponível em tempos de execução
	D24	Gestão de riscos
Estimativa	D01	Qualidade do código-fonte legado
	D02	Complexidade do código fonte legado
	D03	Falta de documentação
	D04	Má qualidade da documentação
	D09	Falta de conhecimento nas tecnologias do sistema alvo
	D10	Falta de conhecimento no negócio
	D11	Falta de conhecimento nas tecnologias do sistema legado
	D12	Falta de conhecimento do sistema
	D14	Escopo mal definido
	D15	Acréscimo de muitas funcionalidades novas em relação ao sistema legado
	D16	Falta de tempo para realizar a estimativa
	D17	Estimar esforço para a fase de engenharia reversa
	D18	Falta de dados para realizar a estimativa
	D19	Tamanho do projeto
	D20	Complexidade do projeto de Reengenharia
D24	Gestão de riscos	
Monitoramento e Calibragem	D22	Falta de similaridade entre projetos
	D24	Gestão de Riscos

Outro problema identificado foi a falta de Gestão do Conhecimento do processo de estimativa (quando existente), uma vez que as lições aprendidas ficam atualmente registradas apenas pelas pessoas envolvidas diretamente no projeto. Isto apresenta um grande risco de perda de conhecimento para a organização, no caso do afastamento ou desligamento dessas pessoas.

Sendo assim, este modelo propõe a realização de um subprocesso de Planejamento Estratégico, a ser realizado pela alta gestão da organização e pelos gerentes de projeto (e outros responsáveis pela estimativa de esforço em reengenharia).

A Figura 7.2 apresenta o processo de Planejamento Estratégico.

O Planejamento Estratégico envolve as etapas de:

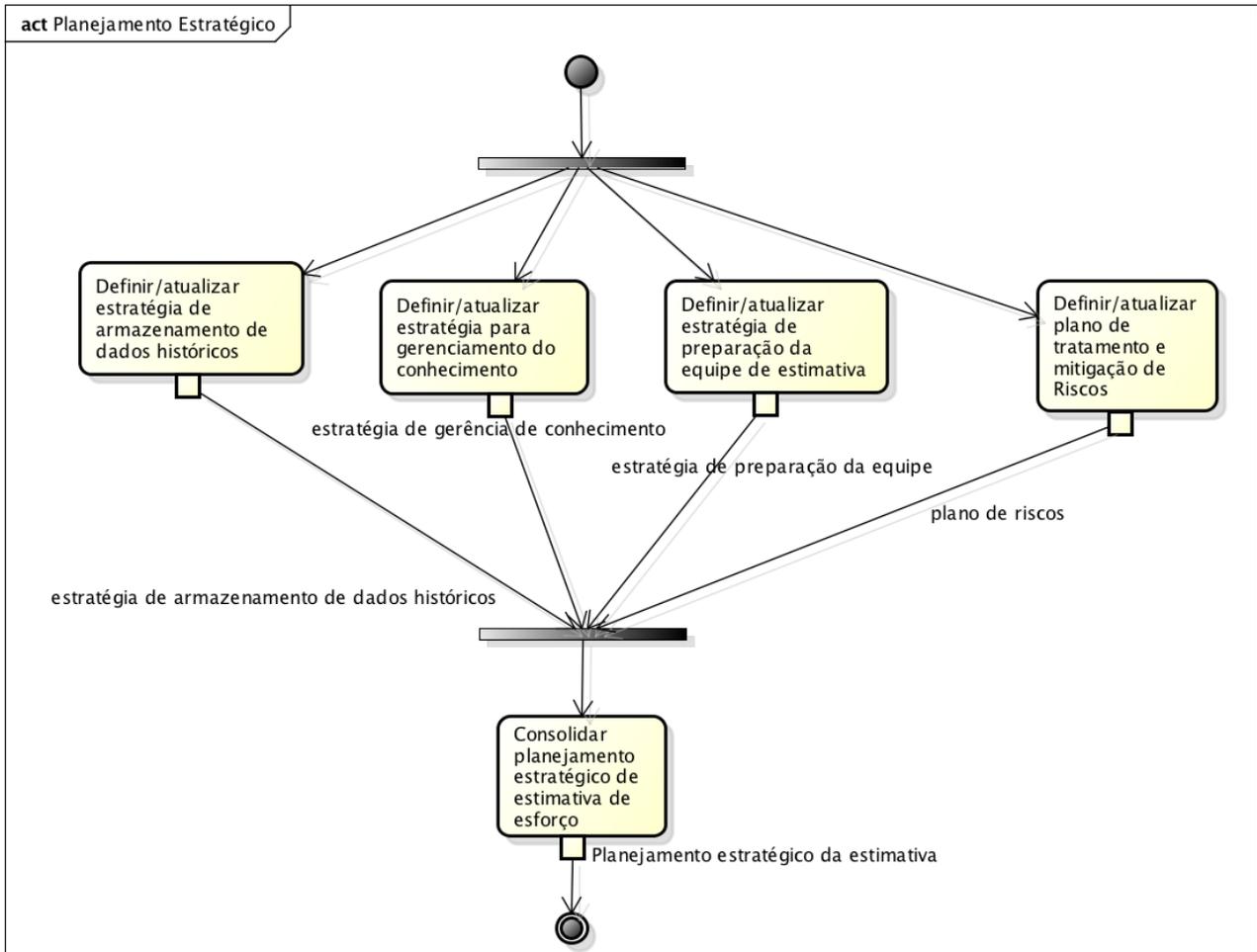


Figura 7.2 – Planejamento Estratégico

- Definir/atualizar estratégia de armazenamento de dados históricos
- Definir/atualizar estratégia de gerenciamento de conhecimento
- Definir/atualizar estratégia de preparação da equipe de estimativa
- Definir/atualizar plano de tratamento e mitigação de riscos
- Consolidar planejamento estratégico da estimativa de esforço

### 7.2.1 Definir/atualizar estratégia de armazenamento de dados históricos

Em um primeiro momento, caso a organização não possua uma base de dados das métricas dos projetos, deve-se planejar a criação desta base. A estratégia de criação da base histórica de métricas deve envolver a definição de quais dados deverão ser armazenados, o formato de armazenamento, a forma de acesso e manipulação dos dados.

Estabelecer, manter e utilizar uma base histórica de projetos (BP11) é uma boa prática amplamente divulgada na literatura e adotada pela maioria das organizações direta ou indiretamente. O indiretamente implica que a organização mantém registro dos dados do projeto de maneira descentralizada (espalhados por ferramentas e planilhas) e sem uma gestão adequada destes dados e, com isso, não consegue se beneficiar completamente das informações contidas nestes dados.

### 7.2.2 Definir/atualizar estratégia de gerenciamento de conhecimento

De acordo com a (L06), deve haver aprendizagem sobre o processo de estimativa de esforço em reengenharia. Esta aprendizagem pode ser alcançada a partir de uma estratégia de Gerência de Conhecimento.

No processo de estimativa de esforço, o conhecimento frequentemente perdido e/ou mal gerenciado está relacionado a falta de controle sobre as lições aprendidas nos projetos de reengenharia. Exemplos de lições aprendidas não gerenciadas incluem: adequação da técnica de estimativa ao projeto, processo de levantamento de dados para estimativa, satisfação em relação aos valores reais em função do estimado.

Este modelo propõe que a organização elabore uma estratégia de coleta e armazenamento dessas informações. Assim como no caso da base histórica de projetos, a estratégia de gerenciamento de conhecimento deve envolver: a definição de quais dados deverão ser armazenados, o formato de armazenamento, a forma de acesso e manipulação dos dados.

### 7.2.3 Definir/atualizar a estratégia de preparação da equipe de estimativa

Dada a existência de um processo de estimativa e que este processo deve empregar mais de uma técnica para o cálculo da estimativa (ver subprocesso Estimativa), é necessário que a equipe que realiza esta atividade esteja preparada para aplicar tais técnicas, de acordo com as boas práticas (BP03 - Conhecimento do processo de estimativa) e (B04 - Conhecimento de diferentes técnicas de estimativa).

Devido ao tempo limitado para realização das estimativas, esta preparação não deve ocorrer em tempos de projeto. Portanto, é necessário que se estabeleça e mantenha uma estratégia de preparação da equipe, que permita que ela esteja apta no momento da realização do projeto.

#### 7.2.4 Definir/atualizar plano de tratamento e mitigação de riscos

Adicionalmente, deve ser definido um plano para tratamento e mitigação de riscos. Segundo [SNE91], os riscos de um projeto de reengenharia devem ser significativamente menores do que o do desenvolvimento de um software do zero.

Assim, é necessário realizar uma Avaliação de Riscos, onde será analisado o custo/benefício de se realizar o projeto de reengenharia (L07 - Deve haver gestão de riscos sobre o processo de estimativa). De acordo com o modelo proposto neste trabalho, a análise de riscos deve ser feita por projeto, durante o subprocesso de Estimativa. O que se propõe no subprocesso de Planejamento Estratégico é que seja feito um plano de mitigação e tratamento de riscos, para caso uma determinada situação ocorra durante o projeto, exista um planejamento geral do que deve ser realizado, uma vez que muitos riscos são comuns a todos os projetos de reengenharia.

Dentre os riscos relacionados a estimativa de esforço, identificados como comuns aos projetos de reengenharia e que devem ser mitigados e/ou tratados, estão:

1. **Possuir insumos para realização da estimativa, tais como código-fonte legado e documentação do sistema legado:** um projeto de reengenharia sempre parte de um sistema legado. Porém, no momento da estimativa nem sempre esse sistema está disponível. No caso de organizações que trabalham com pré-venda de soluções, a estimativa deve ser realizada antes do início efetivo do projeto e muitas vezes o cliente, por questão de confidencialidade, não disponibiliza o código-fonte. Já a falta de documentação é um desafio recorrente de projetos de reengenharia, uma vez que quase nenhum sistema possui e quando possui está desatualizada.
2. **Conhecimento nas tecnologias do sistema legado:** embora um sistema legado não se caracterize exatamente por ser antigo, na maioria das vezes o é. Assim, um sistema escrito em uma linguagem de programação pouco conhecida, cujos profissionais da organização não estão aptos a analisar, pode se tornar um grande desafio a ser tratado, já que afeta tanto o processo de realização das estimativas (dificultando a análise do código e a extração das métricas) quanto o valor das mesmas (elevando os valores, dada a inexperiência da equipe na tecnologia).
3. **Conhecimento sobre o domínio do negócio:** os projetos de reengenharia são dos mais diversos domínios (por exemplo: bancários, gestão de recursos, previdência, mídia, etc). Assim como o conhecimento da tecnologia empregada no sistema legado afeta a estimativa, o conhecimento do domínio do sistema também afeta, pois cada domínio tem todo um glossário e regras próprias, cujo não entendimento pode dificultar a realização e os valores da estimativa.

4. **Possuir um fator de aceleração:** este risco afeta mais diretamente o valor da estimativa, uma vez que o fator de aceleração é o que faz com que um projeto de reengenharia tenha um esforço (e conseqüentemente custo) menores do que de um redesenvolvimento, já que envolve algum grau de reaproveitamento de recursos do sistema legado para o sistema alvo (novo). Um exemplo de fator de aceleração é a existência de apoio ferramental para realização da reengenharia em si, principalmente a parte de engenharia reversa. Sistemas legados envolvem diferentes tecnologias e domínios e as ferramentas existentes hoje em dia são muito específicas em relação a isso. Logo, é importante que a organização tenha um planejamento sobre o que fazer na falta desses fatores de aceleração.

#### 7.2.5 Consolidar planejamento estratégico de estimativa de esforço

Uma vez que a organização estabeleça as estratégias de armazenamento de conhecimento, de armazenamento de dados históricos, de preparação da equipe de estimativa e de planejamento de tratamento e mitigação de riscos, deve-se iniciar a melhoria contínua destes a partir dos dados obtidos no subprocesso de Aprendizagem. O objetivo é que o que foi planejado nesta fase seja revisto para comportar as novas experiências obtidas com os projetos concluídos. Tais revisões podem incluir alteração nos riscos, alteração dos dados a serem armazenados na base histórica, alteração nos dados a serem armazenados na base de conhecimento.

### 7.3 Planejamento da Estimativa

Este subprocesso é realizado no início do projeto, no início de um módulo ou, como no caso de organizações como a Organização A, durante a pré-venda do projeto. O momento da realização varia de acordo com o processo de reengenharia da organização. O ideal que é se tenha uma ideia clara da demanda a ser desenvolvida, ou seja, deve-se ter conhecimento do escopo do projeto (L01 - Deve haver uma definição clara do escopo do projeto de reengenharia), principalmente no que tange o tipo de reengenharia que deve ser realizada (reengenharia de código e/ou de dados, e se será necessário fazer a reengenharia de documentação).

A Figura 7.3 apresenta o subprocesso de Planejamento da Estimativa.

A seguir são descritas as atividades relacionadas ao subprocesso:

- Analisar demanda
- Consultar base de conhecimento

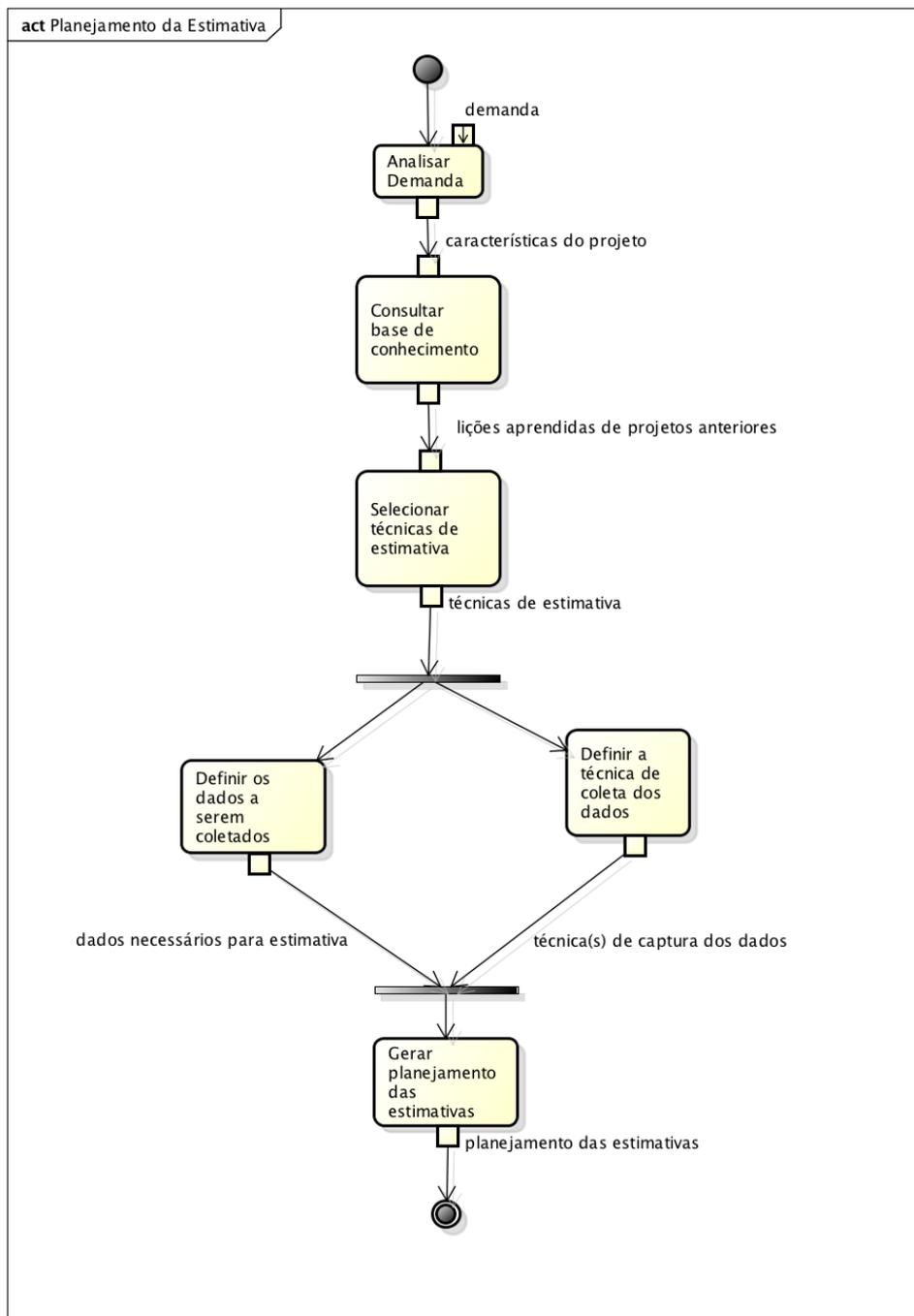


Figura 7.3 – Planejamento da Estimativa

- Selecionar técnicas de estimativa
- Definir os dados a serem coletados
- Definir as técnicas de coleta dos dados
- Gerar planejamento das estimativas

### 7.3.1 Analisar Demanda

A demanda do projeto (que pode ser o escopo, uma descrição em alto nível feita pelo cliente e registrada em uma ata de reunião, dentro outras) deve ser analisada e a partir dela devem ser extraídas as características do projeto, com o maior detalhamento possível. Informações importantes que devem ser levadas em consideração nesta fase são: o escopo da reengenharia, a quantidade e tipo de documentação disponível, porte do projeto, domínio de desenvolvimento e as tecnologias envolvidas.

### 7.3.2 Consultar base de conhecimento

Além das características do projeto, o responsável pela estimativa pode se embasar em lições aprendidas de projetos anteriores para selecionar as técnicas de estimativa a serem empregadas. Exemplos de lições aprendidas que podem ser utilizadas nesta fase incluem: qual técnica usar para um determinado tipo de projeto, quais as melhores combinações de técnicas, qual o tempo médio para a aplicação de uma determinada técnica, etc.

### 7.3.3 Selecionar técnicas de estimativa

Esta atividade está relacionada com a (BP12 - Utilizar mais de uma técnica de estimativa) e com a (BP13 - Selecionar a técnica de acordo com as características projeto). A primeira prática visa minimizar o viés de se utilizar apenas uma técnica de estimativa, já que nenhuma é 100% assertiva. Assim, de acordo com [BAC00] uma composição de técnicas não garante um acerto total, mas tende a permitir uma aproximação maior do valor real, uma vez que cada técnica pode analisar o problema sob uma perspectiva.

Além de utilizar mais de uma técnica de estimativa deve-se selecionar as técnicas de acordo com o projeto. Este alinhamento permite tirar um maior proveito do que se tem disponível sobre o projeto, selecionando-se uma técnica que possa fazer uma melhor utilização dessas informações.

### 7.3.4 Definir os dados a serem coletados

Uma vez definidas as técnicas, deverão ser definidos quais dados deverão ser coletados para a aplicação de cada técnica. Isto é necessário pois cada uma tem seu

conjunto de parâmetros e estes devem ser informados com precisão, para se obter um melhor resultado.

A variedade de técnicas disponíveis é muito grande e cada técnica requer um conjunto de parâmetros de entrada próprio. Os tipos de parâmetros que podem ser considerados são diversos. Alguns exemplos de parâmetros de entrada comuns em técnicas de estimativa de esforço são: grau de experiência da equipe, uso ou não de ferramentas de apoio, etc.

Vale ressaltar a maioria das técnicas utiliza como base o tamanho do projeto. No caso da reengenharia há dois tamanhos: o do sistema atual e o do sistema alvo (novo).

### 7.3.5 Definir as técnicas de coleta dos dados

Assim como cada técnica de estimativa de esforço tem seus parâmetros de entrada específicos, cada parâmetro pode requerer uma técnica de coleta de dados diferente. Dentre as técnicas comumente utilizadas estão a Análise do Sistema Legado (código-fonte), Análise da Documentação e Entrevista com os *Stakeholders*. A aplicação de cada uma dessas técnicas é discutida no subprocesso Estimativa. Porém, é importante que durante o Planejamento da Estimativa, o responsável pelas estimativas tenha em mente os dados necessários e como e onde obter estes dados.

Como dito na seção 7.3.4, o tamanho é o parâmetro base para a maioria das técnicas de estimativa de esforço. No caso do tamanho do sistema legado, este pode ser medido diretamente pela análise do código-legado (caso esteja disponível) ou estimado a partir da documentação do projeto. Já o tamanho do sistema alvo (sistema após a reengenharia) deve ser estimado a partir de uma técnica de estimativa de tamanho, como contagens de linha de código, APF e UCP.

### 7.3.6 Gerar planejamento das estimativas

O resultado do conjunto de atividades realizadas deve compor o Planejamento das Estimativas: um documento que servirá como entrada para a realização da estimativa em si. É importante frisar que, além da demanda do projeto, a base de conhecimento também deverá estar disponível como entrada para este subprocesso e pode servir como apoio à tomada de decisão em quaisquer das atividades a serem realizadas.

## 7.4 Estimativa

Este subprocesso é onde efetivamente vai ser realizada a estimativa de esforço, com base no Planejamento da Estimativa realizado na etapa anterior. A Figura 7.4 apresenta o subprocesso de Estimativa.

A seguir são descritas as atividades envolvidas das três etapas principais deste subprocesso:

- Levantamento de Dados e Análise de Riscos
- Estimativa
- Comparação e Validação

### 7.4.1 Levantamento de Dados e Análise de Riscos

Nessa etapa são levantados os dados (parâmetros) necessários para aplicação de cada técnica, de acordo com o definido no Planejamento da Estimativa. Além disso, são analisados os riscos do projeto que poderão interferir nas estimativas.

#### Levantamento de Dados

O levantamento dados é o momento onde o responsável pela estimativa obtém os dados necessários para a aplicação das técnicas. Geralmente, nesta fase, há apoio da equipe de Análise, já que esta atividade pode envolver coleta de dados com as partes interessadas no sistema (*stakeholders*) e análise do sistema legado.

Um possível desafio desta fase é o tempo disponível para fazer o levantamento dos dados, principalmente se esta estimativa estiver sendo realizada antes do início do projeto e se o nível de conhecimento da equipe sobre o projeto for baixo. Não é possível especificar o tempo necessário para levantar os dados, dado que vários fatores como complexidade do sistema legado, qualidade do código, qualidade da documentação, disponibilidade dos entrevistados, dentro outros, devem ser levados em consideração. Porém, uma boa prática identificada neste caso é a realização de um projeto preliminar, também chamado pré-projeto, antes da realização das estimativas (L02 - A realização de um projeto preliminar é essencial para o sucesso da estimativa, principalmente quando não houver familiaridade com o tipo de projeto).

O projeto preliminar consiste em selecionar um módulo do sistema legado e aplicar o processo de reengenharia sobre este módulo para que, a partir daí, se avalie melhor a

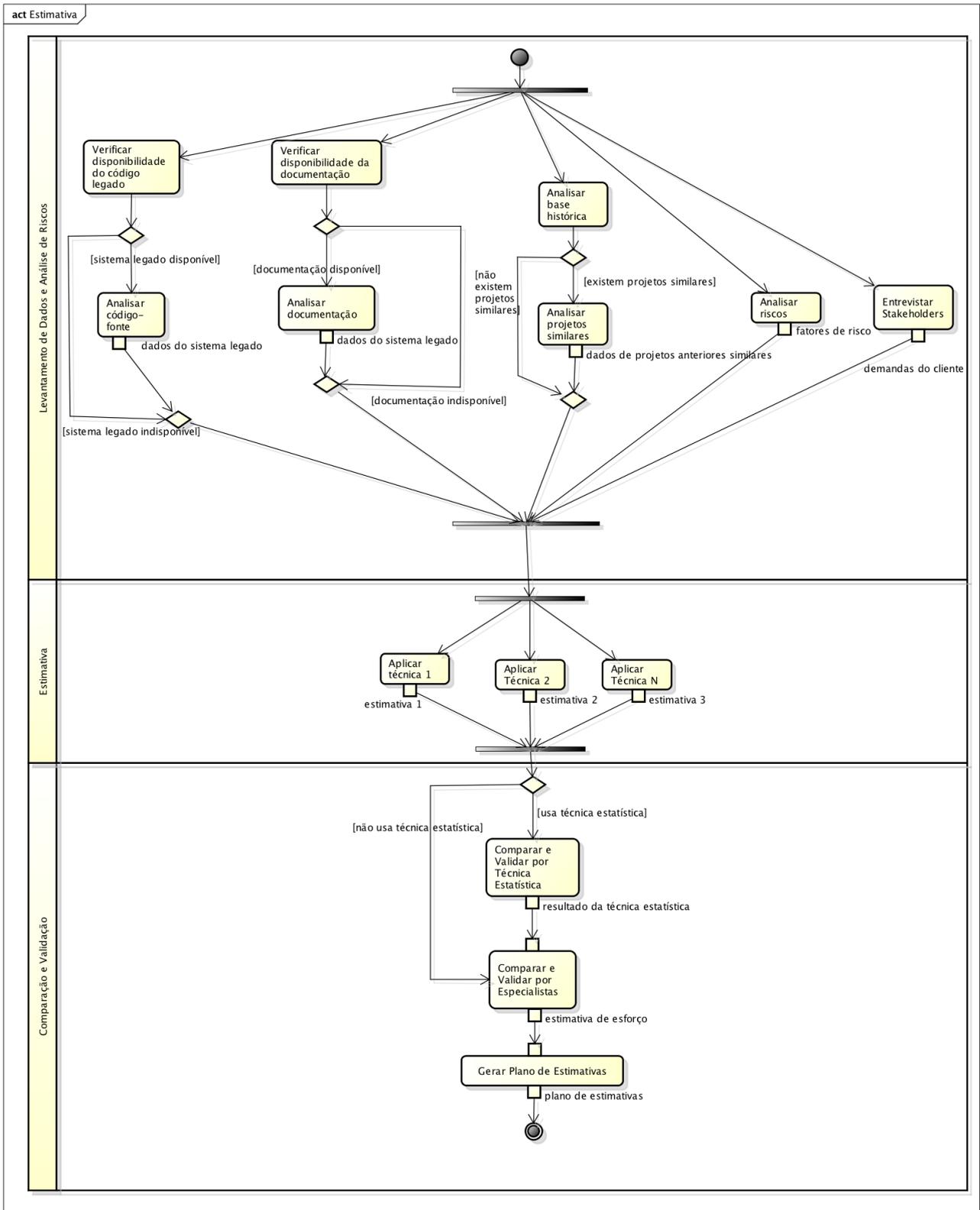


Figura 7.4 – Estimativa de Esforço

complexidade do projeto como um todo e se possa obter uma maior assertividade nas estimativas.

Uma boa prática que pode substituir a realização de um projeto preliminar, no caso da falta de conhecimento ser relacionada apenas ao funcionamento do sistema a ser migrado, é realizar um *workshop* de imersão (BP09 - Realizar *workshop* para entendimento do sistema a ser migrado).

Para o levantamento dos dados, o conhecimento do domínio do negócio (BP05), o conhecimento das tecnologias do sistema legado (BP06) e do sistema alvo (BP07) são diferenciais positivos em favor da equipe que realiza a estimativa.

Em relação as técnicas de levantamento de requisitos, as principais são as seguintes:

1. **Analisar código-fonte:** consiste na análise do código-fonte do sistema legado, de onde os responsáveis pela análise podem extrair fatores quantitativos e qualitativos para servirem como base na aplicação da técnica. Dentre estes dados estão: complexidade do código legado, número de linhas de código, número de transações, qualidade do sistema. Nesta fase, pode-se contar com o apoio do time que realizou a manutenção do sistema, caso ele esteja disponível, já que esta equipe tem um amplo conhecimento dos detalhes de implementação do sistema (BP10 - Ter suporte da equipe de manutenção do sistema).
2. **Analisar Documentação:** o sistema possuir documentação e ela estar atualizada (BP01) pode ser uma grande vantagem para coletar informações sobre este sistema, logo, é importante analisar esta documentação como complemento à análise do sistema. Porém, deve-se atentar para o fato de que a documentação pode não estar atualizada em relação ao sistema e, sendo assim, ela não pode ser a única forma de coleta de dados.
3. **Entrevistar *stakeholders*:** é o método mais tradicional de levantamento de informações sobre o projeto. Na reengenharia, esta etapa é necessária principalmente quando há acréscimo de novas funcionalidades no sistema alvo. Mesmo em casos que o projeto de reengenharia consista apenas em uma migração de código é essencial realizar entrevistas com as partes interessadas no sistema, para complementar os dados que serão obtidos pela análise no mesmo.

Dentre os parâmetros a serem identificados para estimativa de esforço, o principal é o tamanho, que deve ser identificado a partir de uma estimativa. Esta estimativa pode ser derivada por contagem de linhas de código, e técnicas como APF e UCP. O detalhamento da estimativa de tamanho está fora do escopo deste trabalho.

## Analisar Riscos

Complementarmente ao levantamento de dados, deve ser feita a análise dos riscos do projeto e esta análise também deve ser levada em consideração na realização da estimativa.

Nesta fase, deve-se identificar os riscos e o calcular o peso de cada um para o projeto. O Plano de Tratamento e Mitigação de Riscos, elaborado e mantido no Planejamento Estratégico, deve servir como guia para esta avaliação.

## Analisar Projetos Anteriores

Com base nas características do projeto a ser desenvolvido, os responsáveis por realizar a estimativa devem verificar na base histórica de projetos se há similaridade entre o projeto atual e algum já desenvolvido pela organização. Se houver, as informações deste(s) projeto(s) devem ser resgatadas para, juntamente com as características do projeto atual, auxiliarem na realização da estimativa.

Em relação a similaridade entre projetos, esta pode ser em termos de tipo de negócio, tecnologias utilizadas, características da solução, etc. Aqui, o ponto crítico é identificar os projetos similares. Atualmente isto é predominantemente realizado manualmente, com o auxílio de especialistas que conhecem os projetos passados. Porém, existe todo um campo de estudo de estimativa de esforço baseada em aprendizagem de máquina que, dentre outras coisas, busca soluções para encontrar a similaridade entre projetos (Apêndice A).

### 7.4.2 Estimativa

É a atividade onde as técnicas de estimativa selecionadas serão efetivamente aplicadas para gerar a estimativa de esforço do projeto ou de um módulo do projeto. Nesta etapa, cada técnica deve ser aplicada separadamente, seguindo as suas especificações e utilizando os parâmetros coletados no Levantamento de Dados. Podem ser utilizadas ferramentas de apoio específicas para estimativa, embora a maioria das empresas trabalhe com planilhas Excel.

### 7.4.3 Comparação e Validação

Nesta etapa os valores estimados pelas técnicas de estimativa devem ser comparados e validados afim de gerar os valores finais de estimativa (L03 - A estimativa calculada deve ser revisada antes de aprovada para o planejamento do projeto). Para isto, pode ser

aplicada uma abordagem estatística como média ou uma técnica como a PERT [PP01]. Independentemente de ser aplicada uma técnica estatística é importante que seja feita a avaliação por especialistas, pois além dos fatores técnicos, que são usados como parâmetros pelas técnicas de estimativa, é importante que os fatores não-técnicos que podem afetar as estimativas também sejam ponderados. Exemplos desses fatores envolvem: interesses organizacionais, interesses do cliente, interesses pessoais da equipe, dentre outros.

### Gerar Plano de Estimativas

Por fim, uma vez realizada a estimativa e estabelecido os valores que são aplicados para o projeto ou para a etapa do projeto que se esteja estimando, deve-se registrar tais decisões em um Plano de Estimativas, este plano pode ainda ser estendido para incluir as estimativas de custo e tempo, que são derivadas do esforço, mas que estão fora do escopo deste trabalho.

## 7.5 Monitoramento e Calibragem

A estimativa de esforço deve ser calculada para o projeto como um todo ou para uma determinada parte deste projeto. Independente disto, deve-se monitorar o andamento deste projeto, comparando-se o que foi estimado com os valores reais, para avaliar se o projeto está fluindo como planejado ou se é necessária a realização de uma calibragem nas estimativas (L04 - Deve haver monitoramento constante das estimativas do projeto). A frequência do monitoramento é uma decisão gerencial e deve estar de acordo com o modelo de processo de reengenharia adotado. Porém, uma boa prática é que este monitoramento seja constantemente pois, assim, é possível perceber e contornar mais rapidamente possíveis desvios no projeto.

A Figura 7.5 apresenta o subprocesso de Monitoramento e Calibragem.

O subprocesso tem como entrada as métricas de execução do projeto e o Plano Estimativas. O subprocesso é composto por três etapas:

1. Monitoramento
2. Calibragem
3. Atualização da Base

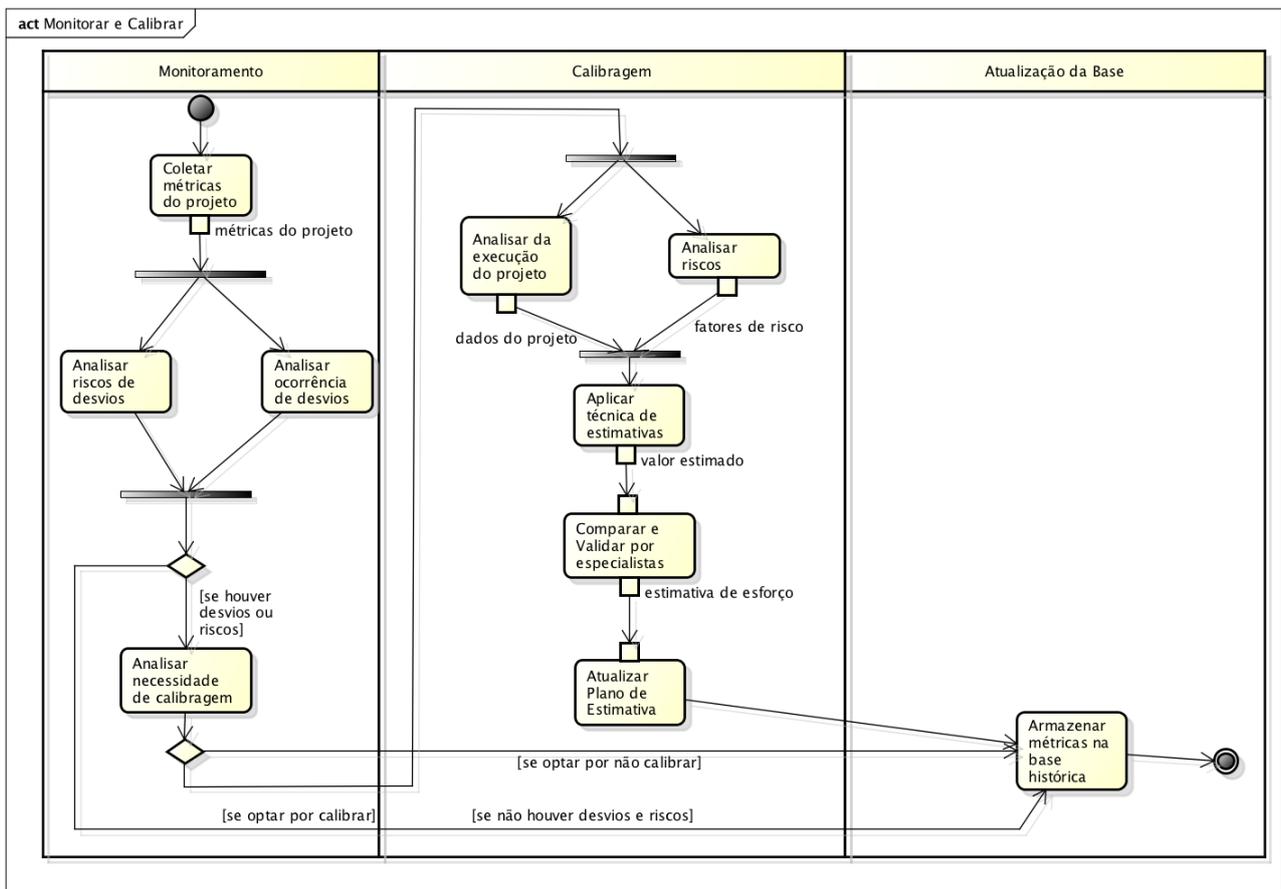


Figura 7.5 – Monitoramento e Calibragem

### 7.5.1 Monitoramento

#### Analisar ocorrência de desvios

Com base nas métricas de andamento do projeto deve-se avaliar a ocorrência de desvios dos valores estimados. Dificilmente os valores estimados serão exatamente iguais aos realizados. Porém, é importante que de maneira geral, para o período em que foi realizada a estimativa, tenha havido um equilíbrio no esforço despendido nas tarefas, de maneira que o esforço total esteja dentro do previsto.

#### Analisar riscos de desvios

Mesmo nos casos em que o projeto não apresente desvios nas estimativas é necessário avaliar os riscos de que estes desvios venham a ocorrer nas próximas atividades. Portanto, deve ser realizada uma análise dos riscos atuais e futuros do projeto, dentro do período da estimativa realizada. O plano de mitigação e tratamento de riscos de estimativas deve ser utilizado como apoio nesta atividade.

## Analisar necessidade de calibragem

A necessidade de realização de calibragem deve ser considerada caso haja riscos de desvios futuros ou já estejam ocorrendo desvios nas estimativas. Embora seja uma boa prática, muitas organizações que trabalham com projeto de escopo fechado encontram dificuldades em justificar este reajuste junto ao cliente, principalmente se os desvios forem causados por falhas no processo. Porém, deve-se analisar minuciosamente as causas do desvio e/ou risco e a possibilidade de calibragem, já que optar por não alterar as estimativas durante o projeto aumenta o risco de insucesso do mesmo.

### 7.5.2 Calibragem

A Calibragem nada mais é do que a realização do subprocesso de Estimativa novamente, mas usando os dados do projeto em execução como entrada para o cálculo das estimativas (L05 - Caso haja necessidade de recalibrar a estimativa, deve-se usar os dados do projeto). Esta boa prática permite que haja um aprendizado na realização das estimativas, buscando aumentar sua assertividade.

### 7.5.3 Atualização da base

Independentemente de haver ou não Calibragem nos valores da estimativa, as métricas do projeto, coletadas até aquele ponto de monitoramento, deverão ser armazenadas de acordo com o previsto na estratégia de armazenamento de dados históricos. Estes dados servirão não só para embasar estimativas de projetos futuros, mas para calibragem das estimativas do próprio projeto, conforme visto na etapa de Calibragem.

## 7.6 Aprendizagem

O último ciclo do modelo proposto ocorre após o desenvolvimento do projeto e diz respeito ao aprendizado, sendo caracterizado por um processo de *Avaliação e Feedback*. Este processo é importante à medida que se busca sempre uma melhoria no processo de desenvolvimento e no produto final. No decorrer dos estudos, foi identificado que as organizações, com ênfase na Organização A, que foi analisada no Estudo de Caso, enfrentam grandes desafios no que tange a estimativa de esforço nos projetos de reengenharia. Estes desafios motivam a aplicação de boas práticas e geram lições aprendidas sobre os projetos. Porém, este conhecimento é apenas tácito.

O objetivo deste subprocesso é tornar explícito o conhecimento adquirido durante a realização do projeto de reengenharia, acerca da estimativa de esforço (L06 - Deve haver aprendizagem sobre o processo de estimativa de esforço em reengenharia).

A Figura 7.6 apresenta o subprocesso de Aprendizagem.

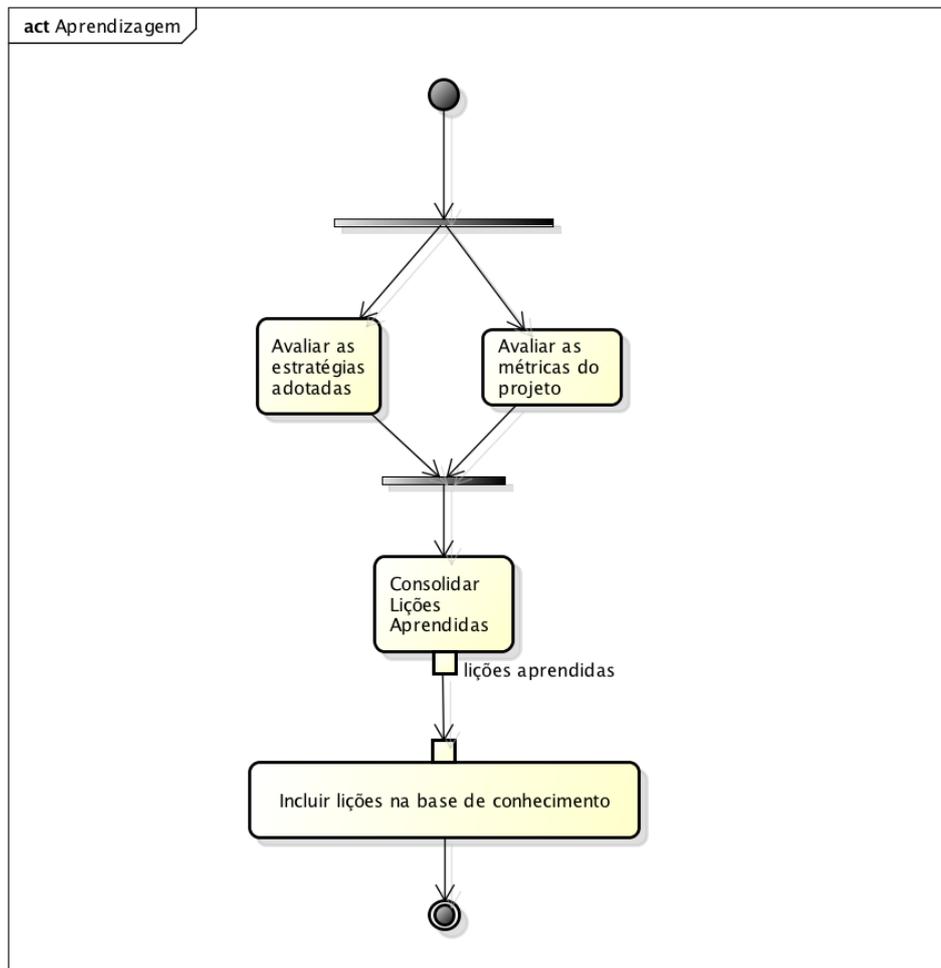


Figura 7.6 – Aprendizagem

As atividades envolvidas neste subprocesso são:

- Avaliar as estratégias adotadas
- Avaliar as métricas do projeto
- Consolidar as lições aprendidas
- Incluir lições na base de conhecimento

### 7.6.1 Avaliar as estratégias adotadas e Avaliar as métricas do projeto

Ao final do projeto deve haver uma reunião envolvendo os responsáveis pela realização e acompanhamento das estimativas naquele projeto, onde serão discutidas as estratégias de estimativa adotadas para o projeto (técnicas, práticas aplicadas, etc), juntamente com os resultados dos valores reais de esforço obtidos no projeto (confrontados com os valores estimados).

### 7.6.2 Consolidar as lições aprendidas e Incluir lições aprendidas na base conhecimento

A partir da avaliação das estratégias adotadas no projeto e das métricas obtidas, devem ser geradas e registradas as lições aprendidas sobre estimativa de esforço naquele projeto. Tais lições realimentarão o Planejamento Estratégico do processo de estimativa de esforço, possibilitando que haja uma melhoria contínua neste processo, com base nos erros e acertos cometidos dos projetos concluídos.

Dentre as organizações analisadas nos estudos empíricos, somente a Organização D aplica este processo parcialmente: anualmente são analisados os dados dos projetos finalizados, para melhorar a técnica de estimativa aplicada na organização.

## 7.7 **Modelo de Estimativa de Esforço em Diferentes Contextos de Processos de Reengenharia**

Os processos de desenvolvimento de software, incluindo o de reengenharia, geralmente são adaptados ao contexto organizacional. No caso do processo de reengenharia, a adaptação geralmente ocorre de acordo com o tipo de reengenharia ser realizada, ou seja, se é uma reestruturação de código, de dados, se deve haver reestruturação da documentação, etc.

Como um dos objetivos do modelo proposto era o de se manter flexível a essas mudanças de estrutura do processo de reengenharia, ele não foi relacionado a um processo específico. Porém, como este modelo foi baseado no estudo da prática de estimativa de esforço em algumas organizações de desenvolvimento de software, cabe apresentar uma visão geral de como este modelo poderia se comportar no contexto das organizações estudadas.

As organizações apresentam basicamente dois cenários de desenvolvimento de software: as organizações A e B realizam desenvolvimento para clientes externos, logo, seus projetos são de escopo fechado, com o esforço sendo calculado antes do início do

projeto e mantido durante a execução do mesmo. Já as organizações C e D desenvolvem software para clientes internos, o que permite uma maior flexibilidade na realização das estimativas, podendo estas serem replanejadas a cada módulo do projeto.

A Figura 7.7 apresenta uma visão geral de como o modelo proposto pode se comportar no contexto de projetos com escopo fechado e no de projetos com escopo aberto. Os números 1, 2, 3 e 4 correspondem aos subprocessos de Planejamento da Estimativa, Estimativa, Monitoramento e Calibragem, e Aprendizagem, respectivamente.

O subprocesso de Planejamento Estratégico foi propositalmente omitido, pois conforme descrito, este processo deve ser realizado anteriormente ao desenvolvimento de um projeto em particular.

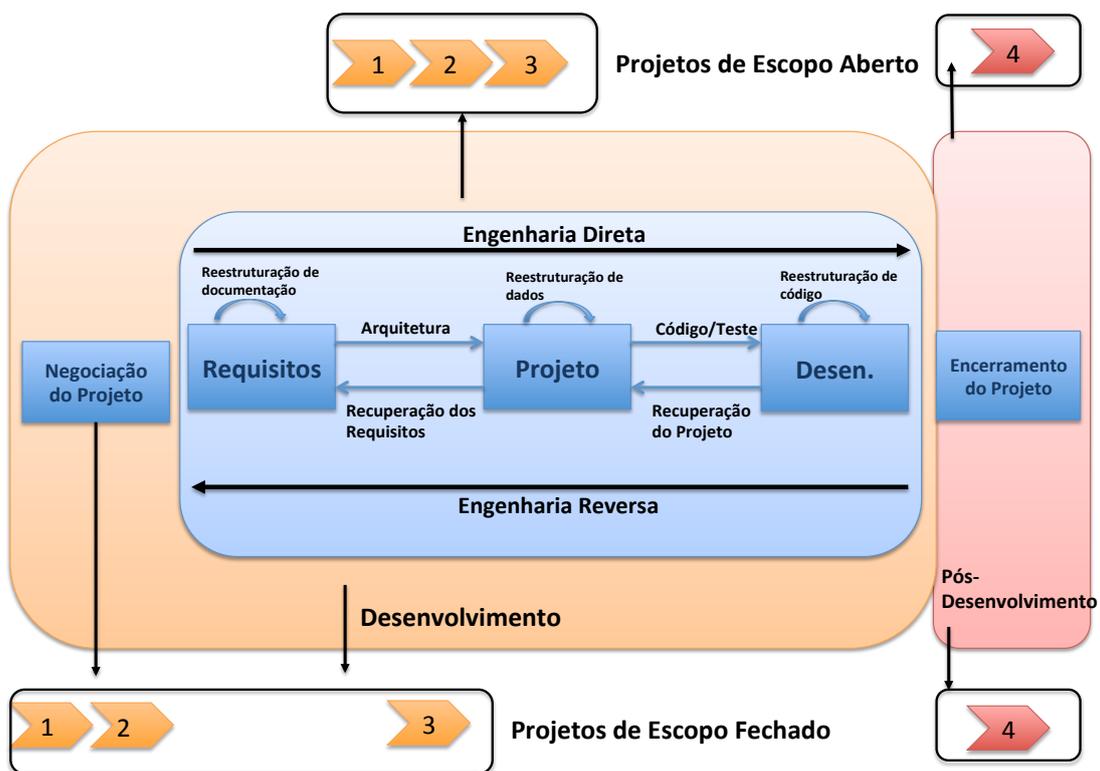


Figura 7.7 – Modelo de Estimativa de Esforço em Diferentes Contextos de Processos de Reengenharia

A seguir, é feita uma descrição de como o modelo proposto pode ser mapeado para esses contextos.

### 7.7.1 Projetos de Escopo Fechado

No contexto de projetos de escopo fechado, a estimativa de esforço é planejada (subprocesso Planejamento) e realizada (subprocesso Estimativa) antes do início do projeto, em uma etapa de Negociação do Projeto. Isto ocorre pois as estimativas obtidas neste momento (esforço, custo, prazo) servirão como base para a negociação do contrato junto ao cliente.

Neste tipo de projeto, o principal problema para a estimativa é que nem cliente nem fornecedor conseguem antever todo o escopo do projeto. Com isso, as estimativas feitas nesta fase podem se mostrar inadequadas a medida em que o projeto avança e sua real complexidade é identificada.

O modelo proposto auxilia neste problema sugerindo boas práticas para aumentar a confiança nos dados obtidos nesta fase tais como, realização de pré-projeto, realização de *workshop*, utilizando do conhecimento de especialistas no sistema e alocando equipe com conhecimento no tipo de projeto a ser desenvolvido.

Outro ponto forte do modelo proposto, para este contexto, é o monitoramento do projeto (subprocesso Monitoramento e Calibragem), que auxiliará na identificação de desvios ou riscos de desvios nas estimativas realizadas. Este acompanhamento constante também é útil para se estabelecer pontos de atualização da base histórica do projeto, principalmente se esta coleta não é feita automaticamente. Isto evita que dados desatualizados ou incorretos sejam inseridos ao final do projeto, por falta de planejamento de atualização constante da base.

Embora a calibragem das estimativas não seja uma alternativa bem vista neste tipo de projeto, principalmente do ponto de vista do cliente, ela pode ser uma alternativa inevitável, diante da ocorrência de grandes desvios, como foi o caso do Projeto 2, analisado durante o Estudo de Caso (Capítulo 5). Assim, o modelo proposto visa garantir que se é inevitável que ocorra calibragem, que essa seja feita com o maior rigor possível e utilizando os dados do próprio projeto como base.

Por fim, o aprendizado das lições do projeto (subprocesso Aprendizagem) auxiliará na melhoria do processo e na tomada de decisão para projetos futuros.

### 7.7.2 Projetos de Escopo Aberto

No contexto de projetos com o escopo aberto, a cada módulo de execução do projeto pode ser feito o planejamento (subprocesso Planejamento) e a realização da estimativa (subprocesso Estimativa), assim como o monitoramento e calibragem (subprocesso Monitoramento e Calibragem).

Assim como nos projetos de escopo fechado, neste tipo de projeto o principal problema para a estimativa é que nem cliente nem equipe de desenvolvimento conseguem antever todo o escopo do projeto. A diferença é que aqui há flexibilidade para o replanejamento das estimativas caso o escopo sofra mudanças significativas.

Esta flexibilidade faz com que o modelo proposto atue como um facilitador do processo de estimativa, guiando o planejamento e o monitoramento deste processo. As boas práticas e lições aprendidas podem ser aplicadas para execução do processo de estimativa da maneira mais completa e eficiente, sem os desafios inerentes de falta de tempo e recursos.

Por fim, assim como nos projetos de escopo fechado, o aprendizado das lições do projeto (subprocesso Aprendizagem) auxiliará na melhoria do processo e na tomada de decisão para projetos futuros

## **7.8 Conclusões do Capítulo**

Este capítulo apresentou a proposta de um modelo de estimativa de esforço para projetos de reengenharia de software, baseado em boas práticas e lições aprendidas da realização deste processo no contexto real de desenvolvimento de software.

Diferentemente das propostas existentes de apoio a estimativa de esforço em reengenharia, este modelo não visa prover uma forma de cálculo da estimativa, mas o processo geral de como essa estimativa deve ser realizada, de maneira a mitigar os possíveis desafios existentes.

Entende-se que a estrutura do processo de estimativa dependerá diretamente do processo de reengenharia adotado, por isso o modelo não foi proposto em relação a um processo de reengenharia específico, mas sim em um nível de abstração que permite aplicar os subprocessos de estimativa de maneira flexível ao processo geral, conforme os exemplos apresentados.

## 8. CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

Este capítulo apresenta as considerações finais deste trabalho, incluindo uma discussão dos resultados, e destaca sua contribuição para pesquisadores e profissionais que trabalham ou são interessados em estimativa de esforço em projetos de reengenharia. As sugestões de trabalhos futuros fornecem indicações de como poderão ser realizadas validação e melhorias no modelo proposto.

### 8.1 Discussão dos Resultados

A gestão bem sucedida de projetos de software começa com uma estimativa precisa do esforço de desenvolvimento [PRE11] [SOM07]. Imprecisão nas estimativas continua a ser um dos fatores-chave que contribuem para falhas de projeto de software [STA13].

No contexto de projetos de reengenharia de software, as técnicas de estimativa de esforço propostas até então são voltadas para converter uma estimativa de tamanho (como um número estimado de linhas de código-fonte) para uma estimativa de esforço (pessoas por hora, dia, mês ou ano). Este tipo de abordagem limita o contexto de estimativa apenas para a fase de cálculo dela.

Este trabalho teve como objetivo fornecer uma opção a abordagem de estimativa de esforço como sendo apenas o cálculo da estimativa, a partir da apresentação de um processo completo, que abrange todo o ciclo de reengenharia, desde o seu planejamento até sua finalização, visando a melhoria contínua da estimativa de esforço, a partir de sub-processos de Planejamento Estratégico e Aprendizagem. O modelo proposto foi baseado na realização de estudos empíricos, onde foram identificadas atividades, desafios e práticas relacionados a realização de estimativa de esforço em projetos de reengenharia de software.

A identificação do conhecimento tácito proveniente da experiência dos especialistas entrevistados e a sua consequente transformação em conhecimento explícito a partir do relato dessa experiência e consolidação do modelo é importante para a área de estimativa de esforço em reengenharia de software. Este relato permite utilizar esse conhecimento para apoiar a tomada de decisão na realização dessas estimativas. Além disso, colabora para a pesquisa na área, no sentido de apresentar o estado da prática, seus principais desafios e suas práticas, que podem servir como insumos para o desenvolvimento de novas soluções.

Espera-se que este trabalho possa contribuir para a área de pesquisa em reengenharia, a partir dos protocolos de pesquisa disponibilizados, que permitem replicar e ampliar esta pesquisa para identificação da prática de estimativa de esforço em reengenharia.

Além de contribuir para a indústria de desenvolvimento de software, a partir da utilização do modelo proposto para estabelecer ou melhorar o processo de estimativa de esforço em projetos de reengenharia de software.

A seguir são relatadas as principais contribuições e limitações da pesquisa, bem como os trabalhos futuros que podem ser desenvolvidos a partir dela.

## **8.2 Contribuições**

As principais contribuições deste trabalho são:

1. Proposição de um modelo para estimativa de esforço em projetos de reengenharia de software, que levou em consideração as atividades, boas práticas e lições aprendidas sobre o processo, a partir da realização de estudos empíricos na indústria de desenvolvimento de software
2. A identificação o estado da arte em soluções para estimativa de esforço em projetos de desenvolvimento de software, a partir da realização de uma revisão sistemática da literatura.
3. Disponibilização dos protocolos de estudo de campo e estudo de caso, bem como os instrumentos de coleta de dados, para que outros pesquisadores possam expandir e replicar esta pesquisa.

## **8.3 Limitações e Ameaças a Validade**

Como em todos os estudos de natureza empírica, houve várias ameaças que podem afetar a validade dos resultados. A principal ameaça para a validade dos resultados é o pequeno número de entrevistados que participaram da pesquisa, restringindo a generalização dos resultados obtidos. Entretanto, deve-se destacar que os entrevistados e as organizações analisadas são casos representativos de realização de estimativa de esforço em reengenharia e possibilitaram a obtenção de uma vasta quantidade de informações sobre como este processo ocorre no mercado. Esta abordagem é típica do tipo de pesquisa desenvolvida (exploratória e de base qualitativa), permitindo o uso de inferências nas conclusões obtidas.

Outra ameaça para a validade dos resultados é a subjetividade da classificação de dados, uma vez que a análise qualitativa foi realizada por apenas um pesquisador. O Método de Comparação Constante foi utilizado a fim de diminuir essa ameaça, uma vez que este método requer que toda a análise seja baseada nos dados coletados.

Por fim, não foi possível no prazo da pesquisa a realização de algum tipo de avaliação ou validação do modelo. Porém, tentou-se embasar ao máximo a sua estrutura no que foi considerado como o estado da prática da estimativa em reengenharia, obtido através dos estudos empíricos e do estudo da teoria.

#### 8.4 Trabalhos Futuros

Identifica-se grande potencial de crescimento nesta linha de pesquisa, onde os pontos fortes envolvem uma parceria estável entre academia e a indústria, criando condições de experimentação de aprendizagem. Como pesquisas futuras sugere-se:

- Avaliação do modelo a partir da análise de especialistas, tanto de pesquisadores na área de estimativa de esforço e/ou de reengenharia quanto de profissionais da indústria. Pretende-se utilizar o *feedback* obtido nesta avaliação para aperfeiçoamento do modelo proposto;
- Elaboração e aplicação de um estudo de caso para avaliar o modelo em um contexto real de reengenharia de software;
- Generalização do modelo para o contexto de desenvolvimento de software novo, mantendo as atividades-base e incluindo os fatores que impactam neste tipo de desenvolvimento.



## REFERÊNCIAS BIBLIOGRÁFICAS

- [AAMPR13] AL-ANI, B.; Marczak, S.; Prikladnicki, R.; Redmiles, D. “Revisiting the factors that engender trust of global systems engineers”. In: IEEE 8th International Conference on Global Software Engineering (ICGSE), 2013, pp. 31–40.
- [ABdNT08] Associação Brasileira de Normas Técnicas, A. “NBR ISO 9001: Sistemas de gestão da qualidade-requisitos”. ABNT, 2008, 28p.
- [AO10] ATTARZADEH, I.; Ow, S. H. “Proposing a novel artificial neural network prediction model to improve the precision of software effort estimation”. In: Bio-Inspired Models of Network, Information, and Computing Systems - 5th International ICST Conference, BIONETICS 2010, Boston, MA, USA, December 1-3, 2010, Revised Selected Papers, 2010, pp. 334–342.
- [BAC00] BOEHM, B.; Abts, C.; Chulani, S. “Software development cost estimation approaches — a survey”, *Annals of Software Engineering*, vol. 10–1, nov 2000, pp. 177–205.
- [BBCV06] BALDASSARRE, M. T.; Boffoli, N.; Caivano, D.; Visaggio, G. “Speed: Software project effort evaluator based on dynamic-calibration.” In: ICSM, 2006, pp. 272–273.
- [BBdSdC13] BASGALUPP, M. P.; Barros, R. C.; da Silva, T. S.; de Carvalho, A. C. “Software effort prediction: A hyper-heuristic decision-tree based approach”. In: Proceedings of the 28th Annual ACM Symposium on Applied Computing, 2013, pp. 1109–1116.
- [BBR12] BASGALUPP, M. P.; Barros, R. C.; Ruiz, D. D. “Predicting software maintenance effort through evolutionary-based decision trees”. In: Proceedings of the 27th Annual ACM Symposium on Applied Computing, 2012, pp. 1209–1214.
- [BCH+95] BOEHM, B. W.; Clark, B.; Horowitz, E.; Westland, J. C.; Madachy, R. J.; Selby, R. W. “Cost models for future software life cycle processes: Cocomo 2.0.”, *Ann. Software Eng.*, vol. 1, 1995, pp. 57–94.
- [BCV03] BALDASSARRE, M. T.; Caivano, D.; Visaggio, G. “Software renewal projects estimation using dynamic calibration.” In: ICSM, 2003, pp. 105–115.
- [BDSM12] BENALA, T. R.; Dehuri, S.; Satapathy, S. C.; Madhurakshara, S. “Genetic algorithm for optimizing functional link artificial neural network based software cost estimation”. In: Proceedings of the International Conference on Information Systems Design and Intelligent Applications 2012 (INDIA 2012) held in Visakhapatnam, India, January 2012, 2012, pp. 75–82.

- [BEN09] BENTLEY, C. "Prince2: a practical handbook". Routledge, 2009, 322p.
- [BIG89] BIGGERSTAFF, T. J. "Design recovery for maintenance and reuse", *Computer*, vol. 22-7, 1989, pp. 36-49.
- [BOE84] BOEHM, B. W. "Software engineering economics", *IEEE Transactions on Software Engineering*, vol. SE-10-1, January 1984, pp. 4-21.
- [BRO95] BRODIE, M. L. "Migrating Legacy Systems: Gateways, Interfaces the Incremental Approach". Morgan Kaufmann Publishers, 1995, 120p.
- [BS14] BASTEN, D.; Sunyaev, A. "A systematic mapping of factors affecting accuracy of software development effort estimation", *Communications of the Association for Information Systems (CAIS)*, vol. 34, 2014, pp. 51-86.
- [BST+05] BERGEY, J.; Smith, D.; Tilley, S.; Weiderman, N.; Woods, S. "Why reengineering projects fail", Technical report, Carnegie Mellon University, Software Engineering Institute, 2005, 30p.
- [CC90] CHIKOFFSKY, E. J.; Cross, J. H. "Reverse engineering and design recovery: A taxonomy", *IEEE Software*, vol. 7-1, 1990, pp. 13-17.
- [CLV06] CAIVANO, D.; Lanubile, F.; Visaggio, G. "Software renewal process comprehension using dynamic effort estimation." In: ICSM, 2006, pp. 209-218.
- [CMM10] CMMI Product, T. "Cmmi for development, version 1.3", Relatório Técnico, Carnegie Melon, 2010, 468p.
- [DLPS02] DE LUCIA, A.; Pompella, E.; Stefanucci, S. "Effort estimation for corrective software maintenance". In: Proceedings of the 14th international conference on Software engineering and knowledge engineering, 2002, pp. 409-416.
- [FLGdC11] FACELI, K.; Lorena, A. C.; Gama, J.; de Carvalho, A. "Inteligência Artificial: Uma Abordagem de Aprendizado de Máquina". São Paulo: LTC, 2011, 1 ed., 377p.
- [FOR68] FORRESTER, J. W. "Industrial dynamics-after the first decade", *Management Science*, vol. 14-7, 1968, pp. 398-415.
- [FWD97] FINNIE, G. R.; Wittig, G. E.; Desharnais, J.-M. "A comparison of software effort estimation techniques: using function points with neural networks, case-based reasoning and regression models", *Journal of Systems and Software*, vol. 39-3, 1997, pp. 281-289.
- [GIL08] GIL, A. "Métodos e técnicas de pesquisa social". Atlas, 2008, 216p.

- [GM97] GRAY, A. R.; MacDonell, S. G. "A comparison of techniques for developing predictive models of software metrics", *Information and software technology*, vol. 39–6, 1997, pp. 425–437.
- [HCC08] HUANG, S.-J.; Chiu, N.-H.; Chen, L.-W. "Integration of the grey relational analysis with genetic algorithm for software effort estimation", *European Journal of Operational Research*, vol. 188–3, 2008, pp. 898–909.
- [HOP96] HOPPEN, N. "Um guia para avaliação de artigos de pesquisas em sistemas de informação", *Revista eletrônica de administração*, vol. 2–2, 1996, pp. 31–52.
- [JEN83] JENSEN, R. "An improved macrolevel software development resource estimation model", 1983, pp. 88–92.
- [JH10] JØRGENSEN, M.; Halkjelsvik, T. "The effects of request formats on judgment-based effort estimation", *Journal of Systems and Software*, vol. 83–1, 2010, pp. 29–36.
- [JON91] JONES, C. "Applied Software Measurement: Assuring Productivity and Quality". New York, NY, USA: McGraw-Hill, Inc., 1991, 618p.
- [JØR05] JØRGENSEN, M. "Evidence-based guidelines for assessment of software development cost uncertainty", *Software Engineering, IEEE Transactions on*, vol. 31–11, 2005, pp. 942–954.
- [JS04] JØRGENSEN, M.; Sjøberg, D. I. "The impact of customer expectation on software development effort estimates", *International Journal of Project Management*, vol. 22–4, 2004, pp. 317–325.
- [JS07] JØRGENSEN, M.; Shepperd, M. "A systematic review of software development cost estimation studies", *Software Engineering, IEEE Transactions on*, vol. 33–1, 2007, pp. 33–53.
- [KAJH99] KHOSHGOFTAAR, T. M.; Allen, E. B.; Jones, W. D.; Hudepohl, J. P. "Classification tree models of software quality over multiple releases". In: *Software Reliability Engineering, 1999. Proceedings. 10th International Symposium on*, 1999, pp. 116–125.
- [KC07] KITCHENHAM, B.; Charters, S. "Guidelines for performing systematic literature reviews in software engineering", Relatório Técnico, Keele University and Durham University Joint Report, 2007, 57p.
- [KG11] KUMAR, A.; Gill, B. "Maintenance vs. reengineering software systems", *Global Journal of Computer Science and Technology*, vol. 310–23, 2011.

- [KJZ+11] KHATIBI, V.; Jawawi, D. N. A.; Zaiton, S.; Hashim, M.; Khatibi, E. "A new fuzzy clustering based method to increase the accuracy of software development effort estimation", *World Applied Sciences Journal*, 2011, pp. 1265–1275.
- [KTB08] KULTUR, Y.; Turhan, B.; Bener, A. B. "Enna: software effort estimation using ensemble of neural networks with associative memory". In: Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of software engineering, 2008, pp. 330–338.
- [LP95] LEDERER, A. L.; Prasad, J. "Causes of inaccurate software development cost estimates", *Journal of systems and software*, vol. 31–2, 1995, pp. 125–134.
- [MCG95] MCGRATH, E. "Methodology matters: Doing research in the behavioral and social sciences". In: Readings in Human-Computer Interaction: Toward the Year 2000, 1995, pp. 152–169.
- [MEN14] MENDES, E. "Practitioner's Knowledge Representation: A Pathway to Improve Software Effort Estimation". Springer Publishing Company, Incorporated, 2014.
- [MFCM13] MATOS, O.; Fortaleza, L.; Conte, T.; Mendes, E. "Realising web effort estimation: A qualitative investigation". In: Proceedings of the 17th International Conference on Evaluation and Assessment in Software Engineering, 2013, pp. 12–23.
- [NSB08] NGUYEN, V.; Steece, B.; Boehm, B. "A constrained regression technique for cocomo calibration". In: Proceedings of the Second ACM-IEEE international symposium on Empirical software engineering and measurement, 2008, pp. 213–222.
- [PFL04] PFLEEGER, S. L. "Engenharia de Software: Teoria e Prática". São Paulo: Prentice Hall, 2004, 2 ed., 537p.
- [PMI14] PMI. "Um guia do conjunto de conhecimentos em gerenciamento de projetos". Editora Saraiva, 2014, 5 ed., 496p.
- [PP01] PETERS, J.; Pedrycz, W. "Engenharia de Software". Rio de Janeiro: Campus, 2001, 560p.
- [PQ00] PIEKARSKI, A. E. T.; Quináia, M. A. "Reengenharia de software: o quê, porquê e como", *Revista Ciências Exatas e Naturais*, vol. 1–2, 2000, pp. 33–51.
- [PRE11] PRESSMAN, R. S. "Engenharia de Software". São Paulo: McGraw-Hill, 2011, 7 ed., 780p.
- [PUT78] PUTNAM, L. H. "A general empirical solution to the macro software sizing and estimating problem.", *IEEE Trans. Software Eng.*, vol. 4–4, 1978, pp. 345–361.

- [RH96] ROSENBERG, L.; Hyatt, L. "Software reengineering", Technical report, Software Assurance Technology Center, 1996, 31p.
- [SB95] SUBRAMANIAN, G. H.; Breslawski, S. "An empirical analysis of software effort estimate alterations", *Journal of Systems and Software*, vol. 31–2, 1995, pp. 135–141.
- [SB01] SCHWABER, K.; Beedle, M. "Agile Software Development with Scrum". Upper Saddle River, NJ, USA: Prentice Hall PTR, 2001, 1st ed., 158p.
- [SC90] STRAUSS, A.; Corbin, J. M. "Basics of qualitative research: Grounded theory procedures and techniques". Sage Publications, Inc, 1990, 272p.
- [SEA08] SEAMAN, C. "Qualitative Methods, In: Guide to Advanced Empirical Software Engineering". São Paulo: Springer-Verlag, 2008, 1 ed., 389p.
- [SF95] SRINIVASAN, K.; Fisher, D. "Machine learning approaches to estimating software development effort", *Software Engineering, IEEE Transactions on*, vol. 21–2, 1995, pp. 126–137.
- [SMLE02] SHAN, Y.; McKay, R. I.; Lokan, C. J.; Essam, D. L. "Software project effort estimation using genetic programming". In: Communications, Circuits and Systems and West Sino Expositions, IEEE 2002 International Conference on, 2002, pp. 1108–1112.
- [SNE91] SNEED, H. M. "Economics of software reengineering", *Journal of Software Maintenance: Research and Practice*, vol. 3–3, 1991, pp. 163–182.
- [SNE95] SNEED, H. M. "Planning the reengineering of legacy systems", *IEEE Software*, vol. 12–1, 1995, pp. 24–34.
- [SNE05] SNEED, H. M. "Estimating the costs of a reengineering project". In: WCRE, 2005, pp. 111–119.
- [Sof14] Software, V. "Maxqda: The art of text analysis". Capturado em: <http://www.maxqda.com>, Maio 2014.
- [SOM07] SOMMERVILLE, I. "Engenharia de Software". São Paulo, SP: Addison Wesley, 2007, 8 ed., 578p.
- [SS97] SHEPPERD, M. J.; Schofield, C. "Estimating software project effort using analogies.", *IEEE Trans. Software Eng.*, vol. 23–11, 1997, pp. 736–743.
- [SS08] SINGER, J.; Sim, S. "Software Engineering Data Collection for Field Studies, In: Guide to Advanced Empirical Software Engineering". São Paulo: Springer-Verlag, 2008, 1 ed., 389p.

- [STA13] STANDISH GROUP, T. “Chaos manifesto”. Capturado em: <http://www.versionone.com/assets/img/files/ChaosManifesto2013.pdf>, julho de 2014.
- [SYB08] SEO, Y.-S.; Yoon, K.-A.; Bae, D.-H. “An empirical analysis of software effort estimation with outlier elimination”. In: Proceedings of the 4th international workshop on Predictor models in software engineering, 2008, pp. 25–32.
- [TEN10] TENÓRIO, N. N. “Modelo–e10: Um modelo para estimativas de esforço em manutenção de software”, Dissertação de Mestrado, Faculdade de Informática – PUCRS, Porto Alegre, RS, Brasil, 2010, 134p.
- [TGdA02] TRAVASSOS, G. H.; Gurov, D.; do Amaral, E. “Introdução à engenharia de software experimental”, Relatório Técnico, COPPE, UFRJ, Rio de Janeiro, 2002, 52p.
- [TMKiM12] TSUNODA, M.; Monden, A.; Keung, J. W.; ichi Matsumoto, K. “Incorporating expert judgment into regression models of software effort estimation.” In: APSEC, 2012, pp. 374–379.
- [TRR08] TENÓRIO, N.; Ribeiro, M.; Ruiz, D. “A quasi-experiment for effort and defect estimation using least square linear regression and function points”. In: Software Engineering Workshop, 2008. SEW '08. 32nd Annual IEEE, 2008, pp. 143–151.
- [VIS01] VISAGGIO, G. “Ageing of a data-intensive legacy system: symptoms and remedies”, *Journal of Software Maintenance and Evolution: Research and Practice*, vol. 13–5, 2001, pp. 281–308.
- [WFT+15] WITTEN, I. H.; Frank, E.; Trigg, L.; Hall, M.; Holmes, G.; Cunningham, S. J. “Weka 3: Data mining software in java”. Capturado em: <http://www.cs.waikato.ac.nz/ml/weka/>, janeiro 2015.
- [WLT09] WEN, J.; Li, S.; Tang, L. “Improve analogy-based software effort estimation using principal components analysis and correlation weighting”. In: Software Engineering Conference, 2009. APSEC '09. Asia-Pacific, 2009, pp. 179–186.
- [WM94] WATSON, I.; Marir, F. “Case-based reasoning: A review”, *The knowledge engineering review*, vol. 9–04, 1994, pp. 327–354.
- [YIN10] YIN, R. “Estudo de caso: planejamento e métodos”. Bookman, 2010.
- [ZN13] ZIAUDDIN, Shahid Kama, S. K.; Nasir, J. A. “A fuzzy logic based software cost estimation model”, *International Journal of Software Engineering and Its Applications*, vol. 7–2, 2013, pp. 7–18.

## APÊNDICE A – REVISÃO SISTEMÁTICA: SOLUÇÕES PARA ESTIMATIVA DE ESFORÇO EM PROJETOS DE DESENVOLVIMENTO DE SOFTWARE

Após a realização do estudo de base teórica, sentiu-se necessidade de fazer uma revisão da literatura para o conhecimento da área de estimativa de esforço, com foco nas soluções que foram propostas nesta área, em especial as voltadas para os projetos de re-engenharia de sistemas. Com isso, pretendia-se a identificação de oportunidades na área, bem como de trabalhos relacionados. Desta forma, uma vez que existe um grande corpo de conhecimento na área, optou-se por utilizar o método de pesquisa Revisão Sistemática da Literatura por meio da pesquisa em bases científicas.

### A.1 INTRODUÇÃO

Em um estudo realizado pelo *The Standish Group* em 1995, denominado de *Chaos Report*, cujo foco foi a indústria de software comercial, foi detectado que 31,1% dos projetos de software foram cancelados, antes mesmo de serem concluídos e 52,7% custaram 189% a mais do que o previsto inicialmente, enquanto que apenas 16,2% dos projetos são completados no tempo e custo estimados. Embora muitos destes projetos entregues não reflitam o que foi especificado, uma vez que apenas 42% dos requisitos originais são desenvolvidos.

Embora essa pesquisa tenha sido realizada na década de 90, ainda hoje estes números são reais na indústria de software. O *Chaos Report* de 2012 identifica que apesar de 39% dos projetos serem entregues no prazo, orçamentos e terem recursos necessários para as funções, 43% dos projetos enfrentam situações opostas a esta, pois estão atrasados, acima do orçamento e/ou sem os recursos necessários para sua conclusão e 18% foram cancelados antes do término ou entregues e nunca utilizados [SG12].

Dentre os fatores identificados pelo *The Standish Group* [SG12] como sendo determinantes do insucesso do projeto está predominantemente o estouro do tempo e/ou do orçamento. Os dados mostram que em 2012 cerca de 74% dos projetos ultrapassaram o tempo previsto, enquanto 59% ultrapassaram o orçamento.

Estes dados de projetos de software reais evidenciam o que a literatura acerca de Gerenciamento de Projetos já afirma amplamente: que a estimativa de custos de desenvolvimento de software representa uma importante área de pesquisa e é uma das tarefas mais relevantes no planejamento e gerenciamento de processo de desenvolvimento de software [SOM07] [PRE11] [PFL04].

Dada a relevância da realização de estimativas de esforço muitas soluções foram propostas no intuito de auxiliar este processo. Estas soluções envolvem: (1) modelos algorítmicos, como, por exemplo, COCOMO [BOE81] [BOE95] e SLIM [PUT78]; (2) Julgamento de Especialistas [TSU12] [JØR05]; (3) Regressão Estatística [NGU08] [SEO08] [RIB08]; (4) Modelos Dinâmicos e soluções Orientadas à Aprendizagem de Máquina, que incluem, (4.1) Árvores de Regressão e Classificação [BAS13] [BAS12], (4.2) Lógica Fuzzy [KHA11] [ZIA13] [SAD11] [AZZ09] [BEN12a] [SAX12], (4.3) Redes Neurais Artificiais [ATT12] [KUL08] [SHU08], (4.4) Algoritmos Genéticos [BEN12b] [SHA02] [LIN11a], (4.5) Estimativa por Analogia [SHE96] [MEN00] [WEN09], dentre outras.

No contexto de soluções de apoio a realização de estimativas de esforço, esta revisão sistemática tem como objetivo identificar e categorizar tais trabalhos, as técnicas utilizadas, os domínios de desenvolvimento em que são aplicados, bem como o processo de validação pelo qual estas soluções são submetidas, de maneira a apoiar o desenvolvimento de novas soluções com base nas melhores práticas aplicadas nas soluções já existentes.

Esta revisão está organizada em 7 seções, da seguinte forma:

- SEÇÃO A.1 – apresenta introdução do trabalho;
- SEÇÃO A.2 – apresenta os conceitos principais de estimativa de esforço;
- SEÇÃO A.3 – apresenta o protocolo da revisão sistemática executada;
- SEÇÃO A.4 – apresenta a execução da fase I da revisão;
- SEÇÃO A.5 – apresenta a execução da fase II da revisão;
- SEÇÃO A.6 – apresenta os resultados da revisão;
- SEÇÃO A.7 – apresenta as considerações finais.

## A.2 ESTIMATIVA DE ESFORÇO EM DESENVOLVIMENTO DE SOFTWARE

A gestão bem sucedida de projetos de software começa com uma estimativa precisa do esforço de desenvolvimento [PRE11] [SOM07]. Imprecisão nas estimativas continua a ser um dos fatores-chave que contribuem para falhas de projeto de software [SG12]. Subestimar o esforço a ser gasto gera pressões que comprometem a qualidade do desenvolvimento do projeto, enquanto que, superestimar pode elevar o custo e diminuir a competitividade.

Técnicas de estimativa de esforço para projeto de software são projetadas para converter uma estimativa de tamanho (como um número estimado de linhas de código-fonte) para uma estimativa de esforço (pessoas por hora, dia, mês ou ano).

A previsão exata tem sido difícil, já que muitos desses relacionamentos não são bem compreendidos. Melhorar as técnicas de estimativa disponíveis para gerentes de projeto facilitaria o controle mais eficaz do tempo e os orçamentos no desenvolvimento de software [CW97].

### A.2.1 TÉCNICAS DE ESTIMATIVA DE ESFORÇO

Esta revisão adota a taxonomia sugerida por [ABT00] por entender que esta, além de ser a primeira proposta de taxonomia na área, é a mais adequada ao estado da arte das estimativas classificando, se não todas, grande parte das técnicas existentes na literatura. Assim, as técnicas de estimativas são classificadas como se segue:

- Baseadas em Modelo;
- Dinâmicas;
- Baseadas em Expertise;
- Baseadas em Regressão;
- Orientadas a Aprendizagem de Máquina;
- Compostas.

#### **Baseadas em Modelo**

O final dos anos 70 produziu uma grande variedade de modelos robustos de estimativa como o SLIM [PUT78], Checkpoint [JON97], o PRICE-S [PAR88], SEER [JEN83] e COCOMO [BOE81]. Muitos deles são modelos proprietários e, portanto, não podem ser

comparados e contrastados, em termos de estrutura do modelo. Teoria ou experimentação determinam a forma funcional destes modelos. Esta seção discute SLIM e COCOMO81, dois dos mais populares modelos conhecidos.

### **SLIM**

Larry Putnam desenvolveu o modelo de ciclo de vida de software (Software Life-Cycle Model - SLIM) no final de 1970 [PUT78]. O SLIM é baseado na análise do ciclo de vida de Putnam em termos de uma chamada distribuição Rayleigh da equipe do projeto versus tempo. Ele suporta a maioria dos métodos de estimativa de tamanho populares, incluindo técnicas de estimativa, instruções de código, pontos por função, etc. Ele faz uso de uma curva de Rayleigh para estimar esforço do projeto, cronograma e taxa de defeito. SLIM pode gravar e analisar dados de projetos previamente preenchidos que são então utilizados para calibrar o modelo, ou se os dados não estão disponíveis, um conjunto de perguntas podem ser respondidas para obter estes valores a partir do banco de dados existente.

### **COCOMO**

O modelo de estimativa de custo e cronograma (Constructive Cost Model - CO-COMO) foi originalmente publicado em 1981 [BOE81]. Tornou-se um dos mais populares modelos paramétricos de estimativa de custo da década de 1980. Porém, devido à evolução nos processos de desenvolvimento de software, o COCOMO '81 passou a ter dificuldades em estimar os custos de software desenvolvidos para novos processos de ciclo de vida e competências. O esforço de pesquisa para o COCOMO II foi iniciado em 1994 na Universidade do Sul da Califórnia para abordar as questões sobre modelos não sequenciais e processos rápidos de desenvolvimento, reengenharia, reutilização, abordagens orientadas a objeto, etc. COCOMO II foi publicado em 1995 [BOE95] e tem três submodelos: Composição de Aplicação – mais conveniente para a fase de prototipação em um ciclo de vida espiral - , Design precoce – quando os requisitos são bem conhecidos e a arquitetura do software foi bem explorada - e Pós-arquitetura de modelos – onde se trabalha com o real desenvolvimento e manutenção do produto de software.

## **Dinâmicas**

Técnicas dinâmicas reconhecem explicitamente que esforço e fatores de custo mudam ao longo do desenvolvimento do sistema, isto é, eles são dinâmicos (não estáticos) ao longo do tempo. Esta é uma diferença significativa em relação às outras técnicas, que tendem a confiar em modelos estáticos e previsões com base em “quadros” de uma situação de desenvolvimento em um determinado momento no tempo. No entanto, fatores como prazos, níveis de pessoal, requisitos de design, necessidades de treinamento, orçamento, dentre outros, flutuam ao longo do desenvolvimento e causam flutuações correspondentes na produtividade da equipe do projeto. Isto, por sua vez, tem consequências no prazo e or-

çamento do projeto. As técnicas dinâmicas mais importantes são baseadas na abordagem dinâmica de sistemas cujo modelo foi proposto por Jay Forrester em 1961 [FOR61].

### **Baseadas em Expertise**

Técnicas baseadas em expertise são úteis na ausência de dados empíricos e quantificados. Elas capturam o conhecimento e a experiência de profissionais dentro de um domínio de interesse, fornecendo estimativas com base em uma síntese dos resultados conhecidos de todos os últimos projetos em que o especialista esteja a par ou em que ele ou ela participou. A desvantagem óbvia deste método é que uma estimativa é tão boa quanto o parecer do perito, e não há nenhuma maneira geral para testar essa opinião até que seja tarde demais para corrigir o dano se que a opinião se provar errada. Anos de experiência não necessariamente se traduzem em elevados níveis de competência. Além disso, mesmo a mais competente de pessoas, às vezes, simplesmente estima errado. Duas técnicas foram desenvolvidas para capturar o parecer dos peritos, além de tomar medidas para mitigar a possibilidade de que o julgamento de um especialista seja perdido, estas técnicas são *Delphi* e *Planning Poker*.

#### **Delphi**

A técnica baseada em expertise *Delphi* foi desenvolvido pela *Rand Corporation*, no final da década de 40, como forma de realizar previsões de eventos futuros. Com o passar do tempo, tal abordagem passou a ser utilizada como forma de guiar um grupo determinado de indivíduos a um consenso sobre determinado assunto [ABT00].

O *Delphi* mostra-se útil quando há necessidade de estimar valores sem o apoio de grandes volumes de dados empíricos [ABT00]. Embora limitada, a técnica pode servir de apoio a outras técnicas, como foi o caso de sua utilização na calibragem Bayesiana dos dados do COCOMO 2 e na especificação de informações prioritárias e necessárias à calibragem.

#### **Planning Poker**

Outra técnica baseada em expertise é o *Planning Poker*, largamente utilizado e divulgado pela metodologia ágil *Scrum* [SB02]. O termo *Planning Poker* foi criado por James Grenning e popularizado por Mike Cohn. O *Scrum* prega que a atividade de estimar deve ser realizada por todos os membros envolvidos no processo, durante uma reunião de planejamento, que define um conjunto de tarefas prioritárias a serem realizadas em um período de duas a quatro semanas após a reunião, este período de realização das tarefas é denominado *Sprint*.

A granularidade utilizada na realização das estimativas para cada tarefa no *Planning Poker*, ao contrário do *Delphi*, baseia-se em uma série numérica que visa limitar o número de escolhas possíveis dos participantes buscando agilidade e diminuição no tempo gasto durante as estimativas. As estimativas se baseiam na série numérica de *Fibonacci*.

Assim como no *Delphi*, o *Planning Poker* onera as equipes de desenvolvimento, pois estas necessitam deslocar seus membros para reuniões onde são estimados os valores empíricos das demandas [TEN10]. Realizar as estimativas de forma empírica pode trazer, dentre outros, dois visíveis problemas: i) falta de precisão nas estimativas de prazos, onde estes quando mal mensurados podem causar aumento da pressão na equipe e, conseqüentemente, desgaste de seus membros por trabalhar em demasia para cumpri-lo; e ii) dependência do conhecimento humano, o que se torna prejudicial para as estimativas de esforço, se a rotatividade de pessoal for alta.

### **Baseadas em Regressão**

Modelos de regressão estatística estimam o esforço de desenvolvimento de software como a variável dependente. Tamanho de software (em métricas como linhas de código ou pontos de função) é usado como uma variável independente. Em alguns modelos, outros parâmetros, como a linguagem de programação de desenvolvimento ou sistema operacional pode ser usado como variáveis independentes adicionais para um modelo de regressão múltipla. Os modelos de regressão tem a vantagem de possuir uma base matemática sólida, bem como medidas de qualidade de ajuste (*goodness fit*), ou seja, quão bem a curva corresponde ao conjunto de dados especificado. Porém, modelos de regressão estatística são muito suscetíveis ao efeito de *outliers* (itens de dados que podem estar completamente fora de sintonia com o resto do conjunto de dados). Além disso, um modelo de regressão também precisa de um conjunto de dados relativamente grande que pode ser um problema no campo de estimativa de software [FIN97].

### **Orientadas a Aprendizagem de Máquina**

As técnicas para as estimativas de software baseadas em Aprendizado de Máquina (ML—*Machine Learning*) representam uma alternativa às técnicas tradicionais, as quais são baseadas em regressão ou na expertise humana [TEN10].

O método para a criação de modelos de aprendizagem de máquina é a exploração dos dados históricos do domínio, feita por algoritmos específicos, na tentativa de formular ou inferir um conjunto de regras que permitam a dedução de valores futuros [SRI95].

Dentre as abordagens de aprendizagem de máquina para o desenvolvimento de modelos preditivos de software destacam-se: Redes Neurais Artificiais; Sistemas Fuzzy, Raciocínio Baseado em Casos; Árvores de Classificação e Regressão e Algoritmos Genéticos.

#### **Redes Neurais Artificiais**

De acordo com Gray e McDonell [GRA97] redes neurais artificiais é a técnica de construção de modelos de estimativa mais comumente utilizada como uma alternativa à

regressão de mínimos quadrados. As redes neurais artificiais (RNAs) adotam uma abordagem de aprendizagem derivada de um modelo preditivo. A rede é projetada para um conjunto específico de entrada, por exemplo, pontos por função, linguagem de programação, etc, bem como a saída (s), por exemplo, o esforço de desenvolvimento. A rede recebe um conjunto de processos conhecidos (o conjunto de treino) que é utilizado para "treinar" a rede, ou seja, determinar os pesos associados a cada entrada na rede. Uma vez que a rede está treinada e estável, o esforço de desenvolvimento de um novo caso pode ser previsto, substituindo os valores de entrada relevantes para o caso específico. RNAs são reconhecidas por sua capacidade de fornecer bons resultados quando se trata de problemas onde existem relações complexas entre entradas e saídas, e onde os dados de entrada são distorcidos por altos níveis de ruído [TRE91].

Apesar da robustez da técnica, algumas desvantagens são observadas como a falta de imunidade a problemas comuns em técnicas estatísticas, como existência de *outliers*, valores incompletos ou perdidos. Além disso, tem-se o fato de a rede apresentar um funcionamento dito "caixa-preta", ou seja, não detalham as informações de processamento até chegar ao resultado final, o que pode ser crítico no momento de se realizar uma análise causal, por exemplo [TEN10].

### **Sistemas Fuzzy**

Um sistema fuzzy é um mapeamento de valores em termos linguísticos, por exemplo, "muito baixo", "baixo", "alto" e "muito alto" para um conjunto de valores de variáveis correspondentes. Tanto a entrada quanto a saída do sistema fuzzy pode ser numérica ou linguística [GM97].

A principal vantagem da utilização de sistemas fuzzy para estimativas de software é a sua fácil compreensão, devido ao uso de termos linguísticos, fazendo com que o mesmo possa ser analisado e criticado por pessoas sem conhecimento ou treinamento prévio. Como desvantagem, tem-se a dificuldade de especificação de um sistema que permita uma alta precisão de resultados mantendo uma interface interpretável, onde geralmente, sistemas mais complexos precisam de mais regras, levando a um aumento de complexidade e decréscimo de poder de interpretação [GRA97] [KRI94].

### **Raciocínio Baseado em Casos (Estimativa por Analogia)**

O raciocínio baseado em casos (RBC), também conhecido como Estimativa por Analogia (no contexto de estimativa de esforço), [WAT94] é uma técnica de resolução de problemas que resolve novos problemas adaptando soluções que foram usadas para resolver problemas antigos. RBC recupera um ou mais casos semelhantes ao problema atual e tenta modificar estes casos para ajustar aos parâmetros do problema atual. Na estimativa de esforço de desenvolvimento de software, cada caso pode ser um desenvolvimento de software anterior, enquanto o problema atual é extrair uma estimativa adequada para o projeto atual. Como resultado, o desenvolvimento deve ser mais rápido e a estimativa de tempo ajustada em conformidade.

A vantagem da utilização dessa abordagem é que pode-se justificar decisões com base em casos anteriores utilizados na resolução de um problema. Além disso, a abordagem RBC é intuitivamente semelhante ao julgamento de especialistas adotado em muitas organizações que dependem de uso de desenvolvedores experientes para estimar esforço do projeto. Estes indivíduos estimam adaptando o julgamento de desenvolvimentos de sistemas anteriores.

O problema dessa técnica é que os projetos têm que ter a mesma característica para que as estimativas tenham uma boa proximidade do real, o que nem sempre é possível, principalmente no contexto de reengenharia de sistemas e de manutenção.

### **Árvores de Classificação e Regressão**

Árvores de classificação e de regressão são conceitos similares, mas diferem na representação do atributo que se deseja prever. Ambas se utilizam de um conjunto de dados previamente conhecidos, induzindo as regras necessárias para classificarem esses dados. O método mais comumente utilizado para construção da árvore é o *top-down*. Essa estratégia consiste da análise de qual atributo dos registros de dados melhor divide o conjunto de dados em subpopulações disjuntas. Dentre as abordagens utilizadas para a realização dessa divisão estão: o cálculo do erro médio quadrado, o cálculo de entropia, dentre outros [SRI95] [GRA97].

As árvores de regressão são mais utilizadas em métricas de software do que as de decisão. Isso é dado pela própria necessidade que o domínio impõe de obtenção de resultados numéricos. Além de serem utilizadas para estimar esforço, as árvores de regressão também são utilizadas para estimar defeitos [JON00] [GRA97].

A vantagem em se utilizar as árvores de regressão é que elas são simples de serem aplicadas sobre um conjunto de dados, já que existem várias ferramentas que as implementam, como o WEKA por exemplo [UW13]. Como desvantagens tem-se a dificuldade de compreensão, pois quanto mais níveis e nodos a árvore tiver, menos compreensível ela se torna. Além disso, por depender de dados do projeto, a árvore é sensível à *outliers*, isso quer dizer que a qualidade dos dados é importante para que se obtenham bons resultados com essa técnica.

### **Algoritmos Genéticos**

Algoritmo Genético (GA – *Genetic Algorithm*) é um algoritmo usado para pesquisar a melhor solução de forma aleatória. Ele imita os genes biológicos. Após a avaliação, ele terá a melhor condição. GA mostram os genes por estado de cromossomo. Após a competição, os mais aptos são as melhores soluções. Os cromossomos são gerados por números aleatórios. Seu valor de *fitness* é dado pela função alvo. De acordo com o valor de *fitness*, o cromossomo será cruzado (*crossover*), sofrerá mutação, e será selecionado. O *crossover* vai escolher aleatoriamente dois cromossomos, chamados de pais, e trocar genes entre si. Em seguida, ele irá gerar dois novos cromossomos, chamados filhos. Mutação significa

escolher um cromossomo aleatoriamente, e torná-lo um pouco diferente. Por cruzamento e mutação, são produzidos mais cromossomos, e pode-se obter o valor de aptidão de cada cromossomo. Então é aplicado um método de seleção. Os cromossomos selecionados serão usados na próxima avaliação.

Imitando seleção e reprodução biológica, GA pode eficientemente pesquisar através do espaço de soluções de problemas complexos e oferece oportunidade para escapar do ótimo local. Com isso, GA tornou-se um dos algoritmos mais populares para os problemas de otimização [LIN11b].

## **Compostas**

Por fim, [ABT00] afirma que nenhuma técnica em especial é adequada para todas as situações, portanto, na maioria das vezes faz-se uso de uma combinação de técnicas para se obter melhor acurácia nos resultados. Com base nisso, diversas soluções emergiram já propondo composição de técnicas de maneira a tornar a solução proposta mais generalizável aos diversos contextos de desenvolvimento. Assim, técnicas compostas nada mais são do que modelos, metodologias, métodos, dentre outros que combinam duas ou mais técnicas para realização de estimativas de esforço.

## REVISÃO SISTEMÁTICA DA LITERATURA

A revisão sistemática é caracterizada como uma maneira de identificar, avaliar e interpretar toda a pesquisa relevante disponível sobre uma questão de pesquisa em particular [KIT04]. É um estudo secundário que permite, dentre outras coisas, a sumarização das evidências existentes sobre uma determinada tecnologia, a identificação de *gaps* em uma área de pesquisa, que permitam a sugestão de novos temas a serem investigados nessa área, além de prover suporte para o posicionamento de novas atividades de pesquisa.

Para [KIT04] uma revisão sistemática avalia e interpreta toda literatura relevante disponível a respeito de uma questão de pesquisa, tópico de alguma área ou ainda, algum fenômeno de interesse. Tanto a avaliação quanto a interpretação devem ser realizadas a partir de uma fidedigna e rigorosa metodologia, que permita posteriormente alguma auditoria ou continuidade das pesquisas realizadas. Em um local, alguém ou um grupo pode ter feito pesquisas iguais ou semelhantes ou, ainda, complementares para a pesquisa que se pretende fazer. Dessa forma a procura por tais fontes bibliográficas é essencial para que não haja repetição de esforços e evitar a descoberta de ideias já concebidas.

Para dar rigor a metodologia, deve ser elaborado um protocolo que guiará a revisão sistemática e será um dos grandes benefícios gerados por esse método de pesquisa, já que permitirá o reuso dos resultados obtidos. Dessa forma outros pesquisadores interessados pelo mesmo tema podem manuseá-lo. Assim podem, por exemplo, julgar o quão adequado está o protocolo, ou ainda expandir a revisão sistemática sobre o tema.

### A.2.2 TRABALHOS RELACIONADOS

Há muito tempo esse método de pesquisa é utilizado na área da saúde. Na área de Computação e Informática tem sido muito frequente em congressos e periódicos da área de Engenharia de Software [BIO05]. No que diz respeito a revisões sistemáticas acerca de Estimativas de Esforço em Desenvolvimento de Software, foram identificados os trabalhos de [JØR07], [KIT07b] e [TEN10].

Jørgensen [JØR07], identificou 304 trabalhos publicados apenas em periódicos, até o ano de 2004, e tinha como objetivo principal orientar e apoiar pesquisas sobre estimativa de custo de modo geral. Com isso, foram identificados, dentro outras coisas, quais os principais periódicos que publicam sobre o tema, os principais pesquisadores, os métodos de estimativa de esforço mais utilizado e onde são mais frequentemente aplicadas as pesquisas sobre o tema.

Kitchenham et al.[KIT07b] tinham como objetivo determinar em que circunstâncias uma companhia individual poderia adotar modelos de estimativa de esforço de *cross-company*. Para isso foram analisados 10 trabalhos que comparavam as duas abordagens,

*cross-company* e *within-company*, sendo os resultados obtidos inconclusivos para as questões de pesquisa definida.

Por fim [TEN10] visava identificar as iniciativas com relação às estimativas de esforço em projetos de manutenção de software com intuito de estabelecer quais técnicas são mais amplamente utilizadas neste domínio. Neste trabalho, foram identificados 14 trabalhos, com o tempo limitado entre 2003-2009, sendo apenas um desses trabalhos classificado como solução voltada para projeto de manutenção de software, e os demais, como trabalhos voltados para desenvolvimento de sistema novo. Com isso, o autor conclui a escassez de trabalhos sobre o tema.

### A.2.3 PROTOCOLO

O planejamento desta revisão sistemática está baseado nos protocolos apresentados em [KIT07a] e [BIO05].

#### **Formalização da Questão**

- Foco da Questão Essa revisão sistemática tem como objetivo identificar o estado da arte nas pesquisas acerca da realização de estimativas de esforço em projetos de desenvolvimento de software, identificando as soluções que tem sido propostas para a realização desta atividade no decorrer dos anos e as técnicas aplicadas nessas soluções.
- Qualidade e Amplitude da Questão
  1. Problema

A literatura apresenta uma gama de propostas de soluções para estimativas de esforço. O desafio dos pesquisadores é aperfeiçoar tais soluções para que tenham resultados cada vez mais precisos. Assim, torna-se necessário verificar o estado da arte nessa área de pesquisa, explicitando as boas práticas já definidas e as lacunas ainda existentes.
  2. Questões de interesse
    - 1) Quais são as soluções existentes voltadas para auxílio a realização de estimativa de esforço em projetos de desenvolvimento de software?
    - 2) Quais são os tipos de técnicas de estimativa de esforço utilizadas nas soluções existentes e como estas técnicas evoluíram no decorrer dos anos?
    - 3) Quais são os domínios de desenvolvimento de software para quais as soluções são voltadas?
    - 4) Como é realizada a validação das soluções existentes?

### 3. Palavras-chaves

Estimativa de Esforço: Effort Estimation, Estimating Effort, Effort Prediction, Predicting Effort, Cost Estimation, Cost Prediction. Desenvolvimento de Software: Software Development, Software, Software Project, System Development.

### 4. Intervenção

Busca-se identificar e analisar trabalhos que propõem soluções para realização de estimativa de esforço.

### 5. Controle

Seleção de conferências e periódicos, onde será realizada uma busca manual por artigos referentes ao tema de pesquisa.

### 6. Efeito

Consolidação do conhecimento e identificação do estado da arte na pesquisa em soluções para estimativa de esforço em desenvolvimento de software.

### 7. Medida do Resultado

Oportunidades de pesquisas geradas, a partir da publicação de textos (monografias e artigos) relacionados a esta revisão sistemática em veículos de comunicação importantes.

### 8. População de interesse

Projetos de desenvolvimento de software.

### 9. Aplicação

Este trabalho aplica-se a pesquisadores que estejam desenvolvendo pesquisas científicas e necessitem de informações inerentes. Também pode ser utilizado por profissionais de mercado, que possuem interesse em informações importantes a respeito do assunto.

### 10. Desenho do Experimento

Nenhuma meta-análise será realizada.

## **Seleção de Bases de Dados**

### **a. Definição dos Critérios de Seleção das Fontes de Dados**

No que diz respeito aos motores de busca, prezou-se pela disponibilidade de consulta a artigos completos (full papers) através da Web pelo convênio PUCRS–CAPES e existência de mecanismos de busca com suporte a inserção de operadores booleanos (como AND e OR). Além disso, foram selecionadas conferências relacionadas à área de estudo a ser investigada, a partir da lista de classificação de Qualis da CAPES para Ciência da Computação (Triênio 2010-2012) [CAP12] e os 10 periódicos mais importantes sobre o tema de Estimativa de Esforço, identificados por [JØR07]. Tais locais de publicação foram

selecionados para a aplicação de uma busca manual, que viria a complementar a busca automatizada.

## **b. Idiomas das Fontes de Dados**

Inglês

## **c. Identificação de Fontes**

### **i. Métodos de procura**

A pesquisa foi realizada através dos motores de busca das seguintes bibliotecas online: IEEEExplore, ACM Digital Library, CiteseerX library, Springer-Link, ScienceDirect e Scopus. Essas bibliotecas foram escolhidas por serem referências importantes em pesquisas na área de Ciência da Computação

### **ii. String de Busca**

Strings de busca – String PICO

1. População: projetos de desenvolvimento de software P := (Software Development <or> Software Project <or> System Development)
2. Intervenção: estimativa de esforço I := (effort estimation <or> effort estimating <or> >effort prediction <or> effort predicting <or> cost estimation <or> cost prediction)
3. O := (Technique <or> Method <or> Model <or>Strategy <or> Approach <or> Methodology <or> Framework <or> Tool) String de busca final: P <and> I <and> O

### **iii. Lista de Fontes de Dados**

- Bases de Artigos: IEEEExplore, ACM Digital Library, CiteseerX library, Springer-Link, ScienceDirect, Web of Science, Scielo e Scopus.
- Anais de Conferências:
- Periódicos

Os periódicos selecionados para a busca manual foram obtidos no trabalho de [JØR07], que identificou, dentre outras coisas, todos os periódicos onde houve publicações sobre Estimativa de Esforço em Desenvolvimento de Software, até o ano de 2004. Os 10 periódicos destacados pelo autor e selecionados para este trabalho foram responsáveis por 66% destas publicações.

## **d. Seleção de Fontes de Dados após Avaliação**

Todas as fontes acima obedecem ao critério estabelecido em 3.1.2.a.

## **e. Artigo de Controle**

Tabela A.1 – Anais de Conferências Selecionados a Partir da Listagem de Qualis da CAPES

<b>Sigla</b>	<b>Conferência</b>	<b>Qualis</b>
ICSE	ACM/IEEE International Conference on Software Engineering	A1
BPM	International Conference in Business Process Management	A2
CSMR	European Conference on Software Maintenance and Reengineering	A2
ICSM	IEEE International Conference on Software Maintenance	A2
WCRE	IEEE Working Conference on Reverse Engineering	A2
ECIS	European Conference on Information Systems	B1
QEST	International Conference on Quantitative Evaluation of Systems	B1
ICEIS	International Conference on Enterprise Information Systems	B1
EuroSPI	European Conference on Software Process Improvement	B2
MLDM	IAPR International Conference on Machine Learning and Data Mining	B2
BPMDS	International Conference on Business Process Modeling, Development, and Support	B3

Tabela A.2 – Periódicos Selecionados a Partir do Trabalho de [JØR07]

<b>ISSN</b>	<b>Periódico</b>	<b>Qualis</b>
0098-5589	IEEE Transactions on Software Engineering	A1
0378-7206	Information and Management	A1
0740-7459	IEEE Software	A1
0001-0782	Communications of the ACM	A1
0950-5849	Information and Software Technology	A2
1382-3256	Empirical Software Engineering	A2
0164-1212	Journal of Systems and Software	A2
1532-0618	Journal of Software Maintenance and Evolution	B2
1099-1670	American Programmer	B2
0963-9314	Software Quality Journal	B2

Com o intuito de dar uma maior garantia aos resultados gerados pela string de busca, [SHE95] será considerado o artigo de controle, ou seja, deverá aparecer nos resultados provenientes de todas as bases. Caso ele não apareça, a string de procura deve ser alterada. Este artigo foi escolhido por ser amplamente referenciado como exemplo de solução de estimativa de esforço e cuja técnica apresentada foi estendida, comparada e debatida em diversas publicações no decorrer dos anos.

## Seleção de Estudos

### Definição de Estudos

#### 1. Critérios para inclusão/exclusão dos estudos

##### Critérios de inclusão dos estudos:

- a. Artigos que apresentem soluções de estimativa de esforço para projetos de desenvolvimento de software em geral;
- b. Artigos que contenham todos os aspectos envolvidos para estimar o esforço em desenvolvimento de software em geral e/ou em reengenharia de sistemas legados, desde a coleta de métricas do projeto, passando pela escolha da técnica e aplicação do modelo preditivo. As soluções propostas nos artigos podem incluir técnicas, métodos, modelos, estratégias ou qualquer outra iniciativa relacionada à estimativas de esforço.

##### Critérios de exclusão dos estudos:

- c. Trabalho que não estejam no formato de artigo completo (*full paper*);
  - d. Trabalhos relacionados a estimativa de esforço, mas que não proponham soluções para realização de estimativas de esforço em projeto de software;
  - e. Trabalhos cujo conteúdo esteja relacionado às estimativas de defeitos;
  - f. Trabalhos que indicam os desafios e as direções a serem tomadas na área de pesquisa em questão (*roadmaps*).
2. Definição dos tipos de estudos

Serão selecionados estudos do tipo *Solution Proposal*, que segundo [WIE06] são trabalhos onde são propostas soluções para um determinado problema. No caso, o problema seria a realização da estimativa de esforço em projetos de desenvolvimento de software. Tal solução pode ser nova ou uma extensão de uma solução existente, e os benefícios e a aplicabilidade da solução devem ser demonstrados com um exemplo de aplicação ou uma boa linha de argumentação.

#### 3. Procedimento para seleção dos estudos

**Bases de Dados:** submissão da *string* de busca nos mecanismos de busca oferecidos em cada uma das fontes selecionadas; leitura dos metadados de cada artigo; leitura de título e o resumo (*abstract*) de cada artigo aprovado na primeira triagem; leitura integral dos artigos remanescentes.

**Conferências e Periódicos:** busca dos anais de todas as edições da conferência e busca de todas as edições do periódico; leitura dos metadados de cada artigo; leitura de título e o resumo (*abstract*) de cada artigo aprovado na primeira triagem; leitura integral dos artigos remanescentes.

A seleção de estudos será dividida em duas fases. A primeira fase será guiada pelo formulário descrito na Seção A.8, para seleção dos artigos. Dessa forma, na fase I, para a seleção dos estudos de interesse, o pesquisador lerá o título e a seção *abstract* de cada um deles, e preencherá os critérios de seleção/qualidade definidos na seção A.8. Ao lado de cada artigo foram inseridas as respostas para os critérios de inclusão e exclusão, definidos na seção A.3.1.3 – a – i. Somente aqueles artigos que responderem “Sim” para o critério a serão selecionados a participar da próxima fase.

O refinamento da revisão sistemática seguirá sendo realizado a partir da leitura completa dos artigos (fase II), selecionados na primeira fase. Os artigos que não obedecerem o critério b também serão excluídos da revisão sistemática. Para cada artigo selecionado serão respondidos os critérios de avaliação de qualidade do estudo descritos na Seção A.9.

Com o objetivo de facilitar a operacionalização do processo de revisão sistemática foi utilizada a ferramenta *StArt (State of the Art through Systematic Review)*, que tem como objetivo dar suporte a realização da revisão sistemática [ZAM10]. A versão atual da ferramenta *StArt* apóia as três etapas do processo de revisão, sendo que no Planejamento, o pesquisador preenche o protocolo. Na Execução, o pesquisador adiciona e avalia os artigos que compõem a revisão sistemática e extrai informações daqueles que são relevantes ao tópico de pesquisa abordado. Na Sumarização, são apresentados gráficos e tabelas que dão uma visão geral sobre a revisão sistemática e auxiliam a descrever o estado da arte do tema pesquisado.

#### 4. Critérios de Qualidade das Fases da Revisão Sistemática

Os critérios de qualidade que serão avaliados nas duas fases estão listados abaixo:

##### **Fase I**

- Apresenta uma solução para realização de estimativa de esforço para projetos de desenvolvimento de software?

##### **Fase II**

- Método de Pesquisa: apresenta como a pesquisa foi conduzida. Os possíveis valores de coluna (Comparativo / Estudo de Caso / Protótipo/ Formalização Matemática/ Literatura), conotam respectivamente: a comparação de diferentes propostas, a realização de um estudo de caso em um ambiente real, proposta de um protótipo, proposta de um modelo matemático, pesquisa na literatura para identificação de possibilidade de pesquisa?
- Foco de Contribuição: informa o contexto da contribuição (Científica / Mercado)?

- Classificação da técnica: classifica a solução quanto à técnica de estimativa de esforço empregada (Baseadas em Modelo / Dinâmicas / Baseadas em Expertise / Baseadas em Regressão/ OAM / Compostos/Outra)?
- Classificação do tipo de solução: classifica a solução proposta (técnica, metodologia, método, ferramenta, modelo, abordagem, *framework*, outra)?
- Domínio: informa se a proposta de estimativas de esforço se insere no contexto de desenvolvimento de software novo, manutenção ou reengenharia de software (Desenvolvimento Novo / Modernização/ Manutenção);
- Representação: informa o domínio do valor da estimativa de esforço obtido pela proposta, onde numérica é um valor que representa o esforço de trabalho em horas para uma pessoa, e categórico é um valor textual que representa um intervalo de valores (Numérica / Categórica)?
- Base Histórica: informa se o trabalho se baseia em dados anteriores de projetos (Sim / Não)?
- Calibragem: informa se a proposta permite a calibragem da técnica (Sim / Não);?
- Possui Avaliação: avalia o desempenho da solução em relação a soluções anteriormente propostas (Sim / Não)?
- Parâmetros Utilizados: informa quais foram os parâmetros utilizados para o cálculo da estimativa de esforço?

#### A.2.4 AVALIAÇÃO DO PLANEJAMENTO

Esta fase do processo de planejamento da revisão sistemática é essencial, a fim de garantir a consistência do mesmo e capturar percepções sobre a sua viabilidade.

O protocolo dessa revisão sistemática foi apresentado impresso ao coordenador do grupo de pesquisas GPIN/PUCRS (Grupo de Pesquisa em Inteligência de Negócio), para uma primeira discussão e validação. Posteriormente, o protocolo foi discutido com um docente da área de Engenharia de Software, que tem grande experiência em trabalhos sobre Revisões Sistemáticas, capturando também suas percepções acerca do protocolo.

#### A.2.5 CRONOGRAMA

A Tabela A.3, ilustrada abaixo, exhibe o cronograma geral da revisão sistemática.

Tabela A.3 – Cronograma da Revisão Sistemática

<b>Atividades</b>	<b>08/13</b>	<b>09/13</b>	<b>10/13</b>	<b>11/13</b>	<b>12/13</b>	<b>01/14</b>	<b>02/14</b>	<b>03/14</b>
Planejamento e definição do protocolo	X	X	-	-	-	-	-	-
Execução da Fase I	-	X	X	X	-	-	-	-
Execução da Fase II	-	-	-	-	X	X	-	-
Escrita dos Resultados	-	-	-	-	-	X	X	X

### **A.3 EXECUÇÃO DA FASE I**

#### **A.3.1 BUSCA POR STRING**

A string de busca, definida no protocolo, foi executada sobre as seis bases de dados escolhidas nessa revisão sistemática, quais sejam, ACM Digital Library, CiteSeerX, IEEEExplore, Springer, Scopus e Science Direct. Os seis motores de busca dessas bases de dados permitiram que a string fosse colocada conforme a definição do protocolo, a partir do uso da seção de "pesquisa avançada" de cada uma delas:

- ACM Digital Library, seção Advanced Search;
- CiteSeerX, seção Advanced Search;
- IEEEExplore, seção Command Search;
- Scopus, seção Advanced Search;
- Springer Link, seção Advanced Search;
- Science Direct, seção Expert Search.

A seguir são apresentadas as strings de busca aplicadas em 10/08/2013.

- ACM:  
Digital Library (Abstract: "Software Development" or Abstract:"Software Project" or Abstract:"System Development") and (Title:"effort estimation" or Title:"effort estimating" or Title:"effort predicting" or Title:"effort prediction" or Title:"cost estimation" or Title:"cost prediction") and (Title:"Technique" or Title:"Method" or Title:"Model" or Title:"Strategies" or Title:"Approach" or Title:"Methodology" or Title:"Framework" or Title:"Tool")

- CiteSeerX:

title:(“Software Development” OR “Software Project” OR “System Development” OR “Software Reengineering” OR “Technique” OR “Method” OR “Model” OR “Strategies” OR “Approach” OR “Methodology” OR “Framework” OR “Tool”) AND abstract:(“Software Development” OR “Software Project” OR “System Development” OR “Technique” OR “Method” OR “Model” OR “Strategies” OR “Approach” OR “Methodology” OR “Framework” OR “Tool”) AND keyword:(“effort estimation” OR “effort predicting” OR “cost estimation” OR “cost prediction”)

- IEEExplore:

((“Abstract”：“Software Development” OR “Abstract”：“Software Project” OR “Abstract”：“System Development”) AND (“Document Title”：“effort estimation” OR “Document Title”：“effort predicting” OR “Document Title”：“cost estimation” OR “Document Title”：“cost prediction”) AND (“Abstract”：“Technique” OR “Abstract”：“Method” OR “Abstract”：“Model” OR “Abstract”：“Strategies” OR “Abstract”：“Approach” OR “Abstract”：“Methodology” OR “Abstract”：“Framework” OR “Abstract”：“Tool”))

- Scopus:

(TITLE-ABS-KEY(“Software Development”) OR TITLE-ABS-KEY(“Software Project”) OR TITLE-ABS-KEY(“System Development”)) AND (TITLE-ABS-KEY(“effort estimation”) OR TITLE-ABS-KEY(“effort estimating”) OR TITLE-ABS-KEY(“effort prediction”) OR TITLE-ABS-KEY(“effort predicting”) OR TITLE-ABS-KEY(“cost estimation”) OR TITLE-ABS-KEY(“cost prediction”)) AND (TITLE-ABS-KEY(“Technique”) OR TITLE-ABS-KEY(“Method”) OR TITLE-ABS-KEY(“Model”) OR TITLE-ABS-KEY(“Strategies”) OR TITLE-ABS-KEY(“Approach”) OR TITLE-ABS-KEY(“Methodology”) OR TITLE-ABS-KEY(“Framework”) OR TITLE-ABS-KEY(“Tool”))

- Springer:

(“Software Development” OR “Software Project” OR “System Development”) AND (“effort estimation” OR “effort predicting” OR “cost estimation” OR “cost prediction”) AND (“Technique” OR “Method” OR “Model” OR “Strategies” OR “Approach” OR “Methodology” OR “Framework” OR “Tool”)

- Science Direct:

Science Direct (“software development” OR “software project” OR “system development”) AND (“effort estimation” OR “effort prediction” OR “cost prediction” OR “cost estimation”) AND (“technique” OR “method” OR “model” OR “strategies” OR “approach” OR “methodology” OR “framework”)[All Sources(Computer Science)]

Após executar a *string* em cada um dos locais acima mencionados, a quantidade de artigos retornada por cada motor de busca apresentou disparidade, conforme demonstrado na Tabela A.4.

Tabela A.4 – Número de artigos originalmente extraídos dos motores de busca

<b>BASES</b>	<b>Scopus</b>	<b>IEEEExplore</b>	<b>ACM Digital Library</b>	<b>CiteSeerX</b>	<b>Springer</b>	<b>Science Direct</b>
<b>Nº DE ARTIGOS</b>	980	187	71	113	96	109

### A.3.2 BUSCA MANUAL

Conforme especificado na seção de Seleção de Bases de Dados, além da busca através da *string* foi realizada uma busca manual em anais de conferência e periódicos selecionados a partir do Qualis-Capes para o triênio de 2010-2012 [CAP12] e do trabalho de [JØR07], respectivamente. Tal medida, prevista por Kitchenham [KIT07a] e aplicada em Kitchenham et al. [KIT07b] é comumente utilizada quando o pesquisador já tem conhecimento prévio de fontes potenciais para a pesquisa. No caso desta pesquisa, a busca manual foi aplicada com o intuito de garantir que os principais veículos de Engenharia de Software, que tratam do tema de interesse, seriam cobertos pela revisão. A seção “Lista de Fonte de Dados”, indica as conferências e periódicos selecionados, cuja lista foi revisada pelo orientador desta pesquisa.

Para a realização da busca manual foi realizado o levantamento dos anais das conferências e de todos os volumes dos periódicos. Então, foram aplicados os critérios de exclusão e os critérios de inclusão de estudo, no título das publicações. A Tabela A.5 apresenta o número de artigos levantados em cada conferência e periódico.

### A.3.3 ARTIGO DE CONTROLE

As strings de busca que foram aplicadas nos motores de busca da ACM Digital Library, IEEE Xplore e Scopus, selecionaram o artigo de controle dos autores Shepperd, Schofield *et al.* [SHE95]. No entanto, os motores de busca CiteSeerX, Springer e Science Direct não o retornaram. A justificativa disso é que o artigo em questão, embora importantíssimo para a área, não encontra-se nestas três bases de dados. Para comprovar esta afirmação, foram feitas buscas sobre ele a partir do seu título e também do nome do autor. O artigo foi retornado na busca manual nos anais da *International Conference on Software Engineering*.

Tabela A.5 – Número de Artigos Encontrados Manualmente

<b>Tipo de Publicação</b>	<b>Nome da Publicação</b>	<b>Número de Artigos</b>
Periódico	Empirical Software Engineering	7
Periódico	Software Quality Journal	2
Periódico	IEEE Transactions on Software Engineering	8
Periódico	Information and Software Technology	4
Periódico	Journal of Systems and Software	5
Anais de Conferência	International Conference on Software Engineering	2
Anais de Conferência	IEEE International Conference on Software Maintenance	3
Anais de Conferência	IEEE Working Conference on Reverse Engineering	1

#### A.3.4 ARTIGOS RELACIONADOS

Portanto, a fase I de análise dessa revisão sistemática foi realizada sobre um total de 1588 artigos. Para cada execução da *string* em um motor de busca, foi gerado um arquivo contendo o BibTex dos resultados obtidos. Tal arquivo era então submetido à ferramenta StArt, onde foi previamente indicado os motores de busca que seriam utilizados. A Figura A.1 apresenta a interface de *upload* dos artigos na ferramenta StArt. Além do *upload* dos arquivos BibTex, resultante da busca nos motores de busca, também foi realizado o cadastro dos artigos encontrados manualmente.

Os dados obtidos de cada artigo variaram de acordo com a qualidade do BibTex fornecido pelo motor de busca. Porém, todos tinham as informações básicas de Nome do Autor, Título, Palavras-Chave, Local de Publicação, Ano e Tipo. O resumo não foi fornecido diretamente por todas as bases, caso da Springer e da ACM digital library. Nesse caso, o resumo foi obtido manualmente e cadastrado na ferramenta. A Figura A.2 mostra a interface de apresentação dos dados dos artigos.

Com base nessas informações básicas foi realizada a avaliação dos artigos na Fase I, aplicando-se os critérios de exclusão e os critérios de inclusão de estudo, no título e *abstract* das publicações. A Figura A.3 apresenta a aplicação dos critérios na fase I.

Dessa forma, foi possível, de maneira mais eficiente, definir quais artigos seriam selecionados para a segunda fase da revisão sistemática, além de permitir que algumas estatísticas sobre os artigos fossem geradas inicialmente. Também foi realizado o tratamento de duplicatas de artigos, ou seja, artigos que retornaram em mais de um motor de busca e na busca manual. Nessa atividade, percebeu-se que todos os artigos obtidos na busca ma-

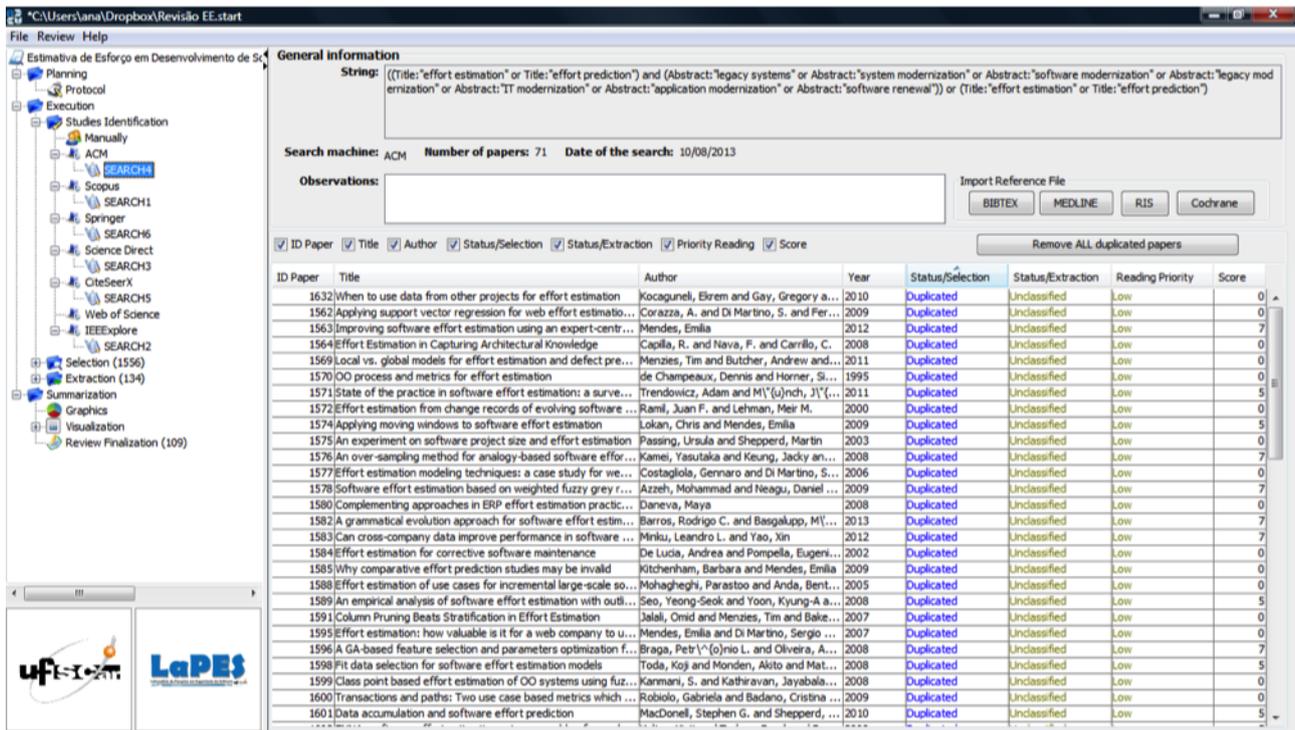


Figura A.1 – Interface de Upload de Arquivos StArt

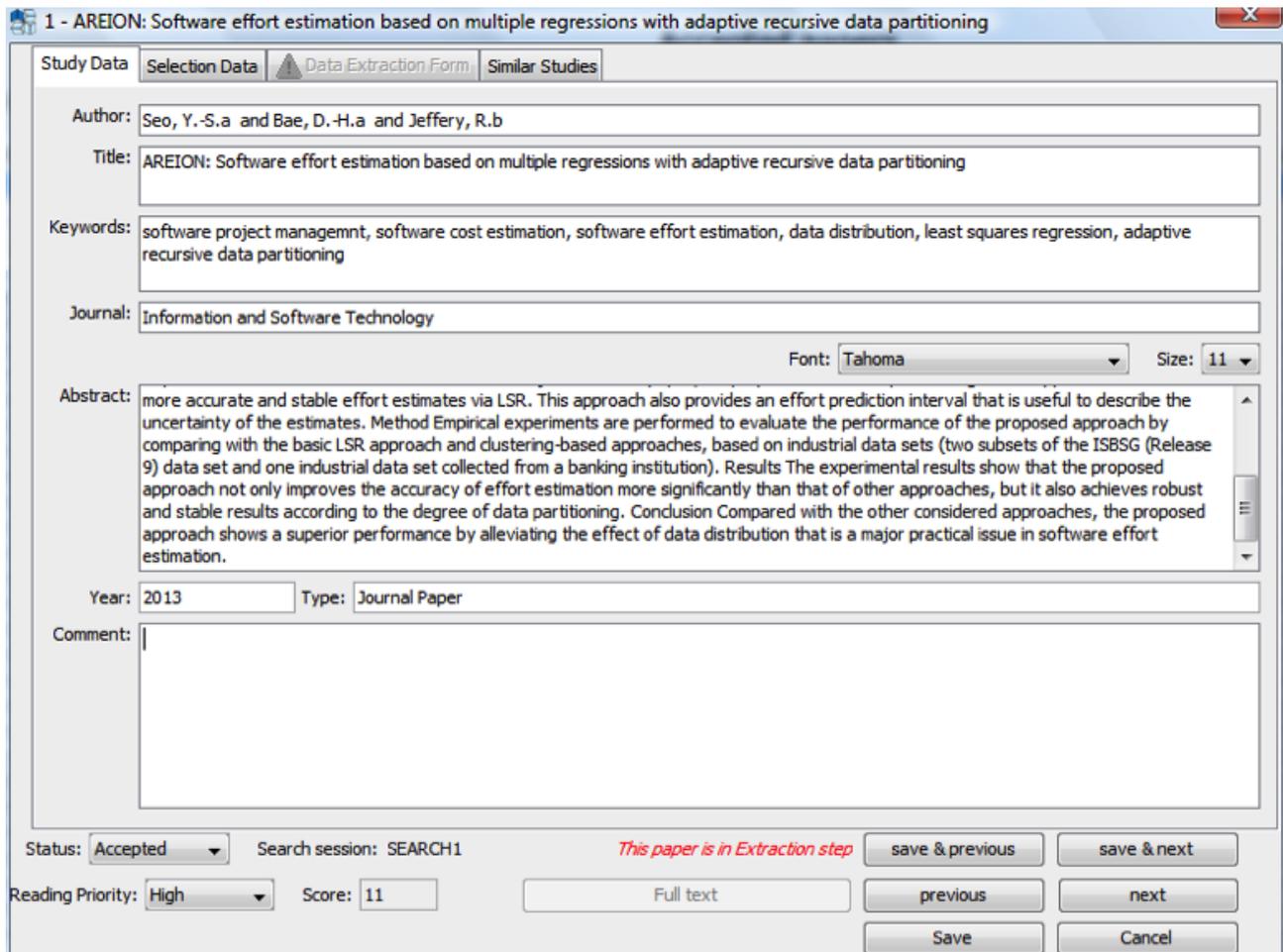


Figura A.2 – Informações Gerais dos Artigos

Figura A.3 – Aplicação dos Critérios de Inclusão e Exclusão na Fase I

nual foram retornados por algum dos motores de busca, enfatizando assim, a importância de tais trabalhos.

Dos 1588 artigos revisados, 132 artigos foram escolhidos para a Fase II da revisão sistemática. O critério para que um paper fosse selecionado a participar da Fase II era que deveria responder SIM para a pergunta abaixo.

- Apresenta uma solução para realização de estimativa de esforço para projetos de desenvolvimento de software?

Nessa fase, 1092 artigos foram rejeitados, sendo 1057 pelo critério de exclusão d (relacionado a estimativa de esforço, mas não apresenta uma solução para o problema), 31 pelo critério e (trabalhos voltados para estimativa de defeitos) e 4 pelo critério f (apresenta desafios e direcionamentos na área). Além disso, foram encontradas 330 duplicatas de artigo.

## A.4 EXECUÇÃO DA FASE II

Após a execução da Fase I dessa Revisão Sistemática, foi realizada a leitura detalhada de cada um dos 132 artigos selecionados, a fim de caracterizá-los segundo os critérios definidos no protocolo da revisão e descritos na Seção A.9. Tais critérios foram escolhidos com base na revisão sistemática realizada por Tenório Jr. [TEN10] e na revisão sistemática realizada por [JØR07]. Nesse capítulo, os aspectos analisados foram consolidados em diversas seções, com o propósito de ilustrar o panorama da pesquisa a respeito de Soluções de Apoio à Estimativa de Esforço em Projetos de Desenvolvimento de Software.

### A.4.1 AVALIAÇÃO DE QUALIDADE

Uma vez selecionados, cada um dos 132 artigos foram avaliados de acordo com os 12 critérios de qualidade indicados na Seção A.9. Em conjunto, estes 12 critérios fornecem uma medida do grau em que cada um dos artigos contribui para o resultado desta revisão. Cada um dos 12 critérios foi graduado em uma escala dicotômica (“Sim” ou “Não”). Nesta fase artigos foram retirados e adicionados artigos de acordo com o a seção a seguir.

#### **Retirando e Adicionando Artigos**

Após a leitura completa dos textos constatou-se que os artigos [KUM11], [KUL09] e [GAL07] tratavam-se de duplicatas dos artigos [16], [50] e [94], respectivamente. Com exceção do último, os artigos duplicados continham alterações no título e no resumo do trabalho, mas mantendo os mesmos resultados e sendo publicados em veículos diferentes, por isso, optou-se por excluir os artigos em duplicata, conservando as versões publicadas nos veículos melhores classificados pela CAPES.

Além disso, após a leitura do texto na íntegra, verificou-se que os artigos [ALJ13], [SAX12], [THA12], [KOS12], [CAT11], [PUR09], [JOR03] e [LOP12] não se tratavam de Solution Proposal, ou seja, não apresentavam uma solução para realização de estimativa de esforço, mas sim estudos comparativos entre soluções ou estudos de caso para aplicação de uma solução apresentada anteriormente, por isso, tais artigos também foram excluídos.

Já os artigos [KAS09], [NAR12], [SAN09] e [JUN08] foram excluídos por não corresponderem ao critério de inclusão b, e não terem descritos os dados necessários para avaliação deste artigo nesta revisão sistemática. Enquanto que os artigos [LUC02] e [LIU09] tiveram sua avaliação na primeira fase revisada, pois entendeu-se que as soluções não eram aplicadas ao contexto de desenvolvimento de software, como sugeria o título e o abstract. Os artigos [GAN13], [TOD13], [FU12], [GON12], [CHE11], [ATT11], [PAP09] e [PRI96]

não foram obtidos para leitura na íntegra, não podendo, portanto, serem avaliados nesta fase da pesquisa.

Por fim, os artigos “Software Engineering Economics” e “Software Function, Source Lines of Code, And Development Effort Prediction: a Software Science Validation” não foram retornados na aplicação das strings de busca e na busca manual no Journal IEEE Transactions on Software Engineering, por não serem Solution Proposal em estimativa de esforço, porém, a partir da leitura dos trabalhos selecionados na fase I, identificou-se que esses artigos eram marcos de referência do modelo COCOMO81 e da técnica de Estimativas por Pontos por Função, respectivamente. Uma vez que estas soluções foram propostas inicialmente em livros e que estes foram os primeiros artigos dos autores sobre os temas, optou-se por inserir estes trabalhos na lista final da revisão.

No total, 25 artigos foram excluídos e 2 artigos foram incluídos nesta fase, restando 109 a serem avaliados como diretamente relacionados a proposta de solução para estimativa de esforço em desenvolvimento de software.

A Seção A.12 mostra a avaliação quantitativa dos 109 artigos selecionados, de acordo com os critérios de qualidade propostos, neste caso, foi atribuído peso 1 para as respostas do tipo “sim” e peso 0 para as respostas do tipo “não”. Já a Seção A.13 contém a listagem destes artigos.

#### A.4.2 EXTRAÇÃO DOS DADOS

Durante esta fase, os dados foram extraídos de cada um dos 109 artigos incluídos nesta revisão, para tal foi utilizada, assim como na fase I, a ferramenta StArt, uma vez que esta permite o cadastro de um formulário de extração personalizado. A Figura A.4 apresenta a interface do formulário na ferramenta e a Seção A.11 uma síntese dos dados desse formulário. Desta maneira foi possível registrar os detalhes completos dos artigos em análise e responder mais especificamente as questões propostas na pesquisa.

A seguir serão feitas algumas considerações, baseadas nos 109 artigos selecionados, a respeito da área de Estimativa de Esforço em Desenvolvimento de Software sobre grupos de pesquisa mais atuantes, periódicos e conferências mais ativas nesse contexto, motor de busca que deu melhor resultado para esse tema de pesquisa, dentre outras.

#### A.4.3 MOTOR DE BUSCA E BUSCA MANUAL

A Tabela A.6 mostra os artigos distribuídos pelos motores de busca e pela busca manual. Observa-se que o motor Scopus, individualmente ou concomitante com outra base,

65 - Towards an early software estimation using log-linear regression and a multilayer perceptron model

Study Data Selection Data **Data Extraction Form** Similar Studies

Classificação da Proposta

- Empíricas
- Compostas
- Dinâmicas
- Expertise
- Orientado a Aprendizagem de Máquina
- Regressão Estatística
- Outra

Descrição

Apresenta um modelo de regressão log-linear, baseado no modelo de casos de uso. Usa uma abordagem fuzzy para calibrar o fator de produtividade no modelo de regressão. Um modelo de redes neurais com multicamada de perceptron realiza a predição de esforço.

Domínio: Desenvolvimento Novo

Representação: Numérica

Base Histórica: Sim

Combinação de Técnicas: Sim

Passível de Calibragem: Não

Estimativas são avaliadas?: Não

Método de Pesquisa: Literatura

Classificação do Tipo de Solução: Modelo

Status: Accep... Search session: SEARCH1 *This paper is in Summarization step* save & previous save & next

Reading Priority: High Score: 3 Full text previous next Save Cancel

Figura A.4 – Extração dos Dados na Ferramenta StArt

foi responsável pelo levantamento de 102, dos 109 artigos, cerca de 94% do total. Através da tabela, pode-se observar, também, que todos os artigos encontrados na busca manual foram retornados por algum dos motores de busca.

#### A.4.4 ANO

O gráfico da Figura A.5 ilustra a evolução desse tema nos últimos 35 anos (1978-2013).

Ao analisar a Figura A.5, constata-se que tem havido um interesse maior em soluções voltadas para estimativa de esforço em desenvolvimento de software. A partir de 2010 um número maior de artigos foi aceito nos veículos de publicação, e especialmente em 2012 e 2013 esse crescimento foi bastante evidente, levando em consideração que o levantamento dos artigos foi realizado em Agosto de 2013.

Tabela A.6 – Número de Artigos Encontrados em Motores de Busca e na Busca Manual

<b>Motor</b>	<b>Número de Artigos</b>
Scopus	52
IEEE	1
Science Direct	1
Springer	2
Scopus e IEEE	22
Scopus ScienceDirect	8
Scopus e ACM	7
Scopus e Springer	2
Scopus e Manual	6
Scopus, IEEE e CiteSeerX	2
Scopus, IEEE, ACM e Manual	1
Scopus, IEEE, CiteSeerX e Manual	1
Scopus, Science Direct e Manual	1
IEEE e CiteSeerX	1
<b>TOTAL</b>	<b>109</b>

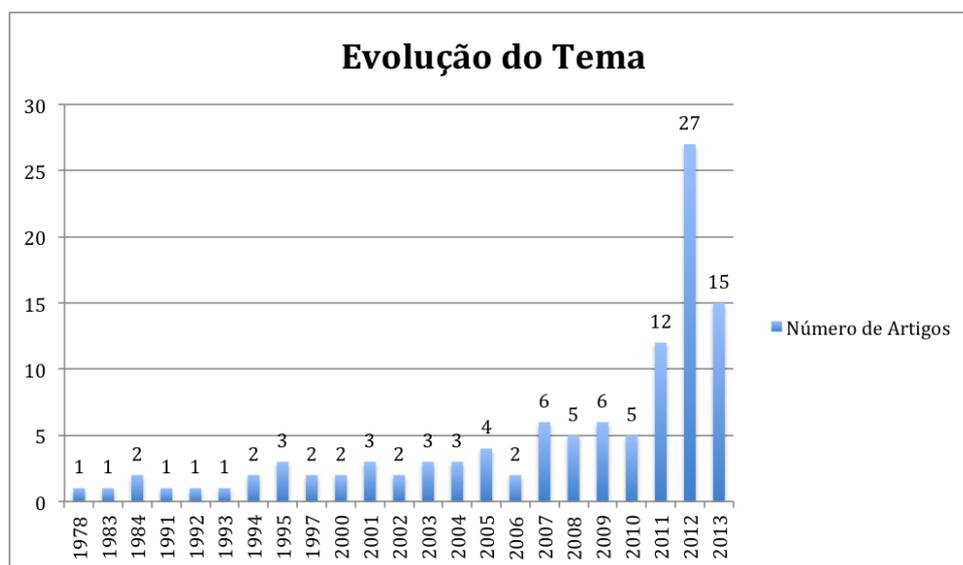


Figura A.5 – Evolução do número de artigos ao longo dos anos.

#### A.4.5 TIPO DE PUBLICAÇÃO

A partir da Figura A.6, ilustrada abaixo, é possível observar que o tema estudado tem tido melhor receptividade em periódicos do que em conferências.

Dos 49 artigos encontrados em conferências, aquelas que mais abriram espaço para o tema de soluções para estimativa de esforço em projetos de desenvolvimento de software foram: International Conference on Tools with Artificial Intelligence, Asia-Pacific Software Engineering Conference, IEEE International Conference on Software Maintenance,

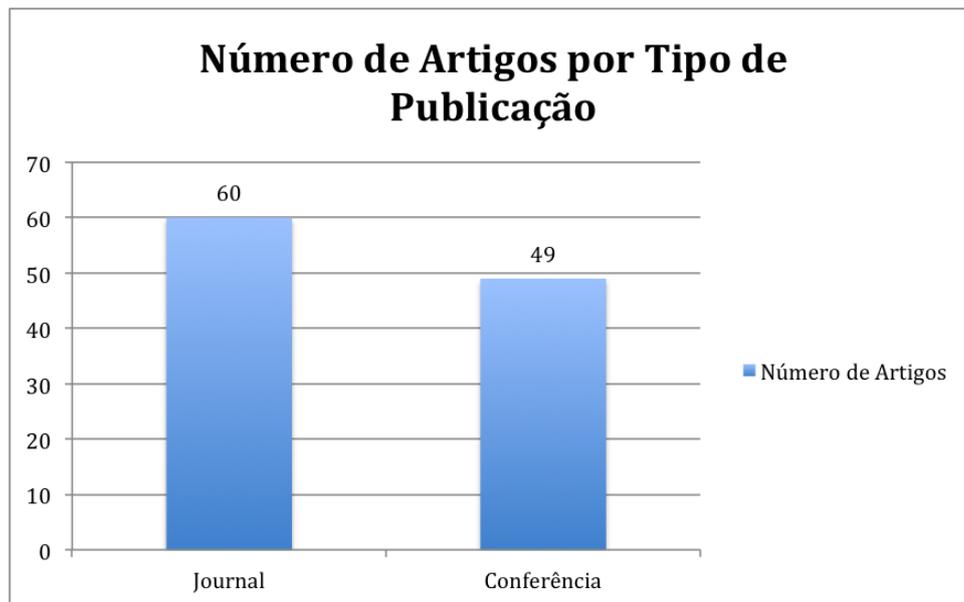


Figura A.6 – Comparativo entre os tipos de publicação e o número de artigos.

ACM International Conference Proceeding Series, ACM Symposium on Applied Computing, EUROMICRO Conference on Software Engineering and Advanced, International Conference on Eletronics Computer Technology, International Conference on Machine Learning and Applications e International Conference on Software Engineering, conforme mostra a Tabela A.7, que apresenta, também, a proporção de artigos encontrados em cada conferência em relação ao total de artigos de conferências. Além dessas conferências, outras 29 foram identificadas, cada uma com 1 artigo. A lista completa de conferências é disponibilizada na Seção A.10.

Tabela A.7 – Número de artigos separados por nome de conferência

Conferência	Artigos	%
International Conference on Tools with Artificial Intelligence	4	8
IEEE International Conference on Software Maintenance	3	6
Asia-Pacific Software Engineering Conference	3	6
ACM International Conference Proceeding Series	2	2
ACM Symposium on Applied Computing	2	2
EUROMICRO Conference on Software Engineering and Advanced Applications	2	2
International Conference on Eletronics Computer Technology	2	2
International Conference on Machine Learning and Applications	2	2

Em relação aos periódicos encontrados, os que mais se destacaram foram Empirical Software Engineering, Lecture Notes in Computer Science, IEEE Transactions on Software Engineering, Communications in Computer Information, Information and Software Technology, Lecture Notes in Business Information Processing, Journal of Systems and

Software, International Journal of Software Engineering and Its Applications, Software Quality Journal, Journal of Computer Science and Technology e European Journal of Computer Science and Technology. Esta lista difere um pouco da apresentada por [JØR07], que foi utilizada como base na busca manual de *journals* (apresentada na seção A.3.1.2), por não haver destacado os periódicos Information and Management, IEEE Software, Communications of the ACM, Journal of Software Maintenance e Evolution e American Programmer. Vale destacar, porém, que o trabalho de [JØR07], ao contrário deste, não tinha como objetivo identificar somente soluções e sim toda a gama de trabalhos acerca do tema. A Tabela A.8 apresenta o número de artigos publicados nestes veículos e o quanto este número representa em % em relação ao total de artigos publicados em periódicos.

Tabela A.8 – Número de artigos organizados por periódicos.

<b>Periódico</b>	<b>Artigos</b>	<b>%</b>
IEEE Transactions on Software Engineering	8	13
Empirical Software Engineering	6	10
Journal of Systems and Software	4	7
Communications in Computer and Information Science	3	5
Lecture Notes in Computer Science	3	5
Information and Software Technology	3	5
Lecture Notes in Business Information Processing	2	3
European Journal of Scientific Research	2	3
International Journal of Software Engineering and Knowledge Engineering	2	3
Software Quality Journal	2	3

Além dos periódicos identificados na Tabela A.9, outros 25 foram contabilizados, cada com 1 artigo. A lista completa de periódicos está na Seção A.10.

#### A.4.6 DEPARTAMENTOS, INSTITUTOS e FACULDADES

Dos 109 artigos selecionados, foram extraídos os departamentos de pesquisa (institutos e faculdades também) que os produziram, a fim de identificar um panorama global de pesquisa na área. Observa-se um total de 152 departamentos distintos envolvidos nos artigos. Nesse contexto, chama a atenção o Departamento de Engenharia de Software da Universiti Teknologi Malaysia, que participa de 7 artigos diferentes. Seguindo-se a esse departamento, encontra-se o Departamento de Engenharia da Computação, da Islamic Azad University, com 6 publicações distintas. Cinco departamentos possuem 4 artigos cada: CSE em Gudlalleru Engineering College, Departamento de Engenharia da Computação da Bagazici University, Departamento de Informática da Aristotle University of Thessaloniki, Departamento de Engenharia de Software da University of Malaya e NFA Estimation Inc.

A Tabela A.9 ilustra os principais departamentos encontrados.

Tabela A.9 – Departamentos, faculdades e institutos interessados pelo tema.

<b>Departamentos</b>	<b>País</b>	<b>Art.</b>
Department of Software Engineering, Faculty of Computer Science and Information System, Universiti Teknologi Malaysia	Malásia	7
Department of Computer Engineering, Bardsir Branch, Islamic Azad University	Irã	6
Department of Informatics, Aristotle University of Thessaloniki	Grécia	4
NFA Estimation Inc.	Canadá	4
CSE Dept, Gudlavalleru Engineering College	Índia	3
Department of Computer Engineering Bogazici University	Turquia	3
Department Of Computer Science and Engineering Anil Nee-rukonda Institute Of Technology Sciences	Índia	3
Department of ECE, University of Western Ontario	Canadá	3
Department of Software Engineering, Faculty of Computer Science and Information Technology, University of Malaya	Malásia	3
Dept of Info. and Computer Science, King Fahd University of Petroleum and Minerals	Arábia Saudita	3
Dipartimento di Informatica – Università di Bari	Itália	3
Departamento de Ingeniería, Pontificia Universidad Católica del Perú	Peru	2
Department of Computer Science and Engineering Motilal Nehru National Institute of Technology	Índia	2
Department of Computing, The Hong Kong Polytechnic University	Hong Kong	2
Department of ECE, Western University	Canadá	2
Department of Information Management, National Taiwan University of Science and Technology	China	2
Dept. of Computer Science and Engineering, Tatung University	China	2
Graduate School of Information Science, Nara Institute of Science and Technology	Japão	2
ICMC-USP	Brasil	2
Informatics Institute, Middle East Technical University	Turquia	2
Information Systems Department, CUCEA, Guadalajara University	México	2
Information Systems, School of Business Administration, Capital College	EUA	2
Institute of Mathematical Sciences and Computing University of São Paulo	Brasil	2
Medi-Caps Institute of Technology and Management	Índia	2
MIS and Decision Sciences, Eberly College of Business and Information Technology, Indiana University of Pennsylvania	EUA	2
School of Computer Science and IT, Devi Ahilya University Indore	Índia	2
School of Computer Science and Software Engineering, Monash University	Malásia	2

#### A.4.7 AUTORES

A partir desta revisão também foi possível descobrir quais os autores que mais publicam nos veículos de comunicação consultados, a respeito do tema estudado. Foram encontrados 236 autores distintos. No entanto, apenas 47 autores possuem mais de 1 publicação sobre o tema, o que demonstra bastante heterogeneidade na área, em relação aos autores que publicam artigos. A Tabela A.10 ilustra o nome dos dezessete autores com pelo menos três publicações sobre o tema.

Tabela A.10 – Autores com mais de duas publicações sobre o tema

<b>Autores</b>	<b>Artigos</b>
D. N. A. Jawawi	6
E. Khatibi	6
V. K. Bardsiri	6
S. Z. M. Hashim	5
A. B. Nassif	4
D. Ho	4
L. F. Capretz	4
S. Huang	4
D. Caivano	3
E. Mendes	3
I. Attarzadeh	3
L. Angelis	3
M. P. Basgalupp	3
N. Chiu	3
R. C. Barros	3
S. Dehuri	3
T. R. Benala	3

Cabe salientar o destaque de “D. N. A. Jawawi”, “E. Kathibi” e “V. K. Bardsiri”, autores que mais publicam nessa área. Eles possuem 6 artigos publicados, dando destaque ao Departamento de Engenharia de Software da Universiti Teknologi Malaysia do qual são pesquisadores.

A fim de compreender a interação dos autores sobre “Soluções de Apoio a Estimativa de Esforço em Projetos de Desenvolvimento de Software”, foi gerado um grafo que ilustra a Rede Social (SNA - *Social Network Analysis*) [HAN05] formada pelos autores que pesquisam a respeito desse tópico. Neste caso, por conta do grande volume de autores, optou-se por gerar a rede apenas para aqueles que tem mais de uma publicação sobre o tema, sendo assim, eventualmente, um autor que tenha apenas uma publicação, mas que tenha publicado com um autor que tenha mais de uma, será representado na rede.

Para a construção de tal SNA foi utilizada a ferramenta R [R14], que é um ambiente de software livre para utilização em análises estatísticas e geração de gráficos. Inicialmente

gerou-se uma matriz quadrada de tamanho 118, a fim de capturar o relacionamento dos autores. A Tabela A.11 exibe o relacionamento entre alguns dos 118 autores representados na rede. Através dela é possível, por exemplo, perceber que os autores “D. N. A. Jawawi” e “E. Kathibi” escreveram 5 artigos juntos.

Tabela A.11 – Matriz de Autores para Geração da Rede Social

-	<b>D. N. A. Jawawi</b>	<b>E. Khatibi</b>	<b>V. K. Bardsiri</b>	...	<b>S. Tunalilar</b>
D. N. A. Jawawi	0	5	5	...	0
E. Khatibi	5	0	6	...	0
V. K. Bardsiri	5	6	0	...	0
S. Z. M. Hashim	5	6	6	...	0
...	...	...	...	...	...
S. Tunalilar	0	0	0	...	0

De posse do arquivo acima, executou-se dentro do ambiente R, o *script* descrito na Figura A.7, que gerou a rede social de autores ilustrada na Figura A.8. A rede social da Figura A.8 também sofreu uma alteração manual dos nodos para que ficasse mais visível de enxergar o relacionamento entre os autores.

```
read.delim("C:/autores.tab", row.names = 1)
g<-graph.adjacency(dados,mode="undirected")
V(g)$label <- V(g)$name
tkplot(g, layout=layout.fruchterman.reingold)
```

Figura A.7 – Código para gerar SNA dos autores, dentro do ambiente R.

Observando-se o grafo exibido na Figura A.8, pode-se perceber alguns relacionamentos que se destacam entre os autores. Na marcação (1), por exemplo, “D. N. A. Jawawi”, “S. Z. M. Hashim”, “E. Kathibi” e “V. K. Bardsiri” são os autores com maior interação entre todos, pois publicaram 6 artigos juntos, com exceção de “S. Z. M. Hashim” que participou de apenas 5 das publicações. Pode-se perceber, também, que esses autores só publicam entre si, não havendo interação com outros pesquisadores.

Na marcação (2) há uma grande interação entre os autores: “D. Ho”, “A. B. Nassif” e “L. F. Capretz” na escrita de artigos. Os dois últimos ainda relacionam-se com “M. Azzeq”, que por sua vez tem um trabalho com “P. Cowling” e “D. Neagu”. Em (3),(4), (5), (6), (7) e (8) é possível visualizar que os autores com maior volume de artigos publicados sobre o tema trabalham com um pequeno grupo fechado de colaboradores, não havendo grandes interações entre os pesquisadores da área.

Após a fase II concluída, obteve-se um mapeamento sistemático sobre a área estudada, pautada sobre a análise de diferentes ângulos do tema de pesquisa. Essa análise

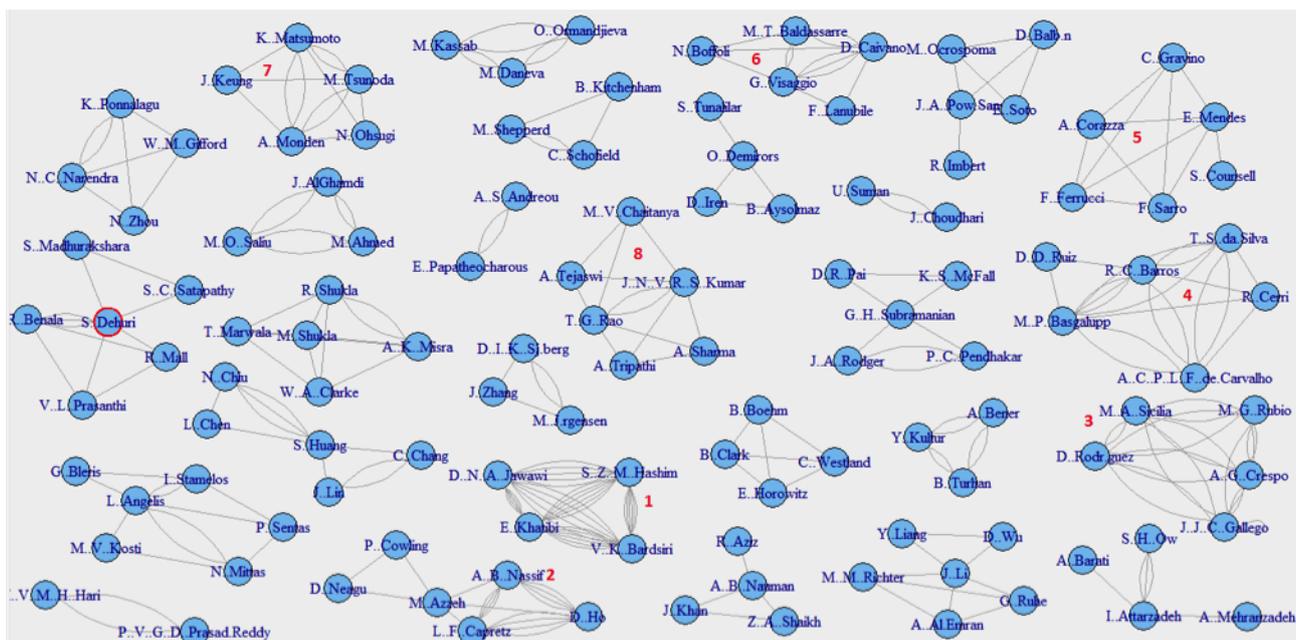


Figura A.8 – Grafo da rede social dos autores que pesquisam sobre Soluções de Apoio a Estimativa de Esforço em Projetos de Desenvolvimento de Software.

foi realizada a partir da caracterização dos 109 artigos, relacionando-os aos motores de busca, ano de publicação, tipo de publicação, departamentos, institutos e faculdades, e por fim, aos autores que os escreveram. O próximo capítulo descreverá os Resultados da Revisão Sistemática, onde há um aprofundamento sobre os assuntos abordados nos 109 artigos lidos.

## A.5 RESULTADOS

Foram identificadas 109 soluções para apoio a realização de estimativa de esforço em projetos de desenvolvimento de software, dentre métodos, modelos, ferramentas, *frameworks*, abordagens, técnicas e metodologias. Os resultados foram condensados em 4 categorias principais, de acordo com as questões de pesquisa da revisão: (1) Visão Geral das Soluções (2); Técnicas de Estimativa de Esforço Aplicadas; (3) Domínio de Desenvolvimento; (4) Validação da Solução.

### A.5.1 VISÃO GERAL DOS ESTUDOS

A Tabela A.12, A.13 e A.14 apresenta uma visão geral das soluções encontrados.

De acordo com as Tabelas A.12, A.13 e A.14 pode-se verificar que 86% dos trabalhos encontrados justificam a necessidade da solução proposta e tem requisitos coletados com base na literatura relacionada, ou seja, fazem uma revisão de trabalhos anteriores e encontram *gaps* a serem solucionados. Comparar soluções existentes e extrair pontos fortes, fracos e possibilidades de melhoria também é uma maneira de coletar requisitos identificada em 12% das soluções. Por fim, foram encontrados 2 trabalhos (cerca de 2%) [2] e [93] que realizaram um estudo de caso para identificação do problema e coleta de requisitos. Com isso, é possível observar que as soluções pouco se baseiam em evidências empíricas em sua formulação, e sim, na literatura.

A Tabela A.15 mostra a frequência de classificação das soluções quanto ao método de pesquisa, a frequência absoluta é dada pela contagem dos trabalhos conforme o método de pesquisa e a frequência relativa é dada pelo percentual em relação à quantidade total de trabalhos encontrados.

Além disso, pode-se observar que quase a totalidade das soluções encontradas, cerca de 98%, tem foco científico. Tais soluções são importantes para que as técnicas de estimativas apresentadas possam ser analisadas e aprimoradas. Por outro lado, trabalhos voltados para a indústria de desenvolvimento de software auxiliam na identificação das vantagens e desvantagens da adoção de cada técnica, bem como na avaliação dos resultados em ambiente real. Neste contexto nota-se a carência de soluções, conforme a apresenta a Tabela A.16.

No que diz respeito ao tipo de solução proposta, modelo e método são as mais comuns, representando 39% e 37% do total, respectivamente. Além disso, foram identificadas abordagens, ferramentas, técnicas, *frameworks* e metodologias, conforme mostra a Tabela A.17.

Tabela A.12 – Visão geral dos resultados - Parte 1 (Artigo 1-39)

Art.	Método de Pesquisa	Domínio	Foco	Técnica	Solução	Valor	Etapa
[1]	Literatura	Manutenção	Científico	OAM	Framework	Numérico	Todas
[2]	Estudo de Caso	Novo	Indústria	Regressão	Modelo	Numérico	Todas
[3]	Comparativo	Novo	Científico	OAM	Framework	Numérico	Todas
[4]	Literatura	Novo	Científico	Composta	Método	Numérico	Todas
[5]	Literatura	Novo	Científico	OAM	Método	Numérico	Plan. e Req.
[6]	Literatura	Novo	Científico	OAM	Método	Numérico	Todas
[7]	Literatura	Novo	Científico	Composta	Modelo	Numérico	Todas
[9]	Literatura	Novo	Científico	Bas. em Modelo	Modelo	Numérico	Todas
[10]	Literatura	Novo	Científico	OAM	Método	Numérico	Todas
[11]	Literatura	Novo	Científico	OAM	Método	Numérico	Todas
[12]	Literatura	Novo	Científico	OAM	Abordagem	Numérico	Todas
[13]	Literatura	Novo	Científico	Composta	Modelo	Numérico	Todas
[14]	Literatura	Novo	Científico	Composta	Método	Numérico	Todas
[15]	Literatura	Novo	Científico	Composta	Método	Numérico	Todas
[16]	Literatura	Novo	Científico	OAM	Método	Numérico	Todas
[17]	Literatura	Novo	Científico	OAM	Método	Numérico	Todas
[18]	Literatura	Novo	Científico	Composta	Modelo	Numérico	Todas
[19]	Literatura	Novo	Científico	OAM	Modelo	Numérico	Todas
[20]	Literatura	Novo	Científico	OUTRA	Método	Numérico	Todas
[21]	Literatura	Novo	Científico	OAM	Framework	Numérico	Todas
[22]	Literatura	Novo	Científico	OAM	Ferramenta	Numérico	Todas
[23]	Comparativo	Novo	Científico	Expertise	Método	Numérico	Todas
[24]	Literatura	Novo	Científico	Composta	Método	Numérico	Todas
[25]	Comparativo	Novo	Científico	OUTRA	Método	Numérico	Todas
[26]	Literatura	Novo	Científico	OAM	Framework	Numérico	Todas
[27]	Literatura	Novo	Científico	OAM	Modelo	Numérico	Todas
[28]	Literatura	Novo	Científico	Composta	Modelo	Numérico	Todas
[29]	Literatura	Novo	Científico	Composta	Método	Numérico	Todas
[30]	Literatura	Novo	Científico	OUTRA	Método	Numérico	Todas
[31]	Literatura	Novo	Científico	OAM	Método	Numérico	Plan. e Req.
[32]	Comparativo	Novo	Científico	OAM	Método	Numérico	Todas
[33]	Literatura	Novo	Científico	OAM	Método	Numérico	Todas
[34]	Comparativo	Novo	Científico	Regressão	Modelo	Numérico	Todas
[35]	Literatura	Novo	Científico	Regressão	Abordagem	Numérico	Todas
[36]	Literatura	Novo	Científico	OUTRA	Técnica	Numérico	Todas
[37]	Comparativo	Novo	Científico	OUTRA	Método	Numérico	Todas
[38]	Literatura	Novo	Científico	Bas. em Modelo	Modelo	Numérico	Todas
[39]	Comparativo	Novo	Científico	Bas. em Modelo	Modelo	Numérico	Todas

Tabela A.13 – Visão geral dos resultados - Parte 2 (Artigo 40-79)

<b>Art.</b>	<b>Método de Pesquisa</b>	<b>Domínio</b>	<b>Foco</b>	<b>Técnica</b>	<b>Solução</b>	<b>Valor</b>	<b>Etapa</b>
[40]	Comparativo	Novo	Científico	Expertise	Método	Numérico	Todas
[41]	Literatura	Novo	Científico	OAM	Método	Numérico	Todas
[42]	Literatura	Novo	Científico	OAM	Modelo	Numérico	Todas
[43]	Literatura	Manutenção	Científico	Composta	Método	Numérico	Todas
[44]	Literatura	Novo	Científico	OUTRA	Metodologia	Numérico	Todas
[45]	Literatura	Novo	Científico	OAM	Modelo	Numérico	Todas
[46]	Literatura	Novo	Científico	Bas. em Modelo	Abordagem	Numérico	Todas
[47]	Comparativo	Novo	Científico	Regressão	Método	Numérico	Todas
[48]	Literatura	Novo	Científico	OAM	Ferramenta	Numérico	Todas
[49]	Literatura	Novo	Científico	OAM	Modelo	Numérico	Todas
[50]	Literatura	Novo	Científico	OAM	Modelo	Numérico	Todas
[51]	Literatura	Novo	Científico	OAM	Modelo	Numérico	Todas
[52]	Literatura	Manutenção	Científico	OAM	Método	Numérico	Todas
[53]	Literatura	Novo	Científico	OAM	Modelo	Numérico	Todas
[54]	Literatura	Novo	Científico	OAM	Modelo	Numérico	Todas
[55]	Literatura	Manutenção	Científico	OAM	Ferramenta	Numérico	Todas
[56]	Literatura	Novo	Científico	OAM	Abordagem	Numérico	Todas
[57]	Literatura	Reeng.	Científico	OUTRA	Método	Numérico	Todas
[58]	Literatura	Novo	Científico	OUTRA	Modelo	Numérico	Todas
[59]	Comparativo	Novo	Científico	OAM	Método	Numérico	Todas
[60]	Literatura	Novo	Científico	OAM	Modelo	Numérico	Todas
[61]	Literatura	Novo	Científico	OAM	Modelo	Numérico	Todas
[62]	Literatura	Novo	Científico	Composta	Abordagem	Numérico	Todas
[63]	Literatura	Novo	Científico	Composta	Framework	Numérico	Todas
[64]	Literatura	Novo	Científico	OAM	Modelo	Numérico	Todas
[65]	Literatura	Novo	Científico	OAM	Método	Numérico	Todas
[66]	Literatura	Novo	Indústria	Composta	Modelo	Numérico	Todas
[67]	Literatura	Novo	Científico	Composta	Modelo	Numérico	Todas
[68]	Literatura	Novo	Científico	OAM	Método	Numérico	Todas
[69]	Literatura	Novo	Científico	OAM	Modelo	Numérico	Todas
[70]	Comparativo	Novo	Científico	OAM	Método	Numérico	Todas
[71]	Literatura	Novo	Científico	OAM	Modelo	Numérico	Todas
[72]	Literatura	Novo	Científico	Composta	Modelo	Numérico	Todas
[73]	Literatura	Novo	Científico	Composta	Modelo	Numérico	Todas
[74]	Literatura	Manutenção	Científico	OUTRA	Técnica	Numérico	Todas
[75]	Literatura	Novo	Científico	OAM	Framework	Numérico	Todas
[76]	Literatura	Novo	Científico	OAM	Modelo	Numérico	Todas
[77]	Literatura	Manutenção	Científico	OAM	Técnica	Numérico	Todas
[78]	Literatura	Novo	Científico	Composta	Método	Numérico	Todas
[79]	Literatura	Novo	Científico	OAM	Modelo	Numérico	Todas

A Tabela A.18 mostra a frequência de como as soluções representam os valores das estimativas, uma vez que o valor é comumente numérico (homens/mês), é interessante

Tabela A.14 – Visão geral dos resultados - Parte 3 (Artigo 80-109)

Art.	Método de Pesquisa	Domínio	Foco	Técnica	Solução	Valor	Etapa
[80]	Literatura	Novo	Científico	Composta	Modelo	Numérico	Todas
[81]	Literatura	Novo	Científico	Composta	Modelo	Numérico	Todas
[82]	Literatura	Novo	Científico	Bas. em Modelo	Framework	Numérico	Todas
[83]	Literatura	Novo	Científico	OAM	Modelo	Numérico	Todas
[84]	Literatura	Novo	Científico	OAM	Método	Numérico	Todas
[85]	Literatura	Novo	Científico	Composta	Modelo	Numérico	Todas
[86]	Literatura	Novo	Científico	Composto	Modelo	Numérico	Todas
[87]	Literatura	Novo	Científico	OAM	Método	Numérico	Todas
[88]	Literatura	Manutenção	Científico	OAM	Modelo	Numérico	Todas
[89]	Literatura	Novo	Científico	Composta	Técnica	Numérico	Todas
[90]	Literatura	Novo	Científico	OAM	Técnica	Numérico	Todas
[91]	Literatura	Novo	Científico	OAM	Método	Numérico	Todas
[92]	Literatura	Novo	Científico	Bas. em Modelo	Modelo	Numérico	Todas
[93]	Estudo de Caso	Novo	Científico	Bas. em Modelo	Modelo	Numérico	Todas
[94]	Literatura	Novo	Científico	Regressão	Método	Numérico	Todas
[95]	Comparativo	Novo	Científico	Regressão	Ferramenta	Numérico	Todas
[96]	Literatura	Novo	Científico	OAM	Método	Numérico	Todas
[97]	Literatura	Novo	Científico	OUTRA	Método	Numérico	Todas
[98]	Comparativo	Reeng.	Científico	Dinâmicas	Método	Numérico	Todas
[99]	Literatura	Reeng.	Científico	Dinâmicas	Método	Numérico	Todas
[100]	Literatura	Reeng.	Científico	Dinâmicas	Ferramenta	Numérico	Todas
[101]	Literatura	Manutenção	Científico	OUTRA	Modelo	Numérico	Todas
[102]	Literatura	Manutenção	Científico	OUTRA	Modelo	Numérico	Todas
[103]	Literatura	Novo	Científico	OAM	Modelo	Numérico	Todas
[104]	Literatura	Novo	Científico	Composta	Framework	Numérico	Todas
[105]	Literatura	Novo	Científico	OAM	Modelo	Numérico	Todas
[106]	Literatura	Novo	Científico	Bas. em Modelo	Abordagem	Numérico	Todas
[107]	Literatura	Novo	Científico	Bas. em Modelo	Ferramenta	Numérico	Todas
[108]	Literatura	Novo	Científico	Composta	Método	Numérico	Todas
[109]	Literatura	Novo	Científico	OAM	Método	Numérico	Todas

Tabela A.15 – Distribuição de Frequência (absoluta e relativa) do aspecto “Método de Pesquisa”

Método de Pesquisa	Quantidade de Artigos	%
Literatura	94	86%
Comparativo	13	12%
Estudo de Caso	2	2%
TOTAL	109	100%

Tabela A.16 – Distribuição de Frequência (absoluta e relativa) do aspecto “Foco da Contribuição”

<b>Foco</b>	<b>Quantidade de Artigos</b>	<b>%</b>
Indústria	2	1,83%
Científico	107	98,17%
TOTAL	109	100,00%

Tabela A.17 – Distribuição de Frequência (absoluta e relativa) do aspecto “Tipo de Solução”

<b>Tipo de Solução</b>	<b>Quantidade de Artigos</b>	<b>%</b>
Modelo	42	39%
Método	40	37%
<i>Framework</i>	8	7%
Ferramenta	7	6%
Abordagem	6	6%
Técnica	5	5%
Metodologia	1	1%
TOTAL	109	100%

identificar se existe outra forma de representação. Neste caso, foram encontrados diversos trabalhos que tratam dados categóricos, como [15], [49], [80],[103], dentre outros, porém todos representam o esforço final como um valor numérico.

Tabela A.18 – Distribuição de Frequência (absoluta e relativa) do aspecto “Valor de Representação”

<b>Representação</b>	<b>Quantidade de Artigos</b>	<b>%</b>
Numérico	109	100%
Categórica	0	0%
TOTAL	109	100%

Outra análise a ser feita é em relação a qual fase do ciclo de desenvolvimento de software é voltada a solução, ou seja, se ela é configurada para calcular a estimativa para uma fase específica ou para todas as fases. Com exceção das soluções [5] e [30], que são voltadas exclusivamente para estimar o esforço do planejamento e análise de requisitos, as demais soluções podem ser aplicadas para qualquer etapa do desenvolvimento, conforme pode-se ver na Tabela A.19.

Tabela A.19 – Distribuição de Frequência (absoluta e relativa) do aspecto “Etapa do Desenvolvimento”

<b>Etapa</b>	<b>Quantidade de Artigos</b>	<b>%</b>
Planejamento/Requisitos	2	1,83%
Todas	107	98,17%
TOTAL	109	100,00%

A partir das Tabela A.12, A.13 e A.14 também podemos observar a distribuição quanto às técnicas utilizadas e ao domínio desenvolvimento para qual é voltada a solução. A análise desses aspectos será feita nas próximas seções.

#### A.5.2 Técnicas de Estimativa de Esforço

Conforme discutido na Seção A.2, esta revisão adota a taxonomia sugerida por [ABT00] para classificação das técnicas de estimativa utilizada em cada solução, pois tal classificação, além de ser a primeira proposta de taxonomia na área, é a mais adequada ao estado da arte das estimativas classificando, se não todas, grande parte das técnicas existentes na literatura. Assim, as técnicas de estimativas são classificadas como (1) Baseadas em Modelo; (2) Dinâmicas; (3) Baseadas em Expertise; (4) Baseadas em Regressão; (5) Orientadas a Aprendizagem de Máquina (OAM); e Compostas. Uma descrição mais detalhada sobre cada técnica é dada no decorrer da Seção A.2.

Com base nessa classificação e acrescentando a opção “outra” para o caso de alguma solução não adotar nenhuma das técnicas da taxonomia de [ABT00], foram obtidos os resultados apresentados na Tabela A.20 e na Figura A.9:

Tabela A.20 – Técnicas de Estimativa no Decorrer dos Anos

-	<b>-1989</b>	<b>1990-1999</b>	<b>2000-2009</b>	<b>2010-2013</b>	<b>TOTAL</b>
Baseadas em Modelo	3	2	2	2	9(8%)
Dinâmica	0	0	3	1	4(4%)
<i>Expertise</i>	0	2	0	0	2(2%)
Regressão	0	1	3	2	6(6%)
OAM	0	3	20	30	53(49%)
Composta	0	2	3	18	23 (21%)
Outra	1	0	5	6	12 (11%)

Esta distribuição sugere que:

- Apesar de técnicas baseadas em Expertise serem as mais utilizadas na indústria de software de acordo com o trabalho de Moløkken et al. 2002 [MOL02], a maioria das soluções propostas para realização de estimativa de esforço não se baseiam nessa técnica. Uma possível justificativa para este comportamento seria a qualidade empírica das estimativas geradas que trazem maior incerteza quanto a precisão dos dados gerados, além de serem altamente dependente do conhecimento humano, o que é prejudicial para as estimativas, em caso de alta rotatividade da equipe;
- A utilização de técnicas orientadas a aprendizagem de máquina e de técnicas compostas teve uma aumento significativo a partir de 2010. Ressaltando que os resultados de 2013 são parciais (até Agosto).

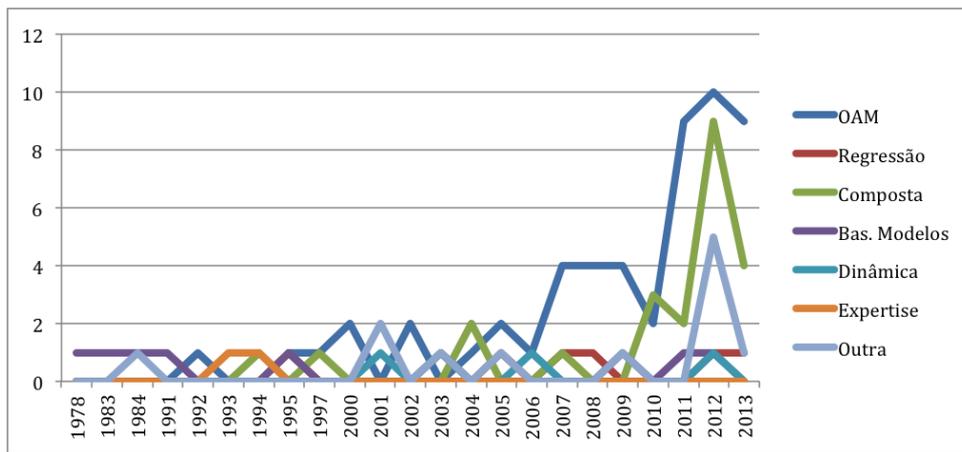


Figura A.9 – Distribuição das Técnicas de Estimativa 1978-2013.

- Regressão Estatística é a base se muitas soluções como COCOMO e soluções orientadas à aprendizagem de máquina. Porém, não há um número significativo de soluções que utilizem análise estatística pura para realização de estimativas;
- Técnicas baseadas em modelo foram as primeiras soluções formais para realização de estimativa de esforço e tiveram grande difusão nos anos 80-90, sendo utilizadas até hoje em soluções que propõe extensões e melhorias nos modelos clássicos. Sobre essa categoria, é importante frisar que foram identificados apenas técnicas baseadas em modelo apresentadas em artigos de periódicos ou conferências, porém, outros modelos como Checkpoint [JON97] e SEER [JEN83], podem ser encontrados em livros sobre o tema;
- O fato de 88% das soluções serem classificadas na taxonomia de [ABT00] reforça a justificativa dada para a adoção da mesma. Porém, é importante enfatizar a existência de trabalhos que utilizem outras propostas de solução para a realização de estimativa de esforço. “Outras” incluem, por exemplo, modelo de previsão vetorial [24], modelos baseados em regras de votação [96], modelos baseados na especificação de requisitos de software [100] e modelos baseados em story points [101].

### Soluções Orientadas à Aprendizagem de Máquina

Destacando-se como a técnica de estimativa de esforço mais utilizada no desenvolvimento de soluções a partir do começo da década de 90, as Orientadas à Aprendizagem de Máquina tem um gama de abordagens que foram adotadas no decorrer dos anos, como Redes Neurais Artificiais, Sistemas Fuzzy, Raciocínio Baseado em Casos; Árvores de Classificação e Regressão e Algoritmos Genéticos.

A Figura A.10 apresenta quais as abordagens de aprendizagem de máquina mais utilizadas nas soluções propostas.

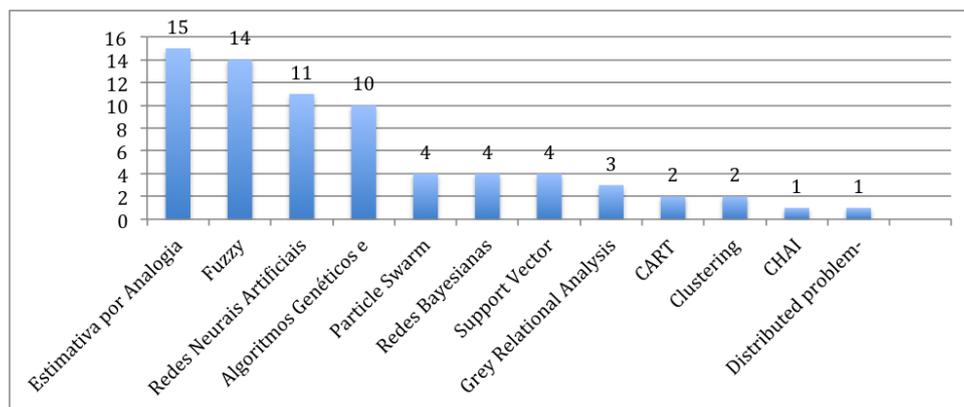


Figura A.10 – Abordagens de Aprendizagem de Máquinas Aplicadas para Estimar Esforço em Desenvolvimento de Software.

A popularidade de Estimativa por Analogia dá-se devido a proximidade desta abordagem com a estimativa por julgamento de especialista, uma vez que nela são usados dados de projetos anteriores para estimar o esforço para um novo projeto a partir da utilização de funções de similaridade. Já a lógica Fuzzy trata um problema comumente encontrado no campo das estimativas por esforço, que é o tratamento dos dados categóricos.

Importante ressaltar dos 53 trabalhos que utilizam OAM, 21 deles combinam mais de uma abordagem como Fuzzy e Redes Neurais [32] [67] [87] [104], ou Estimativa por Analogia e Algoritmos Genéticos [102], por exemplo.

### Soluções Compostas

Além das soluções OAM, as soluções compostas estão em crescente utilização. Tais soluções, como definido por [ABT00] são aquelas que combinam duas ou mais das demais técnicas e tem como vantagem o fato diminuir as desvantagens de uma técnica em particular pela combinação com outras, aumentando, com isso, a precisão da solução.

O gráfico da Figura A.11 mostra as combinações de técnicas mais utilizadas.

Conforme a Figura A.11 podemos notar que a técnica OAM e a baseada em modelo são as que tem maior destaque, sejam em conjunto ou individualmente. A Figura A.12 apresenta as abordagens de OAM (A.12a) e baseadas em modelo (A.12b) mais utilizadas nas combinações.

#### A.5.3 Domínio de Desenvolvimento

Conforme enfatiza a literatura sobre tema, o ciclo de desenvolvimento de software, que inclui as atividades, tarefas, papéis, dentre outros, difere de acordo com o tipo de projeto que está sendo desenvolvido, seja um projeto novo, de manutenção ou de reengenharia

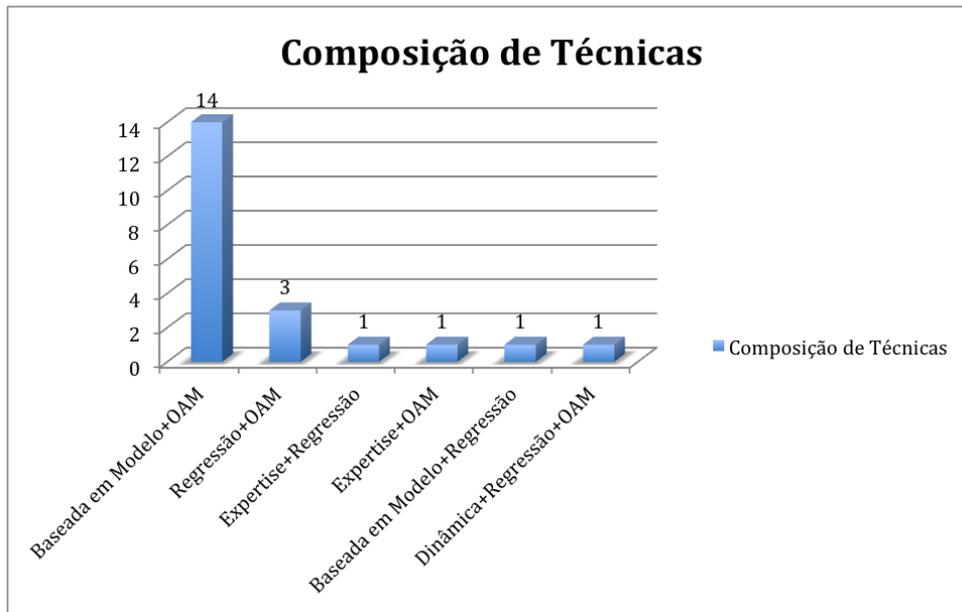


Figura A.11 – Técnicas Combinadas

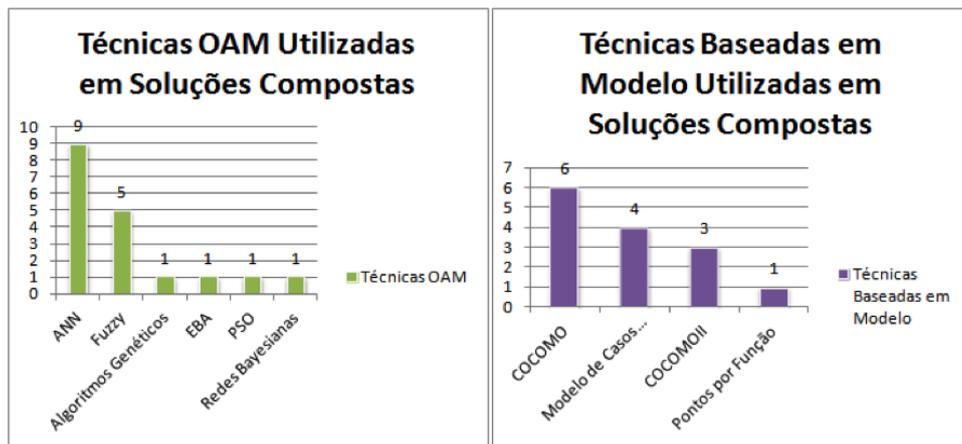


Figura A.12 – Abordagens de OAM e Baseadas em Modelo mais Utilizadas

de sistemas [PRE11], [SOM07]. Por exemplo, o processo de reengenharia tem características que implicam em novas exigências do modelo de estimativa, como componentes de software mais detalhados (ou seja, código) do que aqueles normalmente utilizados nos processos de desenvolvimento (requisitos de sistema e documentos de análise) [PRE11]. Além disso, inclui atividades que dependem de parâmetros de qualidade do sistema legado [VIS01].

Em função disso, uma solução de estimativa de esforço geralmente é voltada para um domínio de desenvolvimento em especial ou possui características que permitem a adequação da solução ao domínio. Conforme apresentam as Tabela A.12, A.13 e A.14, com exceção de [38] em que o modelo COCOMO 2.0 pode ser utilizado tanto em um desenvolvimento de software novo quanto adaptado para o contexto de reengenharia, as demais soluções são categóricas em relação ao domínio para o qual são voltadas. Sendo assim, a

partir da Figura A.13 podemos observar que a maioria das soluções propostas na literatura são aplicáveis ao desenvolvimento de um projeto novo, tendo poucas opções disponíveis para os projetos de manutenção e reengenharia.

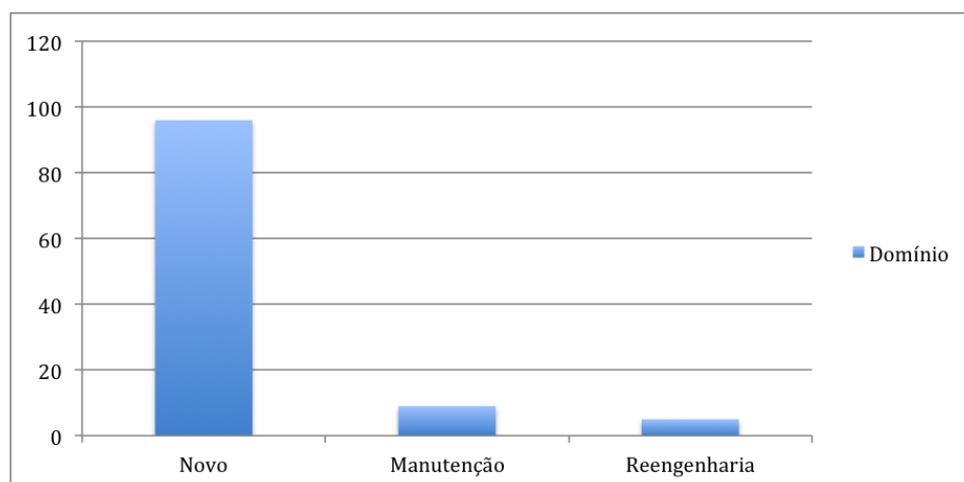


Figura A.13 – Domínios Explorados em Soluções de Estimativa de Esforço.

Outro ponto importante a ser destacado, observável a partir da Figura A.14 são as técnicas de estimativa aplicadas a cada domínio, como discutido na seção anterior as técnicas OAM são as mais utilizadas de maneira geral e com maior crescimento, seguidas das técnicas baseadas em modelo. No entanto tais técnicas tem sido pouco (ou não) exploradas nos domínios de manutenção e reengenharia.

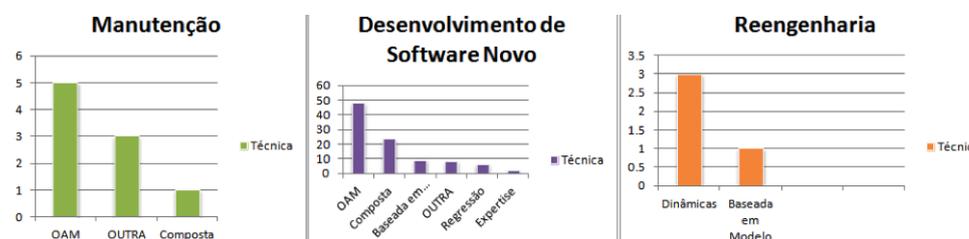


Figura A.14 – Técnicas de Estimativa de Esforço mais Utilizadas de Acordo com o Domínio de Desenvolvimento

#### A.5.4 VALIDAÇÃO DE SOLUÇÕES

É indispensável que uma solução seja ela um modelo, método, metodologia, ferramenta, técnica, *framework*, ou outra, seja validada de maneira a demonstrar que realmente atende as necessidades a que se propõe inicialmente, justificando, com isso, os benefícios de utilização da mesma, em detrimento de outra abordagem. No que diz respeito às soluções voltadas para estimativa de esforço em desenvolvimento de sistemas, a validação é comumente realizada comparando-se a solução proposta com outras soluções que utilizem a mesma técnica.

Dentre as soluções identificadas neste trabalho, cerca 70% (76 trabalhos) realizam validação e a forma como comparam e discutem seus resultados é através do uso de gráficos e tabelas, que analisam diversos vieses (parâmetros). As Tabela A.21, A.22 e A.23 apresentam apenas as soluções propostas que realizam validação, quais as soluções que são usadas para comparação, bem como o tipo de comparação realizada.

A coluna “Compara com” evidencia que as soluções são comparadas com outras dentro da mesma técnica de estimativa aplicada. Como a maioria dos trabalhos são sobre técnicas OAM, as abordagens relacionadas a esta técnica como Estimativa por Analogia (*Estimation by Analogy - EBA*), Árvores de Classificação e Regressão (*Classification and Regression Trees - CART*), Redes Neurais Artificiais (*Artificial Neural Networks - ANN*), Raciocínio Baseado em Casos (*Case-Based Reasoning - CBR*), dentre outras, são as mais aplicadas, juntamente com o modelo COCOMO. De acordo com a coluna “Tipo de Comparação”, os parâmetros analisados são diversos, com destaque para *Mean Magnitude of Relative Error (MMRE)*, *Median Magnitude of Relative Error (MdMRE)* e *Prediction at level (PRED(L))*.

### Tipo de Datasets Utilizados

Os artigos que realizam validação procuram extrair suas análises e discussões a partir da execução de testes sobre datasets, podendo estes serem de dados reais ou de dados sintéticos (artificiais).

Apenas o trabalho [62] utiliza um dataset totalmente artificial, gerado utilizando a equação nominal do COCOMO, definida por Boehm [BOE81]. Trinta dos trabalhos levantados (28%) utilizam dados de projetos particulares, não oferecendo forma de obtenção dos mesmos para reutilização.

No que diz respeito a dados públicos, estes são utilizados por 50 dos trabalhos identificados, sendo que alguns trabalhos utilizam mais de um *dataset*. A Figura A.15 apresenta os *datasets* públicos utilizados para validação de soluções.

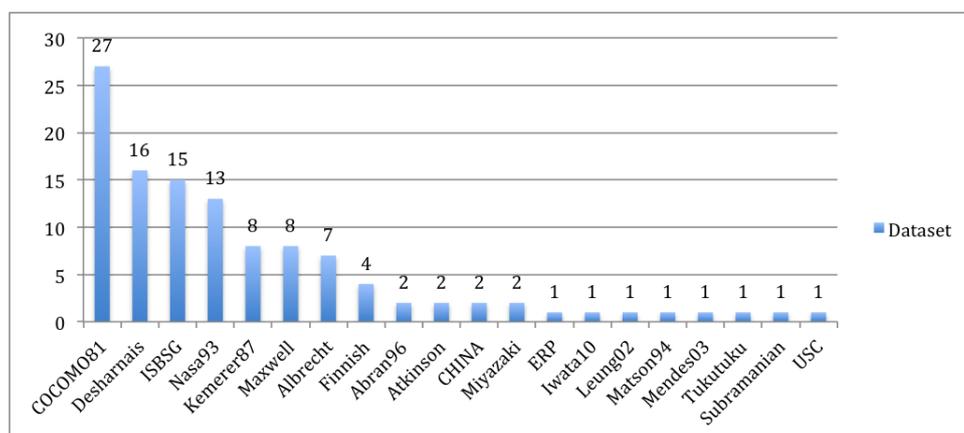


Figura A.15 – Datasets públicos utilizados para validação de soluções

Tabela A.21 – Validação das Soluções - Parte 1

Art.	Compara com	Tipo de Comparação
[1]	EBA e SWR	MMRE e MdMRE
[2]	COCOMO	MRE e fit(ASRE, F e R <sup>2</sup> adj.)
[3]	ANN	RMS e ASREI
[4]	COCOMO81, Halsted, Walston-Felix, Bailey-Basili, Doty, Triangular Member Function e Gbell Function	VAF, MARE e VARE
[5]	CBR, SWR, COCOMO81, Closed Neighbour	MMRE, Pred(25), Strenght, Support, MPS e MPS-W
[6]	EBA, EBA+Clustering, MLR, ANN, CART e SWR	MMRE e Pred(25)
[7]	COCOMOII e Alaa Sheta Model	MMRE, Pred(25) e VAF
[9]	SVM-Regression, ANN e Least-Square Linear Regression	MMRE, RMSE e MAE
[10]	ANN, CART, Algoritmos Genéticos, LABE, NABE, RABE, SWR	MMRE e Pred(25)
[11]	CHAID e CART	MMRE e MBRE
[14]	EBA, ANN, MLR e SWR	MMRE e Pred(25)
[16]	COCOMO, Early Design Model, Post Arch Model, Doty Model, Mittal Model, Swarup Model	MARE
[17]	COCOMO, ANN	MRS e MMRE
[18]	CART e ANN	Pairwise Difference in Means, Absolute Values of Errors
[19]	DMCoMo	MMRE
[21]	EBA	MMRE e Pred(25)
[22]	COCOMO e TRW	Características (processo de estimativa, fatores ambientais, tomada de decisão, etc)
[23]	MLR e Modelo de Casos de Uso	MMRE, MdMRE, Pred e MSE
[24]	Pontos por Função, MKII e COCOMOII	Correlação R <sup>2</sup> , Standart Error
[25]	COCOMO	Pred(25) e RMSRE
[27]	Modelo de Reddy e Raju, e COCOMO	MRE
[28]	Pontos por Função	R <sup>2</sup> , ANOVA, Estimated Effort e MRE
[29]	Julgamento por Especialista e Regressão	Confidence level, Mean HitRate, Median PIWidht, Median MRE
[31]	Fuzzy e Regressão	ANOVA
[32]	COCOMO e ANN	MMRE e MdMRE
[33]	Pontos por Função	RMSE e MRE
[34]	LSR e LSR clusterizado	-
[38]	Ada COCOMO e COCOMO81	características e funcionalidades do modelo
[44]	Regression (listwise, pairwise, mean imputation)	MAE, VAE, MRE, VRE, Pred(25)
[46]	COCOMO e Pontos por Função	-
[47]	Linear Regression e Stepwise Regression	MMRE e R <sup>2</sup>
[48]	SOM e FANN	MAE, MRE, SDAE e SDRE

Tabela A.22 – Validação das Soluções - Parte 2

<b>Art.</b>	<b>Compara com</b>	<b>Tipo de Comparação</b>
[49]	FPA (mean based), FPA (median based), LS regression, LS regression (no outlier), LMS regression, ANN e Fuzzy	MMRE, Pred(10) e Pred(25)
[50]	ENN, NN, e RT	MMRE, MdMRE e Pred (L)
[51]	CN, COCOMO81 e NDE	MARE
[52]	Karners Model e Schneider Model	MMRE, MRE e Pred(L)
[53]	Regression e UCP	MMER e Pred (L)
[55]	Regression	MMRE e Pred(L)
[58]	COCOMO, Pontos por Função e Expertise	MRE
[59]	Bailey-Basili Estimate, Alaa F. Shete G. E. Model, Harish Model	VAF, MARE e VARE
[60]	Fuzzy Grey Relation Analysis e ANN	MMRE e Pred(L)
[61]	OCFWFLANN, OFWFLANN, FLANN, SWR e CART	MMRE, MdMRE e Pred (L)
[62]	COCOMO	Pred(L), nominal error
[63]	Traditional Analogy, Correlation Weighted Analogy e Correlation Weighted PCA Analogy	MMRE, MRE e Pred(L)
[64]	NW, DW, CW, LW , NLW e MW	MMRE e Pred(L)
[66]	Linear Regression	MRE, MER e BRE
[67]	ANN, ABE, CART, SWR, MLR, ABEMA, ABEI, ABEM	MMRE e Pred(25)
[68]	CBR, CART e ANN	MMRE e Pred(25)
[69]	Euc-CBR, Man-CBR, Min-CBR, Gre-CBR, Gau-CBR e Mah-CBR	MMRE, MdMRE e Pred (L)
[70]	EBA, CART, MLR, SWR, ANN, ANN-C-Means, EBA-PSO, EBA-GA, EBA-ANN, EBA-GREY	MMRE, MdMRE, Pred (25) e BMMRE
[71]	LS e EBA	MAE, MdAE, MMRE, MdMRE e pred(25)
[72]	COCOMO	MRE e pred(25)
[76]	Logistic, J48, CART, BFTree	Accuracy, F-Measure, Precision, Recall e tree size
[78]	COCOMOII	-
[79]	COCOMOII	MMRE e Pred(25)
[80]	COCOMOII	MMER e Pred (L)
[81]	COCOMO	MAPE
[82]	FGRA e ANN	MMRE e Pred(25)
[83]	EBA	MMRE e Pred(25)
[84]	MLR e UCP	MMER e Pred (L)
[85]	ANN, Regression	MRE
[86]	SVR RBF, MLP, M5P e Bagging	MMRE e Pred(25)
[87]	FCM-FLANN, KM-FLANN, FLANN, SWR e CART	MMRE, MdMRE e Pred (L)
[88]	OLS, CART, RBFN	MMRE, MdMRE e Pred (L)

Dentre os datasets de destaque tem-se o COCOMO81, que consiste em um conjunto de dados de 63 projetos de software realizados na empresa TRWAerospace e apre-

Tabela A.23 – Validação das Soluções - Parte 3

Art.	Compara com	Tipo de Comparação
[89]	FLANN, SVR, RBF e CART	MMRE, MdMRE e Pred (L)
[90]	J48, CART, BFTree, LEGAL-Tree	Accuracy, F-Measure, Precision, Recall e tree size
[93]	Regressão passo-a-passo, regressão simples	MMRE e Pred(25)
[95]	Regressão Linear e Regressão Logaritmica	MMRE e Pred(25)
[96]	COCOMO, Linear Regression, ANN, Grey Relational Analysis, CBR, CART GA, EBA	MMRE, MdMRE e Pred (L)
[98]	EBA	MMRE
[100]	Belady-Lehman, Schaefer, AMEffMo, Boehm, SMPEEM, FP	Esforço gerado
[102]	EBA, AMH, AMK, AAE, AAMH, AAMK, ANN, CARL, OLS	MMRE, MdMRE e Pred (L)
[103]	COCOMO	Pred(25)
[104]	Regression, MLP, UCP e Sch	MMER, Pred(L), RMSE, MAE e SD
[107]	Pontos por função, Pontos por Caso de Uso	Esforço obtido
[108]	SVR+TS, SVRrand, MSWR, CBR	MdAR, MAR, MRE e EMRE

sentados por Boehm [BOE81]. Cada projeto é descrito por 17 atributos neste conjunto de dados, sendo que tamanho e esforço são atributos numéricos, enquanto os restantes 15 atributos são drivers de esforço utilizados na equação do COCOMO.

O Desharnais, apresentado em [DES89] consiste em dados de 81 projetos de software comercialmente desenvolvidos na empresa Canadian Software House, entre 1981 e 1988.

ISBSG é a sigla para *International Software Benchmarking Standard Group*, uma companhia localizada na Austrália que coleta dados de projetos de desenvolvimento de software do mundo todo, tendo atualmente dados de cerca de 6500 projetos, dentre projetos de desenvolvimento novo e de manutenção e suporte [ISB+14].

O Nasa93, como o nome indica é um conjunto de dados de projetos da *National Aeronautics and Space Administration*, publicados em [JEF93].

Os demais trabalhos são artigos acadêmicos que aplicaram técnicas de estimativa de esforço utilizando dados de projetos reais e disponibilizaram estes dados para a comunidade científica. Podem ser encontrados em Kemerer87 [KEM87], Maxwell [MAX02], Albretcht [ALB83], Finnish [SHE97], Abran96 [ABR96], Atkinson [ATK94], Myiazaki [MYI94], Iwata [IWA10], Leung02 [LEU02], Matson94 [MAT94], Mendes03 [MEN03], Tukutuku [MEN05], Subramanian [SUB96].

Além de poderem ser encontrados em seus artigos de origem, alguns desses datasets, como COCOMO81, Nasa93, Maxwell, dentre outros, podem ser acessados livremente no repositório de dados PROMISE [MEN12], que também contém os dados de CHINA, ERP e USC.

## A.6 CONSIDERAÇÕES FINAIS

A partir do mapeamento sistemático realizado (fase I) e da posterior revisão sistemática (fase II), conforme definido no protocolo desta revisão, foi possível responder questões de pesquisa propostas, que são:

1. Quais as soluções existentes voltadas para auxílio a realização de estimativa de esforço em projetos de desenvolvimento de software?

Foram identificadas 109 soluções (listadas na Seção A.13, por ordem alfabética de título) dentre modelos, métodos, metodologias, ferramentas, *frameworks*, técnicas e abordagens, no período de 1978 à agosto de 2013.

2. Quais são os tipos de técnicas de estimativa de esforço utilizadas nas soluções existentes e como estas técnicas evoluíram no decorrer dos anos?

Conforme discutido na Seção A.5.4.2 as soluções foram classificadas utilizando a taxonomia de Abts [ABT00] cujas técnicas: baseadas em modelo, orientadas à aprendizagem de máquina, dinâmica, regressão estatística, expertise e composta classificam 88% das soluções identificadas, sendo que outras soluções como, por exemplo, modelo de predição vetorial, modelos baseados em regras de votação, modelos baseados na especificação de requisitos de software, modelos baseados em *story points*, sendo encontradas em casos particulares.

No que diz respeito a evolução das técnicas, temos que as baseadas em modelo foram as primeiras formalmente apresentadas, com destaque para o modelo COCOMO [91] e SLIM [8]. Tais técnicas foram amplamente utilizadas e discutidas nos anos seguintes, até meados da década de 90 quando as primeiras soluções orientadas a aprendizagem de máquina surgiram. Tais soluções incluem diversas abordagens como Redes Neurais Artificiais, lógica Fuzzy, Estimativa por Analogia, e diversas outras, sendo a técnica OAM a mais utilizada atualmente, seja sozinha ou combinada com outra técnica.

3. Quais os domínios de desenvolvimento de software para quais as soluções são voltadas?

Conforme apresentado na Seção A.5.4.3 as soluções para apoio a realização de estimativa de esforço são voltadas para os três principais domínios de desenvolvimento: desenvolvimento de software novo, manutenção de software e reengenharia.

Embora haja soluções que atendam a todos os domínios é grande a diferença entre o número de soluções voltadas para desenvolvimento de software novo (88%) em relação as soluções para manutenção (8%) e reengenharia (5%). Apenas uma das

soluções [91] apresenta de forma explícita a possibilidade de ser utilizada em mais de um domínio.

Além disso, no que diz respeito as técnicas de estimativas aplicadas em cada domínio, observa-se que apenas o desenvolvimento novo, devido a gama de soluções, possui soluções que aplicam todas as técnicas possíveis, principalmente no que tange a técnica de OAM, a mais difundida atualmente. Embora as soluções voltadas para a manutenção já estejam adotando esta técnica também, nenhuma solução de reengenharia identificada usa OAM, deixando a aberta a possibilidade de investigação nessa área.

4. Como é realizada a validação das soluções existentes? Por fim, a Seção A.6.4 apresenta como as soluções identificadas validam seus resultados. Cerca 70% dos trabalhos realizam comparações entre soluções semelhantes, com base em parâmetros de qualidade tais como MMRE, MdMRE e Pred(L). Para tanto, estes trabalhos utilizam *datasets* reais e artificiais, sendo os *datasets* reais divididos entre públicos e privados. Esta pesquisa identificou as referências para os principais *datasets* públicos, bem como um repositório onde estes dados estão disponibilizados.

#### A.6.1 LIMITAÇÕES

Entende-se como limitações desta pesquisa o fato de ela ter sido realizada por apenas um pesquisador. Embora ela tenha sido supervisionada pelo seu orientador, seria de suma importância a discussão entre mais de um pesquisador, com interesse na área, a respeito de decisões a serem tomadas sobre alguns artigos.

#### A.6.2 CONTRIBUIÇÃO

A contribuição desta pesquisa, a partir de um levantamento bibliográfico metódico, foi melhorar conhecimento e ampliar as discussões na área de estimativa de esforço em projetos de desenvolvimento de software, principalmente no que se refere a criação de soluções que visem apoiar essa atividade. Pode-se destacar como pontos principais a identificação das técnicas de estimativa de esforço em maior evidência e crescimento, bem como os domínios de desenvolvimento para os quais as soluções existentes estão sendo voltadas. Neste caso, evidenciando a baixa quantidade de trabalhos voltados para manutenção e reengenharia de sistemas, e abrindo, com isso, a possibilidade de novas pesquisas voltadas para estas áreas. Outro ponto importante seria identificar a forma de validação de soluções voltadas para estimativa de esforço, bem como a identificação dos

principais *datasets* utilizados para estes fins, o que colabora para o aumento da qualidade de soluções futuras nesta área.

## A.7 REFERÊNCIAS BIBLIOGRÁFICAS

[ABR96] A. Abran, A. "Function points analysis: An empirical study of its measurement processes". IEEE Transactions on Software Engineering, vol. 22-12, 1996, pp. 895–910.

[ABT00] C. Abts, B. Boehm e S. Chulani, "Software Development Cost Estimation Approaches – A Survey", Annuals Software Engineering, vol.10 (1-4), October 2000, pp. 177–205.

[ALB83] A.J. Albrecht et al., "Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation". IEEE Transactions on Software Engineering, vol. 9-6, 1983, pp. 639-648.

[ALJ13] H. Aljamaan et al. "An ensemble of computational intelligence models for software maintenance effort prediction."Advances in Computational Intelligence. Springer Berlin Heidelberg, 2013, pp. 592-603.

[ATT11] I. Attarzadeh et al. "Software development cost and time forecasting using a high performance artificial neural network model". In: Intelligent Computing and Information Science. Springer Berlin Heidelberg, 2011, pp. 18-26.

[ATT12] I. Attarzadeh e S. H. Ow. "Proposing a Novel Artificial Neural Network Prediction Model to Improve the Precision". In: BIONETICS 2010, LNICST 87, pp. 334–342, 2012.

[ATK94] K. Atkinson e M. Shepperd, "The Use of Function Points to Find Cost Analogies". In: European Software Cost Modelling Meeting, Ivrea, Italy, 1994.

[AZZ04] M. Azzeh, D. Neagu e P. Cowling, "Software Effort Estimation Based on Weighted Fuzzy Grey Relational Analysis". In: Proceedings of the 5th International Conference on Predictor Models in Software Engineering. ACM, 2009. p. 8.

[BAS12] M. Basgalupp et al. "Predicting Software Maintenance Effort through Evolutionary based Decision Trees". In: Proceedings of the 27th Annual ACM Symposium on Applied Computing. ACM, 2012. p. 1209-1214.

[BAS13] M. Basgalupp et al. "Software effort prediction: a hyper-heuristic decision-tree based approach". In: Proceedings of the 28th Annual ACM Symposium on Applied Computing. ACM, 2013. p. 1109-1116.

[BEN12a] T. Benala et al. "Software Effort Prediction Using Fuzzy Clustering and Functional Link Artificial Neural Networks". In: Swarm, Evolutionary, and Memetic Computing. Springer Berlin Heidelberg, 2012. p. 124-132. [BEN12b] T. Benala. "Genetic Algorithm

for Optimizing Functional Link Artificial Neural Network Based Software Cost”. In: Proceedings of the InConINDIA 2012, AISC 132, pp. 75–82.

[BIO05] J. Biolchini. “Systematic Review in Software Engineering”. Technical Report RT-ES 679/05, COPPE/UFRJ. Rio de Janeiro, May, 2005.

[BOE81] B. Boehm “Software Engineering Economics”, Prentice Hall, 1981.

[BOE95] B. Boehm et al. “Cost Models for Future Software Life-cycle Processes: COCOMO 2.0,”.In: Annals of Software Engineering Special Volume on Software Process and Product Measurement, J.D. Arthur and S.M. Henry (Eds.), J.C. Baltzer AG, Science Publishers, Amsterdam, The Netherlands, Vol 1, 1995, pp. 45 - 60.

[CAP12] CAPES – Classificação Conferências Ciência da Computação. Disponível em: [http://www.capes.gov.br/images/stories/download/avaliacao/Comunicado\\_004\\_2012\\_Ciencia\\_da\\_Computacao.pdf](http://www.capes.gov.br/images/stories/download/avaliacao/Comunicado_004_2012_Ciencia_da_Computacao.pdf). Acessado em 29/07/2013.

[CAT11] C. Catal, M. Aktas. "A Composite Project Effort Estimation Approach in an Enterprise Software Development Project". In: SEKE, 2011, pp. 331-334.

[CHE11] X. Chen et al. “Software Effort Estimation Model Based on Use Case Specification”. In: International Conference on Evaluation on Novel Approaches to Software, 2011.

[DES89] J. M. Desharnais. “Analyse Statistique de la productivité des projets de développement en informatique à partir de la technique des points de fonction”. Master’s Thesis, Université du Montréal, 1989.

[FOR61] Jay Forrester “Industrial Dynamics”, Forrester, J., MIT Press, Cambridge, MA, 1961.

[FIN97] G. Finnie et al. “A comparison of software effort estimation techniques: using function points with neural networks, case-based reasoning and regression models” Journal of Systems and Software, v. 39-3, 1997, pp. 281-289.

[FU12] Ya-fang Fu et al. “Software Effort Estimation Method Based on Grey Relational Analysis”. Journal of Systems Engineering and Electronics , vol. 34-11, 2012, pp. 2384-2389.

[GAN13] M. Ganesh et al. “FAHSCEP: Fuzzy and Analogy Based Hybrid Software Cost Estimation Process”. International Review on Computers and Software, 2013.

[GAL07] J. Gallego et al. “Software Project Effort Estimation Based on Multiple Parametric Model Generated Through Data Clustering”. Journal of Computer Science and Technology, vol. 22-3, 2007, pp. 371-378.

[GON12] I. González-Carrasco et al. “SEffEst: Effort Estimation in Software Projects Using Fuzzy Logic and Neural Networks”. International Journal of Computational Intelligence Systems, vol. 5-7, 2012, pp. 679-699.

[GRA97] A. Gray e S. MacDonell. "A comparison of techniques for developing predictive models of software metrics", *Information and Software Technology*, vol. 39-6, November 1996, 425–437.

[HAN05] R. Hanneman. "Introduction to social network methods". Riverside, CA: University of California, Riverside, 2005. Disponível em <<http://faculty.ucr.edu/hanneman/>>. Acessado em: 20.01.2014.

[ISB+14] ISBSG (2014) International software benchmark and standards group, Data R8, [www.isbsg.org](http://www.isbsg.org), Março, 2014.

[IWA10] K. Iwata, et. al. "Improving accuracy of an artificial neural network model to predict effort and errors in embedded software development projects," *Studies in Computational Intelligence*, vol. 295, 2010, pp. 11–24.

[JEF93] R. Jeffery e J. Stathis. "Specification Based Software Sizing: An Empirical Investigation of Function Metrics". In: *NASA Goddard Software Eng. Workshop*. Greenbelt, Md., 1993.

[JEN83] R. Jensen "An Improved Macrolevel Software Development Resource Estimation Model". In: *Proceedings 5th ISPA Conference*, April 1983, pp. 88-92.

[JON00] W. Jones, T. Khoshgoftaar, E. Allen, e J. Hudepohl. "Classification tree models of software-quality over multiple releases", *IEEE Transactions on Reliability*, vol. 49-1, March 2000, 4–11.

[JØR03] M. Jorgensen et al. "Software Effort Estimation by Analogy and 'Regression Towards the Mean". *Journal of Systems and Software*, vol. 68-3, Dez., 2003, pp. 253-262.

[JØR05] M. Jørgensen. "Evidence-based guidelines for assessment of software development cost uncertainty". *IEEE Transactions on Software Engineering*, v. 31-11, 2005, pp. 942-954.

[JØR07] M. Jørgensen e M. Shepperd. "A Systematic Review of Software Development Cost Estimation Studies". *IEEE Transactions on Software Engineering*, vol. 33-1, 2007, pp. 33-52.

[JON97] J. Jones. "Applied Software Measurement", 1997, McGraw Hill.

[JUN08] Z. Junguang "The Establishment and Application of Effort Regression Equation". In: *International Conference on Computer Science and Software Engineering*. IEEE, 2008, pp. 11-14.

[KAS09] M. Kassab. "Towards an Early Software Effort Estimation Based on Functional and Non-Functional Requirements". In: *Software Process and Product Measurement*. Springer Berlin Heidelberg, 2009, pp. 182-196.

[KEM87] C. Kemerer. "An empirical validation of software cost estimation models". *Communication of the ACM*, vol. 30-5, 1978, pp. 436–445.

[KHA11] V. Khatibi et al. "A New Fuzzy Clustering Based Method to Increase the Accuracy of Software Development Effort Estimation Department of Software Engineering, Faculty of Computer Science and Information System". In: World Applied Sciences Journal 14 (9): 1265-1275, 2011.

[KIT04] B. Kitchenham, "Procedures for Performing Systematic Reviews". Keele University Technical Report TR/SE-0401. UK, July, 2004.

[KIT07a] B. Kitchenham, "Guidelines for performing systematic literature reviews on software engineering". Keele, UK, Keele University, version 2.3, 2007.

[KIT07b] B. Kitchenham et al. "A Systematic Review of Cross- vs. Within-Company Cost Estimation Studies". IEEE Transactions on Software Engineering, vol. 33-5, 2007, pp 316-329.

[KRI94] A. Krishna, S. Kumar, and P. Satsangi. "Fuzzy systems and neural networks in software engineering project management", Journal of Applied Intelligence, vol. 4-1, May 1993, 31–52.

[KOS12] M. V. Kosti et al. "Alternative methods using similarities in software effort estimation". In: Proceedings of the 8th International Conference on Predictive Models in Software Engineering. ACM, 2012, pp. 59-68.

[KUL08] Y. Kultur. "ENNA: Software Effort Estimation Using Ensemble of Neural Networks with Associative Memory". In: Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of software engineering. ACM, 2008. p. 330-338.

[KUL09] Y. Kultur et al. "Ensemble of neural networks with associative memory (ENNA) for estimating software development costs". Knowledge-based systems, vol. 22-6, 2009, pp.395-402.

[KUM11] J. N. Kumar et al. "A novel model for software estimation using exponential regression as firing interval in fuzzy logic". Communications in Computer and Information Sciences, vol. 142, 2011, pp. 118-127. [LEU02] H. Leung. "Estimating maintenance effort by analogy". Empirical Software Engineering, vol. 7-2, 2002, pp.157–175.

[LIN11a] J. Lin et al. "Research on Software Effort Estimation Combined with Genetic Algorithm and Support Vector Regression". In: Computer Science and Society (ISCCS), 2011 International Symposium on. IEEE, 2011. p. 349-352.

[LIN11b] J. Lin e C. Chang. "Genetic Algorithm and Support Vector Regression for Software Effort Estimation". In: Advanced Materials Research, v. 282, p. 748-752, 2011.

[LIU09] J. Liu et al. "A Bayesian Net Based Effort Estimation Model for Service Governance Processes". In: Second International Conference on Information and Computing Science, 2009, pp. 83-86.

[LOP12] C. Lopez-Martin et al. "Software Effort Prediction of Industrial Projects Applying a General Regression Neural Network". Empirical Software Engineering, vol. 17-6, 2012, pp. 738-756.

[LUC02] A. De Lucia et al. "Early effort estimation of massive maintenance processes". In: International Conference on Software Maintenance, 2002, pp. 234-237.

[MAR96] M. Shepperd et al. "Effort estimation using analogy". In: International Conference on Software Engineering, 1996, pp 170–178 [MAR97] M. Shepperd et al. "Estimating software project effort using analogies". IEEE Transactions on Software Engineering, vol. 23-11, 1997, pp. 736–743.

[MAT94] J. E. Matson et al. "Software development cost estimation using function points". IEEE Transactions on Software Engineering, vol. 20-4, 1994, pp. 275–287.

[MAX02] K. Maxwell. "Applied Statistics for Software Managers". Nova Jersey: Prentice-Hall, Englewood Cliffs, 2002.

[MEN00] E. Mendes et al. "Web development effort estimation using analogy". In: Software Engineering Conference, 2000. Proceedings. 2000 Australian. IEEE, 2000. p. 203-212.

[MEN03] E. Mendes et al. "A comparative study of cost estimation models for web hypermedia applications". Empirical Software Engineering, vol. 8-2, 2003, pp. 163–196.

[MEN05] E. Mendes et al. "Investigating web size metrics for early web cost estimation". Journal of Systems and Software, vol. 77-2, 2005, pp. 157–172.

[MEN12] T. Menzies, B. Caglayan, E. Kocaguneli, J. Krall, F. Peters, and B. Turhan, The PROMISE Repository of empirical software engineering data <http://promisedata.googlecode.com>, West Virginia University, Department of Computer Science, 2012.

[MIY94] Y. Miyazaki et al. "Robust regression for developing software estimation models". Journal of Systems and Software, vol. 27-1, 1994, pp. 3–16.

[MOL02] Moløkken, K., 2002. "Expert estimation of Web-development effort: individual biases and group processes". Dissertação de Mestrado, Department of Informatics, University of Oslo.

[NAR12] N. Narendra et al. "Towards a Formal Model for Optimal Task-Site Allocation and Effort Estimation in Global Software Development". In: SRII Global Conference (SRII), 2012 Annual. IEEE, 2012, pp. 470-477.

[NGU08] V. Nguyen, B. Steece, e B. Boehm. "A constrained regression technique for cocomo calibration". In: ESEM '08: Second ACM-IEEE international symposium on Empirical Software Engineering and Measurement, 2008, 213–222.

[PAP09] E. Papathecharous et al. "Hybrid Computational Models for Software Cost Prediction: An Approach Using Artificial Neural Networks and Genetic Algorithms". In: Enterprise Information Systems. Springer Berlin Heidelberg, 2009, pp. 87-100.

[PAR88] Park "The Central Equations of the PRICE Software Cost Model", In: 4th COCOMO Users' Group Meeting, November 1988.

[PFL11] S. Pfleeger. “Engenharia de Software Teoria e Prática”. São Paulo: Prentice Hall, 2004, 2ª edição, 535p.

[PRE11] R. Pressman. “Software Engineering: A Practitioner’s Approach”. Boston: McGraw-Hill, 2011, 7th edition, 889p.

[PRI96] M. Prietula et al. “Software Effort Estimation Using a Case Based Reasoner”. *Journal of Experimental and Theoretical Artificial Intelligence*, vol. 8-3, 1996, pp. 341-363.

[PUR09] M. Purvis et al. "Software effort estimation: Harmonizing algorithms and domain knowledge in an integrated data mining approach". *International Journal of Intelligent Information Technologies*, 2009.

[PUT78] L. H. Putnam. “A general empirical solution to the macro software sizing and estimating problem”. *IEEE transactions on software engineering*, vol. 4-4, 1978, pp. 345-361.

[R14] The R Project for Statistical Computing. Disponível em <[http:// www. r-project .org/](http://www.r-project.org/)>. Acessado em: 20/01/2014.

[RIB08] M. B. Ribeiro, N. N. Tenorio Jr., and D. D. Ruiz. “A quasi-experiment for Effort and Defect Estimation using Least Square Linear Regression and Function Points”. In: *Software Engineering Workshop, Annual IEEE/NASA Goddard*, 2008, 143–151.

[SG12] The Standish Group. “Chaos Report”. Boston. 2012.

[SAD11] M. Sadiq et al. “Prediction of Software Project Effort Using Fuzzy Logic”. In: *Electronics Computer Technology (ICECT), 2011 3rd International Conference on*. Vol. 4. IEEE, 2011.

[SAN09] P. Sandhu et al. "A model for estimation of efforts in development of software systems". *World Academy of Science, Engineering and Technology*, v. 56, 2009, pp. 148-152.

[SAX12] U. Saxena et al. "Software effort estimation using Neuro-fuzzy approach". In: *Software Engineering (CONSEG), 2012 CSI Sixth International Conference on*. IEEE, 2012. pp. 1-6.

[SB02] K. Schwaber and M. Beedle. “Agile Software Development With Scrum”. Upper Saddle River:Prentice Hall, 2002, 1st. edition, 158p.

[SRI95] K. Srinivasan e D. Fisher. “Machine learning approaches to estimating software development effort”, *IEEE Transaction Software Engineering*, vol.21-2, February 1995, pp. 126–137.

[SHA02] Y Shan. “Software Project Effort Estimation Using Genetic Programming”. In: *Communications, Circuits and Systems and West Sino Expositions, IEEE 2002 International Conference on*. IEEE, 2002. p. 1108-1112.

[SHE97] M. Shepperd M, C. Schofield. "Estimating software project effort using analogies". IEEE Transactions on Software Engineering, vol. 23-12, 1997, pp. 736–743.

[SOM07] I. Sommerville. "Engenharia de Software". Pearson Addison-Wesley, 2007, 8ª edição.

[SEO08] Y. Seo, K. Yoon, and D. Bae. "An empirical analysis of software effort estimation with outlier elimination". In: PROMISE '08: 4th International Workshop on Predictor Models in Software Engineering, 2008, 25–32.

[SHE95] M. Shepperd; C. Schofield et al. "Effort estimation using analogy". In: International conference on software engineering, 1995.

[SUB96] G. H. Subramanian e G. Zarnich. "An Examination of Some Software Development Effort and Productivity Determinants in ICASE Tool Projects". Journal of Management Information Systems, vol. 12, 1996.

[THA12] I. Thamarai et al. "Using differential evolution in the prediction of software effort". In: Advanced Computing (ICoAC), 2012 Fourth International Conference on. IEEE, 2012. pp. 1-3.

[TEN10] N. N. Tenório Jr "Modelo-E10: um modelo para estimativas de esforço em manutenção de software". Tese de Doutorado, Programa de Pós Graduação em Ciência da Computação, Pontifícia Universidade Católica do Rio Grande do Sul, 2010, 132 p.

[TOD13] K. Toda et al. "Hybrid Effort Estimation based on Multivariate Linear Regression and Analogy based Estimation". Computer Software, Vol.30-2, 2013, pp.227-233.

[TRE91] D. Tregueiros e R. Berry. "The application of neural network based methods to the extraction of knowledge from accounting reports". In: System Sciences, 1991. Proceedings of the Twenty-Fourth Annual Hawaii International Conference on. IEEE, 1991. p. 136-146.

[TSU12] M. Tsunoda et al. "Incorporating Expert Judgment into Regression Models of Software Effort Estimation". In: 19th Asia-Pacific Software Engineering Conference, 2008, 374-379.

[UW01] UW The University of Waikato. Weka Machine Learning Project. <http://www.cs.waikato.ac.nz/ml/weka/>, 2013. Último acesso: 11/12/2013.

[VIS01] G. Visaggio. "Ageing of a dataintensive legacy system: symptoms and remedies". Journal of Software Maintenance and Evolution: Research and Practice, v. 13, n. 5, p. 281-308, 2001.

[WEN09] J. Wen et al. "Improve Analogy-Based Software Effort Estimation Using Principal Components Analysis and Correlation Weighting". In: Software Engineering Conference, 2009. APSEC'09. Asia-Pacific. IEEE, 2009. p. 179-186.

[WIE06] R. Wieringa et al. "Requirements engineering paper classification and evaluation criteria: a proposal and a discussion", Requirements Engineering, vol. 11-1, 2006, pp.102–107.

[WAT94] I. Watson e F. Marir. "Case-based reasoning: A review", Knowledge Engineering Review, v. 9, n. 4, p. 327-354, 1994.

[ZAM10] A. B. Zamboni et al. "StArt Uma Ferramenta Computacional de Apoio à Revisão Sistemática". In: Brazilian Conference on Software: Theory and Practice - Tools session, 2010.

[ZKKN13] Ziauddin et al. "A Fuzzy Logic Based Software Cost Estimation Model". In: International Journal of Software Engineering and Its Applications Vol. 7, No. 2, March, 2013

**A.8 CRITÉRIOS USADOS PARA ORGANIZAR OS ARTIGOS SELECIONADOS PELAS STRINGS DE BUSCA E PELA BUSCA MANUAL, FASE I**

Tabela A.24 – Critérios de Organização dos Artigos

TÍTULO	-
AUTOR	-
ABSTRACT	-
ANO	-
TIPO DE PUBLICAÇÃO	-
NOME VEÍCULO DE PUBLICAÇÃO	-
MOTOR DE BUSCA	-
LINK	-
Critérios:	1) Apresenta uma solução para a realização de estimativa de esforço em projeto de desenvolvimento de software? ( )Sim ( )Não

## A.9 CRITÉRIOS DE QUALIDADE USADOS NOS ARTIGOS SELECIONADOS PARA A FASE II

Tabela A.25 – Critérios de Qualidade

<b>Critérios de Qualidade</b>	-
1) Informa qual o embasamento teórico para o desenvolvimento da solução? Considerar: - Justifica a necessidade da solução - Informa as fontes de requisitos para desenvolvimento da solução (literatura, estudo de caso, experiência pessoal, etc)	Sim/Não
2) Informa o domínio de desenvolvimento contribuição? Considerar: - Informa se a solução é para desenvolvimento novo, projeto de manutenção ou de reengenharia	Sim/Não
3) Informa o foco da contribuição? Considerar: - Informa se a solução proposta é voltada/aplicada para o mercado ou se é uma contribuição científica somente	Sim/Não
4) Classifica a técnica aplicada na proposta?	Sim/Não
5) Classifica a solução proposta? Considerar: - Informa o resultado científico da solução (abordagem, método, modelo, metodologia, framework, ferramenta, técnica, etc)	Sim/Não
6) Informa os parâmetros utilizados para o cálculo da estimativa de esforço?	Sim/Não
7) Informa o tipo de valor obtido pela estimativa? - Informa se o resultado gerado é um valor numérico e/ou categórico	Sim/Não
8) Informa se o trabalho se utiliza bases de dados históricas? Considerar: - Informa as bases de dados utilizados - Informa as referencias para obtenção da base de dados	Sim/Não
9) Informa se a proposta permite a calibragem da técnica?	Sim/Não
10) Realiza comparação da proposta com outras propostas? Considerar: - Descreve os critérios usados para comparar	Sim/Não
11) Informa a etapa do processo de desenvolvimento de software a qual a proposta é aplicada? Considerar: - Indica uma etapa em particular do ciclo de desenvolvimento de software a qual a solução é aplicada - Caso não indique uma etapa explicitamente, considerar que se aplicada a todas as etapas.	Sim/Não
12) Informa como foi realizada a validação/experimento da proposta? Considerar: - Descreve o processo de validação	Sim/Não

## A.10 LISTAGEM DAS CONFERÊNCIAS E PERIÓDICOS

Tabela A.26 – Lista de Periódicos

<b>Periódicos</b>	<b>Art.</b>
IEEE Transactions on Software Engineering	8
Empirical Software Engineering	6
Journal of Systems and Software	4
Communications in Computer and Information Science	3
Lecture Notes in Computer Science	3
Information and Software Technology	3
Lecture Notes in Business Information Processing	2
European Journal of Scientific Research	2
International Journal of Software Engineering and Knowledge Engineering	2
Software Quality Journal	2
Journal of Computer Science and Technology	1
Advanced Materials Research	1
Advances in Intelligence and Soft Computing	1
Applied Intelligence	1
Applied Mathematics and Information Sciences	1
Australian Journal of Basic and Applied Sciences	1
Decision Support Systems	1
Engineering Applications of Artificial Intelligence	1
Expert Systems	1
IET Software	1
Information Process and Management	1
Information Software and Technology	1
International Journal of Software Engineering and Its Applications	1
Journal of Computer Information Systems	1
Journal of Software	1
Journal of Software Engineering	1
Journal of Systems Integration	1
Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering	1
Malaysian Journal of Computer Science	1
MIS quartely: Management Information Systems	1
Procedia Technology	1
Scientific Research and Essays	1
Software Engineering Special Volume on Software Process and Product Measurement	1
The Journal of Supercomputing	1
World Applied Sciences Journal	1

Tabela A.27 – Lista de Conferências - Parte 1

<b>Conferências</b>	<b>Art.</b>
International Conference on Tools with Artificial Intelligence	4
IEEE International Conference on Software Maintenance	3
Asia-Pacific Software Engineering Conference	3
ACM International Conference Proceeding Series	2
ACM Symposium on Applied Computing	2
EUROMICRO Conference on Software Engineering and Advanced Applications	2
International Conference on Electronics Computer Technology	2
International Conference on Machine Learning and Applications	2
International Advance Computing Conference	1
International Conference on Software Engineering	1
ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing	1
ACM Symposium on the Foundations of Software Engineering	1
Annual Conference on the north american fuzzy information processing society	1
Annual Conference on Systems, programming, and applications: software for humanity	1
Genetic and Evolucionary Computation Conference	1
IEEE International Conference on Communications, Circuits and Systems	1
IEEE International software engineering standards symposium	1
IEEE Pacific Rim International Symposium on Dependable Computing	1
IEEE Software Engineering Workshop	1
India Software Engineering Conference	1
International Conference on Computational Intelligence, Communication Systems and Networks	1
International Conference on Computer and Communication Technology	1
International Conference on Computer and Electrical Engineering	1
International Conference on Genetic and Evolutionary Computing Applying	1
International Conference on Hybrid Intelligence Systems	1
International Conference on New Trends in Information Science and Service Science	1
International Conference on Product Focused Software Process Improvement	1
International Conference on Wireless Communications, Networking and Mobile Computing	1
International e-Conference on Advanced Science and Technology	1
International Symposium on Computer Science and Society	1
International Symposium on Innovations in Intelligent Systems and Applications	1
International Workshop on Software Measurement	
1 National Conference on Computing and Communication Systems	1
Software Engineering Conference	1

Tabela A.28 – Lista de Conferências - Parte 2

<b>Conferências</b>	<b>Art.</b>
International Workshop on Software Measurement	
1 National Conference on Computing and Communication Systems	1
Software Engineering Conference	1
Symposium on Assessment of Quality Software Development Tools	1
Working Conference on Reverse Engineering	1
World Congress on Information and Communication Technologies	1

## A.11 FORMULÁRIO DE EXTRAÇÃO DOS DADOS

Tabela A.29 – Formulário de Extração de Dados

<b>Extração de Dados</b>	-
1) Descrição da Proposta	O que é proposto, quais as características?
2) Classificação da Solução	Classificação de Boehm et al.2000 (Empírica, Dinâmica, Expertise, Regressão, Orientado a Aprendizagem de Máquina, Composta, Outra). No caso de “Outra”, qual?
3) Domínio	Desenvolvimento Novo/ Manutenção/ Reengenharia
4) Representação	Numérica/Categórica
5) Base Histórica	Sim/Não. Caso “sim”, qual?
6) Passível de Calibragem?	Sim/Não
7) Foco da Contribuição	Mercado/ Científica
8) Método de Pesquisa	Comparativo, Revisão da Literatura, Estudo de Caso, Experimento, Outro.
9) Tipo de Solução	Método, Ferramenta, Metodologia, Modelo, Abordagem, Framework, Técnica
10) Fase do desenvolvimento de software a que se destina a solução	Planejamento, Requisitos, Arquitetura, Implementação, Testes, Implantação, Todas.
11) Estimativas são avaliadas?	Sim/Não
12) Descrição da Validação	Como é feita a validação?

## A.12 APLICAÇÃO DOS CRITÉRIOS DE QUALIDADE NOS 109 ARTIGOS SELECIONADOS

Tabela A.30 – Aplicação dos critérios de qualidade nos 109 artigos seleccionados - Parte 1

Art.	1	2	3	4	5	6	7	8	9	10	11	12	Total
-	Fonte	Dom.	Foc.	Téc.	Sol.	Par.	Valor	Dat.	Cal.	Comp.	Eta.	Val.	-
[1]	1	1	1	1	1	1	1	1	0	1	1	1	11
[2]	1	1	1	1	1	1	1	1	0	1	1	1	11
[3]	1	1	1	1	1	1	1	1	0	1	1	1	11
[4]	1	1	1	1	1	0	1	1	0	1	1	1	10
[5]	1	1	1	1	1	0	1	1	0	1	1	1	10
[6]	1	1	1	1	1	0	1	1	0	1	1	1	10
[7]	1	1	1	1	1	1	1	1	0	1	1	1	11
[9]	1	1	1	1	1	1	1	0	0	0	1	0	8
[10]	1	1	1	1	1	0	1	1	0	1	1	1	10
[11]	1	1	1	1	1	0	1	1	0	1	1	1	10
[12]	1	1	1	1	1	0	1	1	0	1	1	1	10
[13]	1	1	1	1	1	0	1	0	0	0	1	0	7
[14]	1	1	1	1	1	1	1	1	0	1	1	1	11
[15]	1	1	1	1	1	0	1	1	0	1	1	1	10
[16]	1	1	1	1	1	0	1	1	1	1	1	1	11
[17]	1	1	1	1	1	0	1	1	0	1	1	1	10
[18]	1	1	1	1	1	0	1	1	0	1	1	1	10
[19]	1	1	1	1	1	0	1	1	0	1	1	1	10
[20]	1	1	1	1	1	0	1	1	0	1	1	1	10
[21]	1	1	1	1	1	0	1	1	0	1	1	1	10
[22]	1	1	1	1	1	1	1	1	0	1	1	1	11
[23]	1	1	1	1	1	0	1	0	0	1	1	0	8
[24]	1	1	1	1	1	0	1	1	0	1	1	1	10
[25]	1	1	1	1	1	0	1	1	1	1	1	1	11
[26]	1	1	1	1	1	0	1	1	0	1	1	1	10
[27]	1	1	1	1	1	0	1	1	0	0	1	1	9
[28]	1	1	1	1	1	0	1	1	0	1	1	1	10
[29]	1	1	1	1	1	0	1	1	0	1	1	1	10
[30]	1	1	1	1	1	0	1	1	0	1	1	1	10
[31]	1	1	1	1	1	1	1	0	0	0	1	0	8
[32]	1	1	1	1	1	0	1	1	1	1	1	1	11
[33]	1	1	1	1	1	0	1	1	0	1	1	1	10
[34]	1	1	1	1	1	1	1	0	0	1	1	1	10
[35]	1	1	1	1	1	0	1	1	0	1	1	1	10
[36]	1	1	1	1	1	1	1	1	0	1	1	1	11
[37]	1	1	1	1	1	1	1	1	1	0	1	1	11
[38]	1	1	1	1	1	1	1	0	0	0	1	0	8
[39]	1	1	1	1	1	0	1	0	0	1	1	0	8
[40]	1	1	1	1	1	1	1	1	0	0	1	1	10

Tabela A.31 – Aplicação dos critérios de qualidade nos 109 artigos selecionados - Parte 2

<b>Art.</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>	<b>12</b>	<b>Total</b>
-	<b>Fonte</b>	<b>Dom.</b>	<b>Foc.</b>	<b>Téc.</b>	<b>Sol.</b>	<b>Par.</b>	<b>Valor</b>	<b>Dat.</b>	<b>Cal.</b>	<b>Comp.</b>	<b>Eta.</b>	<b>Val.</b>	-
[41]	1	1	1	1	1	0	1	1	0	0	1	1	9
[42]	1	1	1	1	1	0	1	0	0	0	1	0	7
[43]	1	1	1	1	1	0	1	0	1	0	1	0	8
[44]	1	1	1	1	1	0	1	1	0	0	1	1	9
[45]	1	1	1	1	1	0	1	1	0	1	1	1	10
[46]	1	1	1	1	1	0	1	0	1	0	1	0	8
[47]	1	1	1	1	1	0	1	0	1	1	1	0	9
[48]	1	1	1	1	1	0	1	1	0	1	1	1	10
[49]	1	1	1	1	1	0	1	1	0	1	1	1	10
[50]	1	1	1	1	1	1	1	1	0	1	1	1	11
[51]	1	1	1	1	1	0	1	1	0	1	1	1	10
[52]	1	1	1	1	1	0	1	1	0	1	1	1	10
[53]	1	1	1	1	1	0	1	1	0	1	1	1	10
[54]	1	1	1	1	1	1	1	1	0	1	1	1	11
[55]	1	1	1	1	1	0	1	0	0	0	1	0	7
[56]	1	1	1	1	1	0	1	1	0	1	1	1	10
[57]	1	1	1	1	1	0	1	0	0	0	1	0	7
[58]	1	1	1	1	1	1	1	0	0	0	1	0	8
[59]	1	1	1	1	1	0	1	1	0	1	1	1	10
[60]	1	1	1	1	1	0	1	1	0	1	1	1	10
[61]	1	1	1	1	1	0	1	1	0	1	1	1	10
[62]	1	1	1	1	1	0	1	1	0	1	1	1	10
[63]	1	1	1	1	1	0	1	1	0	1	1	1	10
[64]	1	1	1	1	1	0	1	1	0	1	1	1	10
[65]	1	1	1	1	1	0	1	1	0	1	1	1	10
[66]	1	1	1	1	1	0	1	1	0	0	1	1	9
[67]	1	1	1	1	1	0	1	1	0	1	1	1	10
[68]	1	1	1	1	1	0	1	1	0	1	1	1	10
[69]	1	1	1	1	1	0	1	1	0	1	1	1	10
[70]	1	1	1	1	1	0	1	1	0	1	1	1	10
[71]	1	1	1	1	1	0	1	1	0	1	1	1	10
[72]	1	1	1	1	1	0	1	1	0	1	1	1	10
[73]	1	1	1	1	1	1	1	1	0	1	1	1	11
[74]	1	1	1	1	1	1	1	0	0	0	1	0	8
[75]	1	1	1	1	1	1	1	0	0	0	1	0	8
[76]	1	1	1	1	1	0	1	1	0	0	1	1	9
[77]	1	1	1	1	1	0	1	1	0	1	1	1	10
[78]	1	1	1	1	1	0	1	1	0	1	1	1	10
[79]	1	1	1	1	1	0	1	1	0	1	1	1	10
[80]	1	1	1	1	1	0	1	1	0	1	1	1	10
[81]	1	1	1	1	1	1	1	1	0	1	1	1	11
[82]	1	1	1	1	1	0	1	1	0	1	1	1	10
[83]	1	1	1	1	1	0	1	1	0	1	1	1	10

Tabela A.32 – Aplicação dos critérios de qualidade nos 109 artigos selecionados - Parte 3

<b>Art.</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>	<b>12</b>	<b>Total</b>
-	<b>Fonte</b>	<b>Dom.</b>	<b>Foc.</b>	<b>Téc.</b>	<b>Sol.</b>	<b>Par.</b>	<b>Valor</b>	<b>Dat.</b>	<b>Cal.</b>	<b>Comp.</b>	<b>Eta.</b>	<b>Val.</b>	-
[84]	1	1	1	1	1	0	1	1	0	1	1	1	10
[85]	1	1	1	1	1	0	1	1	0	1	1	1	10
[86]	1	1	1	1	1	0	1	1	0	1	1	1	10
[87]	1	1	1	1	1	0	1	1	0	1	1	1	10
[88]	1	1	1	1	1	0	1	1	0	1	1	1	10
[89]	1	1	1	1	1	0	1	1	0	1	1	1	10
[90]	1	1	1	1	1	0	1	1	0	1	1	1	10
[91]	1	1	1	1	1	0	1	1	0	1	1	1	10
[92]	1	1	1	1	1	0	1	0	0	0	1	0	7
[93]	1	1	1	1	1	0	1	0	0	0	1	0	7
[94]	1	1	1	1	1	0	1	1	0	1	1	1	10
[95]	1	1	1	1	1	0	1	1	0	0	1	1	9
[96]	1	1	1	1	1	1	1	1	0	1	1	1	11
[97]	1	1	1	1	1	0	1	1	0	1	1	1	10
[98]	1	1	1	1	1	0	1	1	1	0	1	1	10
[99]	1	1	1	1	1	0	1	1	1	1	1	1	11
[100]	1	1	1	1	1	0	1	0	1	0	1	0	8
[101]	1	1	1	1	1	0	1	0	0	1	1	0	8
[102]	1	1	1	1	1	0	1	0	0	0	1	0	7
[103]	1	1	1	1	1	0	1	1	0	1	1	1	10
[104]	1	1	1	1	1	0	1	1	0	1	1	1	10
[105]	1	1	1	1	1	0	1	1	0	1	1	1	10
[106]	1	1	1	1	1	0	1	1	0	0	1	1	9
[107]	1	1	1	1	1	0	1	0	0	0	1	0	7
[108]	1	1	1	1	1	1	1	1	0	1	1	1	11
[109]	1	1	1	1	1	0	1	1	0	1	1	1	10

### A.13 LISTAGEM DOS ARTIGOS SELECIONADOS E LIDOS NA FASE II

[1] Y. F. Li; M. Xie; T. N. Goh “A bayesian inference approach for probabilistic analogy based software maintenance effort estimation”. In: 14th IEEE Pacific Rim International Symposium on Dependable Computing, 2008.

[2] S. Wu; I. Kuan “A component-based approach on effort estimation”. In: International Conference on Wireless Communications, Networking and Mobile Computing, 2008.

[3] P. C. Pendhakar; J. A. Rodger “A distributed problem-solving framework for probabilistic software effort estimation”. Journal: Expert systems, 2012.

[4] CH. V. M. H. Hari; P. V. G. D. Prasad Reddy “A fine parameter tuning for CO-COMO 81 software effort estimation using particle swarm optimization”. Journal: Journal of software engineering, 2012.

[5] J. Li; G. Ruhe et al. “A flexible method for software effort estimation by analogy”. Journal Empirical software engineering, 2007.

[6] V. K. Barsiri; D. N. A. Jawawi et al. “A flexible method to estimate the software development effort based on the classification of projects and localization of comparisons”. Journal: Empirical software engineering, 2013.

[7] Ziauddin; S. Kamal et al. “A fuzzy logic based software cost estimation model”. Journal: International Journal of Software Engineering and Its Applications, 2013.

[8] L. H. Putnam “A general empirical solution to the macro software sizing and estimating problem”. Journal: IEEE transactions on software engineering, 1978.

[9] R. C. Barros; M. P. Basgalupp et al. “A grammatical evolution approach for software effort estimation”. In: Genetic and evolutionary computation conference, 2013.

[10] V. K. Bardsiri; D. N. A. Jawawi et al. “A hybrid method for increasing the accuracy of software development effort estimation”. Journal: Scientific research and Essays, 2011.

[11] E. Papatheocharous; A. S. Andreou “A hybrid software cost estimation approach utilizing decision trees”. Journal: International journal of software engineering and knowledge engineering, 2012.

[12] K. Pillai; S. Nair “A model for software development effort and cost estimation”. Journal: IEEE transactions on software engineering, 1997. [13] W. Han; T. Lu et al. “A new estimation model for small organic software project”. Journal: Journal of software, 2013.

[14] V. K. Bardsiri; D. N. A. Jawawi et al. “A new fuzzy clustering based method to increase the accuracy of software development effort estimation”. Journal: World applied sciences journal, 2011.

[15] S. Malathi; S. Sridhar “A novel approach to estimate the software effort based on FUZZAN technique”. Journal: European journal of scientific research, 2012.

[16] J. N. Kumar; T. G. Rao et al. "A novel method for software effort estimation using inverse regression as firing interval in fuzzy logic". In: International conference on electronics computer technology, 2011.

[17] O. F. Saraç; N. Duru "A novel method for software effort estimation: estimating with boundaries". In: International symposium on innovations in intelligent systems and applications, 2013.

[18] P. C. Pendhakar; G. H. Subramanian et al. "A probabilistic model for predicting software development effort". Journal: IEEE transactions on software engineering, 2005.

[19] P. Pytel; P. Britos et al. "A proposal of effort estimation method for information mining projects oriented to SMEs". Journal: Lecture notes in business information processing, 2013.

[20] V. K. Bardsiri; D. N. A. Jawawi et al. "A PSO-based model to increase the accuracy of software development effort estimation". Journal: Software quality journal, 2013.

[21] L. Angelis; I. Stamelos "A simulation tool for efficient analogy based cost estimation". Journal: Empirical software engineering, 2000.

[22] H. Lee "A structured methodology for software development effort prediction using the analytic hierarchy process". Journal: Journal of systems and software, 1993.

[23] A. B. Nassif; L. F. Capretz et al. "A treeboost model for software effort estimation based on use case points". In: International conference on machine learning and applications, 2012.

[24] T. E. Hastings; A. S. M. Sajeev "A vector-based approach to software size measurement and effort estimation". Journal: IEEE transactions on software engineering, 2001.

[25] M. A. Ahmed; M. O. Saliu, et al. "Adaptive fuzzy logic-based framework for probabilistic software effort estimation". Journal: Information and Software Technology, 2005.

[26] E. Kocaguneli; A. Tosun; A. Bener "AI-based models for software effort estimation". In: EUROMICRO conference on software engineering and advanced applications, 2010.

[27] A. Kaushik; A. K. Soni et al. "An adaptive learning approach to software cost estimation". In: National conference on computing and communication systems, 2012.

[28] G. Robiolo; R. Orosco "An alternative method employing uses cases for early effort estimation". In: IEEE Software engineering workshop, 2007.

[29] M. Jørgesen; D. I. K. Sjøberg "An effort prediction interval approach based on the empirical distribution of previous estimation accuracy". Journal: International and software technology, 2003.

[30] B. Aysoalmaz; D. Iren et al. "An effort prediction model based on BPM measures for process automation". Journal: Lecture notes in business information processing, 2013.

[31] C. López-Mantín; I. López-Matín et al. "Application of gamma classifier to development effort prediction of software projects". Journal: Applied mathematics information sciences, 2012.

[32] S. Huang; N. Chiu "Applying fuzzy neural network to estimate software development effort". Journal: Applied intelligence, 2009.

[33] W. Lee; K. Hsu et al. "Applying software effort estimation model based on work breakdown structure". In: International conference on genetic and evolutionary computing applying, 2012.

[34] Y. Seo; D. Bae et al. "AREION: software effort estimation based on multiple regressions with adaptive recursive data partitioning". Journal: Information and Software Technology, 2013.

[35] K. Ponnalagu; N. C. Narendra "Automated trendline generation for accurate software effort estimation". In: Annual conference on systems, programming and applications: software for humanity, 2012.

[36] A. Wang; H. Dunsmore "Back-to-front programming effort estimation". Journal: Information process management, 1984.

[37] O. Benediktsson; D. Dalcher et al. "COCOMO-based effort estimation for iterative incremental software development". Journal: Software quality journal, 2003.

[38] B. Boehm; B. Clark et al. "Cost models for future software life-cycle process: COCOMO 2.0". Journal: Software engineering special volume on software process and product measurement, 1995.

[39] B. Griech; J. Pomerol "Design and implementation of knowledge-based decision support system for estimating software development work-effort". Journal: Journal of systems integration, 1994.

[40] J. Khan, Z. A. Shaikh et al. "Development of intelligent effort estimation model based on fuzzy logic using Bayesian networks". Journal: Communications in computer and information science, 2011.

[41] A. B. Nauman; R. Aziz "Development of proficient effort estimation Bayesian belief network through meta-model conversation". Journal: Australian journal of basic and applied sciences, 2011.

[42] R. Shukla; M. Shukla et al. "Dynamic software maintenance effort estimation modeling using neural network, rule engine and multi-regression approach". Journal: Lecture notes in computer science, 2012.

[43] S. Tunalilar; O. Demirs "EFES: an effort estimation methodology". In: International workshop on software measurement, 2012.

[44] N. Ohsugi; M. Tsunoda et al. "Effort estimation based on collaborative filtering". In: International conference on product focused software process improvement, 2004.

[45] J. A. Pow-Sang; R. Imbert "Effort estimation in incremental software development projects using function points". Journal: Communications in computer and information science, 2012.

[46] S. Rahhal; N. Madhavji "Effort estimation model for implementing ISO 9001". In: IEEE International software engineering standards symposium, 1995.

[47] M. Shepperd; C. Schofield et al. "Effort estimation using analogy". In: International conference on software engineering, 1995.

[48] K. Iwata; T. Nakashima et al. "Effort prediction models using self-organizing maps for embedded software development projects". In: International conference on tools with artificial intelligence, 2011.

[49] M. T. Su; T. C. Ling et al. "Enhanced software development effort and cost estimation using fuzzy logic model". Journal: Malaysian journal of computer science, 2007.

[50] Y. Kultur; B. Turhan et al. A. B. Bener "ENNA: software effort estimation using ensemble neural networks with associative memory". In: ACM symposium on the foundations of software engineering, 2008.

[51] H. Leung "Estimating maintenance effort by analogy". Journal: Empirical software engineering, 2002.

[52] A. B. Nassif; L. F. Capretz et al. "Estimating software effort based on use case point model using sugeno fuzzy inference system". In: International conference on tools with artificial intelligence, 2011.

[53] A. B. Nassif; L. F. Capretz et al. "Estimating effort an ANN model based on use case points". In: International conference on machine learning and applications, 2012.

[54] R. Shukla; A. K. Misra "Estimating software maintenance effort – a neural network approach". In: India software engineering conference, 2008.

[55] M. Shepperd; C. Schofield "Estimating software project effort using analogies". Journal: IEEE transactions on software engineering, 1997.

[56] H. Sneed "Estimating the cost of a reengineering project". In: working conference on reverse engineering, 2005.

[57] F. Fioravanti; P. Nesi "Estimation and prediction metrics for adaptive maintenance effort of object-oriented systems". Journal: IEEE transactions on software engineering, 2001.

[58] T. Mukhopadhyay; S. Vicinanza et al. "Examining the feasibility of case-based reasoning model for software effort estimation". Journal: MIS quarterly: management information systems, 1992.

[59] P. V. Prasad Reddy; CH. V. M. Hari “Fuzzy based PSO for software effort estimation“. Journal: Communications in computer and information science, 2011.

[60] J. Lin; C. Chang “Genetic algorithm and support vector regression for software effort estimation“. Journal: Advanced materials research, 2011.

[61] T. R. Benala; S. Dehuri et al. “Genetic algorithm for optimizing functional link artificial neural network based software cost estimation“. Journal: Advances in intelligence and soft computing, 2012.

[62] H. K. Verma; V. Sharma “Handling imprecision in inputs using fuzzy logic to predict effort in software development“. In: International advance computing conference, 2010.

[63] J. Wen, S. Li et al. “Improve analogy-based software effort estimation using principal components analysis and correlation weighting“. In: Asia-pacific software engineering conference, 2009.

[64] C. Hsu; C. Huang “Improving effort estimation accuracy by weighted grey relational analysis during software development“. In: Asia-pacific software engineering conference, 2007.

[65] E. Mendes “Improving software effort estimation using an expert-centred approach“. Journal: Lecture notes in computer science, 2012.

[66] M. Tsunoda; A. Monden et al. “Incorporating expert judgment into regression models of software effort estimation“. In: Asia-pacific software engineering conference, 2012.

[67] V. K. Bartsiri, D. N. A. Jawawi et al. “Increasing the accuracy of software development effort estimation using projects clustering“. Journal: IET software, 2012.

[68] S. Huang; N. Chiu et al. “Integration of the grey relational analysis with genetic algorithm for software effort estimation“. Journal: European journal of scientific research, 2008.

[69] D. Wu; J. Li et al “Linear combination of multiple case-based reasoning with optimized weight for software effort estimation“. Journal: The journal of supercomputing, 2013.

[70] V. K. Bardsiri; D. N. A. Jawawi et al. “LMES: a localized multi-estimator model to estimate software development effort“. Journal: Engineering applications of artificial intelligence, 2013.

[71] N. Mittas; L. Angelis “LSEbA: least squares regression and estimation by analogy in a semi-parametric model for software cost estimation“. Journal: Empirical software engineering, 2010.

[72] B. Srinivasan; G. Martin “MONSET – a prototype software development estimating tool“. In: Symposium on assessment of quality software development tools, 1994.

[73] J. Choudhari; U. Suman "Phase wise effort estimation for software maintenance: an extended SMEEM model". In: ACM International conference proceeding series, 2012.

[74] F. Schnitzhofer; P. Schnitzhofer "Pocket estimator – a commercial solution to provide free parametric software estimation combining an expert and learning algorithm". In: EUROMICRO conference on software engineering and advanced applications, 2012.

[75] Y. Singh; A. Kaur et al. "Predicting software development effort using artificial neural network". Journal: International journal of software engineering and knowledge engineering, 2010.

[76] M. P. Basgalupp; R. C. Barros; D. D. Ruiz "Predicting software maintenance effort through evolutionary-based decision trees". In: ACM symposium on applied computing, 2012.

[77] M. Sadiq; F. Mariyam et al. "Prediction of software project effort using fuzzy logic". In: International conference on electronics computer technology, 2011.

[78] I. Attarzadeh; S. H. Ow "Proposing a new high performance model for software cost estimation". In: International conference on computer and electrical engineering, 2009.

[79] I. Attarzadeh, S. H. Ow "Proposing a novel artificial network prediction model to improve the precision of software effort estimation". Journal: Lecture notes of the institute for computer sciences, social-informatics and telecommunications engineering, 2012.

[80] I. Attarzadeh; A. Mehranzadeh et al. "Proposing an enhanced artificial neural network model to improve the accuracy in software effort estimation". In: International conference on computational intelligence, communications systems and networks, 2012.

[81] R. Gulezian "Reformulating and calibrating COCOMO". Journal: Journal of systems and software, 1991.

[82] J. Lin; C. Chang et al. "Research on software effort estimation combined with genetic algorithm and support vector regression". In: International symposium on computer science and society, 2011.

[83] M. Azzeh; D. Neagu et al. "Software effort estimation based on weighted fuzzy grey relational analysis". In: ACM International conference proceeding series, 2009.

[84] A. B. Nassif, L. F. Capretz et al. "Software effort estimation in the early stages of the software life cycle using a cascade correlation neural network model". In: ACIS International conference on software engineering, artificial intelligence, networking and parallel/distributed computing, 2012.

[85] D. R. Pai; K. S. McFall et al. "Software effort estimation using a neural network ensemble". Journal: Journal of software information systems, 2013.

[86] P. L. Braga; A. A. L. Oliveira et al. "Software effort estimation using machine learning techniques with robust confidence intervals". In: International conference on hybrid intelligence systems, 2007.

[87] T. R. Benala; R. Mall et al. "Software effort prediction using fuzzy clustering and functional link artificial neural networks". Journal: Lecture Notes in Computer Science, 2012.

[88] R. Setiono; K. Dejaeger et al. "Software effort prediction using regression rule extraction from neural networks". In: International conference on tools with artificial intelligence, 2010.

[89] T. R. Benala; S. Dehuri. "Software effort prediction using unsupervised learning (clustering) and functional link artificial neural networks". In: World congress on information communication technologies, 2005.

[90] M. P. Basgalupp, R. C. Barros et al. "Software effort prediction: a hyper-heuristic decision tree based approach". In: ACM Symposium on applied computing, 2013.

[91] B. Boehm "Software engineering economics". Journal: IEEE transactions on software engineering, 1984.

[92] A. Albrecht; J. Gaffney "Software function, source lines of code, and development effort prediction: a software science validation". Journal: IEEE transactions on software engineering, 1983.

[93] P. Sentas; L. Angelis "Software productivity and effort prediction with ordinal regression". Journal: Information software and technology, 2005.

[94] J. Gallego; D. Rodríguez et al. "Software project effort estimation based on multiple parametric models generated through data clustering". Journal: Journal of computer science and technology, 2007.

[95] Y. Shan; R. McKay et al. "Software project effort estimation using genetic programming". In: IEEE International conference on communications, circuits and systems, 2002.

[96] S. Kock; J. Mitlohner "Software project effort estimation with voting rules". Journal: Decision Support Systems, 2009.

[97] D. Caivano, F. Lanubile et al. "Software renewal process comprehension using dynamic effort estimation". In: International conference on software maintenance, 2001.

[98] M. T. Baldassarre; D. Caivano et al. "Software renewal projects estimation using dynamic calibration". In: International conference on software maintenance, 2003.

[99] M. T. Baldassarre; N. Boffoli et al. "SPEED: software project effort evaluator based on dynamic calibration". In: International conference on software maintenance, 2006.

[100] A. Tripathi; B. Kumar et al. "SRS based estimation of software maintenance effort". In: International conference on computer and communications technology, 2012.

[101] J. Choudahari; U. Suman "Story points based effort estimation model for software maintenance". Journal: Procedia Technology, 2012.

[102] N. Chiu; S. Huang “The adjusted-based software effort estimation based on similarity distances“. Journal: Journal of systems and software, 2006.

[103] M. O. Saliu; M. Ahmed et al. “Towards adaptive soft computing based software effort prediction“. In: Annual conference on the north American fuzzy information processing society, 2004.

[104] M. Kassad; M. Daneva “Towards an early software estimation using log-linear regression and a multilayer perceptron model“. Journal: Journal of Systems and Software, 2013.

[105] K. T. Wei; M. H. Selamat et al. “Towards integration of FP weights into CP class complexity evaluation for effort estimation“. In: International conference on new trends in information science and service, 2011.

[106] D. Balbín; M. Ocrosopoma et al. “TUPUX: an estimation tool for incremental software development projects“. In: International e-conference on advanced science and technology, 2009.

[107] M. R. Braz; S. R. Vergilio “Using fuzzy theory for effort estimation of object-oriented software“. In: International conference on tools with artificial intelligence, 2004.

[108] A. Corazza; S. Di Martino et al. “Using tabu search to configure support vector regression for effort estimation“. Journal: Empirical software engineering, 2013.

[109] E. Mendes; S. Counsell “Web development effort using analogy“. In: Software Engineering Conference, 2000.

## APÊNDICE B – PROTOCOLO PARA ESTUDO DE CAMPO: ESTIMATIVA DE ESFORÇO EM PROJETOS DE REENGENHARIA DE SOFTWARE

### B.1 OBJETIVO DO ESTUDO DE CAMPO

O objetivo principal desta pesquisa é identificar os fatores relacionados a estimativa de esforço em projetos de reengenharia de software, presentes nas organizações onde serão desenvolvidos os estudos de campo (Organizações A-E). Para alcançar tal objetivo serão realizadas entrevistas semiestruturadas com responsáveis por realizar estimativa de esforço em projetos de reengenharia de software, atuantes na indústria, para identificar como se dá esta atividade nas organizações em que eles atuam.

### B.2 ORGANIZAÇÃO DO PROTOCOLO

O protocolo será organizado como segue:

#### B.2.1 PROCEDIMENTOS

Tabela B.1 – Levantamento das questões e Estruturação do Guia para Entrevista

Participante:	Silvia Nunes
Período:	Janeiro-Fevereiro de 2014

Tabela B.2 – Reuniões para Revisão do Guia para a Entrevista

Participante:	Silvia Nunes e Duncan Ruiz
Período:	Fevereiro de 2014
Forma:	Reunião Presencial

#### B.2.2 ESCOLHA DAS PESSOAS ENTREVISTADAS

Gerentes de Projeto de Software, Líderes de Projeto, ou profissionais afins, que realizem a atividade de estimativa de esforço em projetos de reengenharia de sistemas.

Tabela B.3 – Validação de Face e Conteúdo do Guia para Entrevista

<b>Participantes</b>	<b>Data</b>
Sabrina Marczak (Professora e Pesquisadora da PUCRS/ Especialista)	25 de Março de 2014
Alessandra Dutra (Gerente de Projetos do Centro de Inovação da Microsoft, Professora da PUCRS e FGV/ Especialista)	28 de Março de 2014
Rafael Prikkladnick (Diretor do TECNOPUC, Professor da PUCRS/ Especialista)	8 de Abril de 2014

Tabela B.4 – Pré-Teste (Entrevista Piloto)

Participante:	Marcelo Gomes (Gerente de Projetos HP)
Período:	14 de Abril de 2014
Local:	FACIN-PUCRS

Tabela B.5 – Autorização das Empresas Participantes (via e-mail)

<b>Participantes</b>	<b>Período</b>
Contato 1 (Organização A)	16-24 de Abril de 2014
Contato 2 (Organização B)	2-7 de Maio de 2014
Contato 3 (Organização C)	5-19 de Maio de 2014
Contato 4 (Organização D)	7-21 de Maio de 2014
Contato 5 (Organização E)	6-11 de Junho de 2014

Tabela B.6 – Aplicação das Entrevistas

<b>Participantes</b>	<b>Data</b>	<b>Local</b>
Organização A: Consultor de Tecnologia	28 de Abril de 2014	Sede da empresa, Porto Alegre/RS
Organização A: Gerente de Projetos	9 de Maio de 2014	Sede da empresa, Porto Alegre/RS
Organização B: Gerente de Métricas	8 de Maio de 2014	Sede da empresa, Porto Alegre/RS
Organização B: Gerente de Métricas	23 de Maio de 2014	Sede da empresa, Porto Alegre/RS
Organização D: Coordenadores da Fábrica de Software	23 de Maio de 2014	Sede da empresa, Porto Alegre/RS
Organização E: Gerente de Projetos	17 de Junho de 2014	Sede da empresa, Porto Alegre/RS

### B.2.3 RECURSOS UTILIZADOS

#### Sistema Computacional

- Sistema MaxQDA (tabulação de dados)

#### Recursos Materiais

- Gravador para gravar as entrevistas
- Salas de reuniões
- Computador pessoal

#### B.2.4 QUESTÕES DE PESQUISA

- O que são projetos de reengenharia de sistemas no contexto da organização?
- Como é realizada a estimativa de esforço em projetos de reengenharia de sistemas legados?
- Quais as técnicas aplicadas na realização de estimativa de esforço em projetos de reengenharia de sistemas legados?
- Quais as vantagens e desvantagens das técnicas atualmente aplicadas?

#### B.2.5 METODOLOGIA

A pesquisa foi conduzida a partir da aplicação de entrevistas presenciais, com a utilização de um roteiro semiestruturado. As entrevistas foram gravadas e o áudio foi transcrito. Em uma segunda etapa foi realizada a análise dos dados coletados durante as entrevistas. Esta análise foi baseada no método de comparação constante [SEA08].

#### B.2.6 VALIDAÇÃO DO ROTEIRO

Objetivos da Validação:

- Verificar se as perguntas são compreensíveis.
- Avaliar as prováveis respostas e a eficácia do instrumento.
- Avaliar a confiabilidade e a validade do instrumento.
- Garantir que as técnicas de análise de dados combinam com as nossas respostas esperadas.

Processo:

- Julgamento de Especialistas:
- Entrevista Piloto.

### B.2.7 SELEÇÃO DOS PARTICIPANTES DA PESQUISA

- E-mail para o responsável pela organização/setor;
- E-mail os gerentes;
- Envio do Termo de Confidencialidade.

### B.2.8 APLICAÇÃO DA ENTREVISTA

- Aplicação do roteiro de entrevista;
- Gravação de áudio ou realização de anotações;

### B.2.9 ANÁLISE

- Transcrição das Entrevistas;
- Método: Método de Comparação Constante [SEA08];
- Utilização da Ferramenta MaxQda.

### B.2.10 ROTEIRO DE ENTREVISTA: FASE I

#### Informações Gerais

- Nome;
- Cargo;
- Organização;
- E-mail;
- Tempo no cargo.

## Sobre projetos de reengenharia

1. O que são projetos de reengenharia para você? No que eles diferem dos projetos de manutenção?
2. Quantos projetos de reengenharia de sistemas estão sendo gerenciados por você neste momento?
3. Quais as características principais dos projetos de reengenharia desenvolvidos aqui? (tipos de sistemas, nível de documentação, linguagem do sistema legado, linguagem do sistema novo, porte dos projetos (conceito de porte no contexto da organização em questão)?
4. Qual (ais) o (s) modelo (s) de processo de reengenharia de sistemas utilizado (s)? (big-bang, modular) a. Se houver mais de um, em que contexto cada um é utilizado?

## Sobre estimativa de esforço em projetos de reengenharia de sistemas

5. Em que (quais) momento (s) do ciclo de desenvolvimento é realizada a estimativa de esforço em projetos de reengenharia?
6. Como é realizada (envolvidos, técnicas, apoio ferramental)?
  - Envolvidos;
  - Técnica aplicada;
  - Métricas/Parâmetros;
  - Documentos consumidos e gerados;
  - Ferramentas utilizadas.
7. Qual são as principais vantagens na utilização da técnica atualmente aplicada (quão eficiente é)?
8. E as desvantagens? Na sua opinião, como estas desvantagens podem ser contornadas?
9. Gostaria de utilizar outra técnica? Se sim: Porque não usa? Qual técnica?
10. Além das informações fornecidas acima, você gostaria de acrescentar algum outro fator relevante a realização de estimativa de esforço em projeto de reengenharia na empresa em que você atua?

## B.2.11 ROTEIRO DE ENTREVISTA: FASE II

## Requisitos em Reengenharia

1. Como é feita a especificação (análise) de requisitos para projetos de reengenharia de software?

- Etapas;
- Envolvidos;
- Técnicas aplicadas;
- Ferramentas utilizadas.

Caso não seja mencionado especificamente a Engenharia Reversa como sendo utilizada, questionar o por que.

2. Como é estimado o esforço nesta fase?

- Envolvidos;
- Técnicas aplicadas;
- Métricas/Parâmetros;
- Documentos consumidos e gerados;
- Ferramentas utilizadas.

Caso não seja estimado esforço para esta fase:

- Por quê não é feito?
- Como justificam/negociam o esforço gasto nesta fase junto as partes interessadas pelo sistema?

3. Quanto o esforço aplicado nesta fase impacta no esforço total do projeto?

4. Existe alguma dificuldade na realização de estimativa de esforço nesta fase?

Qual (is)?

Caso exista, o quais as ações tomadas para contornar estas dificuldades?

5. Além das informações fornecidas acima, você gostaria de acrescentar algum outro fator relevante a realização de estimativa de esforço em projetos de reengenharia na empresa em que você atua?

## APÊNDICE C – PROTOCOLO PARA ESTUDO DE CASO: ESTIMATIVA DE ESFORÇO EM PROJETOS DE REENGENHARIA DE SOFTWARE

### C.1 VISÃO GERAL DO ESTUDO DE CASO

O objetivo principal deste estudo é validar e ampliar os fatores relacionados a estimativa de esforço em projetos de reengenharia de software, identificados no estudo de campo inicial. Uma vez que deseja-se identificar tais variáveis em seu contexto real de ocorrência, ou seja, na indústria de desenvolvimento de software, a estratégia de pesquisa escolhida foi o Estudo de Caso, de acordo com o método de Yin (2010).

O objetivo deste protocolo é apresentar o instrumento de coleta de dados, bem como os procedimentos e as regras gerais que deveriam ser seguidas ao utilizar esse instrumento (Yin 2010). O protocolo é uma das táticas principais para aumentar a confiabilidade da pesquisa de estudo de caso e destina-se a orientar o pesquisador ao realizar este estudo.

### C.2 ORGANIZAÇÃO DO PROTOCOLO

O protocolo será organizado como segue:

#### C.2.1 PROCEDIMENTOS

Tabela C.1 – Levantamento das questões e Estruturação do Guia para Entrevista

Participante:	Silvia Nunes
Período:	Agosto de 2014

Tabela C.2 – Reuniões para Revisão do Guia para a Entrevista

Participante:	Silvia Nunes e Duncan Ruiz
Período:	Agosto/Setembro 2014
Forma:	Reunião Presencial

Tabela C.3 – Validação de Face e Conteúdo do Guia para Entrevista

<b>Participantes</b>	<b>Data</b>
Alessandra Dutra (Gerente de Projetos do Centro de Inovação da Microsoft, Professora da PUCRS e FGV/ Especialista)	29 de Setembro de 2014
Rafael Prikladnick (Diretor do TECNOPUC, Professor da PUCRS/ Especialista)	2 de Outubro de 2014

Tabela C.4 – Pré-Teste (Entrevista Piloto)

Participante:	Marcelo Gomes (Gerente de Projetos HP)
Período:	15 de Outubro de 2014
Forma:	FACIN-PUCRS

Tabela C.5 – Autorização da Empresa Participante

<b>Etapas</b>	<b>Período</b>
Contato 1 (Organização A) - Via e-mail	Setembro-Outubro de 2014
Reunião de Apresentação do Estudo de Caso	3 de Outubro de 2014
Ajuste das Agendas	3-20 de Outubro de 2014

Tabela C.6 – Imersão na Organização das Entrevistas

<b>Data</b>	<b>Atividade</b>
22 de Outubro de 2014	Coleta de dados do projeto 1
23 de Outubro de 2014	Coleta de dados do projeto 1
24 de Outubro de 2014	Coleta de dados organizacionais
27 de Outubro de 2014	Coleta de dados do projeto 2
28 de Outubro de 2014	Coleta de dados do projeto 2
29 de Outubro de 2014	Coleta de dados do projeto 2
30 de Outubro de 2014	Coleta de dados do projeto 3
31 de Outubro de 2014	Coleta de dados do projeto 3

## C.2.2 ESCOLHA DAS PESSOAS ENTREVISTADAS

Responsáveis direta e indiretamente pela realização de estimativa de esforço nos projetos de reengenharia de software selecionados como unidades de análise.

## C.2.3 RECURSOS UTILIZADOS

### Sistema Computacional

- Sistema MaxQDA (tabulação de dados)

### Recursos Materiais

Tabela C.7 – Aplicação das Entrevistas

<b>Participantes</b>	<b>Data</b>	<b>Local</b>
Consultor de Tecnologia	7 de Novembro de 2014	Sede da empresa, Porto Alegre/RS
Consultor de Tecnologia	18 de Novembro de 2014	Sede da empresa, Porto Alegre/RS
Gerente de Projetos	19 de Novembro de 2014	Sede da empresa, Porto Alegre/RS
Gerente de Projetos	21 de Novembro de 2014	Sede da empresa, Porto Alegre/RS
Analista de Sistemas	25 de Novembro de 2014	Sede da empresa, Porto Alegre/RS
Arquiteto	25 de Novembro de 2014	Sede da empresa, Porto Alegre/RS

- Gravador para gravar as entrevistas
- Salas de reuniões
- Computador pessoal

#### C.2.4 DIMENSÕES, QUESTÕES E PROVÁVEL FONTE DE COLETA DE DADOS DO GUIA PARA ENTREVISTA SEMI-ESTRUTURADA

Tabela C.8 – Dimensões, Questões e Provável Fonte de Coleta de Dados

<b>Dimensão</b>	<b>Questões</b>	<b>Provável Fonte de Coleta de Dados</b>
Dados Demográficos	Nome, Escolaridade, Tempo de Experiência Profissional, Tempo de Experiência no Cargo	Entrevistas
Projeto de Reengenharia	Quais as características principais deste projeto de reengenharia? (tipo de sistema, nível de documentação, porte do projeto)?	Entrevistas
Estimativa de Esforço em Reengenharia	Qual (ais) atividade(s) de estimativa de esforço você realizou no projeto? (estimar, calibrar, monitorar, etc)?	Entrevista, Análise de Documentação
-	Em que momento do projeto esta (s) atividade (s) foi (ram) realizada (s)?	Entrevista
-	Como foi (ram) realizada (s)? (envolvidos, técnica aplicada, métricas/parâmetros, documentos consumidos e gerados, ferramentas utilizadas).	Entrevista, Análise de documentação
Estimativa de Esforço em Engenharia Reversa	O fato de existir um sistema legado como ponto de partida para o projeto influenciou as estimativas? Como?	Entrevista
Fatores de Impacto	Existiu algum fator relacionado a equipe do projeto que afetou a estimativa de esforço? Qual (ais)?	Entrevista
-	Existiu algum fator relacionado ao cliente que afetou a estimativa de esforço? Qual (ais)?	Entrevista
Dificuldades	Do seu ponto de vista, quais são as principais dificuldades enfrentadas acerca de estimativa de esforço?	Entrevista
-	Em quais atividades no processo de reengenharia cada dificuldade foi encontrada?	Entrevista
Boas Práticas e Soluções	Para cada dificuldade encontrada, qual foi a solução adotada?	Entrevista
-	Como esta solução foi identificada?	Entrevista
Satisfação com o processo de estimativa	Você considera que a estimativa de esforço para este projeto foi realizada foi satisfatória? Se sim, quais os fatores que levaram a isto? Senão, por quê? O que deu errado?	Entrevista
Observação	Além das informações fornecidas acima, você gostaria de acrescentar algum outro fator relevante a realização de estimativa de esforço em projeto de reengenharia?	Entrevista