

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

**MODELO–E10: UM MODELO PARA ESTIMATIVAS DE ESFORÇO
EM MANUTENÇÃO DE SOFTWARE**

NELSON NUNES TENÓRIO JUNIOR

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

Orientador: Prof. Dr. Duncan Dubugras Alcoba Ruiz
Coorientador: Prof. Dr. Marcelo Blois Ribeiro

**Porto Alegre
2010**

Dados Internacionais de Catalogação na Publicação (CIP)

T312m	Tenório Junior, Nelson Nunes Modelo – E10 : um modelo para estimativas de esforço em manutenção de software / Nelson Nunes Tenório Junior. – Porto Alegre, 2010. 134 p. Tese (Doutorado) – Fac. de Informática, PUCRS. Orientador: Prof. Dr. Duncan Dubugras Alcoba Ruiz. 1. Informática. 2. Engenharia de Software. 3. Software – Manutenção. I. Ruiz, Duncan Dubugras Alcoba. II. Título. CDD 005.1
-------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

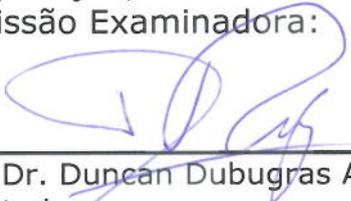
**Ficha Catalográfica elaborada pelo
Setor de Tratamento da Informação da BC-PUCRS**



Pontifícia Universidade Católica do Rio Grande do Sul
FACULDADE DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

TERMO DE APRESENTAÇÃO DE TESE DE DOUTORADO

Tese intitulada "Modelo-E10: Um Modelo para Estimativas de Esforço em Manutenção de Software", apresentada por Nelson Nunes Tenório Junior, como parte dos requisitos para obtenção do grau de Doutor em Ciência da Computação, Sistemas de Informação, aprovada em 30/03/2010 pela Comissão Examinadora:


Prof. Dr. Duncan Dubugras Alcoba Ruiz -
Orientador

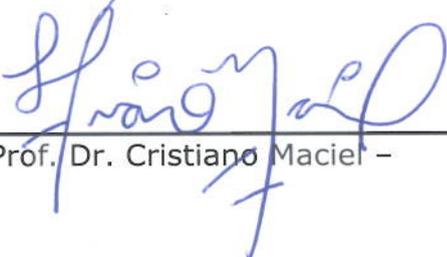
PPGCC/PUCRS


Prof. Dr. Marcelo Blois Ribeiro -
Co-orientador

PPGCC/PUCRS


Prof. Dr. Ricardo Melo Basto -

PPGCC/PUCRS


Prof. Dr. Cristiano Maciel -

UFMT


Profa. Dra. Renata Galante -

UFRGS

Homologada em 18/05/10, conforme Ata No. 08/10 pela Comissão Coordenadora.


Prof. Dr. Fernando Gehr Moraes
Coordenador.

PUCRS

Campus Central

Av. Ipiranga, 6681 - P. 32 - sala 507 - CEP: 90619-900

Fone: (51) 3320-3611 - Fax (51) 3320-3621

E-mail: ppgcc@pucrs.br

www.pucrs.br/facin/pos

*“Esta tese de doutorado é inteiramente dedicada
ao meu amado filho **João Batista Rosa Neto.**”*

Nelson Tenório Jr.

AGRADECIMENTOS

À minha família: meu pai Nelson Tenório (pela carinhosa revisão do português e constante apoio), minha mãe Célia Tenório e meu irmão Crístian Tenório (pela motivação a perseverar em meio às dificuldades); avós do meu filho João Rosa e Helinéa Strobel (pelo apoio às viagens para eu ver meu filho e motivação); mãe do meu filho Marielle Strobel (pela paciência e assistência ao meu filho durante esses anos de doutorado); meu primo Vinícius Nakamura (por ouvir meus desabaços e me motivar); a todos os familiares que entenderam a importância dessa jornada.

Ao meu orientador Duncan Dubugras Alcoba Ruiz e ao meu coorientador Marcelo Blois Ribeiro (pela paciência, pelos ensinamentos, por terem acreditado em mim e abraçado essa causa); à professora Karin Becker (por ter iniciado essa jornada comigo e acreditado que eu poderia chegar até aqui).

Aos meus amigos, em especial: Ana Winck (pela amizade e convivência); Crístian Tristão e Fernanda Figueira (pela companhia nos dois primeiros e solitários anos de doutorado); Christian Quevedo (pelas críticas à parte gráfica); Fernando Castilho e Eliane (pela conversa franca, jantares e por me ensinarem a apreciar uma boa cachaça); Juliano Kunzler (pelo grande apoio na HP e por me proporcionar bons momentos nas crises pessoais); Luciano Bloomberg (pelas discussões sobre a mídia); Lúcio Rogério (pela amizade e força para eu seguir até o fim); Márcio Basgalupp (pela acolhida e amizade de eterna "chinelagem"); Rita Berardi (pela grande amizade, motivação e por me emprestar o ouvido para eu chorar minhas mágoas); Rodrigo Barros (pelos bons conselhos, boas discussões e companhia na academia); Patrícia Hübler (pela ajuda, motivação e apoio); Patrícia Silveira (pela boa convivência na HP); Tiago Silva e Thyago Borges (pela paciência em conviverem comigo e pelos coerentes conselhos); Wilson Xavier (pela pré-apresentação da tese).

À HP: Paulo Priante (por comprar a ideia das estimativas); Taisa Novelo (pela boa convivência e apoio); Ricardo Branco (pelo apoio); Giancarlo Montanari (pela amizade e apoio).

MODELO–E10: UM MODELO PARA ESTIMATIVAS DE ESFORÇO EM MANUTENÇÃO DE SOFTWARE

RESUMO

A indústria de software investe montantes de dinheiro em busca da melhoria de seus produtos e serviços. Atualmente, os clientes estão cada vez mais exigentes e procuram fábricas que possuam um rigoroso controle de qualidade. A qualidade é medida por meio dos modelos de qualidades adotados, como CMMI, ISO, IEC, dentre outros. A exigência dos clientes faz as organizações selarem contratos que medem o nível da qualidade de seus serviços. Esses contratos prevêem como sanções pesadas multas.

O cumprimento de prazos e custos é uma das exigências previstas nesse tipo de contrato. A base para que se cumpram prazos e custos é a estimativa de esforço de desenvolvimento e, após a entrega do produto, o esforço para a manutenção do software. O desafio dos pesquisadores está em apoiar a indústria de software com trabalhos na área de estimativas que possam tornar as mesmas mais precisas. Além disso, as estimativas não devem onerar as equipes de software para não agregar mais custos aos projetos.

Esta tese é fruto de uma pesquisa realizada ao longo de quatro anos, dois dos quais atuando em um projeto de manutenção de software pertencente a um grande banco governamental brasileiro. O projeto era mantido por uma fábrica de software parceira do Programa de Pós-Graduação em Ciência da Computação da Faculdade de Informática da PUCRS. A pesquisa inspirou esta tese de doutorado que apresenta, como uma opção para a área de estimativas, um modelo de esforço em manutenção de software. O modelo aqui apresentado apoia tanto a indústria de software como as pesquisas na área de estimativas.

Palavras-chave: Estimativas de Esforço; Manutenção de Software; Modelos Preditivos; Técnicas de Estimativas; Modelo–E10.

E10–MODEL: A MODEL TO EFFORT ESTIMATION ON SOFTWARE MAINTENANCE

ABSTRACT

The software industry invests amounts of money searching for the improvements on its products and services. Nowadays, clients are more demanding and look for industries that possess a severe quality control. That quality is measured through quality models adopted, such as CMMI, ISO, IEC, among others. The client's demanding makes the organizations contract patterns that evaluate the quality level of its services. These contracts predict as punishments expensive fees.

Observing of costs and deadlines is one of the requirements predicted in such contracts. The basis to its fulfillment (costs and deadlines), is the software development effort estimation and, after the delivery of the product, is the software maintenance effort. The challenge for researchers lies on supporting the software industry with effective works on estimation areas that may become much more accurate. Besides, the estimations shall not be economically heavy to their teams in order not to add more costs to the projects.

This thesis comes from a research that has been developed through the last four years, from which two of them were based on the actions in a software's maintenance project from a great Brazilian public bank. The project was supported by a software industry partner of the Post graduation program in computer science from PUCRS. Such program inspired this thesis that presents as an option to the estimation areas a model of software's maintenance effort. The model here presented supports the software industry as well the researches in the estimation area.

Keywords: Effort Estimation; Software Maintenance; Predictive Models; Estimation Techniques.

LISTA DE FIGURAS

Figura 2.1	Árvore de regressão para estimativas de tempo de desenvolvimento. Adaptação de [GM97].	35
Figura 2.2	Exemplo de gráfico de dispersão com reta de regressão de mínimos quadrados [Moo04].	36
Figura 4.1	Processo de software do projeto P1.	56
Figura 4.2	Estrutura do estudo de caso exploratório.	58
Figura 4.3	Exemplo dos agrupamentos encontrados entre as técnicas de estimativas aplicadas.	66
Figura 4.4	Exemplo dos agrupamentos encontrados para a aplicação da quarta regra.	68
Figura 5.1	Modelo–E10 para estimativas de esforço.	82
Figura 5.2	Exemplo da Retroalimentação.	84
Figura 5.3	Exemplo da aplicação da técnica escolhida.	85
Figura 6.1	Processo de experimentação. Fonte: [HOR ⁺ 00].	93
Figura 6.2	Processo de execução do experimento.	98
Figura 6.3	Histograma da distribuição de esforço real.	99
Figura 6.4	Box plot com os <i>outliers</i> do esforço real.	101
Figura 6.5	Resumo analítico trimestral.	104
Figura A.1	Estrutura do estudo de caso exploratório.	118
Figura A.2	Processo de software do projeto P1.	120
Figura A.3	Cronograma do Estudo de Caso Exploratório.	121

LISTA DE TABELAS

Tabela 2.1	Valores da constante B na Equação do Software. Fonte: [Hod96].	30
Tabela 2.2	Resumo analítico das propostas do estado da arte das estimativas de esforço para atender a manutenção de software.	39
Tabela 3.1	Extração das informações gerais dos estudos selecionados.	46
Tabela 3.2	Distribuição de frequência (absoluta e relativa) do aspecto "Classificação da Solução".	47
Tabela 3.3	Distribuição de frequência (absoluta e relativa) do aspecto "Método de Pesquisa".	47
Tabela 3.4	Distribuição de frequência (absoluta e relativa) do aspecto "Foco da Contribuição".	48
Tabela 3.5	Extração das informações relativas ao contexto das estimativas de ETMS.	48
Tabela 3.6	Extração das informações relativas ao contexto das estimativas de ETMS.	49
Tabela 3.7	Distribuição de frequência (absoluta e relativa) do aspecto "Domínio".	49
Tabela 3.8	Distribuição de frequência (absoluta e relativa) do aspecto "Representação".	49
Tabela 3.9	Distribuição de frequência (absoluta e relativa) do aspecto "Base Histórica".	50
Tabela 3.10	Distribuição de frequência (absoluta e relativa) do aspecto "Combinação".	50
Tabela 3.11	Distribuição de frequência (absoluta e relativa) do aspecto "Ao Menos 3.".	50
Tabela 3.12	Distribuição de frequência (absoluta e relativa) do aspecto "AvaliaçãoMP".	51
Tabela 3.13	Distribuição de frequência (absoluta e relativa) do aspecto "Calibragem".	51
Tabela 4.1	Hipóteses para o estudo de caso exploratório.	59
Tabela 4.2	Modelos, hipóteses e expectativas.	61
Tabela 4.3	Escore da regressão por FP vs. sugestão das estimativas.	62
Tabela 4.4	Escore da regressão por Quantidade de Documentos vs. <i>Expertise</i>	62
Tabela 4.5	Escore da regressão Múltipla (FP e Quantidade de Documentos) vs. <i>Expertise</i>	62
Tabela 4.6	Escore da Regressão Linear, Quantidade de Documento por fase vs. <i>Expertise</i> por fase.	63
Tabela 4.7	Validação das hipóteses para as estimativas de ETMS na fase 1.	64
Tabela 4.8	Técnicas para exemplificar agrupamentos para estimar o ETMS.	65
Tabela 4.9	Exemplo de nova técnica nos agrupamentos para estimar o ETMS.	66
Tabela 4.10	Exemplo da aplicação da terceira regra.	67
Tabela 4.11	Exemplo de valores de esforço para aplicar a quarta regra.	67
Tabela 4.12	Exemplo da aplicação da quarta regra.	68
Tabela 4.13	Continuação da aplicação do quarta regra.	68
Tabela 4.14	Valores de esforço para aplicar a quinta regra.	69
Tabela 4.15	Exemplo da aplicação da quinta regra.	69
Tabela 4.16	Valores de esforço para aplicar a sexta regra.	70

Tabela 4.17	Resumo da aplicação das técnicas de estimativas de ETMS em dados do projeto P1.	71
Tabela 4.18	Escore final da combinação de técnicas de estimativas por fase.	71
Tabela 4.19	Validação das hipóteses para as estimativas de ETMS na Fase 2.	72
Tabela 4.20	Fatores de complexidade para contagem de documentos.	74
Tabela 4.21	Resultados obtidos com a simulação da complexidade de documentos.	74
Tabela 4.22	Validação da hipótese para estimar o valor de ETMS na fase 3.	75
Tabela 4.23	Resumo analítico das técnicas utilizadas no estudo de caso.	76
Tabela 5.1	Exemplo da calibragem de modelos preditivos.	85
Tabela 5.2	Sugestão de técnicas de estimativas para o Modelo–E10.	90
Tabela 5.3	O Modelo–E10 em relação às propostas na área de estimativas.	92
Tabela 6.1	Quantidade de demandas utilizadas no conjunto de testes.	101
Tabela 6.2	Teste de Normalidade <i>Shapiro–Wilk</i> para a variável Esforço Estimado.	102
Tabela 6.3	Teste de hipótese de <i>Wilcoxon</i>	102
Tabela 6.4	Resultado da avaliação do Modelo–E10 com $PRED(x)$ e $MMRE$	103
Tabela 6.5	Análise complementar das estimativas com RMSE.	104

LISTA DE SIGLAS

LOC	<i>Line of Codes</i>
FP	<i>Function Points</i>
UCP	<i>Use Case Points</i>
CMMI	<i>Capability Maturity Model Integration</i>
SLA	<i>Service Level Agreement</i>
ETMS	Esforço de Trabalho de Manutenção de Software
COCOMO	<i>Constructive Cost Model</i>
OAM	Orientado a Aprendizado de Máquina
ML	<i>Machine Learning</i>
BPMP	<i>Back Propagation Multilayer Perceptrons</i>
RBF	<i>Radial Basis Function</i>
SVR	<i>Support Vector Regression</i>
QA	<i>Quality Assurance</i>
OS	Ordem de Serviço
QtdeDoc	Quantidade de Documentos impactados por uma demanda.
MMRE	<i>Mean Magnitude Relative Error</i>
PRED	<i>Percentage Relative Error Deviation</i>
RMSE	<i>Root Mean Squared Error</i>

SUMÁRIO

List of Figures	13
List of Tables	15
List of Abbreviations	17
1. Introdução	23
1.1 Caracterização do Problema	24
1.2 Questão de Pesquisa	25
1.3 Objetivos	25
1.3.1 Objetivo Geral	25
1.3.2 Objetivos Específicos	25
1.4 Metodologia	26
1.5 Organização do Trabalho	26
2. Revisão da Literatura	27
2.1 Classificação das Técnicas de Estimativas	27
2.2 Empíricas e Compostas	28
2.3 Dinâmicas	29
2.4 Baseadas em <i>Expertise</i>	30
2.5 Orientadas a Aprendizado de Máquina	32
2.5.1 Redes Neurais	32
2.5.2 Sistemas Fuzzy	33
2.5.3 Raciocínio Baseado em Casos	33
2.5.4 Árvores de Decisão e de Regressão	34
2.6 Regressão Estatística	35
2.7 Análise das Propostas da Literatura	37
2.8 Considerações Finais do Capítulo	38
3. Revisão Sistemática para Estimativas de Esforço	41
3.1 Protocolo da Revisão	41
3.1.1 Formulação da Questão	41
3.1.2 Seleção das Fontes	42
3.1.3 Seleção dos Estudos	43

3.1.4	Extração de Informações	43
3.1.5	Sumarização dos Resultados	44
3.2	Considerações Finais do Capítulo	51
4.	Estudo de Caso Exploratório	55
4.1	Cenário	55
4.2	Requisitos do Projeto	56
4.3	Planejamento do Estudo de Caso	57
4.4	Definição das Hipóteses e Técnicas	58
4.5	Fase 1 – Regressão Linear Simples e Múltipla	60
4.5.1	Regressão Baseada em Esforço Total	60
4.5.2	Regressão Baseada em Esforço por Fase	63
4.5.3	Hipóteses Validadas com a Fase 1	63
4.6	Fase 2 – Combinação de Técnicas de Estimativas	63
4.6.1	Exemplo da Combinação de Técnicas de Estimativas	65
4.6.2	Resultados Alcançados	70
4.6.3	Hipóteses Validadas com a Fase 2	72
4.7	Fase 3 – Verificação da Complexidade em Documentos de Projeto	72
4.7.1	Classificação de Alterações em Documentos	72
4.7.2	Resultados Alcançados	74
4.7.3	Hipóteses Validadas com a Fase 3	75
4.8	Síntese dos Trabalhos Realizados	75
4.9	Considerações Finais do Estudo de Caso	77
5.	Modelo–E10	79
5.1	Requisitos para Utilização do Modelo–E10	79
5.2	Apresentação	80
5.2.1	Dados para Estimativas	83
5.2.2	Retroalimentação	83
5.2.3	Calibragem	84
5.2.4	Técnicas	85
5.2.5	Modelos Preditivos	86
5.2.6	Agrupamentos	86
5.2.7	Seleção	87
5.2.8	Estimativas	88
5.3	Crterios de Elegibilidade das Técnicas de Estimativas	88
5.4	Considerações Finais do Modelo–E10	91

6. Experimento com o Modelo–E10	93
6.1 Definição	93
6.1.1 Sumário da Definição	94
6.2 Planejamento	94
6.2.1 Contexto da Seleção	94
6.2.2 Formulação das Hipóteses	94
6.2.3 Seleção das Variáveis	95
6.2.4 Seleção da Amostra	95
6.2.5 Esboço do Experimento	96
6.2.6 Instrumentação	97
6.3 Operacionalização	97
6.3.1 Preparação	97
6.3.2 Execução	97
6.3.3 Validação dos Dados	98
6.4 Análise e Interpretação	99
6.4.1 Estatística Descritiva	99
6.4.2 Redução de Dados	100
6.4.3 Determinando o Tipo de Teste de Hipótese	100
6.4.4 Testando Hipóteses	102
6.4.5 Análises Complementares	102
6.5 Sumário e Conclusões	103
7. Conclusões	107
7.1 Contribuição Científica	107
7.2 Contribuição para o Mercado	108
7.3 Trabalhos Futuros	108
7.4 Considerações Finais	109
Bibliografia	111
A. Protocolo do Estudo de Caso Exploratório	117
A.1 Introdução ao Estudo de Caso e Objetivo do Protocolo	117
A.1.1 Questões do Estudo de Caso	117
A.1.2 Estrutura do Estudo de Caso	117
A.1.3 Hipóteses do Estudo de Caso Exploratório	118
A.2 Procedimentos de Coleta de Dados	119
A.3 A Prática na Fábrica de Software	119
A.4 Avaliação	119

A.5	Cronograma do Estudo de Caso Exploratório	120
B.	Proposta de Classificação de Documento para o COCOMO 2.0	123
C.	Complexidade de Documentos	127
C.1	Regra de contagem para Documento de <i>Design</i> (Projeto)	127
C.2	Regra de contagem para Relatórios	128
C.3	Regra de contagem de Lógica	129
C.4	Regra de contagem de OBMaker	130
C.5	Regra de contagem de MO	131
C.6	Regra de contagem de DEIG	132

1. Introdução

Esforço é provavelmente a mais popular métrica de software [LB06]. Ele é definido como a relação entre pessoa/hora, dia, semana, mês ou ano. Contudo, estimar o esforço é um dos principais desafios na área de estimativas de software. Para se estimar o esforço é necessário conhecer outras medidas do software, como por exemplo o tamanho do projeto [LB06]. No mundo tangível, o tamanho é a medida que representa o volume ou a massa [GE06]. No mundo intangível, neste caso software, o tamanho é representado por linhas de código ou funcionalidades [GE06]. As medidas mais comuns para determinarem o tamanho de um software são Linhas de Código (LOC), Pontos por Função (FP) ou Pontos por Caso de Uso (UCP). Essas medidas servem de base para a utilização de técnicas para as estimativas de esforço [GE06].

Estimar o esforço é uma atividade importante no desenvolvimento de software, pois ela é utilizada na composição de informações como custos e prazos, as quais são essenciais para o projeto [Som06] [GJ07]. Infelizmente, as técnicas de estimativas de esforço não produzem resultados satisfatórios, os erros nas estimativas parecem ser regra e não exceção [GJ07] [Dan08].

Em boa parte dos projetos de software, as estimativas de esforço são baseadas em experiências passadas, onde dados de projetos anteriores e similares são analisados para se estimar o esforço de um novo projeto. Todavia, estimativas utilizando esse tipo de técnica não se aplicam quando o novo projeto apresentar características diferentes dos anteriores [Pre09]. Outrossim, as organizações que não coletam e armazenam adequadamente os dados de seus projetos anteriores têm dificuldades em estimar o esforço de novos projetos. É por isso que, as organizações que utilizam modelos de qualidade, como o CMMI, SPICE e ISO, têm menos dificuldade para fazê-lo [SEI06] [ISO98] [IEE98] [Pan04].

Com o crescente uso de computadores e software emergiu a necessidade de manutenção de software nas organizações. A manutenção é um processo de modificação operacional no software, podendo ser: correção de erros, migração para uma nova tecnologia e plataforma, ou ainda, uma adaptação para que sejam levados em consideração novos requisitos. A fase de manutenção de software é a mais trabalhosa e custosa, especialmente no caso de grandes sistemas de software, que têm um ciclo de vida mais longo [KTB08]. Assim, para garantir o controle do processo de manutenção é igualmente necessário um planejamento que envolva as estimativas de tamanho, esforço e custos [SDD⁺02].

Existem várias técnicas para estimar o esforço no desenvolvimento de software. Em geral, para se estimar o esforço de manutenção de software, a indústria aplica as mesmas técnicas de estimativas de esforço utilizadas para seu desenvolvimento. Todavia, quando essas técnicas são utilizadas no contexto de manutenção, a precisão das estimativas tende a ficar comprometida. Isso porque, o valor das métricas na fase de desenvolvimento de software não são os mesmos da fase de manutenção. Por exemplo, o tamanho; uma das medidas base para as estimativas, pode ser distorcido pela reutilização de componentes, o que diminui o tamanho da manutenção à ser realizada [SDD⁺02].

As seções seguintes apresentam a caracterização do problema na área de estimativas de esforço, bem como a questão de pesquisa, seguida dos objetivos e da organização do trabalho.

1.1 Caracterização do Problema

No estado da arte das estimativas de esforço em projetos de software, além das poucas técnicas voltadas à manutenção de software, nota-se a utilização de técnicas de estimativas de esforço, específicas para o desenvolvimento de software, sendo igualmente utilizadas para estimar seu esforço de manutenção, como relata a Seção 2.7 do Capítulo 2. Dessa forma, podem aparecer distorções nos valores estimados, pois a manutenção de software possui algumas particularidades como: custo da alteração, tipo de manutenção, reutilização de código, dentre outros, que as técnicas de estimativas voltadas ao desenvolvimento não levam em consideração [SDD⁺02].

A revisão sistemática, apresentada no Capítulo 3, mostra o desafio de pesquisadores na busca de soluções para estimar o esforço de desenvolvimento de software com maior precisão. Também percebe-se nela a pequena quantidade de propostas específicas para manutenção de software, talvez por não ser dada a devida importância às características que a diferem do desenvolvimento de software [SDD⁺02]. Outrossim, nota-se que, a maior parte dos trabalhos na área negligenciam a utilização conjunta de mais de uma técnica para estimar de esforço, como sugerido por [LB06] [Som06] [Pre09].

Tipicamente, as técnicas disponíveis para estimativas de esforço se baseiam em experiências de projetos anteriores, por meio da coleta, armazenamento e análise de seus dados. Muitas vezes, essas técnicas são calibradas com as atualizações dos dados de projetos mais recentes. A calibragem é importante por tentar impedir a defasagem da técnica em relação à realidade dos projetos em termos de avanços tecnológicos, dentre os quais se destacam: linguagem de programação, bancos de dados e processos de desenvolvimento de software.

A calibragem das técnicas de estimativas não é uma tarefa trivial que possa ser realizada sem demandar um esforço significativo. Talvez, por esse motivo, as técnicas baseadas na *expertise*, como o *Planning Poker* e o *Delphi*, vêm se destacando nas estimativas de esforço na manutenção de software. Se por um lado as técnicas baseadas na *expertise* são boas pela simplicidade e praticidade, por outro, elas possuem três fatores que prontamente as deixam em posição de desvantagem. O primeiro deles é a imprecisão das estimativas, já que tais técnicas são muito dependentes da experiência das pessoas. O segundo fator é a dependência de conhecimento humano, já que os indivíduos mais experientes tornam-se essenciais para que se alcance boas estimativas e isso pode acarretar mais custo para o projeto com relação a contratar ou manter pessoas com o nível de experiência necessário. E por fim, a oneração dos indivíduos, já que as tais técnicas requerem reuniões periódicas quando são realizadas e debatidas as estimativas.

As desvantagens relacionadas à prática de técnicas baseadas na *expertise* foram constatadas no dia a dia de um projeto de manutenção de software. O projeto pertencia a um grande banco estatal brasileiro e envolvia um orçamento bastante representativo na empresa parceira, como relata

o Capítulo 4. Outrossim, o projeto era regido por um contrato de *Service Level Agreement* (SLA), e seu controle era bem rigoroso porque previa multas pesadas caso orçamentos e prazos não fossem devidamente cumpridos.

Por fim, a Seção 1.2 apresenta a questão de pesquisa na tentativa de suprir os problemas da área de estimativas aqui apresentados.

1.2 Questão de Pesquisa

Em meio às necessidades conhecidas da área de estimativas de esforço em manutenção de software, remete-se a seguinte questão de pesquisa: "Como estimar esforço de trabalho em manutenção de software, doravante denominado ETMS, de forma a alcançar boas aproximações em relação ao esforço real, sem depender de uma única técnica de estimativas e da *expertise* e, ainda, possibilitar a calibragem com dados de estimativas anteriores para que não ocorram defasagens dos modelos preditivos utilizados?", onde entende-se que boas aproximações é um valor percentual mínimo de erro nas estimativas, quando comparado com o esforço real, aceitável pela organização.

1.3 Objetivos

As sessões abaixo apresentam os objetivos traçados para composição desta tese.

1.3.1 Objetivo Geral

Esta tese propõe um modelo para a melhoria das estimativas de esforço de trabalho em um projeto de manutenção de uma fábrica de software, por meio da revisão da literatura, da revisão sistemática e de um estudo de caso exploratório.

1.3.2 Objetivos Específicos

Para atingir o objetivo geral e, ainda, responder a questão de pesquisa, os seguintes objetivos específicos foram elaborados:

- Explorar os trabalhos fundamentais da área de estimativas de esforço em projetos de software;
- Realizar uma revisão sistemática para avaliar trabalhos correlatos, a fim de situar a pesquisa em relação ao estado da arte na área;
- Realizar um Estudo de Caso Exploratório em um projeto de uma fábrica de software a fim de levantar as necessidades da prática das estimativas de esforço em manutenção de software;
- Propor um modelo para estimativas de esforço em manutenção de software que responda a questão de pesquisa apresentada na seção 1.2, atenda as necessidades da organização, identificadas no Estudo de Caso do Capítulo 4 e, ainda, esteja alinhado com a literatura seminal, Capítulo 2 e com as recentes pesquisas na área, Capítulo 3;

- Realizar experimentos aplicando o modelo proposto sobre os dados reais de um projeto a fim de rejeitar a hipótese nula e comprovar a hipótese alternativa formuladas para tal;
- Analisar o desempenho dos resultados obtidos com os experimentos, e tirar as devidas conclusões sobre o modelo proposto.

1.4 Metodologia

Para o desenvolvimento desta tese foram realizados: um estudo de caso exploratório, com o intuito de identificar as dificuldades da área de estimativas em um projeto de software de uma organização; uma revisão da literatura para identificar aspectos vistos como importantes nos trabalhos seminais da área de estimativas, onde tais aspectos serviram de base para o alinhamento do estudo de caso com a literatura; uma revisão sistemática para identificar o estado da arte das estimativas de manutenção de software segundo os aspectos identificados nos trabalhos seminais; uma proposta de modelo de estimativas alinhado com a literatura seminal, com o estado da arte e com um ambiente real de estimativas em manutenção de software (estudo de caso exploratório); por fim, um experimento com dados reais de um projeto para a verificação e análise do modelo proposto.

1.5 Organização do Trabalho

Este trabalho está organizado da seguinte forma: o Capítulo 1 faz uma breve introdução sobre as estimativas de esforço em desenvolvimento e na manutenção de software apresentando a caracterização do problema, a questão de pesquisa, os objetivos a serem alcançados para a conclusão da pesquisa e a organização do trabalho; o Capítulo 2 faz uma revisão da literatura básica na área de estimativas de esforço para projetos de software apresentando seus principais trabalhos; o Capítulo 3 faz uma revisão sistemática para avaliar trabalhos correlatos; O Capítulo 4 faz um relato dos trabalhos realizados na fábrica de software da empresa parceira do grupo de pesquisa do autor; o Capítulo 5 apresenta o Modelo–E10, concebido para estimar o ETMS; o Capítulo 6 apresenta um experimento e os resultados obtidos *in-vitro* com os dados reais de um projeto da empresa parceira para validar o Modelo–E10; o Capítulo 7 apresenta as considerações finais da pesquisa seguida da contribuição científica, da contribuição para o mercado, da sugestão de trabalhos futuros e das considerações finais; por fim, são apresentadas as referências bibliográficas utilizadas e os seguintes apêndices:

Apêndice A – Protocolo do Estudo de Caso Exploratório.

Apêndice B – Proposta de Classificação de Documento para o COCOMO 2.0.

Apêndice C – Complexidade de Documentos.

2. Revisão da Literatura

Este capítulo apresenta os conceitos básicos e os trabalhos seminais da área de estimativas de esforço em projetos de software. Algumas técnicas são amplamente utilizadas no mercado, e outros ainda estão apenas no meio científico.

2.1 Classificação das Técnicas de Estimativas

Esta pesquisa adota a taxonomia sugerida por [ABC00] 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, boa parte das técnicas existentes na literatura. Assim, as técnicas de estimativas são classificadas como se segue:

- **Empíricas e Compostas:** classe que engloba principalmente as técnicas tradicionais, como COCOMO 81 e o COCOMO 2 [BCS99], que vêm do acrônimo de Constructive Cost Model. São técnicas calibradas com dados de projetos de software anteriores e que, com a aplicação de diferentes equações, indicam valores estimados para esforço, prazo e custo;
- **Dinâmicas:** classe de técnicas que consideram as mudanças nas estimativas de esforço e custo com o passar do tempo da realização do projeto. Essas técnicas são geralmente representados por equações diferenciais de primeira ordem, como a Equação do Software proposta por Akiyama em [Aki71];
- **Baseadas em *expertise*:** classe de técnicas que foca na análise de especialistas, ou ainda, adota estratégias de dedução sobre uma base de dados históricos, onde regras de inferência ajudam na escolha dos melhores valores de estimativas;
- **Orientadas a Aprendizado de Máquina (OAM):** classe que se utiliza de técnicas das redes neurais, raciocínio baseado em casos, lógica *fuzzy* e árvores de decisão e regressão como estratégia para realização das estimativas;
- **Baseadas em Regressão:** classe das técnicas comumente utilizados, sendo muitas vezes incluídos dentro de técnicas maiores que utilizam a regressão como mais um fator para o cálculo das estimativas. Geralmente, técnicas pertencentes a esta classe se dividem em regressão simples e regressão robusta.

É importante salientar que alguns autores utilizam o termo "modelo de estimativas" [ABC00], outros utilizam o termo "técnicas de estimativas" [Moo04] [Som06]; já outros utilizam "abordagens de estimativas" [JS07]. Sobretudo, esta pesquisa adota o termo "técnica de estimativas" por entender que o termo "modelos de estimativas" é a abstração da aplicação das diferentes técnicas de estimativas, conforme [Moo04] [Som06], descartando assim o termo "abordagem".

2.2 Empíricas e Compostas

A primeira versão do COCOMO, o COCOMO 81, foi proposta pelo Dr. Barry Boehm e derivado da coleta de dados de 161 projetos de software, posteriormente calibrados por meio de um modelo estatístico bayesiano [SS97]. O COCOMO 81 foi considerado a técnica de estimativas de custo mais bem sucedido da década de 80. Porém, com o advento das linguagens orientadas a objetos e o desenvolvimento de novos processos de desenvolvimento de software, iniciaram-se esforços para a criação de uma técnica capaz de suprimir as dificuldades desta com relação a esses novos desafios.

O COCOMO é baseado em uma série de direcionadores de custos que auxiliam na precisão das estimativas. Em sua primeira versão é levado em consideração apenas o desenvolvimento de software. No COCOMO 2, versão mais evoluída da técnica, foi introduzido o suporte a componentes reusáveis, além do reconhecimento de diferentes abordagens de desenvolvimento de software como: prototipação, desenvolvimento pela composição de componentes e uso de linguagens de quarta geração [Pre09] [Som06].

O COCOMO 2, basicamente, é subdividido de acordo com a fase em que se situa o processo de desenvolvimento de software [Som06]:

1. *Application Composition Model (ACM)*: ou nível de prototipação, pois corresponde ao estágio de prototipação em um ciclo de vida espiral, onde busca-se avaliar potenciais problemas de alto risco, como interfaces de usuário, interação entre sistemas, desempenho do produto e tecnologia utilizada;
2. *Early Design Model (EDM)*: ou nível inicial de projeto, quando as exigências são conhecidas e as alternativas de arquitetura do software foram razoavelmente exploradas. Não é possível ainda a realização de estimativas com alto nível de detalhe, pois diversos conceitos operacionais ainda devem ser definidos;
3. *Post Architecture Model (PAM)*: ou nível pós-arquitetural, onde se trabalha com o real desenvolvimento e manutenção do produto de software.

O funcionamento básico do COCOMO 2, independentemente da sua subdivisão e se dá por meio da utilização de medidas de tamanho como FP ou LOC.

A utilização do COCOMO 2 está sujeita às seguintes limitações:

- os direcionadores de custo são baseados em tecnologias envolvidas nos projetos de software e a datação da sua última calibragem é do ano 2000. Novas tecnologias surgiram de lá para cá, o que sugere que a calibragem de seus direcionadores esteja desatualizada; e
- a alta complexidade envolvida em sua utilização devido ao grande número de variáveis externas e direcionadores de custo para o cálculo das estimativas, tornando-o pouco amigável [Som06].

Embora à primeira vista o COCOMO 2 não pareça ser muito amigável, existem diversas vantagens que devem ser consideradas quando da escolha de uma técnica de estimativas [Som06] [Pre09] [BCS99]:

- é bem documentado;
- é disponibilizado livremente, podendo ser incorporado em diferentes softwares para realização de estimativas;
- endereça questões como reutilização de código e é relativamente flexível para diferentes projetos existentes;
- independente do processo de desenvolvimento de software, embora baseado no processo espiral.

Em contraponto com o exposto acima, existem duas desvantagens críticas do COCOMO 2 [Som06] [Pre09] [BCS99]:

- pode ocorrer imprecisão nas estimativas caso os projetos a serem estimados não possuam características similares aos projetos utilizados na sua calibragem;
- alta complexidade devido à agregação de um número enorme de variáveis no cálculo das equações, tornando proibitiva sua adoção em projetos de pequeno porte;
- é dependente de calibragem, processo não menos complexo e que tem influência direta no resultado das estimativas.

2.3 Dinâmicas

A equação do software, proposta por Akiyama em [Aki71], é uma técnica dinâmico, derivado da curva de Putnam–Norden–Rayleigh [Som06], que assume uma distribuição específica de esforço sobre o projeto de desenvolvimento de software. A equação deriva de dados de produtividade de mais de quatro mil projetos de software atuais, e o valor estimado é dado pela Equação 2.1, onde: E é o esforço pessoa/mês ou pessoa/ano; t é a duração do projeto em meses ou ano; B é um fator de características especiais, apresentado na Tabela 2.1; e P é uma constante que representa a produtividade.

A constante P , da Equação 2.1, é o parâmetro de produtividade que indica:

- o processo de maturidade e gerenciamento de práticas;
- a utilização das boas práticas da engenharia de software;
- o nível de utilização das linguagens de programação;
- o estado do ambiente de software;
- a habilidades de experiência do time; e
- a complexidade da aplicação.

$$E = \left[\frac{LOC \times B^{0,333}}{P} \right]^3 \times \left(\frac{1}{t^4} \right) \quad (2.1)$$

Tabela 2.1: Valores da constante B na Equação do Software. Fonte: [Hod96].

Tamanho	Valor de B
5–15K	.16
20K	.18
30K	.28
40K	.34
50K	.37
>70K	.39

Os valores típicos podem ser $P = 2000$ para desenvolvimento de aplicações de tempo real; $P = 10000$ para sistemas de software em telecomunicações; $P = 28000$ para sistemas de software em negócios. O parâmetro de produtividade pode ser derivado de dados históricos locais de esforço de projetos anteriores. É importante observar que a equação tem dois parâmetros independentes; as estimativas baseadas em tamanho (LOC) e a duração em meses ou ano [Pre09] [Hod96].

Embora pareça simples, a utilização da equação apresenta algumas limitações, sendo elas: i) dificuldade das estimativas de tamanho; e ii) a aparente ausência de um método de calibragem de suas constantes, o que faz com que não sejam utilizados os dados atualizados de um projeto.

2.4 Baseadas em *Expertise*

A técnica de *expertise* do *Delphi* foi desenvolvido pela *Rand Corporation*, no final da década de 40, como forma de realizar predições de eventos futuros. Mais recentemente, tal abordagem vem sendo utilizada como forma de guiar um grupo determinado de indivíduos a um consenso sobre determinado assunto [ABC00].

O funcionamento do *Delphi* se dá através dos seguintes passos:

- o coordenador apresenta aos indivíduos especialistas informações relevantes sobre o projeto, bem como uma especificação inicial do mesmo;
- o coordenador promove uma reunião do grupo de indivíduos especialistas, quando são discutidas as estimativas;
- os especialistas preenchem formulários indicando suas estimativas individuais para o esforço total do projeto e do esforço total do desenvolvimento. Os valores indicados pelos especialistas estão contidos em um intervalo: valor limite inferior, valor provável e valor limite superior;
- o coordenador organiza um relatório contendo as estimativas do grupo e as estimativas individuais, tornando público este relatório;

- o coordenador convoca uma reunião durante a qual os especialistas discutem as estimativas atuais.

Esse processo é iterativo até que haja consenso entre os especialistas. As estimativas do grupo são calculadas como média ponderada das estimativas individuais, como mostra a Equação 2.2 [PP00]:

$$estimativas = \frac{limite_{inferior} + (4 \times estimativas_{provável}) + limite_{superior}}{6} \quad (2.2)$$

A variância das estimativas individuais é definida como apresenta a Equação 2.3 :

$$estimativas = \frac{limite_{superior} + limite_{inferior}}{6} \quad (2.3)$$

O *Delphi* mostra-se útil quando há necessidade de estimar valores sem o apoio de grandes volumes de dados empíricos [ABC00]. Embora limitado, 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. Informações adicionais da calibragem Bayesiana do COCOMO 2 podem ser encontradas em [BCS99].

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 define-se como *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 em uma série de *Fibonacci*, como apresenta a Equação 2.4.

$$F_{(n)} = \left\{ \begin{array}{ll} 0, & se \quad n = 0; \\ 1, & se \quad n = 1; \\ F(n-1) + F(n-2) & outros \quad casos. \end{array} \right\} \quad (2.4)$$

Cada número desta série numérica é escrito em cartas montando-se um baralho, utilizado durante a reunião de planejamento. Este baralho também possui duas cartas coringa, sendo uma composta por um ponto de interrogação, que quando utilizada especifica que o membro não consegue estimar a tarefa por motivo a ser explicado na reunião, e uma com a figura de uma xícara de café, que demonstra a necessidade de um descanso para a continuação da reunião.

O resultado final é uma série de *Fibonacci* que sempre se inicia com o valor 0 e 1 e onde os próximos números são formados pela soma dos dois números anteriores. Ex.: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...

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.

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.

2.5 Orientadas a Aprendizado de Máquina

As técnicas para as estimativas de software baseadas em Aprendizado de Máquina (ML–Machine Learning) , representam boas alternativas às técnicas tradicionais, as quais são baseadas em regressão, técnicas estatísticas, ou ainda, na *expertise* humana.

O método para a criação de modelos de ML 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].

Algumas diferentes abordagens de ML para o desenvolvimento de modelos preditivos de software são detalhadas nas seções: 2.5.1 Redes Neurais; 2.5.2 Sistemas Fuzzy; 2.5.3 Raciocínio Baseado em Casos; e 2.1 Árvores de Classificação e Regressão [GM97].

2.5.1 Redes Neurais

As redes *Backpropagation Trained Feed–Forward* ou, simplesmente, redes *backpropagation*, são comumente utilizadas quando o assunto é técnica de estimativas de métricas de software. Essas redes são implementadas através da seleção de uma arquitetura apropriada de neurônios, que inclui, dentre outras coisas, a definição de quantas camadas de neurônios são utilizadas, o número de neurônios em cada camada e como estes se conectam uns com os outros. É necessário, também, definir uma função de transferência, que mapeia a entrada recebida pelo neurônio e propaga a saída correspondente [GM97].

Assim como toda técnica de ML, a rede deve utilizar dados históricos para seu aprendizado. Através de um conjunto de treinamento, a rede aprende por meio do ajuste de seus pesos, de forma a minimizar a distância entre a saída estimada e a saída real. Esse processo ocorre iterativamente até que o poder de generalização da rede seja considerado ótimo.

Embora seja considerada uma técnica robusta para construção de modelos preditivos, [GM97] reforça o fato de que as redes neurais não estão imunes aos problemas que decaem sobre as técnicas estatísticas, como a existência de *outliers*, instâncias vistas como excepcionais à amostra, colinearidade entre as variáveis, valores incompletos ou perdidos. Outro problema identificado é o funcionamento dito caixa preta das redes, pois as mesmas não detalham informações de como chegaram ao resultado final. Esse comportamento é tido como crítico para situações em que se deseja fazer uma análise causal, entendendo e conhecendo as particularidades que levaram determinada métrica a atingir certo valor.

2.5.2 Sistemas Fuzzy

Basicamente, 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. A entrada do sistema *fuzzy* pode ser tanto numérica como linguística, assim como o mesmo pode ser dito para a saída [GM97].

No contexto de métricas de software, um sistema *fuzzy* pode ser entendido como o todo de três componentes básicos:

- Funções de mapeamento de domínio: funções que definem os intervalos de valores que farão parte de determinada categoria;
- Base de regras: conjunto de regras que podem ser obtidas através dos dados históricos do projeto ou da visão do especialista sobre o domínio, que realizará a conexão entre as funções de mapeamento de domínio de entrada para as funções de mapeamento de domínio de saída;
- Processo de *defuzzyficação*: como várias funções de mapeamento de domínio de saída são possíveis, deve haver um componente do sistema responsável pela definição de uma única categoria ou valor numérico de saída do sistema. Este componente é chamado de processo de *defuzzyficação*.

A vantagem da utilização de sistemas *fuzzy* para estimativas de software é 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, 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] [KKS94] [DBP93] [DML⁺04].

2.5.3 Raciocínio Baseado em Casos

O raciocínio baseado em casos é um método de armazenar observações (dados históricos do projeto, no contexto de estimativas), e utilizá-las quando da identificação de uma nova observação, através da recuperação das observações que mais se aproximam da mesma, de forma a permitir uma resolução satisfatória do novo problema [SF95].

Um sistema de raciocínio baseado em casos deve possuir os seguintes componentes:

- Pré-processador: responsável pela preparação dos dados de entrada do sistema;
- Função de similaridade: responsável pela identificação de casos similares;
- Módulo de predição: responsável pelo fornecimento do valor de saída estimado; e
- Atualizador de memória: responsável pela agregação do novo caso à base de casos, caso seja necessário.

A premissa básica deste método, no contexto de estimativas de software, é a simulação de como um especialista utiliza seu conhecimento de projetos anteriores para indicar os valores estimados de determinada métrica. O problema dessa técnica é que os projetos têm que ter a mesma característica para que as estimativas tenham uma boa proximidade do real. Há bons resultados dessa técnica nas estimativas de custos em projetos de desenvolvimento de software com característica *web hypermedia* [MMW02] [KKM⁺08].

2.5.4 Árvores de Decisão e de Regressão

Árvores de decisão 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. Existem vários métodos para a construção dessas árvores. Porém, estes são, em sua grande maioria, uma variação de uma mesma estratégia *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].

Uma árvore de decisão contém nos seus nodos folha, o valor do atributo que se pretende prever classificado como uma categoria (rótulo), enquanto uma árvore de regressão vai utilizar funções lineares em cada nodo folha para geração de um valor contínuo, dentro de um intervalo de valores possíveis.

Para que essas árvores sejam capazes de gerar um modelos preditivos satisfatórios e generalizáveis, é necessário um conjunto de dados para treinamento, que inclui o valor do atributo alvo, e um conjunto diferente para testes, onde o atributo alvo é omitido.

A Figura 2.1 mostra uma árvore de regressão utilizada para estimativas de tempo médio de desenvolvimento em um projeto de software [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 estimarem defeitos [GL97] [JKA⁺00] [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 [UW01]. Existem duas desvantagens em se utilizar essas árvores, a primeira delas está relacionada à facilidade de compreensão, onde há uma dependência do número de níveis e nodos da árvore e quanto mais níveis e nodos a árvore tiver, menos compreensível ela se torna. Outra desvantagem é que, 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 obtenha bons resultados com essa técnica.

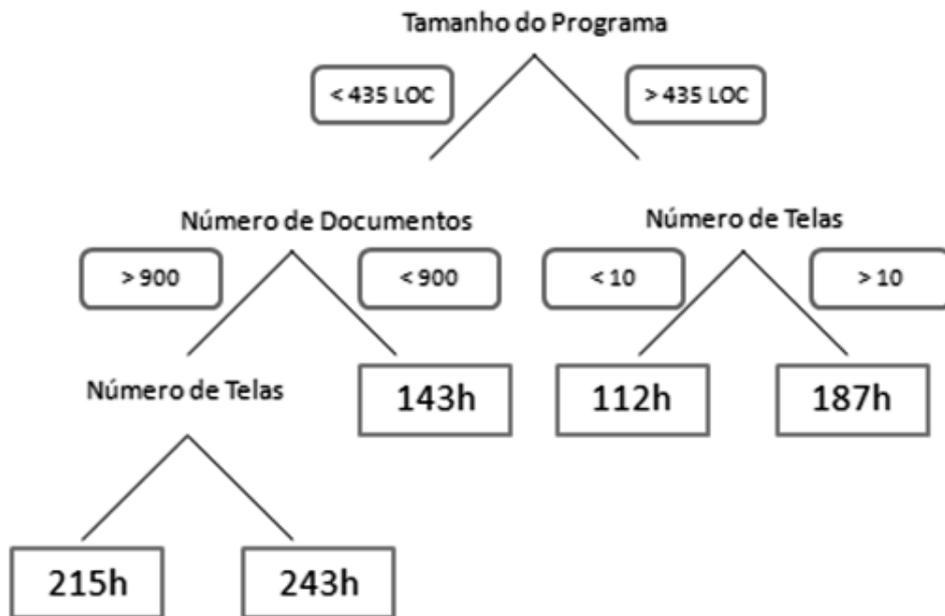


Figura 2.1: Árvore de regressão para estimativas de tempo de desenvolvimento. Adaptação de [GM97].

2.6 Regressão Estatística

Uma reta de regressão resume a relação entre duas variáveis, onde uma delas ajuda a explicar ou prever a outra. Como se tratam de duas variáveis, seu grau de correlação deve ser calculado e, quanto maior esse grau, maior é a influência da variável explicativa na variável resposta. Sendo assim, uma reta de regressão é uma linha, traçada sobre um gráfico de dispersão, que descreve uma variável de resposta y de acordo com o comportamento de uma variável explicativa x [Moo04].

Para padronizar a forma de visualização da regressão, usa-se uma reta de regressão de mínimos-quadrados, onde esta é uma reta que torna menor possível a soma dos quadrados das distâncias verticais dos pontos observados à reta. Para se achar uma reta de mínimos-quadrados faz-se necessário encontrar as médias e desvios padrões das duas variáveis, bem como sua correlação. Os indivíduos têm dados de uma variável explicativa x e uma variável de resposta y . A partir dos dados deve-se calcular as médias \bar{x} e \bar{y} e, ainda, os desvios padrões s_x e s_y das duas variáveis e, também, a correlação r . A reta de regressão de mínimos-quadrados é dada pela Equação 2.5, onde \hat{y} (lê-se "y" chapéu) e fornece uma resposta preditiva. A Figura 2.2 apresenta um exemplo do gráfico resultante da aplicação da regressão de mínimos-quadrados em um determinado conjunto de dados.

$$\hat{y} = a + bx \quad (2.5)$$

Com declividade dada pela Equação 2.6,

$$b = r \frac{s_y}{s_x} \quad (2.6)$$

Com intercepto dado pela Equação 2.7.

$$a = \bar{y} - b\bar{x} \quad (2.7)$$

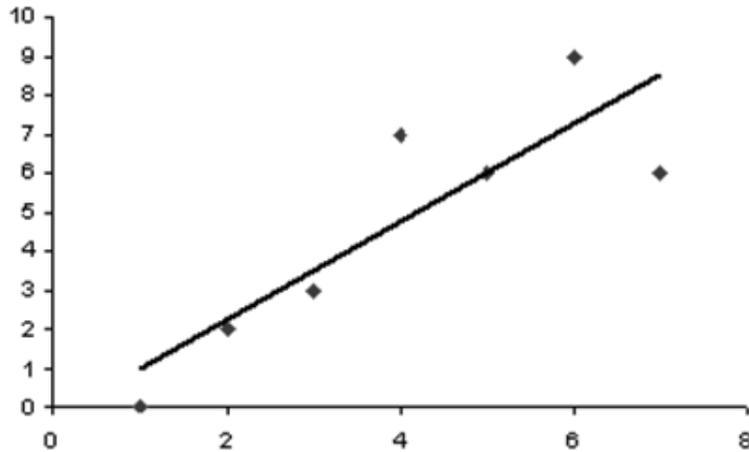


Figura 2.2: Exemplo de gráfico de dispersão com reta de regressão de mínimos quadrados [Moo04].

Após traçar a reta de regressão pode haver a necessidade de melhorar a correlação r entre as variáveis. Isso é feito a partir do ajuste dessa reta, também chamado de limpeza dos dados. Uma técnica de limpeza é a retirada dos valores que se encontram distantes da linha de tendência traçada sobre o eixo x de um gráfico de resíduos, onde o gráfico de resíduo é um gráfico de dispersão dos resíduos de regressão versus a variável explicativa. Sua função é ajudar no ajuste de uma reta de regressão, e o resíduo é dado pela Equação 2.8 [Moo04].

$$resíduo = y_{observado} - y_{predito} \quad (2.8)$$

A regressão é amplamente aplicada em vários domínios de conhecimento. Existem diversos trabalhos envolvendo a regressão linear para estimativas em software [NSB08] [SYB08] [SDD⁺02] [RTR08] [BBT⁺08]. Contudo, deve-se ter alguns cuidados para a utilização da regressão nas estimativas de desenvolvimento de software (PDS), isso porque, talvez não esteja explícita uma correlação linear entre os dados, mas sim, uma correlação exponencial [Kan03]. Assim, as variáveis do modelo devem ser cuidadosamente selecionadas e, além disso, utilizar a regressão faz com que o modelo preditivo tenha uma forte dependência da qualidade dos dados e seja sensível a *outliers* [KKM⁺08].

As seguintes situações que, quando simultâneas, tornam fortemente recomendável o uso da regressão [GM97] [ABC00]:

- quando o projeto dispõe de muitos dados, aumentando o grau de liberdade disponível e permitindo a identificação de padrões nos dados;
- quando os dados são bem comportados, no sentido estatístico da palavra, entende-se que não

existe um número significativo de *outliers*;

- quando um número reduzido de variáveis independentes é suficiente para estimar linearmente a variável dependente, de forma a se ter um resultado interpretável;
- quando não existem valores faltantes das variáveis independentes;
- quando as variáveis independentes são todas contínuas ou todas discretas, pois não existem técnicas estatísticas que permitam a manipulação dos dois tipos de variáveis simultaneamente; e
- quando as variáveis independentes não são correlacionadas entre si, pois variáveis que possuem alto grau de correlação prejudicam a técnica de regressão de mínimos-quadrados.

A principal desvantagem da regressão linear simples ou múltipla é que quando aplicada sobre dados de software ela pode apresentar os seguintes problemas: i) dificuldade na escolha das métricas a serem utilizadas; ii) coexistência de vários processos de desenvolvimento de software; e iii) disponibilidade de dados quantitativos *versus* qualitativos [ABC00].

Segundo [FN99], existem outros problemas relacionados à utilização da regressão, sendo eles:

- Multicolinearidade: é o número de variáveis preditivas altamente correlacionadas positivamente ou negativamente, o que pode distorcer o resultado da predição;
- Ajuste no modelo de dados: por causa da falta de dados, alguns pesquisadores ajustam os dados em seus modelos preditivos e não testam o modelo resultante em uma nova série de dados;
- Remoção de dados dos modelos: podem existir erros de registro pela transcrição de dados. Assim, é necessária uma análise mais detalhada antes que se faça a remoção dos dados; e
- Média dos dados: utilizando médias reduz-se a quantidade de informação disponível para testes, perdendo-se os detalhes, isso pode tornar as conclusões fracamente argumentadas.

2.7 Análise das Propostas da Literatura

Os aspectos relacionados abaixo foram identificados como importantes para a área de estimativas. Tais aspectos foram definidos com base nas deficiências das técnicas de estimativas aqui apresentadas em serem utilizadas para a manutenção de software e, ainda, observadas no Estudo de Caso Exploratório, apresentado no Capítulo 4.

- **Proposta:** apresenta o nome da proposta apresentada na revisão da literatura;
- **Domínio:** informa se a proposta se insere no contexto de desenvolvimento ou manutenção de software. Seus possíveis valores são: (Desenv. / Manut.);

- **Representação:** informa o domínio do valor de ETMS estimado. Seus possíveis valores são: (Numérica / Categórica), 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;
- **Base Histórica:** informa se a proposta utiliza base de dados histórica de projetos anteriores. Seus possíveis valores são: (Sim/Não);
- **Combinação:** informa se a proposta combina diferentes técnicas de estimativas para aumentar a precisão ou se as mesmas são utilizadas isoladamente. Seus possíveis valores são: (Sim / Não);
- **Ao Menos 3:** informa se na proposta da literatura são utilizadas ao menos três técnicas combinadas para compor o valor do esforço estimado de ETMS visando melhorar a precisão, conforme sugere [LB06] [Som06] [Pre09]. Os possíveis valores são: (Sim / Não);
- **Calibragem:** informa se a proposta apresentada é passível de calibragem. Seus possíveis valores são (Sim/Não);
- **AvaliaçãoMP:** informa se a proposta possui algum componente para que sejam avaliadas as estimativas e estas se ajustem com o passar do tempo para melhorar sua precisão. Seus possíveis valores são: (Sim / Não); e
- **Classificação:** refere-se à classificação da proposta segundo a sugestão de [ABC00]. Os possíveis valores são: (Empíricos e Compostos / Dinâmicos / *Expertise* / OAM / Regressão).

Apoiado nos aspectos supracitados, a Tabela 2.2 apresenta um resumo analítico dessas propostas destacando, por sombreado, os aspectos considerados importantes para compor uma técnica de estimativas que atenda a área de manutenção de software.

2.8 Considerações Finais do Capítulo

Para a melhoria da precisão do valor de esforço estimado, a literatura sugere que se utilize pelo menos três técnicas de estimativas [LB06] [Pre09] [Som06]. Todavia, essa deficiência não foi observada em nenhuma das técnicas de estimativas, não havendo, conseqüentemente, uma discussão de como agregar ou retirar tais técnicas e de como selecionar os valores de esforço estimados pelas mesmas.

Essa deficiência também foi constatada por meio do Estudo de Caso Exploratório em um projeto real de uma fábrica de software, apresentado no Capítulo 4, onde surgiu a proposta do Modelo-E10, apresentado no Capítulo 5.

Tabela 2.2: Resumo analítico das propostas do estado da arte das estimativas de esforço para atender a manutenção de software.

Proposta	Domínio	Representação	Base Histórica	Combinação	Ao Menos 3	Calibragem	AvaliaçãoMP	Classificação
COCOMO	Desenv./ Manut.	Numérico	Sim	Não	Não	Sim	Não	Composto
Equação do Software	Desenv.	Numérico	Sim	Não	Não	Não	Não	Dinâmica
Delphi	Desenv./ Manut.	Numérico	Não	Não	Não	Não	Não	<i>Expertise</i>
Redes Neurais	Desenv.	Numérico/ Categórico	Sim	Não	Não	Sim	Não	OAM
Fuzzy	Desenv.	Categórico	Não	Não	Não	Não	Não	OAM
Raciocínio Baseado em Casos	Desenv.	Numérico/ Categórico	Sim	Não	Não	Sim	Não	OAM
Árvores de Decisão	Desenv.	Numérico/ Categórico	Sim	Não	Não	Sim	Não	OAM
Regressão	Desenv.	Numérico	Sim	Não	Não	Sim	Não	Regressão

3. Revisão Sistemática para Estimativas de Esforço

Este capítulo apresenta o protocolo da revisão sistemática realizada na área de estimativas de esforço de forma a referenciar trabalhos de pesquisadores que buscam responder a questão de pesquisa apresentada na seção 1.2 do Capítulo 1.

3.1 Protocolo da Revisão

3.1.1 Formulação da Questão

Questão Foco: identificar iniciativas com relação às estimativas de esforço em projetos de manutenção de software, com o intuito de estabelecer quais técnicas de estimativas estão sendo amplamente utilizadas no contexto de pesquisa na área de estimativas de software.

Qualidade e Amplitude da Questão:

- **Problema:** a literatura apresenta uma gama de propostas de técnicas para estimativas de esforço. O desafio dos pesquisadores é aperfeiçoar tais técnicas 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.
- **Questão:** quais são os tipos de técnicas de estimativas de esforço utilizadas no contexto de manutenção de software que contribuam para pesquisas futuras?
- **Palavras-chave e sinônimos:** *Estimation: Software Effort Estimation, Estimating Software Effort, Software Effort Prediction, Predicting Software Effort.*
Maintenance: Effort Estimation on Software Maintenance, Effort Prediction on Software Maintenance, Estimating Software Maintenance Effort, Predicting Software Maintenance Effort.
- **Intervenção:** identificação e avaliação das soluções expostas nas pesquisas.
- **Controle:** inexistente.
- **Efeito:** consolidação do conhecimento e identificação do estado da arte das estimativas de ETMS.
- **Medição de resultados:** número de trabalhos identificados.
- **População:** artigos publicados em periódicos e anais de conferências relacionados ao tema de pesquisa, com data de publicação posterior a 2007, já que foi utilizada a revisão sistemática de [JS07] para a verificação de trabalhos anteriores a 2007 e não foi encontrado nenhum relevante à área em questão.

- **Aplicação:** pesquisadores da área.
- **Projeto experimental:** não se aplica.

3.1.2 Seleção das Fontes

Definição dos critérios de seleção das fontes: disponibilidade de consulta a artigos completos (*full papers*) através da Web pelo convênio PUCRS–CAPES; existência de mecanismos de busca com suporte a inserção de operadores booleanos (como *AND* e *OR*); e base de dados atualizada com publicações referentes aos últimos 3 anos.

Idioma dos estudos: inglês.

Identificação das fontes: *strings* de busca

- ("*Software Effort*" *OR* (*Maintenance AND (Estimation OR Prediction)*)) *OR* (((*Estimating OR Predicting*) *AND* "*Software Effort*") *OR* *Maintenance*));

Lista de fontes: *IEEEExplore*, *ACM Digital Library*, *Citeseer library*, *CiteseerX library*, *Springer-Link*, *ScienceDirect* e *DBLP*.

Seleção das fontes após avaliação: foram selecionadas as fontes *IEEEExplore*, *ACM Digital Library*, *SpringerLink* e *ScienceDirect*.

Verificação de referências: todas as fontes selecionadas são confiáveis e amplamente utilizadas por pesquisadores do mundo inteiro.

3.1.3 Seleção dos Estudos

Definição dos estudos:

- **Critérios de exclusão dos estudos:**

1. Que não estejam no formato de artigo completo (*full paper*);
2. Que não sejam relacionados às estimativas de esforço em projeto de software; e
3. Cujo conteúdo esteja relacionado às estimativas de custo ou defeitos.

- **Definição do tipo do estudo:** não são considerados trabalhos que indicam os desafios e as direções a serem tomadas na área de pesquisa em questão (*roadmaps*), mas são considerados aqueles que apresentam propostas para a área de estimativas de esforço tanto em desenvolvimento como em manutenção de software ; e

- **Procedimentos para a seleção dos estudos:** execução da string de busca nos mecanismos de busca oferecidos em cada uma das fontes selecionadas; leitura dos meta-dados de cada artigo; leitura de título e o resumo (*abstract*) de cada artigo aprovado na primeira triagem; leitura integral dos artigos remanescentes.

Execução da Seleção:

- **Seleção inicial dos estudos:** a busca em todos os mecanismos resultou em 81 artigos;
- **Avaliação da qualidade dos estudos:** 14 estudos (ou 17,2% da amostra inicial) foram selecionados para a extração de informações; e
- **Revisão da seleção:** a seleção dos estudos foi aprovada pelos professores orientadores do trabalho.

3.1.4 Extração de Informações

Critérios de inclusão e exclusão de informações: as informações extraídas dos estudos devem conter todos os aspectos envolvidos para estimar o esforço em manutenção de software, 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.

Formulários para extração dos dados: abaixo são apresentados os aspectos considerados relevantes neste trabalho para responder a questão de pesquisa apresentada na seção 1.2 do Capítulo 1. Tais aspectos foram definidos conforme observado na Seção 2.7 do Capítulo 2 e no Estudo de Caso Exploratório realizado na empresa parceira, apresentado no Capítulo 4.

- **Proposta:** apresenta o sobrenome do primeiro autor e a referência bibliográfica da proposta;

- **Método de Pesquisa:** apresenta como a pesquisa foi conduzida. Os possíveis valores de coluna (Comparativo / Estudo de Caso / Protótipo), conotam respectivamente: a comparação de diferentes propostas, a realização de um estudo de caso em um ambiente real e a proposta de um protótipo;
- **Foco de Contribuição:** informa o contexto da contribuição (Científica / Mercado);
- **Classificação:** classifica a proposta conforme sugere [ABC00] (Empíricos e Compostos / Dinâmicos / *Expertise* / OAM / Regressão);
- **Descrição:** faz uma breve descrição da proposta;
- **Domínio:** informa se a proposta de estimativas de esforço se insere no contexto de desenvolvimento ou manutenção de software (Desenv. / Manut.);
- **Representação:** informa o domínio do valor de ETMS estimado 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);
- **Combinação:** informa se a proposta combina diferentes técnicas de estimativas com intuito de melhorar a precisão (Sim / Não);
- **Ao Menos 3:** informa se a proposta utiliza pelo menos três técnicas de estimativas visando melhorar a precisão, conforme sugerem [LB06] [Som06] [Pre09] (Sim / Não);
- **Calibragem:** informa se a proposta permite a calibragem da técnica (Sim / Não); e
- **Possui AvaliaçãoMP:** informa se a proposta apresentada consegue avaliar o desempenho dos modelos preditivos das técnicas de estimativas que melhor obtiveram resultados nas estimativas anteriores (Sim / Não).

Execução da extração: vide tabelas 3.1 e 3.6.

Resolução de divergências entre os pesquisadores: não se aplica.

3.1.5 Sumarização dos Resultados

Apresentação dos resultados em tabelas: vide tabelas 3.2 a 3.4.

Comentários finais:

- **Número de estudos:** foram retornados 81 artigos dos quais 14 foram selecionados e outros 67 artigos foram descartados;

- **Possíveis fontes de distorções identificadas:** o número de fontes de busca utilizadas (quatro); a dificuldade de se definir uma string de busca que possibilitasse a identificação dos estudos relevantes em diferentes áreas de pesquisa; a qualidade dos motores de busca das fontes selecionadas; e a influência do autor na seleção dos artigos e na extração das informações;
- **Variação entre revisores:** não se aplica; e
- **Aplicação dos resultados:** os resultados obtidos são aplicados para identificar contribuições;
- **Recomendações:** nenhuma.

A Tabela 3.1 mostra as informações gerais dos trabalhos relacionados com intuito de classificá-los conforme os aspectos relevantes à área de estimativas de software.

Tabela 3.1: Extração das informações gerais dos estudos selecionados.

Proposta	Método de Pesquisa	Foco da Contribuição	Classificação	Descrição
Baskeles [TBB07]	Comparativo	Científico	OAM	Sugere que as estimativas de esforço sejam realizadas com base nos dados históricos de projetos estimados com o COCOMO. Utiliza algoritmos de redes neurais: <i>Back Propagation Multilayer Perceptrons</i> , <i>Radial Basis Function (RBF)</i> e <i>Support Vector Regression(SVR)</i> , todos para gerarem uma árvore de regressão de esforço estimado.
Braga [OBM08]	Comparativo	Científico	OAM	Sugere que para as estimativas de esforço sejam utilizadas técnicas com base em Algoritmos Genéticos proposto com SVR, RBF e Regressão Linear.
Deng [PDP07]	Comparativo	Científico	OAM	Sugere que para as estimativas de esforço, a utilização de algoritmos de ML como: <i>Multi-layer Perceptrons</i> , <i>K-Nearest</i> , <i>Neighbour (k-NN)</i> e RBF.
Gary [BL07]	Comparativo	Científico	<i>Expertise</i>	Sugere que para as estimativas de esforço deve-se avaliar a confiabilidade da <i>expertise</i> , indicando as melhores variáveis a serem utilizadas nas estimativas por meio da mineração de dados.
Jingzhou [LR07]	Protótipo	Científico	OAM	Sugere que para as estimativas sejam utilizadas analogias de métricas de projetos anteriores.
Kultur [KTB08]	Estudo de Caso	Mercado	OAM	Sugere que para as estimativas de esforço sejam utilizadas Redes Neurais Compostas.
Lokan [LM09b]	Comparativo	Mercado	OAM	Sugere para as estimativas o esforço os projetos sejam particionados em ordem cronológica, ao invés utilizar a comparação entre projetos, como feito tradicionalmente. E realiza uma Regressão Linear sobre os dados.
Mendes [LM09a]	Comparativo	Científico	Regressão	Sugere que as estimativas de esforço, para projetos futuros, sejam realizadas com base em projetos anteriores similares. E utiliza a regressão linear para tal tarefa.
Mohammad [ANC08]	Comparativo	Científico	OAM	Sugere que as estimativas de esforço sejam realizadas por analogia, fazendo uma subseleção dos resultados por meio da Lógica <i>Fuzzy</i> .
Mohammad [ANC09]	Comparativo	Científico	Dinâmicos	Sugere que as estimativas de esforço sejam realizadas com base em analogia de projetos anteriores. E utiliza a Lógica <i>Fuzzy</i> e o Modelo <i>Grey</i> , para tal, quando este último gera a equação preditiva de esforço.
Patil [Pat07]	Estudo de Caso	Mercado	Empírico	Sugere que as estimativas de esforço sejam realizadas por meio de medidas no código de linguagens de quarta geração.
Seo [SYB08]	Comparativo	Científico	OAM	Sugere que as estimativas de esforço sejam realizadas por meio de <i>K-means</i> e Redes Neurais, para eliminar <i>outliers</i> . Depois, realiza as estimativas utilizando Redes Neurais e Regressão com Mínimos Quadrados.
Shukla [SM08]	Estudo de Caso	Mercado	OAM	Sugere que para realizar as estimativas de esforço em manutenção de software, corretiva e preventiva, sejam utilizadas Redes Neurais para indicar os direcionadores de custos mais promissores.
YongWang [WSM ⁺ 09]	Comparativo	Científico	Dinâmicos	Sugere que as estimativas de esforço sejam realizadas durante todo o processo de software utilizando uma equação preditiva gerada pelo Modelo <i>Grey</i> .

A Tabela 3.2 mostra a frequência da classificação dos trabalhos relacionados conforme a sugestão de [ABC00]. A frequência absoluta é dada pela contagem dos trabalhos conforme se enquadram na classificação sugerida pela autora. A frequência relativa é dada pelo percentual em relação à quantidade total de trabalhos encontrados.

Tabela 3.2: Distribuição de frequência (absoluta e relativa) do aspecto “Classificação da Solução”.

Classificação da Solução	Quantidade de Artigos	% em Relação ao Total Geral
Técnicas Empíricas	1	7,15
Baseadas em <i>expertise</i>	1	7,15
OAM	9	64
Dinâmicas	2	14,55
Baseadas em Regressão	1	7,15
Técnicas Compostas	0	0
Totais	14	100

A Tabela 3.3 mostra a frequência do Método de Pesquisa utilizado pelos autores. Essa classificação foi utilizada para identificar se os estudos eram comparações entre técnicas de estimativas, aplicação de estudo de caso e protótipo. Essa classificação ajuda a localizar se os trabalhos encontrados possuem foco científico ou de mercado.

Tabela 3.3: Distribuição de frequência (absoluta e relativa) do aspecto “Método de Pesquisa”.

Método de Pesquisa	Quantidade de Artigos	% em Relação ao Total Geral
Comparativo	10	71,4
Estudo de Caso	3	21,4
Protótipo	1	7,2
Totais	14	100

A Tabela 3.4 apresenta a frequência do foco dos trabalhos identificados. Um trabalho com cunho científico é importante para que as técnicas de estimativas apresentadas possam ser analisadas e aprimoradas. Inversamente, o foco no mercado é importante para identificar as vantagens e desvantagens de adoção das mesmas, além da avaliação dos resultados em um ambiente real.

Tabela 3.4: Distribuição de frequência (absoluta e relativa) do aspecto "Foco da Contribuição".

Foco da Contribuição	Quantidade de Artigos	% em Relação ao Total Geral
Científico	10	71,44
Mercado	4	28,56
Totais	14	100

A Tabela 3.6 mostra como as propostas atuais estão em relação às deficiências da área de estimativas, discutidas nas seções 1.1 e 1.2 do Capítulo 1. Uma proposta que enriqueça essa área é aquela que preencha os aspectos aqui apontados.

Tabela 3.5: Extração das informações relativas ao contexto das estimativas de ETMS.

Proposta	Domínio	Representação	Base Histórica	Combinação	Ao menos 3	AvaliaçãoMP	Calibragem
Baskes [TBB07]	Desenv.	Númerica	Sim	Não	Sim	Não	Sim
Braga [OBM08]	Desenv.	Númerica	Sim	Não	Não	Não	Sim
Deng [PDP07]	Desenv.	Númerica	Sim	Não	Sim	Não	Não
Gary [BL07]	Desenv.	Númerica	Sim	Não	Não	Não	Não
Jingzhou [LR07]	Desenv.	Númerica	Sim	Não	Não	Sim	Sim
Kultur [KTB08]	Desenv.	Númerica	Sim	Não	Sim	Não	Sim
Lokan [LM09b]	Desenv.	Númerica	Sim	Não	Não	Não	Sim
Mendes [LM09a]	Desenv.	Númerica	Sim	Não	Não	Não	Sim
Mohammad [ANC08]	Desenv.	Númerica	Sim	Sim	Não	Não	Sim
Mohammad [ANC09]	Desenv.	Númerica	Sim	Não	Sim	Não	Não
Patil [Pat07]	Desenv.	Númerica	Não	Não	Não	Não	Não
Seo [SYB08]	Desenv.	Númerica	Sim	Não	Não	Não	Sim
Shukla [SM08]	Manut.	Númerica	Sim	Não	Não	Não	Sim
YongWang [WSM ⁺ 09]	Desenv.	Númerica	Sim	Não	Não	Não	Sim

Tabela 3.6: Extração das informações relativas ao contexto das estimativas de ETMS.

Proposta	Domínio	Representação	Base Histórica	Combinação	Ao menos 3	AvaliaçãoMP	Calibragem
Baskeles [TBB07]	Desenv.	Númerica	Sim	Não	Sim	Não	Sim
Jingzhou [LR07]	Desenv.	Númerica	Sim	Não	Não	Sim	Sim
Kultur [KTB08]	Desenv.	Númerica	Sim	Não	Sim	Não	Sim
Mohammad [ANC08]	Desenv.	Númerica	Sim	Sim	Não	Não	Sim
Shukla [SM08]	Manut.	Númerica	Sim	Não	Não	Não	Sim

Os problemas identificados com as estimativas de esforço, apresentados na seção 1.1, são inerentes à manutenção de software. Por isso é interessante saber a frequência de trabalhos que tentam estimar o esforço de manutenção em software. Portanto, o Domínio foi classificado como: Esforço de Desenvolvimento e Esforço de Manutenção. Assim, a Tabela 3.7 a frequência dos trabalhos encontrados em seus respectivos domínios.

Tabela 3.7: Distribuição de frequência (absoluta e relativa) do aspecto “Domínio”.

Domínio	Quantidade de Artigos	% em Relação ao Total Geral
Esforço em Desenvolvimento	13	92,8
Esforço em Manutenção	1	7,2
Totais	14	100

A Tabela 3.8 mostra a frequência de como os trabalhos fazem a representação dos valores das estimativas. Isso é importante para identificar se existe uma outra forma de representação de esforço, que não seja numérica, para ser utilizada pelas técnicas de estimativas.

Tabela 3.8: Distribuição de frequência (absoluta e relativa) do aspecto “Representação”.

Representação	Quantidade de Artigos	% em Relação ao Total Geral
Númerica	14	100
Catagórica	0	0
Totais	14	100

A Tabela 3.9 apresenta a frequência com que os trabalhos identificados utilizam dados históricos de projetos como base para estimativas futuras. Isso é importante para definir a estratégia de utilização desses dados.

Tabela 3.9: Distribuição de frequência (absoluta e relativa) do aspecto "Base Histórica".

Base Histórica	Quantidade de Artigos	% em Relação ao Total Geral
Sim	13	7,14
Não	1	92,86
Totais	14	100

A Tabela 3.10 apresenta a frequência com que os trabalhos relacionados combinam as diferentes técnicas de estimativas utilizadas. A combinação de técnicas é importante para realizar sua autoavaliação.

Tabela 3.10: Distribuição de frequência (absoluta e relativa) do aspecto "Combinação".

Combinação	Quantidade de Artigos	% em Relação ao Total Geral
Sim	1	7,15
Não	13	92,85
Totais	14	100

A Tabela 3.11 apresenta a frequência com que os trabalhos identificados utilizam pelo menos três técnicas de estimativas, estando alinhados com as sugestão de [LB06] [Som06]. Isso é importante porque, segundo os autores, quanto mais técnicas se utilizar, maior a possibilidade de convergência entre o esforço estimado e o esforço real, conseqüentemente, a precisão das estimativas é aumentada.

Tabela 3.11: Distribuição de frequência (absoluta e relativa) do aspecto "Ao Menos 3".

Ao Menos 3	Quantidade de Artigos	% em Relação ao Total Geral
Sim	4	28,57
Não	10	71,43
Totais	14	100

Neste contexto, entende-se por AvaliaçãoMP uma forma de manter as técnicas que obtiverem melhor desempenho nas estimativas e de descartar as que não obtiverem resultados satisfatórios. Essa funcionalidade é importante por fazer um ajuste fino nos valores de esforço estimados pelas técnicas utilizadas. Assim, a Tabela 3.12 apresenta a frequência com que os trabalhos relacionados utilizam a AvaliaçãoMP.

Tabela 3.12: Distribuição de frequência (absoluta e relativa) do aspecto “AvaliaçãoMP”.

AvaliaçãoMP	Quantidade de Artigos	% em Relação ao Total Geral
Sim	1	7,15
Não	13	92,85
Totais	14	100

A Tabela 3.13 apresenta a frequência com que os trabalhos identificados utilizam a calibragem das técnicas de estimativas. A calibragem é importante porque ela mantém os dados, utilizados pelas técnicas, atualizados, e como consequência, pode aumentar a precisão das estimativas.

Tabela 3.13: Distribuição de frequência (absoluta e relativa) do aspecto “Calibragem”.

Calibragem	Quantidade de Artigos	% em Relação ao Total Geral
Sim	10	71,43
Não	4	28,57
Totais	14	100

3.2 Considerações Finais do Capítulo

Como mostrado na Tabela 3.6, nenhuma das propostas atuais conseguem atender todos os aspectos observados para responder a questão de pesquisa apresentada na seção 1.2 do Capítulo 1. Assim, faz-se necessário encontrar uma proposta que atenda tais aspectos. Isso porque estes são essenciais na prática das estimativas, como observado no estudo de caso exploratório apresentado no Capítulo 4.

Vale ressaltar, também observando a Tabela 3.6, que poucos são os trabalhos voltados especificamente às estimativas de ETMS. Na realidade, as organizações adaptam técnicas de estimativas para desenvolvimento de software para estimar o esforço de manutenção, como discutido na seção 1.1

do Capítulo 1. Outrossim, a maioria das pesquisas, bem como o mercado, não combinam diferentes técnicas de estimativas de esforço e todos esses fatores comprometem a precisão das estimativas contrariando a sugestão de autores da área: [LB06] [Pre09] [Som06].

Para responder a questão de pesquisa apresentada na Seção 1.2 do Capítulo 1, e, ainda, atender os aspectos relevantes da área de estimativas de esforço em manutenção de software, uma proposta deve:

- **Se basear em dados históricos do projeto:** como apresentado na Tabela 3.9, a maior parte das propostas atuais se baseiam em dados históricos de projetos e, além disso, na classificação da solução apresentada na Tabela 3.2, 7 das 14 propostas são baseadas em OAM que, por sua vez, também utiliza dados históricos. A tendência em se usar bases de dados históricas se deve principalmente à coleta e ao armazenamento de dados de projetos como uma prática regular nas fábricas de software. Isso tudo é suficiente para justificar a utilização de dados históricos de projetos nas estimativas;
- **Ter o domínio na manutenção de software:** como apresentado na Tabela 3.7, das propostas atuais apenas uma é voltada para a manutenção de software, o que deixa evidente a deficiência de trabalhos para projetos com tal característica. Por isso, as organizações utilizam técnicas de estimativas de desenvolvimento de software improvisando-as para a manutenção do mesmo. Assim, é importante preencher essa deficiência e ter uma proposta de estimativas específica para manutenção de software;
- **Ter uma representação numérica:** todas as propostas atuais estimam o valor de esforço de trabalho com uma representação numérica, como apresenta a Tabela 3.8. Essa deficiência também foi observada em uma fábrica de software objeto do Estudo de Caso Exploratório apresentado no Capítulo 4. Assim, seguindo essas tendências e deficiências, uma proposta para estimar o esforço de manutenção em software deve ter como resultado um valor de esforço representado por um valor numérico;
- **Fazer uso de pelo menos três técnicas de estimativas:** a literatura básica na área de estimativas como: [LB06] [Pre09] [Som06], relatam a importância de se utilizar mais de uma técnica para estimar o esforço em projetos de software, contudo, 10 das 14 propostas atuais, negligenciam essa afirmação e utilizam apenas uma técnica para estimar esforço, como apresenta a Tabela 3.11. O ideal é uma proposta que faça o uso de pelo menos três técnicas de estimativas e que, ainda, apresente critérios de adoção e de retirada dessas técnicas;
- **Avaliação do desempenho das estimativas realizadas combinando as diferentes técnicas de estimativas utilizadas:** a Tabela 3.10 mostra que apenas 2 das 14 propostas utilizam e combinam mais de uma técnica de estimativas. A combinação de técnicas é importante porque aumenta a chance de se encontrar um ou mais valores de esforço estimados mais próximos do valor de esforço real. A Tabela 3.12 mostra que também somente 2 das 14

propostas de técnicas de estimativas são passíveis de AvaliaçãoMP. A AvaliaçãoMP é importante porque permite que sejam apenas selecionadas as técnicas com bom desempenho para estimar o esforço de trabalho. Dessa forma, uma proposta de estimativas em manutenção de software deve combinar e avaliar as técnicas utilizada; e

- **Permitir a calibragem constante do modelo:** das propostas atuais, 10 de 14 permitem que os modelos preditivos das técnicas de estimativas utilizadas sejam calibrados, como apresenta a Tabela 3.13. Esse aspecto é importante para manter o modelo preditivo de estimativas atualizado conforme a realidade do projeto, o que torna esse aspecto essencial para uma proposta de estimativas em manutenção de software.

Finalmente, os aspectos supracitados foram ratificados por meio da Revisão da Literatura, Capítulo 2, do Estudo de Caso Exploratório, Capítulo 4, bem como, incorporados ao Modelo–E10 para atender a área de estimativas de ETMS, Capítulo 5.

4. Estudo de Caso Exploratório

Este capítulo apresenta um estudo de caso exploratório realizado em uma fábrica de software. O estudo em questão tem enfoque em melhorar a precisão das estimativas de ETMS, bem como, desonerar os membros da equipe de projeto que, na época deste estudo, era responsável por estimar o ETMS em um dos projetos da organização. Este estudo foi patrocinado pelos gestores de um dos projetos da fábrica de software.

4.1 Cenário

A *Hewlett-Packard Company* (HP) possui no TECNOPUC–Parque Tecnológico da Pontifícia Universidade Católica, em Porto Alegre, RS, uma fábrica de software que atende clientes, desde a esfera governamental até a iniciativa privada. Esses vários projetos, em diferentes áreas de conhecimento, exigem que a organização tenha seu quadro funcional composto por diferentes experiências que, quando em convívio, tornam seu ambiente rico e heterogêneo para o desenvolvimento também de pesquisas.

Visando atrair mais clientes e adquirir maior credibilidade no segmento de desenvolvimento de software, em 2002, a organização iniciou a busca pelo CMM nível 3. Em novembro de 2005, após um grande esforço por parte da organização e de seus colaboradores, foram alcançados os requisitos para elevá-la ao nível 3 do CMM, sendo parte deste esforço em parceria com a PUCRS.

O projeto escolhido pelos gestores da organização para aplicar o estudo de caso em questão é um projeto de manutenção de software de um grande banco estatal, que envolvia um alto orçamento e, ainda, era regido por um contrato de SLA. Por isso, o projeto tinha grande interesse em melhorar as suas estimativas de ETMS. Por questões de confidencialidade, daqui por diante, este projeto é referenciado por projeto P1.

O projeto P1 contava com 63 colaboradores divididos em quatro equipes: Projeto, com 20 membros; Cliente com 12 membros; Servidor com 12 membros; e Testes, com 17 membros. Cada equipe possuía um líder técnico e, além destes, um profissional de *Software Quality Assurance* (SQA) e um subgerente, responsável por organizar e acompanhar o cronograma das demandas.

A equipe de Projeto é responsável por estimar os prazos e realizar criação ou alteração nos documentos de projeto do software; a equipe de Cliente é responsável pelo desenvolvimento ou manutenção do código de telas e regras de negócio na parte cliente da aplicação, com base nos documentos de projeto do software; a equipe de Servidor é responsável pelo desenvolvimento ou manutenção do código da parte de banco de dados da aplicação, com base nos documentos de projeto do software; por fim, a equipe de Testes é responsável pelos testes de integração do software com base nos seus requisitos e documentação de projeto.

O projeto P1 possui um PDS em cascata orientado a entregáveis, onde cada equipe representa uma fase desse PDS. Os projetos têm livre arbítrio para a escolha das ferramentas para auxiliarem

na gestão do PDS sendo que o projeto P1 utiliza o *MS-EPM (Enterprise Project Management)*, para o controle de prazos das atividades; o *MS-Excel* para as estimativas do ETMS e onde são armazenadas as sugestões de estimativas de um membro da equipe de projeto (*expertise*); o *IBM Rational Clear Quest* para o armazenamento de informações referentes a defeitos, tamanhos e esforço; RequisitePro para a gerência dos requisitos; e, por fim, um software próprio para apontamento das horas trabalhadas dos membros das equipes (esforço real).

Tipicamente, o software é dividido em versões, sendo que o projeto deve entregar uma quantidade de FP/mês para o cliente. O FP é distribuído dentro de uma ou mais versões do produto. Para cada versão do produto é determinado um conjunto de requisitos, que são atribuídos a uma demanda chamada de Ordem de Serviço (OS), as quais contêm as alterações a serem realizadas no software. A Figura 4.1 apresenta o PDS do projeto P1.

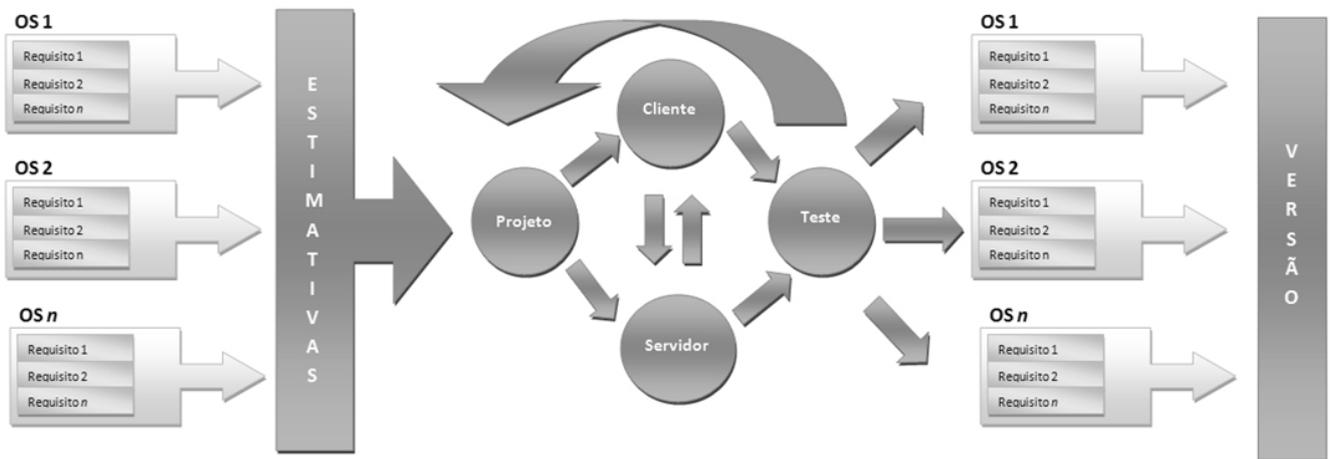


Figura 4.1: Processo de software do projeto P1.

4.2 Requisitos do Projeto

Os requisitos para a busca de técnicas de estimativas foram definidos por meio de reuniões formais com os gestores de projetos, com base na experiência nos projetos da organização. Sendo assim, os requisitos foram:

- Entrada e saída numérica: as técnicas utilizadas devem ter como entrada valores numéricos como FPs, KLOC, dentre outros, os quais se encontram devidamente armazenados na base de dados do projeto. Além disso, os valores de entradas também devem ter algum tipo de relação com o valor do Esforço Real de trabalho na tentativa de se encontrar um padrão dos dados coletados. Já a saída produzida, deve ser um valor numérico dado em horas de trabalho representando o esforço de uma pessoa/hora;
- Justificativa de resultado desnecessária: para o projeto o que importa é o valor do esforço de trabalho em horas, assim, não é necessário inferir como as estimativas são realizada e muito

menos a justificativa de seus resultados;

- Fácil utilização: a técnica adotada tem que ser de fácil utilização para onerar o mínimo possível os responsáveis pelas estimativas;
- Basear-se nos dados históricos do projeto: a sugestão dos gestores foi explorar a base histórica de um dos projetos da organização, alimentada desde 2004. Essa base histórica está em conformidade com as exigências do modelo de qualidade CMM nível 3;
- Atualização constante da técnica (calibragem): com a atualização constante da técnica, uma demanda estimada, depois de finalizada, passa a ser considerada para as próximas estimativas. A calibragem é importante porque com o passar do tempo aumenta as chances de estimativas mais precisas;
- Eliminar a *expertise*: atualmente a organização utiliza uma técnica baseada em *expertise*, a qual leva em consideração as sugestões de estimativas dos membros da equipe com base em sua experiência. É ideal que as técnicas estudadas não sejam baseadas na *expertise*, por dois motivos: i) desonerar as equipes; e ii) não depender de conhecimento humano para as estimativas.

Levando em consideração os requisitos definidos pelos gerentes de projetos, foi realizado o levantamento bibliográfico das técnicas potencialmente promissoras para estimar o ETMS, sintetizado na Tabela 2.2 da seção 2.7 do Capítulo 2.

4.3 Planejamento do Estudo de Caso

A realização deste estudo de caso obedeceu o esquema apresentado na Figura 4.2, como sugere [Yin05]. Nela são mostradas as etapas utilizadas para a realização do estudo de caso no projeto P1. O ponto de partida do estudo de caso foi o levantamento e o estudo das métricas existentes no projeto, realizada na etapa: "Métricas do Projeto". Todas as métricas coletadas estavam devidamente documentadas, o que facilitou o levantamento e entendimento das mesmas.

Na etapa "Métodos de Análise" foram estudadas e definidas as técnicas de estimativas sugeridas pela literatura, e utilizadas neste estudo de caso. As técnicas estatísticas e de mineração foram as mais apropriadas para as estimativas de ETMS, pois as mesmas são ideais para se trabalhar sobre os dados históricos.

A etapa das "Hipóteses" é o passo onde foram definidas as hipóteses alternativas do estudo de caso. A hipótese nula foi formulada após as reuniões com gestores e líderes do projeto P1, onde foi relatado o problema de estimativas do projeto. Já as hipóteses alternativas surgiram por meio dos "Ensaio Empíricos". Estas últimas guiaram a validação dos experimentos realizados no estudo de caso.

Estudo de Caso Exploratório

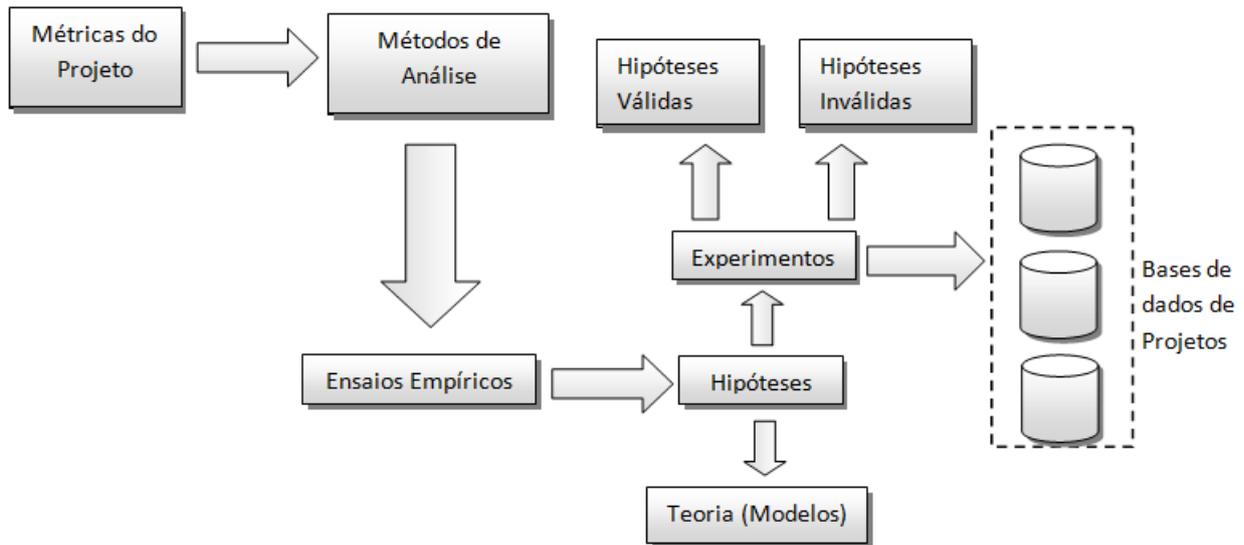


Figura 4.2: Estrutura do estudo de caso exploratório.

Os experimentos foram devidamente planejados com base nas hipóteses formuladas. A princípio a intenção era de validar as hipóteses em outros projetos, contudo, isso não foi possível, por falta de um projeto que tivesse uma base de dados como a do projeto P1.

Os resultados obtidos foram avaliados pela proximidade do valor de ETMS estimado com o valor real de esforço de demandas atuais. A expectativa nessa etapa, era que as técnicas de estimativas utilizadas nos "Ensaio Empírico" e definidas no "Método de Análise", refutassem a hipótese H_0 e ratificassem uma ou mais hipóteses alternativas, apresentadas e discutidas na Tabela 4.1.

O resultado do estudo de caso foi, por inferência, um modelo para estimativas de ETMS. Este modelo é apresentado e discutido em detalhes do Capítulo 5. O protocolo deste estudo de caso exploratório se encontra no Apêndice A.

4.4 Definição das Hipóteses e Técnicas

Por meio dos "Ensaio Empírico" com os dados do projeto P1 emergiram as hipóteses apresentadas na Tabela 4.1. Tais hipóteses foram baseadas nas variáveis existentes na base do projeto P1: *Function Points* (FP) e Quantidade de Documentos (QtdeDoc). A FP era conhecida por determinar o tamanho da demanda a ser implementada; já a quantidade de documentos teve que ser extraída por meio de consultas SQL. A quantidade de documentos foi escolhida porque o projeto P1 possui uma boa documentação do projeto. Além disso, [Pat07] sugere que sejam utilizados os documentos do projeto para a realização das estimativas em ETMS, o que ajudou na escolha dessa variável. Para a obtenção deste último foi realizado um grande esforço de preparação de dados.

A hipótese H_6 só foi utilizada para a fase 3 do estudo de caso, seção 4.7, deste capítulo. E a hipótese nula (H_0) não foi alterada.

Tabela 4.1: Hipóteses para o estudo de caso exploratório.

Hipóteses
H_0 : Nenhuma técnica de estimativas é mais precisa que a técnica atual, baseada em <i>expertise</i> .
H_1 : Conhecida a quantidade de documentos (artefatos de software) impactados por uma demanda é possível determinar seu esforço total de trabalho com mais precisão que a técnica atual.
H_2 : Conhecida a quantidade de documentos por equipe impactados por uma demanda é possível determinar o esforço de trabalho de cada equipe com mais precisão que a técnica atual.
H_3 : Conhecido o tamanho da demanda, em pontos de função, é possível determinar o esforço total de trabalho desta com mais precisão que a técnica atual.
H_4 : Conhecidos o tamanho da demanda, em pontos de função, e a quantidade de documentos relacionados à mesma, é possível determinar o esforço total de trabalho da demanda com mais precisão que a técnica atual.
H_5 : Utilizada mais de uma técnica de estimativas de esforço de trabalho em manutenção de software é possível determinar o esforço de trabalho de cada equipe com mais precisão que a técnica atual.
H_6 : Com base na complexidade dos documentos utilizados em cada fase do projeto é possível determinar o esforço de trabalho de cada equipe com mais precisão que a técnica atual.

Havia também uma hipótese para estimar o esforço de trabalho em manutenção de software utilizando FP por fase, contudo, essa hipótese não evoluiu, porque não foi possível determinar os FP de cada fase.

Diante das hipóteses formuladas, foram escolhidas algumas técnicas para as estimativas de ETMS com base nos requisitos definidos pelos gestores do projeto P1.

As técnicas de estimativas adotadas para este estudo de caso são sugeridas na literatura por autores da área de Engenharia de Software como: [ABC00] [DCS86] [Pre09] [Som06] [LB06] [Kan03] [SDD⁺02]. A principal vantagem de se utilizar as técnicas sugeridas é a maturidade das mesmas na resolução de problemas em diferentes áreas de conhecimento. As técnicas escolhidas foram:

- **Regressão Linear Simples e Múltipla:** A regressão Simples e Múltipla são técnicas estatísticas que se baseiam nos dados do projeto. Uma das principais características destas técnicas é a facilidade de aplicação após a preparação dos dados;
- **Algoritmo *M5P*:** O *M5P* é uma técnica que se baseia em árvore de regressão. Ela analisa os dados históricos do projeto e gera um ou mais modelos preditivos representados por equações lineares para estimar o atributo classe, neste caso, o ETMS. Esta técnica é encontrada na ferramenta *WEKA*, o qual é um software de descoberta de conhecimento em base de dados que contém vários algoritmos de mineração;
- **Árvore *REPTree*:** O *REPTree* é uma técnica de estimativas que gera uma árvore de decisão baseada na redução de ganho/variação dos dados. A profundidade da árvore é dada pelo cálculo de um valor de redução de erro. Ao final se tem uma árvore de decisão onde os nós folhas são os valores das estimativas com base nas variáveis de entradas definidas;
- **Percentual de Esforço por Equipe:** O percentual de esforço por equipe é o percentual médio de esforço despendido de cada equipe no desempenho de suas atividades em determinada OS.

Esse valor é calculado em duas etapas, a primeira delas é calcular a proporção de esforço de cada equipe com base nos dados históricos do esforço real do projeto P1. A segunda etapa é estimar o esforço total da OS, por meio da regressão linear, e calcular o ETMS com base na proporção por equipe;

- **Quartil 1 (25%):** O primeiro quartil (Q1) é uma técnica estatística que identifica o valor a partir do qual se encontram os 25% dos valores mais baixos, computados do 1º ao 25º termo de uma amostra ordenada. A aplicação dessa técnica permite obter um valor que represente o valor mais baixo de esforço real para que este seja utilizado para calcular o ETMS com base na proporção de esforço real por equipe;
- **Mediana:** A mediana, ou segundo quartil (Q2), é o valor do termo do meio de uma amostra de valores ordenados. Essa técnica permite obter o valor mediano de ETMS, mais precisamente o valor do meio de uma amostra ordenada. Com isso, pretende-se obter valores intermediários de esforço para que sejam balanceadas as estimativas;
- **Quartil 3 (75%):** O terceiro quartil (Q3) é uma técnica estatística que identifica o valor a partir do qual se encontram os 25% dos valores mais elevados, a partir do 75º termo até o último de uma amostra ordenada. A aplicação dessa técnica permite obter um valor que represente o valor mais alto de esforço real para que este seja utilizado para calcular o ETMS com base na proporção de esforço real por equipe;

A Tabela 4.2 apresenta as expectativas na aplicação das técnicas definidas para responder as hipóteses formuladas.

A avaliação das técnicas de estimativas para ETMS aplicadas nas fases 1, 2 e 3, foi realizada pelo cálculo do módulo da diferença entre o valor de esforço estimado de ETMS, utilizando a técnica definida na respectiva fase com o valor do esforço real. Esses valores foram comparados com a diferença, também em módulo, do esforço real com a *expertise* dos membros da equipe. O escore foi composto pela quantidade de valores estimados das OSs que se aproximaram do esforço real.

4.5 Fase 1 – Regressão Linear Simples e Múltipla

Essa primeira fase visou encontrar uma forma de melhorar a precisão das estimativas de ETMS por meio da regressão linear nos dados históricos do projeto P1. A riqueza de dados do projeto foi o fator determinante para a aplicação da técnica regressão linear. Os atributos disponíveis na base do projeto para realizar a regressão linear eram FP e Quantidade de Documentos (QtdeDoc) .

4.5.1 Regressão Baseada em Esforço Total

Para gerar o modelo preditivo da regressão simples baseado em FP, primeiramente foi montada uma consulta SQL na base de dados do projeto. Ao executar a consulta foi encontrado um total de 574 OSs, com uma correlação de 0,40, isso representa uma baixa correlação entre os dados,

Tabela 4.2: Modelos, hipóteses e expectativas.

Técnica	Hipóteses	Expectativa
Regressão	H_1, H_2, H_3, H_5	Uma equação que expresse um padrão e forneça um valor de esforço de trabalho total estimado para toda a demanda.
Regressão Múltipla	H_4, H_5	Uma equação que expresse um padrão e forneça um valor de esforço de trabalho para cada fase da demanda. A soma de seus valores individuais representa o valor total de esforço de trabalho da demanda.
<i>M5P</i>	H_1, H_2, H_3, H_4, H_5	Equações que expressam padrões conforme o comportamento dos dados, não necessariamente único. Estas, quando aplicadas sobre as variáveis definidas na hipótese, podem fornecer um valor de esforço de trabalho para cada fase e total da demanda, dependendo de uma ou mais características da demanda.
<i>REPTree</i>	H_1, H_2, H_3, H_4, H_5	Uma árvore de regressão que expresse um padrão e forneça valores de esforço de trabalho da demanda, tanto total como por equipe.
% Esforço por Equipe	H_5	A proximidade do percentual médio de esforço por equipe do esforço real de trabalho da demanda. Este valor é calculado sobre o total de esforço de trabalho estimado para uma demanda.
Quartil 1 (25%)	H_5	A proximidade do valor do percentual médio de esforço por equipe do esforço real, representando os menores valores de esforço. Este valor pode ser utilizado para calcular os valores de esforço da demanda para determinada equipe.
Mediana	H_5	A proximidade do valor central do percentual médio por equipe do esforço real da demanda, podendo este ser utilizado para estimar o esforço de trabalho por equipe, bem como o esforço de trabalho total da demanda.
Quartil 3 (75%)	H_5	A proximidade do valor do percentual médio de esforço por equipe do esforço real, representando os maiores valores de esforço. Este valor pode ser utilizado para calcular os valores de esforço da demanda para determinada equipe.

principalmente pela presença de *outliers*. Realizando a limpeza nos dados, conforme recomenda [Moo04], o número de OSs caiu para 336 e a correlação da variável FP com a variável esforço foi de 0,51, melhorando a correlação entre as variáveis e tornando-as estatisticamente significativas. Essas OSs, doravante chamadas de conjunto de treino, foram utilizadas para encontrar a equação de regressão linear, que representa o modelo preditivo da técnica.

Aplicando a equação encontrada em um conjunto de vinte OSs, doravante denominado de conjunto de testes, obtiveram-se os resultados apresentados na Tabela 4.3.

Na Tabela 4.3 observa-se que a *expertise* teve um score maior do que a técnica de regressão linear baseada em FP. Dessa forma, não se obteve sucesso para refutar a hipótese H_0 , chegando à conclusão de que, para este caso em particular, as estimativas com regressão linear baseada no tamanho da demanda não produz bons resultados, rejeitando-se a hipótese H_3 .

Em busca de melhores resultados foi usada a variável explicativa QtdeDoc ao invés de FP. Para gerar o modelo preditivo utilizando a regressão linear, agora com base em QtdeDoc, foi montada uma consulta SQL na base de dados do projeto e o resultado foi um total de 544.

Tabela 4.3: Escore da regressão por FP vs. sugestão das estimativas.

Modelo	Escore
Regressão por FP	4
<i>Expertise</i>	16

Realizando a limpeza nos dados, o número de demandas caiu para 523 e a correlação da variável QtdeDoc com a variável de esforço real foi de 0,86. Aplicando as equações encontradas no conjunto de treino, obteve-se o resultado apresentado na Tabela 4.4. Esses resultados foram mais promissores que aqueles apresentados na Tabela 4.3. Dessa forma, a precisão da técnica baseada na quantidade de documentos foi significativamente melhor que a precisão da técnica baseada em FP. Com isso foi possível refutar a hipótese H_0 e ratificar a hipótese alternativa H_1 .

Tabela 4.4: Escore da regressão por Quantidade de Documentos vs. *Expertise*.

Modelo	Escore
Quantidade de Documentos	14
<i>Expertise</i>	6

No último experimento realizado utilizou-se regressão linear múltipla. Esta técnica levou em consideração mais de um atributo para gerar o modelo preditivo. Para este caso foram considerados os atributos FP e QtdeDoc juntos. Para gerar o modelo preditivo utilizando a regressão múltipla foi realizada uma consulta SQL na base de dados do projeto, e o resultado foi um total de 296 OSs retornadas. Após a limpeza dos dados, permaneceram 286.

Os resultados obtidos foram menos promissores que àqueles com a regressão linear baseada em QtdeDoc, como pode ser observado na Tabela 4.5. Assim, para este caso, a hipótese H_0 foi ratificada e a hipótese H_4 foi rejeitada.

Tabela 4.5: Escore da regressão Múltipla (FP e Quantidade de Documentos) vs. *Expertise*.

Modelo	Escore
Quantidade de Documentos	9
<i>Expertise</i>	11

Os resultados até aqui obtidos foram apresentados em uma reunião com os gestores e os líderes técnicos do projeto P1 em 18/08/2007. Quando estimado por total de esforço de trabalho da demanda, os valores encontrados tiveram uma precisão maior do que a *expertise* dos membros da equipe.

Apesar dos resultados alcançados serem promissores, os gestores e os líderes técnicos do projeto manifestaram a necessidade em se estimar o ETMS das OSs por equipe. Nesse sentido, foi sugerida a aplicação da regressão linear para estimar ETMS para cada fase do projeto, e não mais para o total de OSs.

4.5.2 Regressão Baseada em Esforço por Fase

Tendo em vista a necessidade em se estimar o ETMS por fase, aplicou-se então uma regressão linear simples em cada fase do projeto P1, quando considerada apenas a QtdeDoc, visto que não havia como identificar o FP de cada fase.

A Tabela 4.6 apresenta os resultados obtidos, assim, nas estimativas da equipe de projeto a hipótese H_2 foi validada e a hipótese H_0 foi rejeitada. Nas demais estimativas a hipótese alternativa H_2 foi rejeitada.

Tabela 4.6: Escore da Regressão Linear, Quantidade de Documento por fase vs. *Expertise* por fase.

	Quantidade OS	Correlação	Escore <i>Expertise</i>	Escore Regressão Linear
Projeto	491	0,72	6	14
Servidor	636	0,52	12	8
Cliente	432	0,42	15	5
Testes	423	0,68	13	7

O resultado desse experimento foi apresentado aos líderes e gestores do projeto P1 na reunião do dia 27/08/2007. Apesar de resultados promissores com a utilização da regressão linear simples com base em na QtdeDoc, como mostrado na Tabela 4.6, os líderes e os gestores do projeto P1 não ficaram convencidos de que o resultado obtido era confiável, por isso, sugeriram investigar outras técnicas de estimativas.

4.5.3 Hipóteses Validadas com a Fase 1

Nessa primeira fase, os ganhos com as estimativas de ETMS foram significativos no tocante às estimativas do esforço total, porém não tão significantes quando o esforço passou a ser estimado por fase. A Tabela 4.7 apresenta as hipóteses validadas, onde a coluna "Esforço" mostra o tipo de esforço estimado, podendo ser total ou por fase; a coluna "Técnica" mostra quais técnicas foram utilizadas para se estimar o ETMS; a coluna "Atributos" mostra quais atributos foram utilizados pelas respectivas técnicas; e, por fim, a coluna "Hipóteses Válidas" mostra quais hipóteses formuladas foram ratificadas com a aplicação das respectivas técnicas de estimativas de ETMS.

4.6 Fase 2 – Combinação de Técnicas de Estimativas

As estimativas por regressão simples e múltipla, quando aplicadas por fase, não produziram resultados satisfatórios. Em meio a isso, a alternativa foi combinar diferentes técnicas de estimativas com intuito de aumentar o escore em relação à *expertise*, que neste contexto são àquelas estimativas mais próximas do esforço real, pois segundo [LB06] [Pre09] [Som06], utilizar mais de uma técnica pode fazer com que os valores de esforço estimado convirjam para os valores de esforços reais.

Tabela 4.7: Validação das hipóteses para as estimativas de ETMS na fase 1.

Esforço	Técnicas	Atributos	Hipóteses Validadas
Total	Regressão Linear	FP	H_0
		Quantidade de Documentos	H_1
	Regressão Linear Múltipla	Quantidade de Documentos	H_0
Projeto	Regressão Linear Simples	Quantidade Documentos	H_2
Servidor			H_0
Cliente			H_0
Testes			H_0

Assim, como a *expertise* é empírica e não combina diferentes técnicas e, ainda, o uso da regressão linear não produziu resultado satisfatório, há certa evidência de que a precisão das estimativas atuais pode melhorar com a combinação de diferentes técnicas.

Um problema enfrentado na combinação de diferentes técnicas de estimativas está em determinar quais valores estimados são mantidos ou descartados. Em [WF05] são apresentadas técnicas de agrupamento para encontrar similaridade entre objetos. Transpondo essa ideia para as estimativas de ETMS, é possível agrupar os valores de esforço estimado de diferentes técnicas e, a partir disso, escolher os valores mais promissores para serem utilizados na composição do valor de esforço estimado.

Para considerar os indivíduos pertencentes a um mesmo agrupamento, determina-se o grau de similaridade entre os mesmos. Por exemplo, pode-se sugerir um intervalo de 1 a 9 para determinar um grau de similaridade representado pela distância desses indivíduos. Dessa forma, valores de esforço entre 1 e 9 pertencem ao mesmo agrupamento, acima dessa distância os valores de esforço pertencem a outro agrupamento. O cálculo do grau de similaridade é realizado por meio da distância Euclidiana, como sugere [WF05] [LR07]. Logo, a regra geral para se manter os valores de esforço é: "Quanto maior o número de indivíduos em um agrupamento, maior a chance de utilização dos valores de esforço deste agrupamento para compor o estimado de ETMS".

A composição do valor final do ETMS com base na combinação de técnicas de estimativas devem obedecer as seguintes regras:

1. Considerar um ou mais grupos com o maior número de valores agrupados;
2. Caso um valor pertença a dois ou mais agrupamentos, esse valor é considerado em todos os agrupamentos aos quais pertença;
3. Caso os grupos tenham o mesmo número de indivíduos, considera-se o grupo que tem as menores distâncias entre os indivíduos, dado pelo menor valor das somas das distâncias. Selecionando assim, o grupo mais denso (indivíduos mais próximos);

4. Caso a soma de todas das distâncias seja igual para agrupamentos distintos, soma-se apenas as menores distâncias de cada agrupamento, até que seja possível identificar um valor de desempate entre as distâncias somadas;
5. Caso ainda ocorra empate entre os grupos em todas as somas das distâncias, calcula-se a média aritmética da mediana de todos os agrupamentos eleitos, pois a mediana permite selecionar a medida de tendência central de cada agrupamento, já a média aritmética, é um valor médio que nos permite representar todos os outros valores da distribuição;
6. Se os agrupamentos não satisfizerem nenhuma das regras anteriores, calcula-se a média aritmética de todos os valores de esforço estimado.

As regras devem ser aplicadas na sequência sugerida. O resultado da aplicação das regras é um valor único de esforço estimado para cada fase do projeto P1.

4.6.1 Exemplo da Combinação de Técnicas de Estimativas

A Tabela 4.8 apresenta um exemplo fictício para ilustrar o funcionamento das estimativas de ETMS utilizando a combinação de diferentes técnicas de estimativas. Supondo que os dados são de uma OS em uma determinada fase do projeto, na coluna "Técnicas" da respectiva tabela, encontra-se o nome da técnica utilizada; já na coluna "Esforço" é apresentado o valor de esforço estimado pela técnica aplicada.

Tabela 4.8: Técnicas para exemplificar agrupamentos para estimar o ETMS.

Técnicas	Esforço
A	32
B	7
C	3
D	36
E	12
F	77

A Figura 4.3 apresenta as técnicas agrupadas por valores similares, neste caso, um intervalo de 1 a 9, utilizando a distância Euclidiana. Com isso, surgem três grupos distintos: agrupamento 1 $\{B, C, E\}$, agrupamento 2 $\{A, D\}$ e agrupamento 3 $\{F\}$. Aplicando a primeira regra observa-se que o agrupamento 1, composto pelas técnicas $\{B, C, E\}$, e mostrado na Figura 4.3, possui a maior quantidade de indivíduos para compor o valor de ETMS. Os demais agrupamentos são descartados e o total de esforço estimado é a média dos valores das técnicas $\{B, C, E\}$ que foi de 7,33 horas.

Caso ocorra um empate na quantidade de valores agrupados, o desempate é realizado pela soma dos valores das distâncias de cada agrupamento. Para apresentar esse exemplo foi inserida na Tabela 4.9 uma nova técnica chamada de G , com valor estimado de 40 horas.

Pelo cálculo das distâncias descobre-se que a técnica G pertence ao agrupamento 2. Existem então dois agrupamentos com o mesmo número de indivíduos. A segunda regra não se aplica nesse

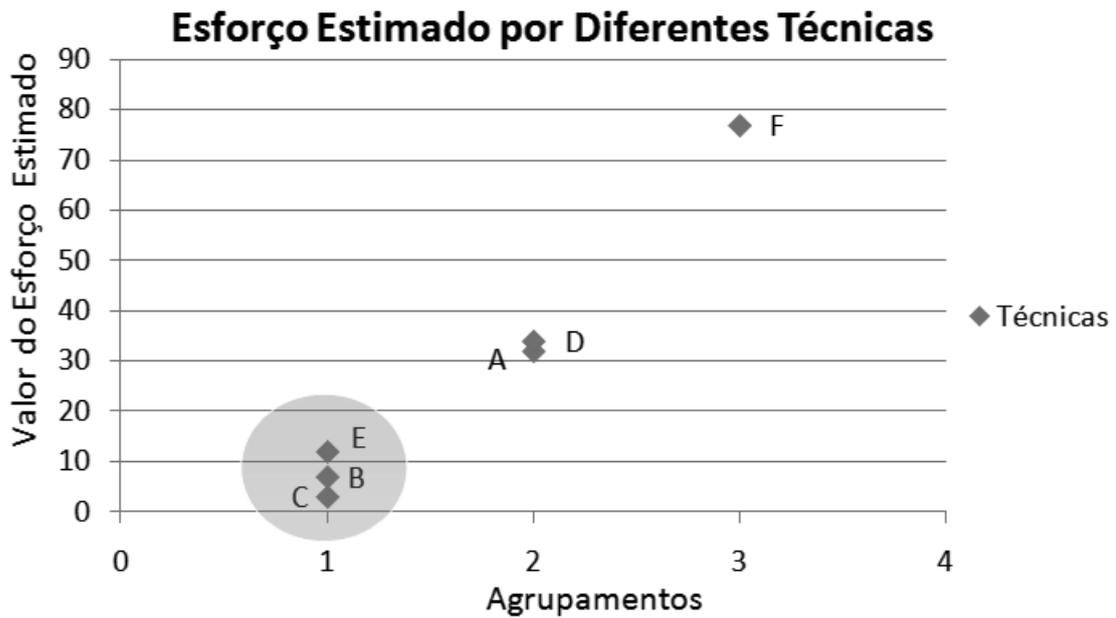


Figura 4.3: Exemplo dos agrupamentos encontrados entre as técnicas de estimativas aplicadas.

Tabela 4.9: Exemplo de nova técnica nos agrupamentos para estimar o ETMS.

Técnicas	Esforço
A	32
B	7
C	3
D	36
E	12
F	77
G	40

caso porque os agrupamentos são distintos. Assim, aplicando a terceira regra, é efetuada a soma das distâncias de cada grupo e mantido o grupo que tiver a menor distância entre seus indivíduos, como mostra a Tabela 4.10. Já as regras 4, 5 e 6 não são aplicadas.

Com isso, descobre-se que o agrupamento 2 é mais denso que o agrupamento 1. Portanto, as técnicas $\{A, D, G\}$ são selecionadas para sugerir o valor estimado de ETMS, encontrado por meio da média dos valores de esforço estimado das técnicas pertencentes ao agrupamento 2, que para este exemplo foi de 36,33 horas.

Tabela 4.10: Exemplo da aplicação da terceira regra.

Agrupamento	Indivíduos	Valor das Distâncias	Soma das Distâncias
Agrupamento 1	Distância {B, C}	4	18
	Distância {C, E}	9	
	Distância {B, E}	5	
Agrupamento 2	Distância {A, D}	4	17
	Distância {D, G}	5	
	Distância {A, G}	8	

A quarta regra é aplicada quando ocorre um empate entre o valor da soma das distâncias de diferentes agrupamentos mas de mesmo número de indivíduos. Quando isso ocorre, retira-se os indivíduos com as maiores distâncias de cada agrupamento, verificando se há ou não empate na soma das distâncias restantes. Caso ainda persista o empate, retira-se então, a segunda maior distância. Essa iteração se segue até que não haja mais empate nas distâncias, ou até que os indivíduos dos agrupamentos sejam únicos. A Tabela 4.11 apresenta um exemplo dos valores de esforço para a aplicação da quarta regra.

Tabela 4.11: Exemplo de valores de esforço para aplicar a quarta regra.

Técnicas	Esforço
A	32
B	7
C	3
D	36
E	12
F	77
G	41

Na Figura 4.4, pode-se observar dois agrupamentos de indivíduos com os valores estimados de ETMS, encontrados pela combinação de diferentes técnicas de estimativas.

Aplicando a primeira, segunda e terceira regras, encontra-se um empate entre a soma dos valores das distâncias de cada agrupamento, como mostra a Tabela 4.12.

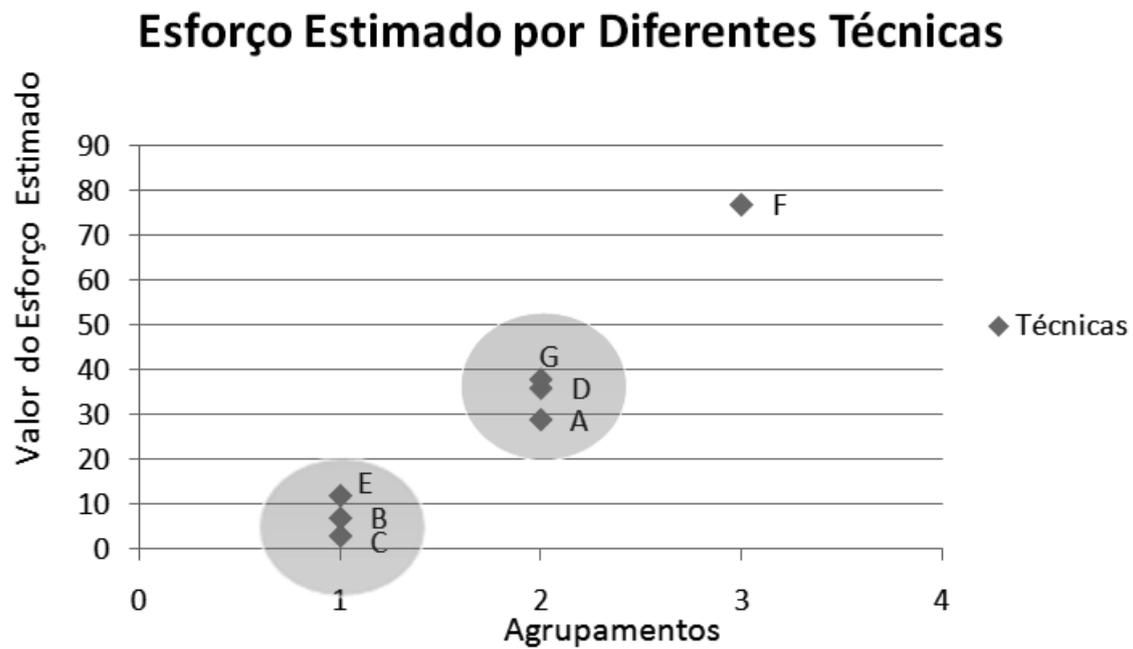


Figura 4.4: Exemplo dos agrupamentos encontrados para a aplicação da quarta regra.

Tabela 4.12: Exemplo da aplicação da quarta regra.

Agrupamento	Indivíduos	Valor das Distâncias	Soma das Distâncias
Agrupamento 1	Distância {B, C}	4	18
	Distância {C, E}	9	
	Distância {B, E}	5	
Agrupamento 2	Distância {A, D}	6	18
	Distância {D, G}	3	
	Distância {A, G}	9	

Com a aplicação da quarta regra, desconsideram-se os maiores valores de distância. Neste exemplo, no agrupamento 1 descarta a distância entre $\{C, E\}$ e no agrupamento 2 descarta a distância entre $\{A, G\}$, ambos com o valor de 9. A Tabela 4.13 apresenta os novos valores da soma das distâncias.

Tabela 4.13: Continuação da aplicação do quarta regra.

Agrupamento	Indivíduos	Valor das Distâncias	Soma das Distâncias
Agrupamento 1	Distância {B, C}	4	9
	Distância {B, E}	5	
Agrupamento 2	Distância {A, D}	6	9
	Distância {D, G}	3	

Ainda não se sabe qual o agrupamento mais denso, pois a soma das distâncias continua tendo o mesmo valor. Sendo assim, descarta-se o maior valor das distâncias novamente, neste caso $\{B, E\}$ para o agrupamento 1 e $\{A, D\}$ para o agrupamento 2. Então, os valores de distância restantes são: 4, distância entre $\{B, C\}$ e 3 entre $\{D, G\}$ dos agrupamentos 1 e 2 respectivamente.

O agrupamento 2 é considerado o mais denso, pois existe uma menor distância entre seus indivíduos. O valor estimado de esforço de trabalho total é calculado pela média dos valores encontrados por cada técnica de estimativas dada pelo agrupamento escolhido, agrupamento 2, dado por $\{A, D, G\}$. Dessa forma, o valor total de esforço estimado foi de 37 horas.

A quinta regra garante que, caso haja empate em todas as distâncias, valores médios dos agrupamentos podem sugerir um bom valor de esforço por meio da média aritmética das medianas dos agrupamentos selecionados. Para exemplificar esse cenário são considerados os seguintes valores estimados de ETMS apresentados na Tabela 4.14.

Tabela 4.14: Valores de esforço para aplicar a quinta regra.

Técnicas	Esforço
A	32
B	7
C	3
D	36
E	12
F	77
G	33

A Tabela 4.15 mostra um exemplo de empate que pode existir entre as distâncias dos dois agrupamentos identificados pelo primeiro critério.

Tabela 4.15: Exemplo da aplicação da quinta regra.

Agrupamento	Indivíduos	Valor das Distâncias	Soma das Distâncias
Agrupamento 1	Distância $\{B, C\}$	4	18
	Distância $\{C, E\}$	9	
	Distância $\{B, E\}$	5	
Agrupamento 2	Distância $\{A, D\}$	9	18
	Distância $\{D, G\}$	5	
	Distância $\{A, G\}$	4	

Dessa forma, o valor da mediana dos valores de esforço estimado de cada agrupamento são, respectivamente, 7 horas do agrupamento 1 e 33 horas do agrupamento 2. O valor do esforço estimado é calculado pela média entre 7 e 33, e seu valor final é de 20 horas.

A sexta regra é utilizada quando todos os agrupamentos são compostos por valores únicos. Isso indica que a distância entre esses valores é maior do que o intervalo de 1 a 9 definido. Esse exemplo é mostrado na Tabela 4.16.

Tabela 4.16: Valores de esforço para aplicar a sexta regra.

Técnicas	Esforço
A	48
B	21
C	33
D	7
E	60
F	120
G	98

Neste caso, é calculada a média aritmética de todos os valores e sugerida como valor estimado de ETMS da demanda. O valor calculado nesse exemplo foi de 55,2 horas.

4.6.2 Resultados Alcançados

Para aplicar a combinação de várias técnicas foram seguidos os seguintes passos:

1. Conjunto de treino: composto por um conjunto de OSs advindas da base histórica do projeto, desde sua implantação até o ano de 2007. A Tabela 4.17 apresenta um resumo com as técnicas de estimativas aplicadas, as hipóteses a serem ratificadas, as suas fases e a quantidade de demanda utilizada para compor o conjunto de treino;
2. Conjunto de testes: o conjunto de testes foi composto pelas 20 mais recentes OSs do projeto P1. Onde estas foram requisitadas em novembro de 2007.

Tabela 4.17: Resumo da aplicação das técnicas de estimativas de ETMS em dados do projeto P1.

Técnicas	Hipóteses	Fases	Quantidade de Demandas
Regressão Linear	H_1	-	523
	H_2	Projeto	491
		Cliente	367
		Servidor	432
		Testes	423
H_3	-	336	
Regressão Linear Múltipla	H_4	-	355
M5P e REPTree	H_1	-	636
	H_2	Projeto	491
		Cliente	367
		Servidor	432
		Testes	423
	H_3	-	573
H_4	-	355	
% Esforço Equipe Quartil 1 (25%) Mediana Quartil 3 (75%)	H_5	Projeto	229
		Cliente	
		Servidor	
		Testes	

A Tabela 4.18 apresenta os escores obtidos com a utilização da combinação de várias técnicas de estimativas. Nota-se que, para todas as fases do projeto P1, os resultados por meio da combinação de várias técnicas de estimativas foi significativamente superior aos das estimativas realizadas por meio da *expertise*. Isso significa que a hipótese H_5 foi validada e a hipótese nula H_0 rejeitada.

Tabela 4.18: Escore final da combinação de técnicas de estimativas por fase.

	Quantidade OS	<i>Expertise</i>	Estimativas com Várias Técnicas
Projeto	20	7	13
Servidor	14	6	8
Cliente	18	8	10
Testes	19	8	11

A partir dos resultados alcançados os gestores e líderes técnicos do projeto pediram que fosse implementada uma planilha utilizando a combinação de técnicas de estimativas, pois estas tiveram resultados muito mais promissores ao trabalho realizado na Fase 1, apresentada na seção 4.5.

Sendo assim, as planilhas foram implementadas e utilizadas para realizarem estimativas de ETMS para o projeto P1. Apesar disso, os gerentes do projeto pediram um estudo de técnicas de estimativas que não se baseasse somente nos dados históricos. O principal argumento dos gestores do projeto foi o de validar se realmente não havia nenhuma outra técnica que pudesse produzir resultados melhores que a combinação das técnicas de estimativas.

4.6.3 Hipóteses Validadas com a Fase 2

A Tabela 4.19 apresenta as hipóteses validadas nesta fase, quando os ganhos foram mais significativos que na primeira fase.

Tabela 4.19: Validação das hipóteses para as estimativas de ETMS na Fase 2.

Esforço	Técnicas	Atributos	Hipóteses Validadas
Projeto	Regressão Linear Simples e Múltipla, <i>M5P</i> ,	Quantidade Documentos	H_5
Servidor	<i>REPTree</i> , %Esforço por Equipe, Quartil1,		
Cliente	Mediana, Quartil3		
Testes			

4.7 Fase 3 – Verificação da Complexidade em Documentos de Projeto

O projeto P1 possui uma série de documentos estruturados com seções bem definidas, contendo todas as regras de negócio e todos os pseudocódigos da aplicação. O time de projeto é o responsável por fazer a criação e a manutenção desses documentos. Em 14/10/2007 foi dado início aos trabalhos de estimativas com base na complexidade da alteração de cada documento de projeto associado a uma demanda. Os tipos de documentos existentes no projeto P1 são: Documento de Projeto (DD), Documento de Interface com o Usuário (DEIG) e Documento de Módulos. Nesta fase é utilizada a hipótese H_6 , apresentada na seção 4.4 deste capítulo.

4.7.1 Classificação de Alterações em Documentos

Uma classificação de complexidade baseada nos direcionadores de custos do COCOMO 2.0, apresentada no Apêndice B, é proposta por [Cla96]. Essa classificação se mostrou apropriada para ser utilizada nos documentos do projeto P1 porque sua proposta se parece com a forma como os membros da equipe realizavam suas estimativas.

Cada uma das classificações foi mapeada para dentro dos documentos de projeto. Esse mapeamento foi realizado com a ajuda do líder técnico e membros da equipe de projeto, e os líderes técnicos das equipes de Cliente, Servidor e Testes, além do SQA do projeto. Para a realização dessa fase ocorreram 15 encontros semanais quando foram desenvolvidas as seguintes atividades:

- Mapeamento da classificação proposta por [Cla96] para cada tipo de funcionalidade em cada tipo de documento do projeto P1;
- Classificação da complexidade das alterações em cada tipo de documento do projeto;
- Escrita de um manual explicativo para classificar e contar a complexidade das alterações dos documentos;
- Criação de uma interface para o cálculo das estimativas baseada na complexidade das alterações nos documentos de projeto;
- Apresentação da interface para os líderes de equipe e coleta das sugestões de melhorias; e
- Implantação do projeto piloto para coletar dados e avaliar as estimativas pela complexidade da alteração no documento.

O Apêndice C apresenta um exemplo do guia de classificação da complexidade dos documentos, composto por:

- nome do documento no cabeçalho;
- função, onde sua complexidade foi determinada por meio de reuniões com membros da equipe;
- classificação da complexidade de alteração dada por cada equipe para a função;
- granularidade definida para a complexidade da alteração; e
- alteração impactada na funcionalidade denominada de OBMaker, responsável pela geração de código e que tem uma contagem própria que deve ser considerada para documentos de projeto (Apêndice C.4 – Regra de contagem para OBMaker).

Os fatores de ponderação foram considerados como frações de horas e mapeados para tal, como apresenta a Tabela 4.20. Além disso, nas reuniões os líderes de equipe afirmaram que o tempo mínimo de uma demanda era de 30 minutos, portanto, foi criado o fator de ponderação "Extra Baixo" com um valor de 0,5 horas para representar demandas pequenas.

Para utilizar a técnica com base em complexidade de documentos, os seguintes passos foram seguidos:

1. Conhecer as alterações que por ventura venham a ser realizadas nos documentos;
2. Encontrar e analisar os documentos impactados pela demanda que se deseja estimar o ETMS;

3. Classificar e contar as alterações a serem realizadas por meio da técnica aqui apresentada; e
4. Multiplicar os valores classificados e contados por seus respectivos fatores de ponderação.

Tabela 4.20: Fatores de complexidade para contagem de documentos.

Complexidade	Valor	Tempo
Extra Baixo	0,50	30 min.
Muito Baixo	0,75	45 min.
Baixo	0,88	53 min.
Nominal	1,00	1 hora
Alto	1,15	1h 10 min.
Muito Alto	1,30	1h 20 min.
Extra Alto	1,65	1h 40 min.

Ao final, o valor de ETMS de uma OS é dado pela soma dos fatores de ponderação para cada tipo de alteração do documento.

4.7.2 Resultados Alcançados

Devido ao grande esforço necessário para a realização da classificação de alterações por documento em todas as fases do projeto P1, o escopo do experimento foi reduzido a 19 demandas e somente à fase de projeto. Os resultados estão na Tabela 4.21.

Tabela 4.21: Resultados obtidos com a simulação da complexidade de documentos.

Fase	Quantidade OS	Escore <i>Expertise</i>	Escore Complexidade Documentos
Projeto	19	5	14

Os resultados foram apresentados em 22/01/2008 e muito bem recebidos pela equipe do projeto, pois o esforço estimado de trabalho chegou muito próximo do esforço real, quando comparado às técnicas aplicadas anteriormente.

Contudo, esta técnica pode não ter bons resultados em outras fases, já que nas fases anteriores as melhores estimativas alcançadas eram na fase de Projeto. Essa técnica chegou a ser implantada como um piloto no projeto P1 mas por onerar muito os membros da equipe, os gerentes do projeto decidiram por abandoná-la e ficaram com a combinação de várias técnicas.

Esta técnica de classificação das alterações nos documentos revelou que, muitas vezes, o projetista "gasta" exatamente o valor estimado do ETMS para realizar as alterações em uma demanda. Essa informação foi confirmada pelos líderes de equipe do projeto em reuniões formais.

Este fato é explicado por [Gol97], como "Síndrome de Estudante", quando a pessoa se comporta como um estudante, deixando sempre para executar suas tarefas na última hora e utilizando todo o tempo estimado para realizá-la.

4.7.3 Hipóteses Validadas com a Fase 3

Os ganhos com as estimativas de ETMS nessa terceira fase, como na segunda fase, foram significativos. A Tabela 4.22, apresenta as hipóteses validadas utilizando a complexidade de documentos para estimar o valor de ETMS.

Tabela 4.22: Validação da hipótese para estimar o valor de ETMS na fase 3.

Esforço	Técnicas	Atributos	Hipóteses Validadas
Projeto	Complexidade de Documentos	N/A	H_6

4.8 Síntese dos Trabalhos Realizados

No momento da realização do estudo de caso, a sugestão das estimativas de ETMS é baseada na análise dos artefatos de software, quando o projetista avalia empiricamente o esforço necessário para realizar as alterações em uma demanda. Na tentativa de melhorar a precisão das estimativas e acabar com o empirismo no projeto P1, foram sugeridas técnicas de estimativas apresentadas neste capítulo e alinhadas com os requisitos definidos pelos seus gestores. Uma síntese dos trabalhos realizados é apresentada na Tabela 4.23.

Tabela 4.23: Resumo analítico das técnicas utilizadas no estudo de caso.

Técnica	Aplicação	Implantada	Vantagens/ Desvantagens	Finalidade
Regressão	-Estimar o esforço total de uma demanda. -Estimar o esforço de uma demanda por equipe.	Sim, aplicada em conjunto com outras técnicas.	-Vantagens: simplicidade de utilização, fácil calibragem, fácil entendimento, desonera membros da equipe de projeto. -Desvantagens: sensível à <i>outliers</i> , dificuldade em identificar os atributos, necessidade de muitos dados.	Auxiliar no cálculo do % de esforço por equipe e dos quartis (Q1 e Q3), aplicado na seção 4.4.
Regressão Múltipla	-Estimar o esforço total de uma demanda.			Calcular o valor de esforço estimado por equipe com base nos FP e Quantidade de Documentos.
<i>M5P</i>				
<i>REPTree</i>		Não, pois monta a árvore de decisão montada pode ser muito complexa, de difícil interpretação e implementação, dependendo da quantidade de dados.	-Vantagens: simplicidade de utilização, fácil calibragem, desonera membros da equipe de projeto. -Desvantagens: dificuldade em identificar os atributos, necessidade de muitos dados.	Montar uma árvore de decisão com os dados do esforço por equipe.
% Esforço por Equipe		Sim, aplicada em conjunto com outras técnicas.	-Vantagens: simplicidade de utilização, fácil calibragem, desonera membros da equipe de projeto. -Desvantagens: grande esforço na preparação dos dados, sensível a <i>outliers</i> .	Calcular o esforço médio gasto por equipe para implantar uma demanda.
Quartil 1	-Estimar o esforço de uma demanda por fase.			Calcular o percentual de esforço com base nos menores valores esforço gasto por fase para finalizar uma demanda.
Mediana				Calcular o percentual de esforço, com base na mediana dos esforços de demandas finalizadas.
Quartil 3				Calcular o percentual de esforço, com base nos maiores valores de esforço gasto por equipe para finalizar uma demanda.
Complexidade de Documentos				-Vantagens: não é sensível a <i>outliers</i> , não necessita de dados de projetos. -Desvantagens: dificuldade na utilização, difícil calibragem, dependência de intrusão humana, não desonera membros da equipe de projeto.

4.9 Considerações Finais do Estudo de Caso

As técnicas de estimativas exploradas para este estudo de caso contribuem para um projeto de software à medida em que se consegue boas aproximações do esforço de trabalho real e, ainda, desonera equipes, diminuindo a relocação de recursos para realizar estimativas.

As lições aprendidas com a aplicação deste estudo de caso foram:

1. O esforço estimado deve ser numérico: o esforço tem que ser um valor numérico, dado em horas ou em dias, e não categórico, como: alto, médio e baixo, por exemplo. Isso porque um valor de hora deve ser assumido para ser colocado em um cronograma de acompanhamento de demandas. Sendo assim, algumas técnicas de mineração de dados, como a classificação, não podem ser utilizadas para se estimar ETMS neste cenário;
2. As OSs pequenas distorcem as estimativas: observou-se que demandas pequenas, demandas de até 8 horas, consomem um tempo para serem estimadas que pode levar até 50% do tempo de trabalho necessário para finalizar a demanda. Neste caso, descobriu-se que os projetistas, recursos responsáveis pelas estimativas, lançavam o valor trabalhado como estimado, distorcendo alguns dos resultados obtidos;
3. A independência de fatores humanos é ideal: como uma demanda levava exatamente a quantidade de tempo estimada para ser implementada, descobriu-se a "Síndrome de Estudante". Sendo assim, ficou claro que quanto mais as estimativas forem independentes de fatores humanos, menos esse problema aparecerá;
4. As técnicas baseadas em dados históricos permitem realimentação: quando se trabalha com base nos dados de projetos pode-se calibrar o modelo preditivo constantemente, de forma iterativa, assim, o modelo preditivo sempre se mostrará atualizado, havendo uma grande possibilidade de convergência com o esforço real; e
5. A combinação e o agrupamento de diferentes técnicas de estimativas é promissora: combinar diferentes técnicas de estimativas, bem como agrupar os valores considerando sua similaridade pela distância euclidiana, é uma forma de descobrir um conjunto de valores para serem utilizados nas estimativas de ETMS. Assim, acredita-se que, quanto mais técnicas e mais valores próximos agrupados, maior a possibilidade de convergência para o esforço real de trabalho.

Por fim, o resultado deste estudo de caso exploratório foram duas publicações no *SEW'08 (Software Engineering Workshop)*. Os artigos publicados foram: *A Quasi-Experiment for Effort and Defect Estimation using Least Square Linear Regression and Function Points* [RTR08] e *Issues on Estimating Software Metrics in a Large Software Operation* [BBT⁺08].

5. Modelo–E10

As atuais técnicas de estimativas, apresentadas pela literatura, são utilizadas pela indústria de software de forma isolada, o que pode comprometer a precisão das estimativas [LB06] [Som06]. Quando se utiliza mais de um modelo de estimativas a probabilidade de melhorar a precisão das mesmas aumenta significativamente [LB06] [Pre09].

Portanto, para responder a questão de pesquisa apresentada no Capítulo 1 seção 1.2, o presente capítulo apresenta o Modelo–E10, um modelo de estimativas capaz de alcançar boas aproximações do esforço de trabalho estimado sem depender da *expertise* de pessoas e, ainda, flexível a ponto de combinar diferentes técnicas de estimativas e de se autoajustar para melhorar estimativas futuras.

5.1 Requisitos para Utilização do Modelo–E10

Para a utilização do Modelo–E10, os seguintes requisitos devem ser satisfeitos:

- **Definição do Nível de Exigência:** o "Nível de Exigência" é um percentual parametrizável que indica a tolerância de erro do valor do esforço estimado em relação ao valor de esforço real das demandas. Os gerentes de projeto devem definir qual é o "Nível de Exigência" das estimativas do Modelo–E10. Esse parâmetro permite definir se um valor estimado para uma demanda é considerado bom ou ruim. O Modelo–E10 considera boas estimativas se o esforço estimado para uma demanda, em relação ao esforço real da mesma, estiver dentro do Nível de Exigência. Inversamente, o modelo considera estimativas ruins, aquelas onde o esforço estimado da demanda, também em relação ao seu esforço real, estiver fora do Nível de Exigência. Por exemplo, se o Nível de Exigência é de 20%, um bom valor estimado é qualquer valor de esforço que esteja dentro dos 20% em relação ao esforço real da demanda. Os valores que estiverem fora desses 20%, tanto para cima quanto para baixo, são considerados valores estimados ruins;
- **Coleta de Dados do Projeto:** a coleta regular dos dados do projeto que se deseja estimar é essencial para o bom funcionamento do Modelo–E10, pois ela mantém os dados do projeto sempre atualizados e o Modelo–E10 depende destes para realizar suas estimativas;
- **Métricas de Projeto Coletadas:** para iniciar a utilização do modelo deve-se coletar no mínimo o Tamanho e o Esforço Real das demandas. Quanto mais métricas forem coletadas melhor é para explicar a relação entre as métricas e o esforço a estimar. Quanto mais forte a relação entre as métricas e o esforço, melhor a possibilidade de precisão do modelo. Caso existam muitas métricas, o cálculo da correlação estatística entre cada métrica e o esforço real da demanda pode auxiliar na escolha das métricas para compor o modelo [Moo04];

- **Ter Volume de Dados Estatisticamente Significativo:** para a utilização do Modelo–E10 são necessários dados de pelo menos 30 demandas finalizadas. Esses dados são utilizados pelas técnicas de estimativas incorporadas ao modelo. Esse valor de 30 demandas torna os dados do modelo estatisticamente relevante [Moo04];
- **Escolha das Técnicas de Estimativas:** o Modelo–E10 é uma proposta flexível projetada para agregar diferentes técnicas de estimativas ou retirar àquelas que não estejam tendo um bom desempenho. Na seção 5.3 é realizada uma discussão de como agregar ou retirar técnicas de estimativas para serem utilizadas pelo Modelo–E10, bem como uma sugestão de quais técnicas podem ser adotadas inicialmente no modelo;
- **Adotar no mínimo três técnicas:** para o bom funcionamento do Modelo–E10, são necessárias no mínimo três técnicas de estimativas. Essa quantidade de técnica é sugerida por [LB06] [Pre09]. Três técnicas geram no mínimo três modelos preditivos, o que aumenta a quantidade de valores de ETMS do modelo, e possibilita boas aproximações do esforço real da demanda; e
- **Definição de uma Ferramenta para aplicar as Técnicas de Estimativas:** o Modelo–E10 utiliza técnicas de estimativas oferecidas por ferramentas de terceiros. Como o modelo se baseia em dados de projetos, as técnicas de estimativas mais utilizadas são àquelas baseadas em estatística e mineração de dados. Porém, não há nenhuma restrição para utilização de outras técnicas de estimativas, como o COCOMO por exemplo. A ferramenta sugerida para a aplicação de técnicas de estimativas baseadas em mineração de dados e estatística é a ferramenta *WEKA* [UW01], a qual é gratuita e possui uma gama de algoritmos de mineração que podem ser amplamente utilizados para estimar o ETMS. Caso essas ferramentas não ofereçam alguma das técnicas de estimativas que se deseje utilizar, a sugestão é que se implemente tal técnica pela parte interessada em utilizá-la.

5.2 Apresentação

O Modelo–E10 é um modelo com o propósito de estimar o ETMS. O modelo pode trabalhar com dois tipos de fontes de dados: "Dados de Projetos" e "Dados de Outras Técnicas". Os "Dados de Projetos" são aqueles coletados por um projeto de uma organização e armazenados em um ou mais repositórios de dados. Geralmente um banco de dados ou ferramentas diversas para acompanhamento do projeto. É sobre os "Dados de Projetos" que atuam as técnicas de estimativas para tentar encontrar um valor do ETMS próximo ao esforço real.

Por sua vez, os "Dados de Outras Técnicas" são gerados por técnicas externas às utilizadas pelo Modelo–E10, como por exemplo: o COCOMO 2.0 ou o *Planning Poker* do SCRUM. Há apenas duas exigências para a utilização dos valores estimados advindos de técnicas externas: i) que os valores gerados por essas técnicas sejam numéricos; ii) que o valor do esforço de diferentes técnicas

de estimativas esteja expresso em uma única unidade de tempo, podendo ser em meses, dias, horas ou minutos.

A Figura 5.1 apresenta o Modelo–E10, onde o mesmo é constituído de oito fases, sendo elas:

1. **Dados para Estimativas** – Nesta fase os Dados de Projetos devem ser devidamente extraídos e disponibilizados para o Modelo–E10;
2. **Retroalimentação** – Nesta fase, os dados das mais recentes e melhores estimativas são agregados aos dados das estimativas dos períodos anteriores;
3. **Calibragem** – Nesta fase os resultados da Retroalimentação são devidamente armazenados em um repositório de calibragem e uma avaliação dos melhores modelos preditivos é realizada;
4. **Técnicas** – Nesta fase diferentes técnicas de estimativas são utilizadas pelo Modelo–E10;
5. **Modelos** – Nesta fase são aplicados os modelos preditivos encontrados pelas técnicas utilizadas;
6. **Agrupamentos** – Nesta fase os valores gerados pelos modelos preditivos são agrupados conforme o Nível de Exigência definido;
7. **Seleção** – Nesta fase os agrupamentos com maior número de indivíduos são selecionados; e
8. **Estimativas** – Nesta fase os valores de um ou mais agrupamentos são devidamente combinados para comporem um valor único de ETMS.

Nas seções subsequentes são detalhadas cada uma das fases descritas acima.

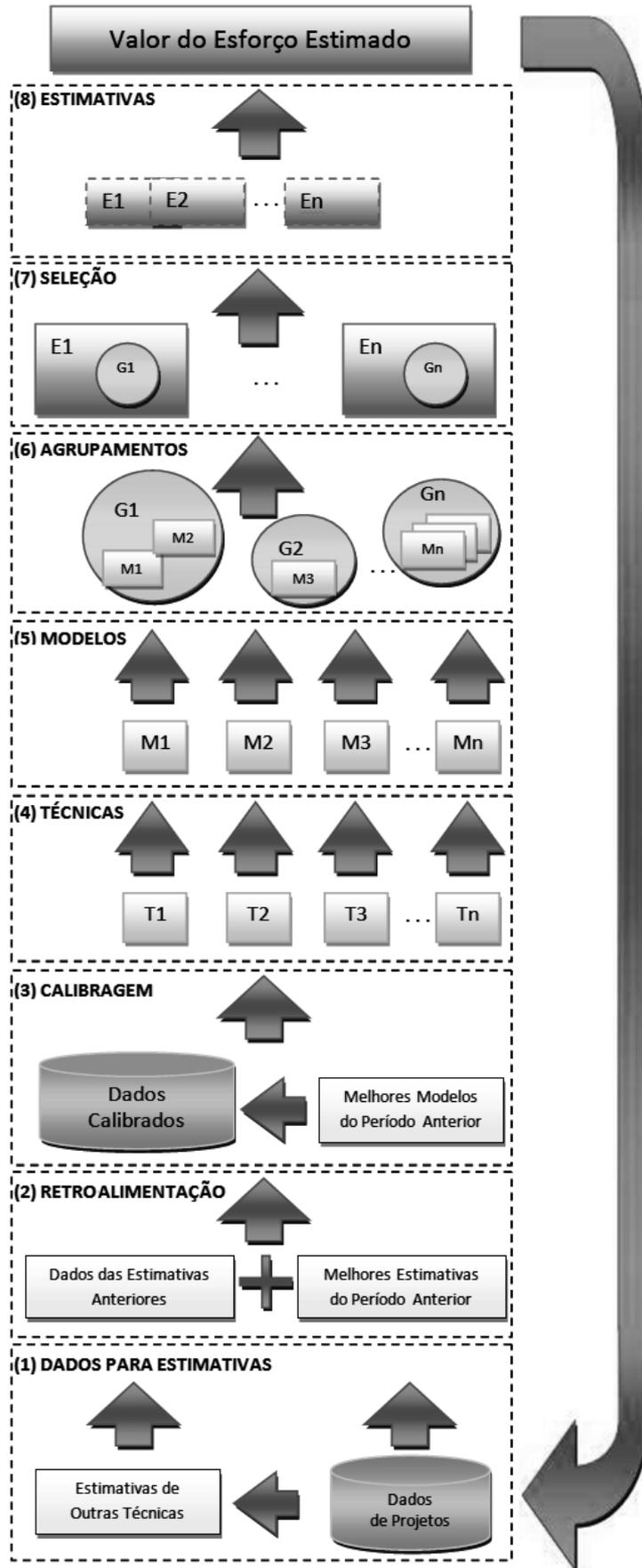


Figura 5.1: Modelo-E10 para estimativas de esforço.

5.2.1 Dados para Estimativas

Os dados utilizados para estimar o ETMS podem estar devidamente armazenados em uma base de dados centralizada, como nas organizações avaliadas no nível 2 do CMMI; ou não, como é o caso de organizações que gerenciam projetos de software utilizando ferramentas para controle de cronograma ou planilhas para acompanhamento de atividades, prazos e custos.

Assim, os dados devem ser coletados de forma a possibilitarem fácil acesso e não necessariamente precisam estar em um banco de dados organizacional ou de projeto. Contudo, quando estes dados estão disponibilizados em um repositório central a extração destes é facilitada.

A saída desta fase é um conjunto de Dados de Projetos, que pode ser: identificador da demanda, esforço real, FP e QtdeDoc, por exemplo e, em se tratando de valores advindos de Estimativas de Outras Técnicas, pode se ter, além dos supracitados, o esforço estimado pela técnica externa ao Modelo-E10. Esses valores de esforço são endereçados à fase de Retroalimentação do Modelo-E10.

5.2.2 Retroalimentação

A Retroalimentação é a união dos dados de todas as demandas anteriormente estimadas, com os dados das demandas que obtiveram melhores estimativas do período imediatamente anterior ao que se está estimando. Neste contexto, um período é uma fatia de tempo definida pela organização. Esses dados, quando unidos, são utilizados pelas técnicas de estimativas para gerar os modelos preditivos a serem utilizados nas estimativas do período corrente. Portanto, a Retroalimentação é uma forma de garantir que as técnicas de estimativas aplicadas gerem os modelos preditivos para o Modelo-E10 com base no esforço real atualizado das demandas, que são àqueles definidos na Seção 5.1 item "Métricas de Projeto Coletadas".

A Figura 5.2 ilustra a Retroalimentação. A calibragem inicial, representada na Figura 5.2(a), contém os dados das demandas anteriores que estejam dentro do Nível de Exigência do projeto. A Figura 5.2(b) representa as estimativas de esforço do período corrente baseadas na calibragem inicial. Após a entrega das demandas, seleciona-se as estimativas que estiverem dentro do Nível de Exigência do projeto sendo estas agregadas aos dados das demandas anteriores, e formando um novo conjunto de dados para serem utilizados nas estimativas do próximo período, como apresenta a Figura 5.2(c).

ID Demanda	Esforco_Real	FP	QtdeDoc
OS-001	20	10	6
OS-002	54	17	10
OS-003	35	12	8
OS-004	43	10	14

(a) Dados da calibragem inicial.

ID Demanda	Esforco_Real	FP	QtdeDoc
OS-008	12	10	4
OS-012	55	7	16
OS-014	40	9	12
OS-015	38	11	7

(b) Dados das boas estimativas.

ID Demanda	Esforco_Real	FP	QtdeDoc
OS-001	20	10	6
OS-002	54	17	10
OS-003	35	12	8
OS-004	43	10	14
OS-012	55	7	16
OS-015	38	11	7

(c) Dados após a retroalimentação.

Figura 5.2: Exemplo da Retroalimentação.

A Retroalimentação é uma das principais vantagens do modelo porque com as estimativas atualizadas cresce a possibilidade do modelo estimar o ETMS convergindo para o valor do esforço real das demandas. Além disso, as técnicas de estimativas atualmente difundidas no mercado, como o *Planning Poker*, não oferecem a funcionalidade de Retroalimentação, e outros processos devem ser realizados em paralelo, como por exemplo, armazenar os valores estimados e aplicar técnicas de estimativas sobre esses dados. Já o COCOMO, também difundido no mercado, pode ser retroalimentado e calibrado com dados atualizados do projeto, porém, esse processo não é trivial de ser realizado.

A saída desta fase é um conjunto de dados, mostrado na Figura 5.2(c), para ser armazenado no repositório da fase Calibragem, denominado Dados Calibrados.

5.2.3 Calibragem

Na fase de Calibragem duas ações são realizadas. A primeira delas é o armazenamento dos dados advindos da fase Retroalimentação. A segunda é a avaliação dos modelos preditivos, gerados pela aplicação das técnicas de estimativas apresentada na seção 5.2.4, e que estimaram dentro do Nível de Exigência do projeto. Ambas as informações são armazenadas no repositório Dados Calibrados e são utilizadas para realizar as estimativas de demandas do próximo período.

A avaliação dos modelos preditivos é realizada em duas etapas: na primeira é calculado, para cada demanda estimada, o percentual do valor de esforço estimado em relação ao esforço real, como mostra a Equação 5.1.

$$\%EsforcoEstimado = \frac{\|ValorEsforcoReal - ValorEsforcoEstimado\|}{ValorEsforcoReal} \quad (5.1)$$

A segunda etapa é a contagem da quantidade das demandas, cujo valor do percentual de esforço estimado está dentro do Nível de Exigência do projeto. Essa contagem é chamada de fator de ponderação e, quanto maior esse fator, maior será o peso do modelo preditivo no próximo período

de estimativas. Supondo que um conjunto de seis demandas tenham tido as aproximações do esforço real apresentadas na Tabela 5.1, e o Nível de Exigência seja de 20%, os modelos preditivos "M5P-Tamanho" e "M5P-QtdeDoc" terão peso 4, e o modelo preditivo "RL-QtdeDoc" terá peso 1 para o próximo período de estimativas.

Tabela 5.1: Exemplo da calibragem de modelos preditivos.

ID Demanda	Esforço Real	Esforço Estimado dos Modelos Preditivos			Avaliação dos Modelos Preditivos		
		M5P-Tamanho	M5P-QtdeDoc	RL-QtdeDoc	M5P-Tamanho	M5P-QtdeDoc	RL-QtdeDoc
OS-030	20	18	15	10	0,10	0,16	0,33
OS-035	25	10	17	28	0,60	0,70	0,64
OS-036	18	22	25	13	0,20	0,13	0,48
OS-041	44	49	50	82	0,11	0,02	0,64
OS-042	23	15	12	20	0,34	0,20	0,66
OS-043	78	79	77	89	0,01	0,02	0,15
Peso dos Modelos Preditivos					4	4	1

Só é possível avaliar os modelos preditivos de demandas que já foram entregues. Também é importante salientar que na primeira execução do Modelo-E10, o peso de todos os modelos preditivos tem o valor de 1.

Não há atualmente nenhuma técnica de estimativas difundida no mercado que realize a avaliação constante dos modelos preditivos por ela gerados, fazendo desta, uma contribuição importante do Modelo-E10 para a área de estimativas. O resultado desta fase é o armazenamento dos dados das estimativas, bem como o valor do peso de cada modelo preditivo, onde este último é passado à fase de Agrupamentos.

5.2.4 Técnicas

A escolha das técnicas é um dos requisitos para a adoção do modelo, como apresentado na seção 5.1. Sendo assim, nesta fase são aplicadas as diferentes técnicas de estimativas sobre os dados do projeto, armazenados no repositório Dados Calibrados. O valor das métricas do projeto são os dados de entrada das técnicas. Logo, para a utilização das técnicas torna-se necessário informar o valor das métricas coletadas (variáveis independentes), para que as mesmas gerem o modelo preditivo. No exemplo da Figura 5.3, é mostrada uma técnica onde se informa a variável independente FP e a dependente Esforço Real. Quando aplica-se a técnica sobre os dados de projeto é produzido um modelo preditivo que, neste caso, é uma equação linear.

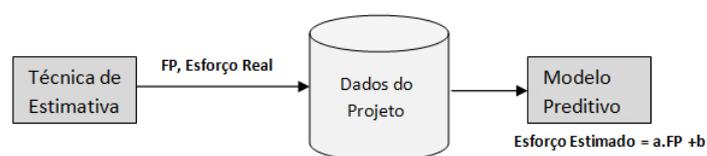


Figura 5.3: Exemplo da aplicação da técnica escolhida.

O Modelo–E10 possibilita a utilização de diferentes técnicas de estimativas, inclusive àquelas utilizadas pelo mercado. Essa é das vantagens em relação às observadas na literatura e no mercado.

A saída produzida por esta fase é um ou mais modelos preditivos para estimar o ETMS para tantas quantas forem as técnicas utilizadas sobre os dados do projeto. Este resultado é transmitido à fase Modelos Preditivos.

5.2.5 Modelos Preditivos

Nesta fase são aplicados os modelos preditivos da fase anterior sobre os dados de entrada do projeto. Os dados de entrada dos modelos preditivos são os dados atuais das demandas, previamente conhecidos, informados pelo usuário. Por exemplo, para se estimar o ETMS de uma demanda deve-se informar ao modelo preditivo o valor da métrica tamanho da demanda, em FP. Com esse dado informado, o modelo preditivo sugere um valor de ETMS.

Geralmente os dados de entrada são calculados por outras técnicas. O tamanho em FP, por exemplo, é calculado como sugerido pela técnica de *Function Points* regida pelo IFPUG [IFP05]. Supondo que o modelo preditivo gerado conforme ilustra a Figura 5.3, seja este dado pela Equação 5.2:

$$EsforçoEstimado = 4FP + 8,5 \quad (5.2)$$

Supondo também que o tamanho da demanda em FP seja igual a 18, logo, a aplicação do modelo preditivo é informar o FP para a Equação 5.2. O Esforço Estimado por esse modelo preditivo para a demanda de tamanho 18 é de 80,5 horas de trabalho para uma pessoa.

Os modelos preditivos têm a vantagem de estarem sempre atualizados pelo Modelo–E10 por meio da Retroalimentação. Assim, cada modelo preditivo vai gerar um valor estimado de ETMS e haverão tantos valores de ETMS quantos forem os modelos aplicados. A saída desta fase é um conjunto de valores de estimativas de ETMS endereçados à fase Agrupamentos.

5.2.6 Agrupamentos

Um agrupamento é dado por um conjunto de indivíduos com certo grau de similaridade. Os agrupamentos do Modelo–E10 são construídos utilizando três elementos: i) o Nível de Exigência do Modelo–E10; ii) o valor estimado por cada modelo preditivo; e iii) média ponderada dos valores do modelo preditivo realizado na fase de Calibragem das técnicas utilizadas. A construção dos agrupamentos foi uma evolução do Estudo de Caso apresentado no Capítulo 4, seção 4.6, onde os mesmos eram encontrados com base na distância de Euclidiana, como relatado naquele capítulo. Assim, a Equação 5.3 apresenta o cálculo da proximidade para determinar se um indivíduo pertence ou não a um agrupamento, onde $ETMS$ é o Esforço Estimado de Manutenção em Software, NE é o Nível de Exigência parametrizado no Modelo–E10, \overline{MP} é a média ponderada do ETMS conforme o fator de ponderação dos modelos preditivos utilizados e i representa a técnica que gerou o modelo preditivo.

$$(ETMS_i - ETMS_{i+1}) \leq (NE \times \overline{MP}_i) \quad (5.3)$$

O agrupamento é uma das principais contribuições do Modelo–E10 diferenciando-o das demais propostas existentes na literatura e utilizadas no mercado. Isso porque os valores agrupados refletem as "opiniões" dos modelos preditivos para estimar o ETMS. Quanto mais opiniões de diferentes modelos preditivos convergirem, maior é a probabilidade dos mesmos estimarem próximo ao esforço real da demanda.

Não há um limite para a quantidade de agrupamentos, como também pode ocorrer de não ser encontrado nenhum agrupamento. Caso este último ocorra, cada valor é considerado um agrupamento. A saída desta fase são os agrupamentos encontrados, endereçados à fase Seleção.

5.2.7 Seleção

A fase de Seleção é responsável por eleger os agrupamentos que contém os valores para compor o ETMS. É nessa fase que os agrupamentos com melhor chance de realizar boas estimativas são selecionados. A chance é determinada pela proximidade dos valores, sendo assim, quanto mais valores próximos um do outro um agrupamento tiver, maior a chance em realizar estimativas próximas do esforço real.

As regras para a seleção dos agrupamentos foram geradas por meio do estudo de caso apresentado no Capítulo 4 e apresentadas na seção 4.6. Contudo, elas foram aperfeiçoadas para o Modelo–E10. A regra 2 do estudo de caso, por exemplo, deixou de existir porque já estava implícita nos agrupamentos, com isso, das seis regras originalmente propostas, restaram apenas cinco, apresentadas abaixo, e devendo ser executadas na ordem disposta:

1. **Considerar o agrupamento com a maior quantidade de indivíduos:** essa regra é importante por encontrar uma tendência nos agrupamentos. Os agrupamentos com o maior número de indivíduos similares indicam que diferentes modelos preditivos convergiram para um mesmo valor estimado de ETMS e, por isso, o agrupamento com maior número de indivíduos é selecionado;
2. **Caso os agrupamentos tenham o mesmo número de indivíduos, considera-se o grupo que tem as menores distâncias entre os indivíduos, dado pelo menor valor das somas das distâncias entre os mesmos:** essa regra garante que, quando houver empate entre o número de indivíduos de diferentes agrupamentos, o Modelo–E10 sempre selecione os agrupamentos mais densos, os quais são àqueles cujos indivíduos estão mais próximos uns dos outros. Selecionar agrupamentos com valores estimados de ETMS mais próximos significa menos dispersão de valores e maior convergência entre os mesmos. Esse critério de desempate do modelo garante que o grupo mais convergente é considerado, o que aumenta a possibilidade de se estimar próximo do esforço real;

3. **Caso a soma de todas as distâncias seja igual para agrupamentos distintos, soma-se apenas as menores distâncias de cada agrupamento, até que seja possível identificar um valor de desempate entre as distâncias somadas:** essa regra procura a menor distância existente entre os indivíduos de um mesmo grupo, caso ainda ocorra empate com o uso da regra anterior. Procurar a menor distância é uma forma de identificar qual agrupamento possui indivíduos mais próximos, e o agrupamento com essa característica passa a ser aquele com a maior possibilidade de estimar o ETMS próximo do esforço real;
4. **Caso ocorra ainda o empate entre os agrupamentos, são selecionados os agrupamentos com o mesmo número de indivíduos calculando a mediana dos agrupamentos e a média das medianas encontradas:** certamente, se dois ou mais agrupamentos forem selecionados há divergência entre os valores estimados de ETMS de cada agrupamento. Para isso, essa regra tenta suavizar essa divergência calculando a mediana de cada agrupamento e, posteriormente, a média das medianas de cada agrupamento. Para o Modelo–E10 isso é importante para evitar que o valor estimado de ETMS fique muito distante do valor de esforço real;
5. **Caso não exista nenhum agrupamento, todos os valores calculados são considerados grupos e calcula-se a média aritmética de todos os indivíduos:** essa regra é aplicada quando ocorre divergência entre todos os valores estimados de ETMS pelos modelos preditivos. Assim, a média é aplicada para suavizar os valores estimados centralizando-os em um valor único estimado. Para o Modelo–E10 isso é importante porque em caso de valores muito divergentes a média suaviza a distância entre os mesmos, o que aumentam as chances do valor estimado de ETMS ficar mais próximo do esforço real.

A saída desta fase são os valores do ETMS, onde estes são endereçados à fase de Estimativas.

5.2.8 Estimativas

Nesta fase os valores do ETMS encontrados em cada agrupamento são "fundidos" por meio do cálculo da média aritmética entre os mesmos. A vantagem em se utilizar a média aritmética é que esta permite centralizar valores de esforço estimado que por ventura estejam muito distante um do outro. Dessa forma, mesmo o modelo não estimando bem, a média tenta reduzir o erro, pois o valor estimado é um valor intermediário entre o menor e o maior valor sugerido pelos agrupamentos.

Finalmente, a saída desta fase é um valor estimado de ETMS informado ao usuário e armazenado na base de dados do projeto. Como o Modelo–E10 é iterativo, para as próximas estimativas todas suas fases são executadas novamente.

5.3 Critérios de Elegibilidade das Técnicas de Estimativas

Nesta seção são discutidos alguns critérios denominados de Critérios de Elegibilidade. Estabelecer estes critérios é importante para que se tenha uma forma sistemática de se inserir ou retirar as

técnicas de estimativas do Modelo–E10, evitando assim, o empirismo no momento de se definir tais técnicas.

Os Critérios de Elegibilidade servem para responder duas questões que emergiram por meio do Estudo de Caso Exploratório do Capítulo 4. A primeira delas é: **Em meio a uma grande quantidade de técnicas de estimativas, quais delas adotar para o Modelo–E10?** Respondendo essa primeira questão, os seguintes pontos devem ser considerados:

- **Utilizar os dados do projeto:** as técnicas devem se basear nos dados de projeto porque esses dados, quando devidamente coletados e analisados, podem revelar padrões de comportamento. Os padrões podem ser descobertos por meio da mineração de dados, a qual utiliza técnicas estatísticas e de inteligência artificial. Existem diversas técnicas de mineração de dados que podem ser utilizadas para as estimativas. Geralmente as organizações que utilizam um modelo de qualidade como o CMMI, possuem coleta e armazenamento dos dados, assim, não há dificuldade em extrair os dados para as técnicas de estimativas serem aplicadas sobre os mesmos.
- **Não ser sensível a volume de dados estatisticamente insignificante:** a técnica pode trabalhar com um volume de dados pequeno desde que seja estatisticamente significativo. Inicialmente a sugestão é que se tenha dados de pelo menos 30 demandas para iniciar o Modelo–E10.
- **Trabalhar com valores numéricos:** os dados de projeto são suas métricas, estas possuem valores numéricos que expressam uma medida de grandeza dentro do projeto. Para se estimar o esforço faz-se necessário informar os dados de entrada para uma demanda. Conhecidos os dados de entrada é calculada a saída, neste caso, o esforço estimado, que em grande parte das técnicas sugeridas pela literatura produzem uma saída numérica, que pode estar representado em dias, em meses, em horas ou até mesmo em minutos trabalhados.

A segunda questão é: **Quando e como adotar e retirar técnicas do Modelo–E10?** Para se adotar uma técnica basta que esta atenda os Critérios de Elegibilidade descritos na primeira questão. Para a retirada de técnicas é necessário avaliar os modelos preditivos gerados pelas mesmas no fim de cada rodada de estimativas. O Modelo–E10 disponibiliza uma avaliação dos modelos preditivos gerados onde é possível saber o desempenho de cada modelo preditivo da rodada anterior. Esse desempenho só pode ser analisado depois que as demandas estiverem sido finalizadas, e seus esforços reais devidamente alimentados na base dados de projeto. O desempenho dos modelos preditivos é medido pelo seu fator de ponderação, que indica quais estão com baixo desempenho. Um modelo preditivo deve ser retirado do Modelo–E10 quando: em seis rodadas consecutivas de estimativas (rodadas essas suficientes para se ter um volume de dados retroalimentados considerável), o seu fator de ponderação tiver o valor igual a um.

A Tabela 5.2 apresenta uma sugestão de um conjunto de técnicas de estimativas que podem ser adotadas pelo Modelo–E10. A tabela está organizada com a coluna "Técnica", que sugere três

técnicas baseadas em estatística e mineração de dados; seguida da coluna "Justificativa", que justifica a escolha da técnica sugerida; na sequência, a coluna "Requisito", que apresenta os requisitos básicos para que a técnica seja utilizada; e, por fim, a coluna "Para o Modelo–E10", que apresenta a vantagem e desvantagem de utilizar tais técnicas sugeridas para o Modelo–E10.

Tabela 5.2: Sugestão de técnicas de estimativas para o Modelo–E10.

Técnica	Justificativa	Requisito	Para o Modelo–E10
Regressão Linear Simples	<ul style="list-style-type: none"> -Gera uma equação para preditiva representar o comportamento dos dados em função do ETMS; -Gera um modelo preditivo para cada variável independente; -Fácil aplicação. 	<ul style="list-style-type: none"> -Definir a variável preditiva e uma ou mais variáveis explicativas; -Extrair dados do projeto; - Utilizar uma ferramenta como o <i>MS–Excel</i> ou o SPSS. 	<p>Vantagem: as diferentes técnicas representam geram diferentes modelos preditivos que expressam diferentes opiniões nas estimativas de ETMS quanto maior a diversidade de valores estimados maior é a possibilidade de convergência do valor estimado de ETMS para o esforço real;</p>
<i>M5P</i>	<ul style="list-style-type: none"> -Gera uma árvore de regressão com uma ou mais equações lineares para representar o ETMS; -Gera um modelo preditivo para uma variável independente ou um conjunto delas; -Fácil aplicação. 	<ul style="list-style-type: none"> -Definir a variável preditiva e as variáveis explicativas; -Gerar um arquivo ".csv" com os dados do projeto; -Ferramenta para aplicação do algoritmo. 	<p>Desvantagem: os modelos gerados pelas técnicas são sensíveis à <i>outliers</i> por isso limpeza nos dados podem ser necessárias, além disso, no caso das árvores de regressão e decisão, dependendo do seu tamanho, pode ser difícil de implementar seu comportamento porque aumenta sua complexidade.</p>
<i>Decision Stump</i>	<ul style="list-style-type: none"> -Busca o comportamento dos dados expresso em uma árvore de decisão binária para conjuntos de dados do projeto; - Por ser uma técnica de ML expressa uma representação diferente das anteriores para o comportamento dos dados; -Fácil aplicação. 		

As técnicas sugeridas na Tabela 5.2 atendem os Critérios de Elegibilidade e tiveram um bom desempenho tanto no estudo de caso apresentado no Capítulo 4, como no experimento realizado no 6. Porém, as técnicas Percentual de Esforço por Equipe, Quartil 1 (25%), Mediana e Quartil 3 (75%), utilizadas no Capítulo 4, não foram sugeridas aqui, porque as mesmas não obtiveram desempenho satisfatório para estimar o valor de ETMS.

O *REPTree* foi substituído pelo *Decision Stump*, porque a complexidade de implementação do primeiro é alta quando se utiliza um conjunto de variáveis independentes. Já o último gera, uma árvore binária mais simples de entender e implementar.

5.4 Considerações Finais do Modelo–E10

O Modelo–E10 é sistemático para estimativas de ETMS evitando que as estimativas sejam realizadas de forma empírica. Outrossim, por se basear em dados históricos de projetos, o modelo pode fazer uso de várias técnicas de estimativas baseadas em mineração de dados. Essa é uma das vantagens do Modelo–E10, pois permite que seja gerado pelo menos um modelo preditivo para cada técnica utilizada. Essa vantagem alinha o Modelo–E10 com a literatura, onde os autores sugerem que a utilização de mais de uma técnica de estimativas tornam os valores estimados mais convergentes com o esforço real [LB06] [Pre09] [Som06].

Outra vantagem do Modelo–E10 é ter seus modelos preditivos sempre atualizados, já que para as estimativas futuras é realizada a Retroalimentação, onde as melhores estimativas do período anterior compõem o conjunto de dados utilizados no próximo período de estimativas. Outrossim, os modelos preditivos são avaliados quanto ao seu desempenho; com isso é feita a ponderação dos modelos preditivos que obtiveram boas aproximações do esforço real.

Os resultados dos modelos preditivos são agrupados conforme seu grau de similaridade, dado pela multiplicação do nível de exigência do Modelo–E10 com sua média ponderada. Essa característica do modelo é fundamental para estimar o valor do ETMS, pois quanto mais indivíduos estiverem nos agrupamentos, maior a chance dos diferentes modelos preditivos conseguirem boas aproximações do esforço real. A seleção escolhe os agrupamentos com o maior número de indivíduos e calcula a média aritmética entre os mesmos para informar ao usuário o valor estimado de ETMS.

O Modelo–E10 não exclui a utilização de outras técnicas de estimativas como o COCOMO 2.0 ou o *Planning Poker*, pois estas podem ser agregadas ao modelo desde que atendam os Critérios de Elegibilidade aqui apresentados.

As desvantagens do Modelo–E10 são, principalmente, em relação aos requisitos necessários para sua utilização. A primeira delas diz respeito à preparação dos dados de demandas para a primeira iteração do modelo. Este trabalho é manual e laborioso, pois necessita de conhecimento na base do projeto e de construção de rotinas SQL para a extração dos dados. Outra desvantagem diz respeito à definição das variáveis independentes, utilizadas pelas técnicas de estimativas, uma tarefa criteriosa e igualmente laboriosa, a qual deve ser realizada por meio do cálculo da correlação entre as mesmas e o esforço real.

A Tabela 5.3 apresenta como o modelo Modelo–E10 contribui para a área de estimativas segundo os aspectos apresentados nos Capítulos 2 e 3 e, ainda, observados no Estudo de Caso Exploratório do 4. Por isso, o Modelo–E10 consegue responder a questão de pesquisa apresentada na Seção 1.2 do Capítulo 1.

Finalmente, o Capítulo 6 apresenta um experimento *in vitro* realizado com o Modelo–E10 utilizando-o para estimar o ETMS do projeto P1.

Tabela 5.3: O Modelo–E10 em relação às propostas na área de estimativas.

Proposta	Domínio	Representação	Base Histórica	Combinação	Ao menos 3	AvaliaçãoMP	Calibragem
COCOMO	Desenv./ Manut.	Numérica	Sim	Não	Não	Não	Sim
Equação do Software	Desenv.	Numérica	Sim	Não	Não	Não	Não
Delphi	Desenv./ Manut.	Numérica	Não	Não	Não	Não	Não
Redes Neurais	Desenv.	Numérica	Sim	Não	Não	Não	Sim
Fuzzy	Desenv.	Categórica	Não	Não	Não	Não	Não
Raciocínio Baseado em Casos	Desenv.	Numérica/ Categórica	Sim	Não	Não	Não	Sim
Árvores de Decisão	Desenv.	Numérica/ Categórica	Sim	Não	Não	Não	Sim
Regressão	Desenv.	Numérica	Sim	Não	Não	Não	Sim
[OBM08]	Desenv.	Numérica	Sim	Não	Não	Não	Sim
[SYB08]	Desenv.	Numérica	Sim	Não	Não	Não	Sim
[LM09a]	Desenv.	Numérica	Sim	Não	Não	Não	Sim
[BL07]	Desenv.	Numérica	Sim	Não	Não	Não	Não
[LR07]	Desenv.	Numérica	Sim	Não	Não	Sim	Sim
[KTB08]	Desenv.	Numérica	Sim	Não	Sim	Não	Sim
[SM08]	Manut.	Numérica	Sim	Não	Não	Não	Sim
[ANC08]	Desenv.	Numérica	Sim	Sim	Não	Não	Sim
[WSM+ 09]	Desenv.	Numérica	Sim	Não	Não	Não	Sim
[LM09b]	Desenv.	Numérica	Sim	Não	Não	Não	Sim
[Pat07]	Desenv.	Numérica	Não	Não	Não	Não	Não
[ANC09]	Desenv.	Numérica	Sim	Não	Sim	Não	Não
[TBB07]	Desenv.	Numérica	Sim	Não	Sim	Não	Sim
[PDP07]	Desenv.	Numérica	Sim	Não	Sim	Não	Não
Modelo–E10	Manut.	Numérica	Sim	Sim	Sim	Sim	Sim

6. Experimento com o Modelo–E10

Os experimentos foram realizados seguindo o processo de experimentação sugerido por [HOR⁺00] e apresentado na Figura 6.1. As seções subsequentes apresentam o protocolo de experimento utilizado neste trabalho.

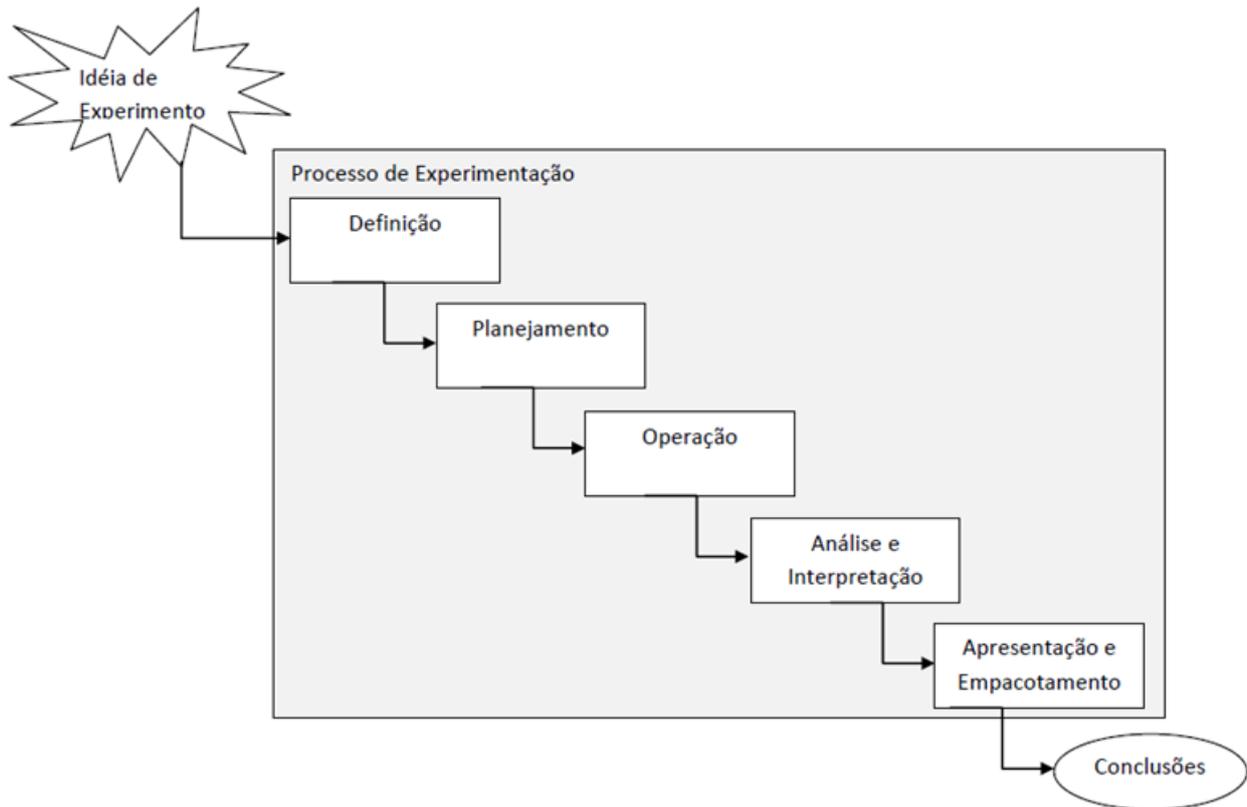


Figura 6.1: Processo de experimentação. Fonte: [HOR⁺00].

6.1 Definição

A motivação do experimento é avaliar o resultado das estimativas de esforço realizadas pelo Modelo–E10 sobre os dados reais de uma fábrica de software.

Objeto de Estudo. O objeto de estudo é o valor estimado de esforço de trabalho em projetos de manutenção de software. O esforço é provavelmente a mais popular métrica que se tem em software. É a partir do esforço que são definidos prazos e custos [Pre09] [Som06] [LB06].

Propósito. O propósito do experimento é avaliar se o Modelo–E10 pode ser amplamente utilizado nos projetos de uma fábrica de software, tendo o potencial de ao menos substituir a *expertise* de membros da equipe no que tange estimar o ETMS.

Perspectivas. Do ponto de vista de pesquisa, a perspectiva é apresentar o Modelo–E10 como uma alternativa viável para se estimar o ETMS de um projeto com boas aproximações do esforço

real. Do ponto de vista de mercado é oferecer um modelo, que possa ser amplamente aplicado em uma organização com boas aproximações do esforço real e sem onerar suas equipes.

Foco. O principal efeito estudado no experimento é o desempenho do Modelo–E10 sobre os dados reais de projeto. Aqui, três aspectos são enfatizados: i) a aproximação do esforço estimado com o esforço real utilizando diferentes técnicas de estimativas e, ainda, autoajustando o modelo para que seus modelos preditivos evoluam; ii) a possibilidade de substituir a *expertise* de um projeto para não haver mais dependência do conhecimento humano; e iii) a desoneração das equipes responsáveis pelas estimativas.

Contexto. O experimento é executado no contexto de estimativas de esforço de trabalho em manutenção de software. Além disso, o experimento é conduzido com dados de um projeto da HP–*Consulting*–Porto Alegre, o qual mantinha à época, um software de operação bancária pertencente a um grande banco estatal brasileiro. O projeto possuía um grande orçamento e uma equipe com mais de 60 pessoas envolvidas diretamente no projeto. Os dados do projeto eram devidamente coletados e armazenados em repositórios do próprio projeto, e foram coletados no período janeiro de 2006 ao primeiro trimestre de 2008.

6.1.1 Sumário da Definição

O sumário da definição é:

Analisar os resultados do Modelo–E10

para o propósito de *avaliação*

à respeito *das estimativas de esforço, que é a métrica sendo avaliada, e seu desvio em relação ao esforço real*

do ponto de vista dos *profissionais responsáveis pelas estimativas*

no contexto de *Estimativas de Esforço em Manutenção de Software*.

6.2 Planejamento

6.2.1 Contexto da Seleção

O experimento é focado em estimar o ETMS e é executado em laboratório, por isso é classificado como *in vitro*. Outrossim, o experimento endereça um problema para estimar o valor de ETMS, e utiliza dados reais de um dos projetos da HP–*Consulting*–Porto Alegre.

Usar dados de projeto de software, no contexto experimental, permite que outros pesquisadores tenham uma oportunidade de reproduzirem o experimento em questão para que, no futuro, possam avaliar e sugerir melhorias no Modelo–E10.

6.2.2 Formulação das Hipóteses

Um aspecto importante do experimento é saber como e o quê se quer formalmente avaliar no Modelo–E10. Isso é realizado por meio da formulação das hipóteses baseadas no seguinte

pressuposto: "Uma fábrica de software utiliza a *expertise* de seus membros de equipe para estimar o esforço de manutenção de software por fase, como apresentado no Capítulo 4. É sabido que existem três problemas ao se utilizar a *expertise* para estimar o ETMS. O primeiro deles é relativo à precisão das estimativas, que se mostram um tanto distorcidas, dando a falsa impressão que a precisão do membro da equipe é muito maior do que qualquer modelo que se possa adotar. Este primeiro problema é conhecido como "Síndrome de Estudante" [GE06]. O segundo problema é a oneração dos membros das equipes, pois os mesmos têm que vasculhar documentos de sistema além de se reunirem para estimar tempo necessário para realizar uma demanda, isso aumenta o custo do projeto, visto que os membros de equipe poderiam estar trabalhando na implementação das demandas. O terceiro e último problema é a dependência de conhecimento humano, que pode aumentar o custo do projeto por tornar seus membros essenciais e insubstituíveis.

Baseado no pressuposto, é formulada a hipótese com a definição das medidas necessárias para sua avaliação.

Hipótese Nula H_0 : As estimativas de esforço de manutenção das demandas são mais próximas do esforço real ($\mu_{EsforçoReal}$) quando baseadas na *expertise* (μ_{EEE}) do que quando baseadas no modelo proposto ($\mu_{Modelo-E10}$). Portanto, na visão organizacional, não há melhoria na precisão das estimativas de ETMS quando se utiliza o Modelo-E10 ao invés da *expertise* do projeto P1.

$$H_0: \Delta_{Expertise} = \Delta_{Modelo-E10}$$

Hipótese alternativa $H_1: \Delta_{Expertise} > \Delta_{Modelo-E10}$

, onde $\Delta_{Expertise} = |\mu_{EsforçoReal} - \mu_{EEE}|$ e $\Delta_{Modelo-E10} = |\mu_{EsforçoReal} - \mu_{Modelo-E10}|$.

Medidas necessárias: Pontos de Função (FP) e Quantidade de Documentos impactados por uma demanda (QtdeDoc).

- FP: representa o tamanho da demanda a ser implementada em pontos de função;
- QtdeDoc: representa a quantidade de documentos de sistema impactados por uma demanda.

6.2.3 Seleção das Variáveis

A variável dependente é o esforço de trabalho estimado para cada fase de manutenção de uma demanda. O esforço é computado em horas trabalhadas por uma pessoa. As variáveis independentes são FP e QtdeDoc, devidamente coletadas pelo projeto P1.

6.2.4 Seleção da Amostra

A amostra escolhida para o experimento foram as demandas do ano de 2005 a 2008, pois os dados anteriores à 2005, apesar de existirem, foram migrados de bases de dados antigas do projeto para uma nova base. Analisando as informações das demandas anteriores à 2005 foram

encontrados dados faltantes como: tamanho, data da demanda, identificador da demanda, dentre outros. Provavelmente esses dados faltantes foram decorrência de incompatibilidades da migração, e, por isso, as demandas anteriores à 2005 foram descartadas do experimento.

As demandas foram divididas em conjunto de treino e teste. Para o conjunto de treino do modelo foram separadas as demandas pertencentes ao ano de 2005, pois o volume de dados é considerável para aplicar as técnicas de estimativas do Modelo-E10. Assim, o conjunto de treino teve 37 demandas selecionadas, e as mesmas serviram para a calibragem inicial do Modelo-E10.

Para o conjunto de testes foram escolhidas as demandas do período de 2006 a 2008, totalizando 118 demandas; estas foram separadas em seus respectivos meses obedecendo a sua ordem cronológica de entrada. Esse procedimento simulou a realidade do projeto P1 e a quantidade de demandas que por mês era bem variada.

6.2.5 Esboço do Experimento

As variáveis dependentes e independentes foram escolhidas para o experimento em questão. Ainda, foram definidas as escalas de mensuração para as variáveis. Os princípios gerais do experimento são:

Randomização. Os sujeitos (demandas) não são selecionados aleatoriamente. Foi obedecida uma ordem cronológica na seleção. Essa ordem é importante, porque simula o ambiente real do projeto P1 na fábrica de software, o que torna a validação do Modelo-E10 a mais realista possível.

Agrupamentos. Há um agrupamento sistemático que deve ser realizado para melhorar a visualização dos resultados. Isso foi identificado em um estudo de caso realizado anteriormente na fábrica de software. Como existem 24 meses a serem analisados, uma análise realizada mês a mês torna difícil a interpretação dos resultados. Para tanto, as demandas foram agrupadas por ano e trimestre para a execução do experimento.

Balanceamento. A princípio não há como saber se a quantidade de demanda existente em cada mês influencia no resultado final das estimativas de esforço. Portanto, nenhum balanceamento dos dados foi realizado. Sendo assim, em diferentes meses há diferentes quantidades de demandas.

Tipos de padrão do esboço. Abaixo são apresentados os padrões de esboço do experimento para refutar ou ratificar as hipóteses nula e alternativa deste experimento. São realizados uma série de testes estatísticos para verificar se a *expertise* tem melhor aproximação do esforço real do que o Modelo-E10.

1. Em primeiro lugar, é realizado um teste de normalidade nos dados, para determinar o tipo de teste que deve ser realizado na tentativa de rejeitar a hipótese nula da amostra, com distribuição normal ou não;
2. Em seguida, são testadas as hipóteses nula e alternativa;
3. Após o teste estatístico, são realizadas outras análises medindo o desempenho do Modelo-E10 em relação à *expertise*. As análises complementares utilizadas são: i) *PRED* e *MMRE*; ii)

um resumo analítico de desempenho por período de tempo; e iii) o cálculo do *RMSE* das demandas que estiverem fora do Nível de Exigência do projeto;

6.2.6 Instrumentação

Os objetos são os dados das demandas devidamente armazenados no repositório do projeto P1. O repositório é composto por tabelas que ficam armazenadas em uma base de dados *MS-SQL Server 2000* e acessada pelas ferramentas *MS-Project Manager* e *IBM-RequisitePro*.

Os dados foram extraídos por meio de consultas SQL executadas no banco *MS-SQL Server 2000*. Nesses dados foi encontrada uma variável dependente (esforço real) e duas independentes (FP e QtdeDoc). Também foi encontrado o esforço estimado pela *expertise*; dado esse, essencial para medir o desempenho do Modelo-E10.

6.3 Operacionalização

6.3.1 Preparação

A população do experimento é composta apenas pelos dados da equipe de Projeto, pois os líderes de equipe relataram a existência de distorções nas estimativas realizadas nas fases de Cliente, Servidor e Testes. Assim, esses dados são devidamente extraídos das bases do projeto P1, por meio de consultas SQL, e exportados para planilhas do *MS-Excel*.

Com os dados devidamente armazenados na planilha, é utilizada a ferramenta *WEKA* [UW01] para calibrar o modelo, gerando assim as equações para realizar as estimativas. Tais equações são chamadas de modelos preditivos. Os modelos preditivos são aplicados sobre os dados das demandas que se pretende estimar, e que estão no *MS-Excel*. Para tanto é necessário informar ao modelo preditivo as variáveis independentes que, neste caso, são: FP e QtdeDoc.

No *MS-Excel* foi montado um conjunto de planilhas de dados mensais das demandas. Cada planilha mensal contém uma macro para realizar as estimativas. Cada planilha mensal está associada a uma planilha de avaliação dos melhores modelos preditivos do mês atual. Essa implementação está em conformidade com o Modelo-E10 apresentado no Capítulo 5.

6.3.2 Execução

O Modelo-E10 necessita de uma calibragem inicial com os dados das demandas do projeto. A calibragem do Modelo-E10 deve ser realizada com os dados das demandas do ano de 2005 totalizando 37 demandas. A Figura 6.2 apresenta um diagrama de atividades do processo de execução do experimento.

Um arquivo ".csv" é gerado para que seja aberto e executado pela ferramenta *WEKA* [UW01], utilizada para executar as técnicas de estimativas e gerar os modelos preditivos. Cada modelo preditivo é aplicado informando os FP e a QtdeDoc da demanda que se quer estimar. Aplicando o

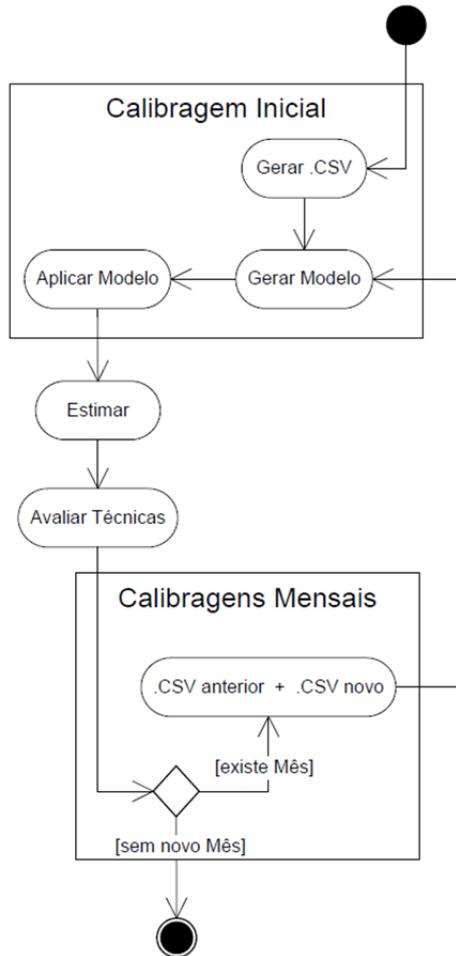


Figura 6.2: Processo de execução do experimento.

modelo se tem o valor do esforço estimado para um homem/hora. Estimado o esforço das demandas, confronta-se o valor encontrado com o esforço real, e se tem a avaliação das técnicas utilizadas.

Após a avaliação das técnicas utilizadas, o passo seguinte é alterar esse arquivo ".csv" adicionando as melhores estimativas do período atual. Este, então, passa a conter os dados das demandas da calibragem inicial e os dados do esforço real dessas últimas demandas estimadas. Assim, os novos modelos preditivos são gerados pelas técnicas de estimativas utilizadas na ferramenta *WEKA* [UW01] e o ciclo recomeça. A execução desse processo de experimentação é iterativa até que não haja mais demandas para se estimar. A ordem das estimativas é cronológica e mensal, simulando a realidade do projeto P1.

6.3.3 Validação dos Dados

Os dados das demandas ficavam devidamente armazenados nas bases do projeto P1 e foram extraídos por meio de consultas SQL a essas bases. Houve a ajuda do SQA do projeto, um especialista que conhecia bem os dados do projeto P1. Além do SQA, havia uma boa documentação sobre os dados armazenados, que foi amplamente explorada para se extrair as informações referentes aos

dados do projeto P1. Por último, os líderes de equipe contribuíram para validar se os dados foram extraídos corretamente da base do projeto. Para tanto, foram realizadas algumas reuniões formais nas quais essa extração foi validada.

6.4 Análise e Interpretação

6.4.1 Estatística Descritiva

O esforço de trabalho real das 118 demandas utilizadas no experimento gera o histograma apresentado na Figura 6.3, com média de aproximadamente 56,5 horas; desvio padrão de aproximadamente 52,29 horas; mediana de aproximadamente 37,7 horas; curtose de aproximadamente 3 indicando que a distribuição é leptocúrtica e possui um desvio padrão menor em relação à uma distribuição mesocúrtica ou platicúrtica; e coeficiente de assimetria de aproximadamente 1,75, indicando que a distribuição é assimétrica positiva.

Pela característica do histograma apresentado na Figura 6.3 suspeita-se que o teste realizado para a validação da hipótese seja não-paramétrico, ou seja, a amostra não possui distribuição normal. Para ratificar essa suspeita foram formuladas as hipóteses descritas na seção 6.2.2.

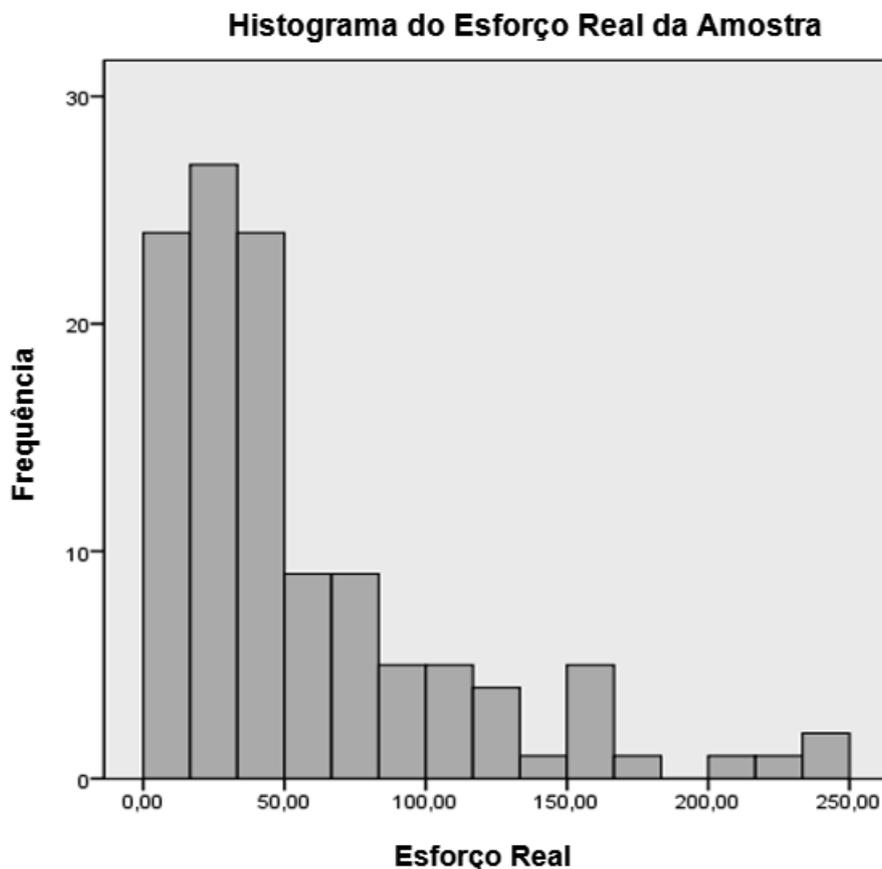


Figura 6.3: Histograma da distribuição de esforço real.

6.4.2 Redução de Dados

Originalmente a quantidade de demandas para o experimento era de 172, porém, após o Estudo de Caso, apresentado no Capítulo 4, foram identificados três fatos, abaixo relacionados, que poderiam distorcer os resultados do experimento. Esses fatos contribuíram para a retirada de 17 demandas, das quais poderiam distorcer os resultados. Ao final, restaram 155 demandas, com 37 delas do ano de 2005 e compondo o conjunto de treino.

- **Síndrome de Estudante:** são identificados como "Síndrome de Estudante" as demandas que tiveram o esforço real com proximidade de até 4 horas das horas estimadas pelo *expertise*. Além disso, são considerados *outliers* as estimativas que acertaram o esforço real de trabalho, pois não é possível identificar se o acerto realmente existiu ou se interferiu nos valores estimados.
- **Mudança do responsável pelo SQA do projeto:** no início de 2007 houve a troca da pessoa responsável pela qualidade do projeto. Esta também era responsável por manter os dados do projeto. Talvez pela inexperiência do novo membro, alguns dados do *baseline* original, que contém as estimativas originais das demandas do projeto, foram sobrepostos pelos dados do *baseline* revisado, que contém informações sobre o replanejamento do esforço das demandas. Assim, o *baseline* original do primeiro trimestre de 2007 foi praticamente todo sobreposto pelo *baseline* revisado, e o esforço estimado, quando não acertou, se aproximou muito do esforço real. Esse comportamento pode dar a impressão de que as estimativas realizadas com a *expertise* da equipe são muito melhores que qualquer técnica que possa ser adotada e, por isso, os dados referentes ao primeiro trimestre de 2007 foram descartados do experimento.
- **Contas de chegada:** segundo informações em reuniões formais com os líderes de equipe, demandas consideradas pequenas, de até 8 horas, em sua maior parte, não eram estimadas. Assim, o valor real de trabalho era lançado como estimado. Outrossim, a única equipe que aparentemente seguia o processo de estimativas das demandas corretamente, era a equipe de projeto. Em grande parte das demandas, as equipes de Cliente, Servidor e Testes, lançavam o esforço real como o estimado pela *expertise*. Esse problema foi corrigido pelo novo responsável de SQA do projeto, mas a maioria dos dados pertencentes às estimativas dessas equipes ficou comprometida.

Com 118 demandas restantes, fez-se uma análise para descobrir se haviam mais *outliers*, obtendo-se o gráfico *Box plot* apresentado na Figura 6.4. A partir desse gráfico foi possível identificar outros *outliers*, contudo, esses foram mantidos para não afetar o volume de dados do experimento.

A Tabela 6.1 apresenta a quantidade das demandas utilizadas no conjunto de testes do experimento.

6.4.3 Determinando o Tipo de Teste de Hipótese

Dependendo da disposição dos dados (normalidade), é definido o tipo de teste para a hipótese nula. Sendo assim, é definida uma hipótese nula e uma hipótese alternativa para a realização desse

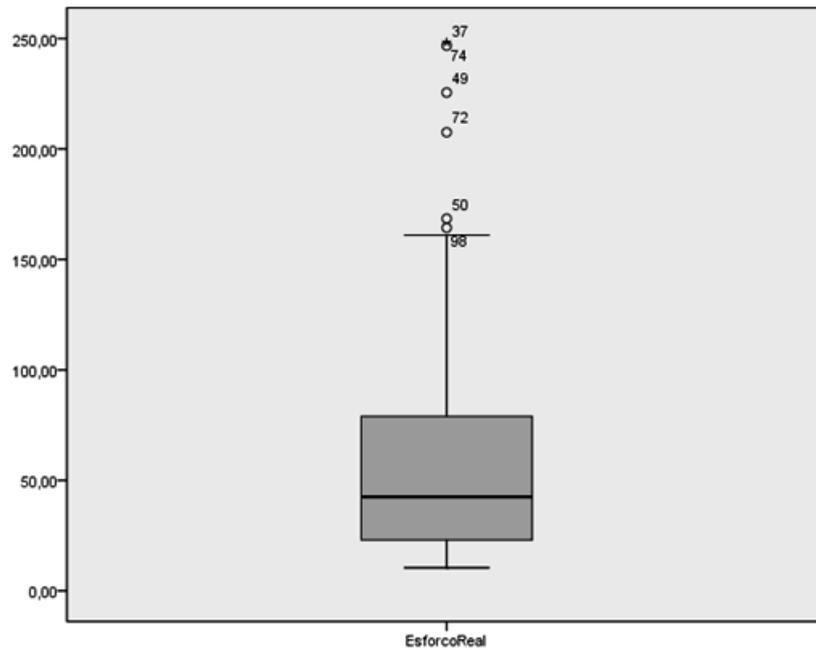


Figura 6.4: Box plot com os *outliers* do esforço real.

Tabela 6.1: Quantidade de demandas utilizadas no conjunto de testes.

Ano	Trimestre	Quantidade de Demanda
2006	1°	14
	2°	20
	3°	24
	4°	18
2007	1°	-
	2°	11
	3°	3
	4°	18
2008	1°	7
Total		118

teste, sendo elas:

- H_0 : a distribuição é normal;
- H_1 : a distribuição não é normal.

Existem duas formas para se avaliar a aderência da amostra à normalidade: o Teste de *Kolmogorov–Smirnov* e o Teste de *Shapiro–Wilk*. O primeiro é utilizado para identificar a normalidade em pequenas amostras; já o segundo em amostras acima de 50 indivíduos. Como a amostra tem 118 indivíduos, o teste realizado é o de *Shapiro–Wilk*. A Tabela 6.2 apresenta o resultado do Teste de *Shapiro–Wilk* de aderência à normalidade.

Observa-se que a significância dos dados do teste de *Shapiro–Wilk* é inferior ao nível de significância de 0,05 ou 5%, tanto para a *expertise* quanto para o Modelo–E10. Com esta informação,

Tabela 6.2: Teste de Normalidade *Shapiro–Wilk* para a variável Esforço Estimado.

Variável	Técnica	Estatística	Graus de Liberdade	Significância
Esforço Estimado	<i>Expertise</i>	0,456	118	0,000
	Modelo–E10	0,732	–	0,000

há fortes indícios para rejeitar a hipótese nula sobre a distribuição da normalidade, sendo assim, o teste para a validação da hipótese nula formulada na seção 6.2.2 é não-paramétrico. Neste caso, é utilizado o teste de *Wilcoxon*, pois o mesmo é apropriado para amostras que não têm aderência à distribuição normal e, ainda, pode ser utilizado em amostras que sejam dependentes uma da outra, que é o caso das amostras desse experimento.

6.4.4 Testando Hipóteses

Para testar a hipótese nula é realizado o teste de *Wilcoxon*. O critério para rejeição de H_0 em favor de H_1 é: $H_1: \Delta_{Expertise} > \Delta_{Modelo-E10}$ rejeita-se H_0 se $p\text{-valor} < 0,05$. Se esse fato for comprovado, vai indicar que, em média, o Modelo-E10 estima mais próximo do esforço real do que a *expertise* dos membros do projeto P1. A Tabela 6.3 apresenta o resultado do teste de *Wilcoxon*.

Tabela 6.3: Teste de hipótese de *Wilcoxon*.

Variável	Técnica	p-valor (Bi-caudal)
Esforço Estimado	Modelo–E10	0,05

O $p\text{-valor}$ apresentado na Tabela 6.3 é um valor bi-caudal ou bilateral. A hipótese H_1 trata de um teste unilateral, portanto o $p\text{-valor}$ deve ser dividido por dois. Logo, $p\text{-valor} = \frac{0,05}{2}$, ou seja, $p\text{-valor} = 0,025$. Com base nesse valor consegue-se rejeitar a hipótese nula em um nível de significância de 2% em favor de H_1 . Pressupõe-se então, que o Modelo–E10 estima em média mais próximo do esforço real do que a *expertise* dos membros do projeto P1.

6.4.5 Análises Complementares

Além do teste estatístico foi realizada uma série de análises complementares com intuito de verificar o desempenho do Modelo–E10 em relação à *expertise*. Para o projeto P1 superestimar as estimativas de ETMS é melhor que subestimar, visto que, o projeto é regido por um contrato de SLA, e está sujeito à pesadas multas quando ocorre atraso nas entregas de suas demandas.

Inicialmente é realizada a avaliação do Modelo–E10 e da *expertise* por meio do *Mean Magnitude Relative Error (MMRE)* e do *Percentage Relative Error Deviation (PRED)*. O *MMRE* apresenta a média do erro relativo e o *PRED(x)* o percentual de instâncias que estão dentro da margem de erro aceitável, onde x é a margem de erro a ser considerada aceitável. Uma margem de erro

aceitável é um valor entre 25% e 40% [PK08] [GM97]. Em seguida é avaliado o desempenho do Modelo-E10 e da *expertise* por meio da média das estimativas, sumarizadas por trimestre.

A Tabela 6.4 apresenta o resultado do *MMRE* e o $PRED(x)$ após a execução do experimento. Como o projeto P1 permite uma tolerância de 20% na margem de erro das estimativas, os valores de x foram $PRED(20)$, $PRED(25)$ e $PRED(30)$, respectivamente, alinham o experimento com as sugestões da literatura.

Tabela 6.4: Resultado da avaliação do Modelo-E10 com $PRED(x)$ e *MMRE*.

	MMRE	PRED(20)	PRED(25)	PRED(30)
<i>Expertise</i>	57%	18%	25%	34%
Modelo-E10	77%	22%	29%	34%

Como mostra a Tabela 6.4, o Modelo-E10 obteve um desempenho tão bom quanto a *expertise*. Isso pode ser observado por meio do *PRED*, que indica o percentual de acerto das instâncias estimadas. Contudo, o *MMRE* do Modelo-E10 foi mais alto que o da *expertise*. Isso indica que o Modelo-E10 é menos preciso que a *expertise*. Este fato não invalida o resultado obtido, pois [KFS⁺03] realizou experimentos com o *MMRE* e relata que quando este é utilizado para avaliar modelos, tende a identificar os modelos que subestimam como superior ao modelo verdadeiro, e modelos que superestimam como inferior ao verdadeiro. Assim, quando analisadas as demandas estimadas, descobriu-se que o modelo superestima o esforço das demandas em 70% dos casos. Isso explica o desempenho do pior do *MMRE* do Modelo-E10 em comparação com a *expertise*, e comprova o relato de [KFS⁺03].

A Figura 6.5 apresenta um resumo analítico trimestral do desempenho do Modelo-E10 e da *expertise*, ambos em relação ao esforço real de trabalho. Como pode-se perceber o Modelo-E10 é melhor que a *expertise* em 5 trimestres, enquanto que a *expertise* supera o Modelo-E10 em 3 trimestres. Como relatado na 6.4.2, havia problemas de distorções no Trimestre 01/2007, por isso, o mesmo foi retirado da análise.

A Tabela 6.5 apresenta uma análise dos resultados absolutos obtidos. Foram analisadas apenas as demandas onde o Modelo-E10 e a *expertise* não estimaram dentro do Nível de Exigência do projeto P1, ou seja, errar em 20% para cima ou para baixo do esforço real. Também foi calculado o erro quadrático médio de ambos *Root Mean Squared Error* para se saber qual modelo teve menos erros nas estimativas, que tiveram resultados aquém do Nível de Exigência do projeto P1. Com esses resultados pode-se inferir que para o projeto P1, o Modelo-E10 é mais adequado, por superestimar mais e, em média, errar menos. Isso não ocorre com a *expertise*, pois ela subestima mais e tem um erro quadrático maior quando comparado ao Modelo-E10.

6.5 Sumário e Conclusões

No experimento foi investigada a seguinte hipótese:

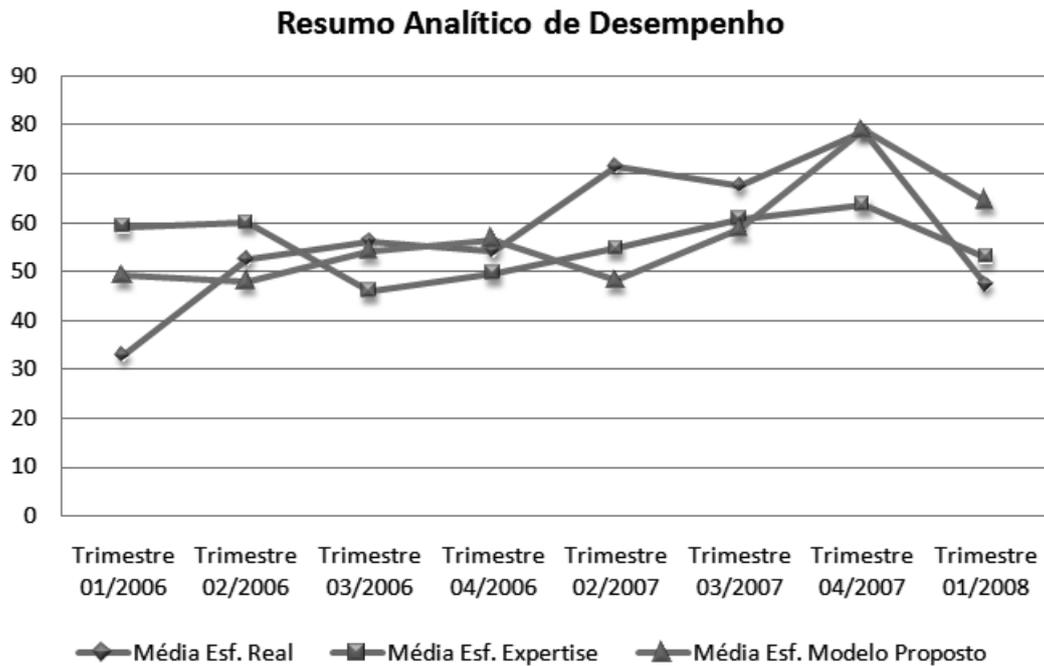


Figura 6.5: Resumo analítico trimestral.

Tabela 6.5: Análise complementar das estimativas com RMSE.

	Modelo-E10	Expertise
Superestimado	56	42
Subestimado	20	34
Erro Quadrático Médio (RMSE)	42,24	65,83

1. O Modelo-E10 estima o esforço de trabalho em manutenção de software mais próximo do esforço real do que a *expertise* de membros da equipe.

Inicialmente foi realizado o teste de *Shapiro-Wilk* para verificar o comportamento dos dados das demandas selecionadas para o experimento, quando descobriu-se que o esforço real das demandas não aderiam a uma distribuição normal. Portanto, o teste de hipótese caracterizou-se por ser não-paramétrico. As amostras são classificadas como relacionadas, pois todas tinham seus valores de esforço real, bem como seus valores de esforço estimado pela *expertise*. Com essas características dos dados, o teste de *Wilcoxon* é o mais recomendado para testar a hipótese em favor de H_1 . Aplicando o teste, este mostrou-se em favor da hipótese H_1 com um nível de significância de 2%.

Com a hipótese H_0 negada, outras análises foram realizadas para medir o desempenho do Modelo-E10 *versus* a *expertise*. Para tanto foram calculadas as medidas de precisão $PRED(x)$ e $MMRE$ sugeridas por [GM97] [PK08]. O desempenho do Modelo-E10, foi tão bom quanto o da *expertise*, e o percentual de instâncias que estavam dentro dos níveis de exigência do projeto P1 foi 4% maior em favor do Modelo-E10, como apresentado na Tabela 6.4. Também foram comparadas as diferenças absolutas do esforço real com o Modelo-E10 *versus* o esforço real com a *expertise* das

demandas. E ainda, foi realizado um teste de *RMSE* para as demandas que estimaram fora do Nível de Exigência do projeto P1, quando constatou-se que o desempenho do Modelo–E10 obteve um erro quadrático médio menor que o da *expertise*, indicando ser mais preciso dentre as demandas que estavam fora do Nível de Exigência do projeto.

Finalmente, pode-se inferir que o Modelo–E10 não só pode ser utilizado pelo projeto P1 para as estimativas de esforço de trabalho em manutenção de software, como também, por qualquer projeto que utilize os atributos FP e QtdeDoc.

7. Conclusões

Dentro dos trabalhos apresentados pela literatura percebe-se o empenho de pesquisadores na tentativa de encontrar e aplicar técnicas de estimativas de esforço em projetos de software.

O Modelo–E10 foi concebido por meio de um estudo empírico dado pela observação dos problemas referentes às estimativas de esforço em um grande projeto de software da HP–*Consulting*, conforme relatado pelos seus gerentes e mostrado no Estudo de Caso do Capítulo 4. Os problemas enfrentados pelo projeto como: obter boas aproximações, desonerar membros das equipes e necessidade de autoajuste para não desatualizar seus modelos preditivos, sem desconsiderar as pesquisas na área, foram as premissas básicas para construir a questão de pesquisa apresentada na seção 1.2 do Capítulo 1.

Este capítulo apresenta as conclusões do trabalho realizado, bem como a Contribuição Científica, a Contribuição para o Mercado, a sugestão de Trabalhos Futuros e as Considerações Finais da tese.

7.1 Contribuição Científica

Uma das deficiências observadas na área das estimativas é que, em geral, os trabalhos estão mais concentrados nas estimativas de esforço em desenvolvimento e não em manutenção de software. Empregar as técnicas de estimativas de esforço em desenvolvimento na manutenção de software não é uma boa prática, pois estas podem distorcer o valor do esforço estimado. Isso porque as estimativas de esforço para projetos de desenvolvimento se baseiam em dados de projetos anteriores e similares, enquanto que para a manutenção são baseados em dados histórico do projeto em questão.

Outra deficiência da área é que a maioria dos trabalhos utilizam as técnicas para estimar o esforço de forma isolada, quando a literatura sugere que se utilize diferentes técnicas de estimativas em conjunto. Além disso, as propostas encontradas na literatura deixam lacunas a serem preenchidas como: Quais dados devem ser devidamente coletados para a aplicação das mesmas? Que tipo de técnicas de estimativas podem ser adotadas por um projeto? Quais os requisitos de um projeto para que as técnicas sejam aplicáveis em seu contexto? Como adotar ou descartar uma técnica de estimativas para um projeto? Qual a quantidade mínima de dados que um projeto deve ter para aplicar as técnicas de estimativas adotadas?

Este trabalho contribui cientificamente porque propõe um modelo para estimativas de ETMS alinhado com as sugestões da literatura na área, possibilitando a utilização de mais de uma técnica e discutindo as características para a adoção e descarte das mesmas. Outrossim, o modelo possui uma proposta de retroalimentação que busca impedir a desatualização de seus modelos preditivos, e igualmente possibilita a agregação das boas práticas das técnicas de estimativas propostas na literatura, desde que cumpram os critérios de elegibilidade estabelecidos.

Por fim, o Modelo–E10 foi concebido por um método científico, com uma avaliação criteriosa e sistemática do modelo por meio de experimentação, apresentada no Capítulo 6, onde o resultado

obtido com o modelo foi satisfatório.

7.2 Contribuição para o Mercado

Os experimentos preliminares realizados com os dados reais do projeto P1, apresentados nos Capítulo 4, serviram de insumos para se propor o Modelo–E10, apresentado no Capítulo 5. Os resultados obtidos nos experimentos do Estudo de Caso mostraram que a utilização dos dados de um projeto poderia proporcionar aproximações do esforço real de trabalho consideravelmente boas. Além disso, ao se basear nos dados do projeto, as estimativas ganham agilidade em seu cálculo, já que podem ser aplicadas técnicas estatísticas e de mineração utilizando ferramentas de terceiros. Isso faz com que o esforço de membros de equipe seja substancialmente reduzido com a utilização do modelo.

O Modelo–E10 foi concebido de forma a permitir que técnicas de estimativas disseminadas no mercado como o COCOMO e o *Planning Poker*, também possam ser utilizadas em conjunto com outras técnicas. Isso permite que um projeto não necessite abandonar sua atual técnica de estimativas, agregando a mesma no Modelo–E10, desde que esta satisfaça os critérios de elegibilidade, apresentados na seção 5.3 do Capítulo 5.

Um projeto de software, que possui uma coleta de dados regular e mantém o histórico desses dados devidamente atualizado, é um potencial candidato para adotar o Modelo–E10.

7.3 Trabalhos Futuros

Uma das sugestões de trabalhos futuros é realizar a mineração de dados para avaliar os modelos preditivos, das técnicas de estimativas utilizadas, com melhor desempenho. Isso porque o modelo implementado só avalia os modelos preditivos com base na medição de seu desempenho no período imediatamente anterior ao que se está estimando. Acredita-se que a avaliação desses modelos preditivos seria mais precisa se, para cada modelo, fosse utilizada uma ou mais técnicas de mineração de dados sobre os dados históricos de desempenho, composto por todos os períodos anteriores.

Há indícios de que o Modelo–E10 possa ser utilizado em diferentes cenários de projetos em manutenção de software, isso porque, o modelo tem como base os dados históricos de projetos, o que o torna dependente da qualidade e do tipo dos dados coletados, e não da característica do projeto em si. Dessa forma, uma segunda sugestão é realizar experimentos com o Modelo–E10 aplicando-o em projetos de manutenção com outros cenários. Isso garantiria sua generalização, caso os resultados alcançados fossem satisfatórios como os alcançados com o P1.

Uma terceira sugestão de trabalho futuro é a modelagem de um repositório para estimativas de software e a implementação de uma ferramenta que possa realizar as estimativas. Não há na literatura, uma sugestão de um repositório para armazenamento de estimativas, tampouco o de uma ferramenta para realizar estimativas. Um repositório é providencial porque possibilita a extração de conhecimento sobre as estimativas realizadas, o que pode ajudar a encontrar causas raiz para más estimativas, por exemplo, onde técnicas de mineração de dados ou de aprendizado de máquina

poderia ser utilizada para este fim. No que diz respeito à ferramenta, esta seria essencial para desonerar ainda mais os membros de equipe e facilitar a extração, transformação e carga (processo de ETL) dos dados de projetos que por ventura estivessem em diferentes repositórios. Outrossim, a ferramenta poderia auxiliar na análise causal em problemas ocorridos nas estimativas por meio de relatórios diversos.

7.4 Considerações Finais

O Modelo–E10 se diferencia dos demais trabalhos existentes na literatura por permitir a utilização de diferentes técnicas de estimativas de esforço combinando-as, de modo a ajustar os valores estimados, levando em consideração as boas práticas de cada técnica.

Apesar de [FM07] afirmar a falta de evidências para substituir a *expertise* por outras técnicas de estimativas, a literatura afirma que a *expertise* é suscetível a "Síndrome de Estudante" [Gol97]. Isso foi constatado no Estudo de Caso realizado em uma fábrica de software, apresentado no Capítulo 4. Outrossim, o experimento com o Modelo–E10 alcançou resultados promissores para inferir que este modelo pode pelo menos substituir a *expertise*. Isso reduz o impacto da "Síndrome de Estudante", possibilitando um maior comprometimento no cumprimento de prazos e custos por parte dos projetos.

Finalmente, o Modelo–E10 não foi proposto com o propósito de ser uma solução para os problemas existentes na área de estimativas de ETMS, mesmo porque [Pre09] afirma que as estimativas em projetos de software estão longe de serem uma ciência exata. Entretanto, o Modelo–E10 é sólido o bastante para ser utilizado pelo mercado e por pesquisadores, quando ambos poderão explorá-lo o suficiente e proporem melhorias e adaptações para sua utilização neste amplo cenário das estimativas de esforço em manutenção de software.

Bibliografia

- [ABC00] Chris Abts, Barry Boehm, and Sunita Chulani. "Software Development Cost Estimation Approaches – A Survey", *Annuals Software Engineering*, vol. 10(1-4), October 2000, 177–205.
- [Aki71] Fumio Akiyama. "An example of software system debugging". In: *IFIP World Computer Congress - International Federation for Information Processing*, 1971, 353–358.
- [ANC08] Mohammad Azzeh, Daniel Neagu, and Peter Cowling. "Improving analogy software effort estimation using fuzzy feature subset selection algorithm". In: *PROMISE '08: Proceedings of the 4th international workshop on Predictor models in software engineering*, 2008, 71–78.
- [ANC09] Mohammad Azzeh, Daniel Neagu, and Peter Cowling. "Software effort estimation based on weighted fuzzy grey relational analysis". In: *PROMISE '09: Proceedings of the 5th International Conference on Predictor Models in Software Engineering*, 2009, 1–10.
- [BBT⁺08] Rodrigo C. Barros, Márcio P. Basgalupp, Nelson N. Tenorio Jr., Duncan Ruiz, and Karin Becker. "Issues on estimating software metrics in a large software operation". In: *Software Engineering Workshop, Annual IEEE/NASA Goddard*, 2008, 152–160.
- [BCS99] Barry Boehm, Sunita Chulani, and Bert Steece. "Bayesian analysis of empirical software engineering cost models", *IEEE Transactions on Software Engineering*, 25-4, July 1999, 573–583.
- [BL07] Gary Boetticher and Nazim Lokhandwala. "Assessing the reliability of a human estimator". In: *PROMISE '07: Proceedings of the Third International Workshop on Predictor Models in Software Engineering*, 2007, 5.
- [Cla96] Bradford Clark. "Cost Modeling Process Maturity - COCOMO 2.0". In: *IEEE Proceedings Aerospace Applications Conference*, 1996, 347–369.
- [Dan08] Maya Daneva. "Complementing approaches in erp effort estimation practice: an industrial study". In: *PROMISE '08: Proceedings of the 4th international workshop on Predictor models in software engineering*, 2008, 87–92.
- [DBP93] Giuseppe Dimarco, Farokh Boman Bastani, and Alberto Paquini. "Experimental evaluation of a fuzzy-set based measure of software correctness using program mutation". In: *15th international Conference on Software Engineering*, 1993, 45–54.

- [DCS86] Hubert Earl Dunsmore, Samuel Daniel Conte, and Vincent Shen. *“Software Engineering Metrics and Models”*. Redwood City:Benjamin-Cummings Publishing Co., 1986, 1st edition, 396p.
- [DML⁺04] Scott Dick, Aleksandra Meeks, Mark Last, Horst Bunke, and Abraham Kandel. “Data mining in software metrics databases”, *Fuzzy Sets and Systems*, vol. 145-1, July 2004, 81–110.
- [FM07] Kristian M. Furulund and Kjetil Molkken-stvold. “Increasing software effort estimation accuracy using experience data, estimation models and checklists”. In: *QSIC '07: Proceedings of the Seventh International Conference on Quality Software*, 2007, 342–347.
- [FN99] Norman Fenton and Martin Neil. “A critique of software defect prediction models”, *IEEE Transaction Software Engineering*, vol. 25-5, September 1999, 675–689.
- [GE06] Daniel Galorath and Michael Evans. *“Software Sizing, Estimation, and Risk Management”*. Boston:Auerbach Publications, 2006, 1st edition, 541p.
- [GJ07] Stein Grimstad and Magne Jørgensen. “Inconsistency of expert judgment-based estimates of software development effort”, *Journal of Systems and Software*, vol. 80-11, November 2007, 1770–1777.
- [GL97] Swapna Gokhale and Michael Lyu. “Regression tree modeling for the prediction of software quality”. In: *30th ISSAT International Conference no Reliability '97*, 1997, 31–36.
- [GM97] Andrew Gray and Stephen MacDonell. “A comparison of techniques for developing predictive models of software metrics”, *Information and Software Technology*, vol. 39-6, November 1996, 425–437.
- [Gol97] Eliyahu Moshe Goldratt. *“Critical Chain”*. North River:North River Press, 1997, 1st edition, 246p.
- [Hod96] Jonathan Hodgson. “The Computational Software Equation”. <http://www.sju.edu/~jhodgson/se/compeq.html>, 10 1996. Último acesso: 01/12/2009.
- [HOR⁺00] Martin Höst, Magnus C. Ohlsson, Björn Regnell, Claes Wohlin, Per Runeson, and Anders Wesslén. *“Experimentation in software engineering: an introduction”*. Norwell:Kluwer Academic Publishers, 2000, 1st edition, 204p.
- [IEE98] IEEE Computer Society. “IEEE Standard for Software Quality Metrics Methodology”. New York:IEEE Inc., 1998, 32.

- [IFP05] IFPUG International Function Point Users' Group. "Function Points Manual". <http://www.ifpug.org/>, 2005. Último acesso: 20/12/2009.
- [ISO98] ISO/IEC. "ISO/IEC TR 15504 Software Process Assessment". The International Organization for Standardization and The International Electrotechnical Commission, 1998.
- [JKA⁺00] Wendell Jones, Taghi Khoshgoftaar, Edward Allen, and John Hudepohl. "Classification-tree models of software-quality over multiple releases", *IEEE Transactions on Reliability*, vol. 49-1, March 2000, 4–11.
- [JS07] Magne Jørgensen and Martin Shepperd. "A systematic review of software development cost estimation studies", *IEEE Transactions on Software Engineering*, vol. 33-1, January 2007, 33–53.
- [Kan03] Stephen H. Kan. "*Metrics and Models in Software Quality Engineering*". Boston:Project Management Institute Inc., 2003,2nd edition,564p.
- [KFS⁺03] Barbara Kitchenham, Tron Foss, Erik Stensrud, and Ingunn Myrtveit. "A simulation study of the model evaluation criterion MMRE", *IEEE Transaction Software Engineering*, vol. 29-11, November 2003, 985–995.
- [KKM⁺08] Yasutaka Kamei, Jacky Keung, Akito Monden, and Ken-ichi Matsumoto. "An over-sampling method for analogy-based software effort estimation". In: *ESEM '08: ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*, 2008, 312–314.
- [KKS94] Ananda Krishna, Satish Kumar, and Prem Satsangi. "Fuzzy systems and neural networks in software engineering project management", *Journal of Applied Intelligence*, vol. 4-1, May 1993, 31–52.
- [KTB08] Yigit Kultur, Burak Turhan, and Ayse Basar Bener. "Enna: software effort estimation using ensemble of neural networks with associative memory". In: *SIGSOFT '08/FSE-16: 16th ACM SIGSOFT International Symposium on Foundations of software engineering*, 2008, 330–338.
- [LB06] Linda Laird and Carol Brennan. "*Software Measurement and Estimation - A Practical Approach*". New York:John Wiley and Sons, 2006, 1st edition, 257p.
- [LM09a] Chris Lokan and Emilia Mendes. "Applying moving windows to software effort estimation". In: *ESEM '09: 3rd International Symposium on Empirical Software Engineering and Measurement*, 2009, 111–122.
- [LM09b] Chris Lokan and Emilia Mendes. "Investigating the use of chronological split for software effort estimation", *IET Software*, vol. 3-5, April 2009, 422–434.

- [LR07] Jingzhou Li and Guenther Ruhe. "Decision support analysis for software effort estimation by analogy". In *PROMISE '07: Proceedings of the Third International Workshop on Predictor Models in Software Engineering*, 2007, 6.
- [Moo04] David Moore. "*The Basic Practice of Statistics*". New York: WH Freeman and Company, 2004, 3th edition, 691p.
- [MMW02] Nile Mosley, Emilia Mendes, and Ian Watson. "A comparison of case-based reasoning approaches". In: *WWW '02: Proceedings of the 11th International Conference on World Wide Web*, 2002, 272–280.
- [NSB08] Vu Nguyen, Bert Steece, and Barry 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.
- [OBM08] Adriano Oliveira, Petrônio Braga, and Silvio Meira. "A GA-based feature selection and parameters optimization for support vector regression applied to software effort estimation". In: *SAC'08: ACM Symposium on Applied Computing*, 2008, 1788–1792.
- [Pan04] C. Ravindranath Pandian. "*Software Metrics - A Guide to Planning, Analysis, and Application*". Boca Raton:Auerbach Publications, 2004, 1st edition, 286p.
- [Pat07] M.V. Patil. "Software effort estimation and risk analysis - a case study". In: *ICTES 2007 - Information and Communication Technology in Electrical Sciences*, 2007, 1002–1007.
- [PDP07] Martin Purvis, Da Deng, and Maryam Purvis. "Software metric estimation: An empirical study using an integrated data analysis approach". In: *International Conference on Service Systems and Service Management*, 2007, 1–6.
- [PK08] Dan Port and Marcel Korte. "Comparative studies of the model evaluation criterions MMRE and PRED in software cost estimation research". In: *ESEM '08: Second ACM-IEEE international symposium on Empirical software engineering and measurement*, 2008, 51–60.
- [PP00] James Peters and Witold Pedrycz. "*Software Engineering: an engineering approach*". New York:Wiley, 2000, 1st edition, 702p.
- [Pre09] Roger Pressman. "*Software Engineering: A Practitioner's Approach*". Boston:McGraw-Hill, 2009, 7th Edition, 889p.
- [RTR08] Marcelo Blois Ribeiro, Nelson N. Tenorio Jr., and Duncan 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.

- [SB02] Ken Schwaber and Mike Beedle. *“Agile Software Development With Scrum”*. Upper Saddle River:Prentice Hall, 2002, 1st. edition, 158p.
- [SDD⁺02] Silvio Stefanucci, Andrea De Lucia, Massimiliano Di Penta, and Gabriele Ventuni. “Early effort estimation of massive maintenance processes”. In: *Software Maintenance, IEEE International Conference on*, 2002, 234–237.
- [SEI06] SEI(Software Engineering Institute). “CMMI for development, version 1.2”. Pittsburgh:Carnegie Mellon, 2006, 573p.
- [SF95] Krishnamoorthy Srinivasan and Douglas Fisher. “Machine learning approaches to estimating software development effort”, *IEEE Transaction Software Engineering*, vol. 21-2, February 1995, 126–137.
- [SM08] Ruchi Shukla and Arun Kumar Misra. “Estimating software maintenance effort: a neural network approach”. In: *ISEC '08: Proceedings of the 1st India software engineering conference*, 2008, 107–112.
- [Som06] Ian Sommerville. *“Software Engineering 8”*. Boston:Addison Wesley, 2006, 8th edition, 840p.
- [SS97] Martin Shepperd and Chris Schofield. “Estimating software project effort using analogies”, *IEEE Transaction Software Engineering*, vol. 23-11, November 1997, 736–743.
- [SYB08] Yeong-Seok Seo, Kyung-A Yoon, and Doo-Hwan 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.
- [TBB07] Burak Turhan, Bilge Baskeles, and Ayse Bener. “Software effort estimation using machine learning methods”. In: *ISCIS'2007. 22nd International Symposium on Computer and Information Sciences*, 2007, 1–6.
- [UW01] UW The University of Waikato. Weka Machine Learning Project. <http://www.cs.waikato.ac.nz/ml/weka/>, 2001. Último acesso: 26/01/2010.
- [WF05] Ian H. Witten and Eibe Frank. *“Data Mining – Practical Machine Learning Tools and Techniques with JAVA Implementations”*. Morgan Kaufmann, 2nd edition, 2005, 525p.
- [WSM⁺09] Yong Wang, Qinbao Song, S. MacDonell, M. Shepperd, and Junyi Shen. “Integrate the gm(1,1) and verhulst models to predict software stage effort”, *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 39-6, November 2009, 647–658.
- [Yin05] Robert Yin. *“Estudo de Caso: Planejamento e Métodos”*. Porto Alegre:Bookman, 2005, 3rd edition, 212p.

A. Protocolo do Estudo de Caso Exploratório

A.1 Introdução ao Estudo de Caso e Objetivo do Protocolo

Este documento é um guia de procedimentos para o estudo de caso exploratório que visa melhorar as estimativas de um projeto de software para um grande banco estatal brasileiro, doravante denominado projeto P1. O software é mantido pela fábrica de software da Hewlett -Packard Company (HP), situada no TECNOPUC - Parque Tecnológico da Pontifícia Universidade Católica, em Porto Alegre, RS. Este estudo de caso se dará no projeto P1 da HP e tem o início programado para 16/05/2007. Sua duração é de aproximadamente 10 meses, com o término previsto para 01/03/2008.

O protocolo encontra-se organizado com segue: na seção A.1.1 são apresentadas as questões levantadas pelos gestores do projeto para a melhoria da estimativa do esforço em manutenção de software; na seção 1.2 é apresentada a estrutura do estudo de caso; na seção 1.3 é apresentada a hipótese formulada para este estudo de caso exploratório; na seção 2 é apresentado o procedimento para coleta de dados; na seção 3 é apresentada a prática na fábrica de software onde é realizado um relato da organização do projeto P1; na seção 4 é apresentada a forma da avaliação dos resultados deste estudo de caso; por fim, na seção 5 é apresentado o cronograma do deste estudo de caso exploratório com as atividades a serem realizadas e seus respectivos prazos.

A.1.1 Questões do Estudo de Caso

O estudo de caso exploratório é realizado para responder as seguintes questões:

1. Como estimar com mais precisão o Esforço de Trabalho em Manutenção de Software (ETMS) do projeto P1?
2. Como utilizar os dados do Projeto P1 para estimar e não depender mais do conhecimento humano, doravante chamado de expertise, nas estimativas de ETMS?
3. Como não onerar os membros da equipe responsável pelas estimativas de ETMS com a adoção de uma ou mais técnicas de estimativa?

A.1.2 Estrutura do Estudo de Caso

O estudo de caso deve seguir a estrutura apresentada na Figura A.1, abaixo.

O primeiro passo é realizar um estudo das Métricas do Projeto P1. Todas as métricas coletadas são devidamente documentadas no Documento de Métricas do Projeto. Esses documentos devem ser consultados para esclarecimentos sobre as métricas coletadas, as dúvidas recorrentes dos documentos devem ser sanadas com a pessoa responsável pelo Software Quality Assurance (SQA) do projeto P1.

Estudo de Caso Exploratório

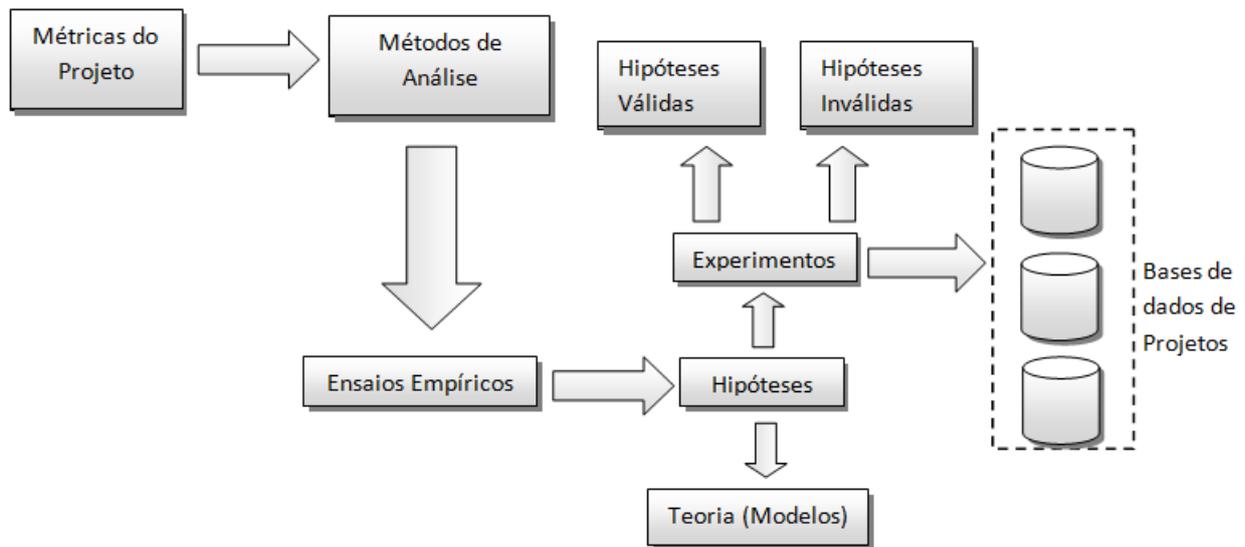


Figura A.1: Estrutura do estudo de caso exploratório.

Em seguida, o passo Métodos de Análise diz respeito à definição das técnicas de estimativas, sugeridas pela literatura, a serem utilizadas para realizar estimativas com base nos dados do projeto. Os resultados obtidos serão medidos pela proximidade do valor estimado com o valor real de esforço de demandas atuais. As técnicas estatísticas parecem ser as mais apropriadas para a estimativa de ETMS, pois as mesmas são específicas para trabalhar com dados históricos.

Nos Ensaio Empírico serão aplicadas as técnicas de estimativas, definidas pelos Métodos de Análise, sobre os dados do projeto P1. Conforme avançam os Ensaio Empírico as hipóteses alternativas do estudo de caso serão definidas e refinadas. Essas hipóteses guiarão a validação dos experimentos realizados no estudo de caso. As técnicas de estimativas utilizadas nos ensaios empíricos e definidas no Método de Análise devem refutar a hipótese nula H_0 , apresentada na seção 1.3, e auxiliar na formulação de uma ou mais hipóteses alternativas.

Os Experimentos farão o uso dos dados armazenados na base do projeto P1. Por meio deles as hipóteses formuladas serão validadas ou refutadas. Caso as hipóteses alternativas sejam validadas, um modelo de estimativas de ETMS será: proposto, testado e implantado no projeto P1.

A.1.3 Hipóteses do Estudo de Caso Exploratório

Como o estudo de caso é exploratório, as hipóteses alternativas deverão ser definidas depois dos ensaios empíricos com os dados do projeto P1. Contudo, faz-se necessária uma hipótese inicial para ser refutada. Assim, a hipótese formulada para este estudo de caso é a seguinte:

H_0 : Nenhuma técnica de estimativa é mais precisa que a técnica atual, baseada em sugestões dos membros da equipe, denominada expertise.

A.2 Procedimentos de Coleta de Dados

Os dados para a realização do estudo de caso serão levantados por meio de entrevistas com líderes de equipe e gestores do projeto P1, por meio de leitura da documentação do projeto e uma análise da base de dados do projeto.

As reuniões acontecerão em um período semanal com uma hora de duração. Nessas reuniões devem estar presentes os líderes de equipe e os gestores do projeto. O pesquisador deve se preparar para as reuniões por meio de leitura da documentação do projeto e do entendimento da base de métricas do mesmo. Caso necessário, os líderes de equipe devem dispor de um trinta minutos diário para tirar dúvidas do pesquisador no que tange à documentação, base de métricas e processos do projeto P1. O pesquisador deve reservar o tempo dos líderes com antecedência.

A.3 A Prática na Fábrica de Software

A organização é uma fábrica de software de grande porte avaliada no CMM3. O seu principal cliente é um banco estatal que envolve um orçamento anual de centenas de milhares de reais.

O projeto P1 possui 63 colaboradores divididos em quatro equipes: Projeto, com 20 membros; Clientee com 12 membros; Servidor com 12 membros; e Testes, com 17 membros. Cada equipe possui um líder técnico e além destes, o projeto possui um profissional de SQA e um subgerente responsável por organizar e acompanhar o cronograma das demandas.

O projeto P1 tem um PDS cascata e orientado a entregáveis. Os projetos têm livre arbítrio para a escolha das ferramentas a serem utilizadas para gerenciar seu PDS. As ferramentas utilizadas pelo projeto P1 são: o MS-EPM (Enterprise Project Management), para o controle dos prazos das atividades; o Ms-Excel para as estimativas de esforço de trabalho na manutenção do software e onde são armazenadas as estimativas de uma demanda com base na expertise de um membro da equipe de projeto; o IBM Rational Clear Quest para o armazenamento de informações referentes a defeitos, tamanhos e esforço; RequisitPro para a gerência dos requisitos; e, por fim, um software próprio para apontamento de horas, onde é lançado o as horas trabalhadas, ou esforço real, de cada membro da equipe.

Tipicamente, o produto é dividido em versões, sendo que o projeto deve entregar um valor de pontos por função/mês ao clientee, os quais são distribuídos dentro de uma ou mais versões do produto. Para cada versão do produto é determinado um conjunto de requisitos que são atribuídos a uma Ordem de Serviço (OS), onde estas contêm as manutenções que devem ser realizadas no software. A Figura A.2 apresenta o PDS do projeto P1.

A.4 Avaliação

Os resultados serão apresentados em reuniões periódicas e avaliados pelos gestores e líderes do projeto P1. Com base nos resultados haverá a decisão de utilizar ou não uma ou mais proposta do pesquisador.

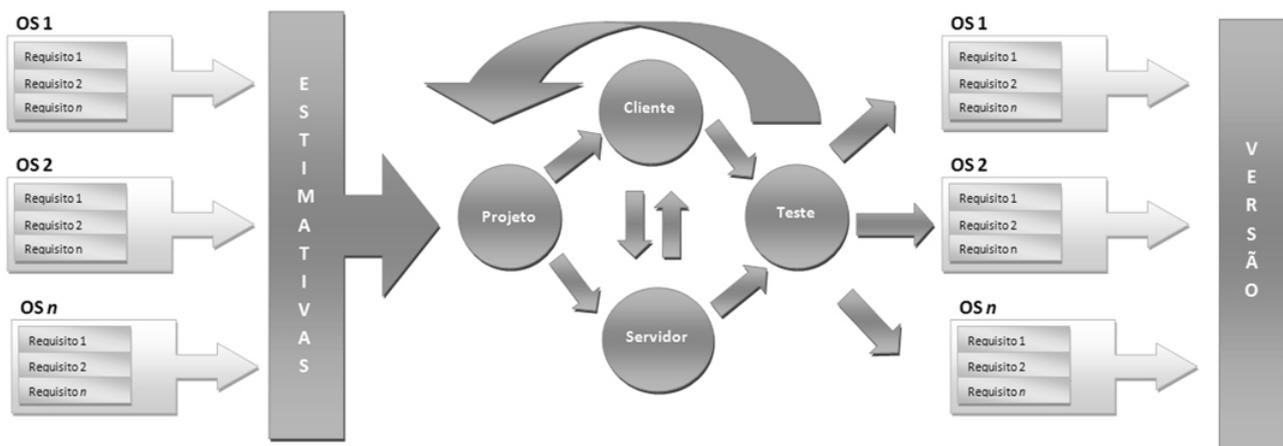


Figura A.2: Processo de software do projeto P1.

As reuniões para apresentação dos resultados seguem o cronograma apresentado na seção A.5. Os entregáveis são relatórios comparativos dos resultados obtidos com as novas técnicas de estimativas e a técnica anterior.

A proximidade do esforço estimado com o esforço real é o fator determinante para definir se uma nova técnica é melhor ou não do que a técnica atual. Testes estatísticos serão realizados para ratificar ou refutar a hipótese H_0 e, ainda, para determinar se os resultados obtidos não ocorreram por acaso. Sendo assim, os seguintes testes serão realizados:

- Significância: para avaliar a evidência fornecida pelos dados sobre alguma afirmação relativa a população;
- $PRED(x)$: é o percentual de instâncias cujo erro calculado se encontra abaixo de x ;
- MMRE: é o percentual do erro relativo de cada instância, independente de quantos ficaram abaixo do valor x .

A.5 Cronograma do Estudo de Caso Exploratório

O cronograma para a realização do estudo de caso exploratório do projeto P1 é apresentado na Figura . O mesmo possui um conjunto de dez atividades a serem realizadas em nos meses de junho de 2007 a março de 2008.

A atividade A1 envolve um estudo das métricas do projeto P1 por meio da leitura da documentação do projeto, a qual explica os processos e métricas do mesmo.

A atividade A2 prevê reuniões semanais de uma hora com os gestores e líderes do projeto P1, essas reuniões têm a finalidade de alinhamento da pesquisa com as expectativas dos gestores do projeto, de tirar eventuais dúvidas do pesquisador e de reportar o andamento da pesquisa.

Na atividade A3 será realizada uma revisão literária para definir as técnicas de estimativas de ETMS. As técnicas definidas darão subsídios para a realização da atividade A4. Nesta última serão

Atividades	Meses de 2007/2008									
	Jun	Jul	Ago	Set	Out	Nov	Dez	Jan	Fev	Mar
A1. Estudo das Métricas do Projeto P1 (Documentação, Processos e Base de Dados)										
A2. Reuniões semanais com as lideranças do projeto										
A3. Métodos de Análise (Técnicas de Estimativas)										
A4. Construção dos comandos para extração dos dados da base do projeto P1										
A5. Ensaios Empíricos										
A6. Experimentos para validação das hipóteses										
A7. Relatório com os resultados obtidos por meio dos experimentos										
A8. Proposta de um Modelo de Estimativas										
A9. Construção do Modelo de Estimativas										
A10. Implantação do Modelo de Estimativas										

Figura A.3: Cronograma do Estudo de Caso Exploratório.

criadas consultas SQL para serem disparadas contra o banco de dados de métricas do projeto P1 visando extrair os dados necessários a serem utilizadas pelas técnicas de estimativas definidas na atividade A3.

Os Ensaios Empíricos previstos na atividade A5 servirão para aplicar as técnicas de estimativas, definidas pelos Métodos de Análise, sobre os dados do projeto P1. Conforme avançam os Ensaios Empíricos as hipóteses alternativas serão definidas e refinadas. Essas hipóteses servirão de guia para a realização da atividade A6. Na atividade A6 serão realizados experimentos na tentativa de refutar a hipótese H0 e ratificar as hipóteses alternativas advindas pela realização da atividade A5.

Um relatório do estudo de caso deverá ser entregue ao final da atividade A6. Este relatório deverá conter um relato de todas as técnicas utilizadas, hipóteses refutadas e validadas, os resultados obtidos com cada técnica e as considerações finais com sugestões para a melhoria da atual técnica de estimativa de ETMS.

Caso os resultados obtidos com a execução das atividades de A1 a A7 sejam promissores, serão realizadas a atividade A8, A9 e A10 onde, respectivamente, será proposto, implementado e implan-

tado um modelo de estimativas de ETMS para o projeto P1. Caso não sejam alcançados resultados satisfatórios o pesquisador deverá voltar à pesquisa na literatura sobre outras formas de estimativas para ETMS das quais não se baseiem em dados do projeto.

B. Proposta de Classificação de Documento para o COCOMO 2.0

	Operações de Controle	Operações Computacionais	Operações Dependentes de Dispositivos	Operações de Gerenciamento de Dados	Operações de Gerenciamento de Interfaces com o Usuário	Fatores de Ponderação
Muito Baixo	Linhas de código poucos aninhamentos de operadores: DO, CASE, IF/THEN/ELSE. Composição de módulos com chamadas simples de procedimentos ou scripts.	Avaliação de expressões simples como: $A=B+C*(D-E)$.	Leituras simples e sentenças com formato simples.	Matrizes simples em memória e consultas/atualizações simples em banco de dados.	Formulários simples para entrada de dados e relatórios simples.	0.75
Baixo	Aninhamento direto de operadores estruturados. Maior parte de predicados simples.	Avaliação de expressões moderadas como: $D=\text{SQRT}(B**2-4*A*C)$.	Desnecessidade de processador ou dispositivo de E/S específicos. E/S realizada em nível de GET e PUT.	Um subconjunto de arquivos simples sem mudança de estrutura, sem edição e sem arquivos intermediários. Consultas e atualizações moderadas.	Uso de interfaces gráficas simples.	0.88
Nominal	Grande parte de aninhamentos simples. Alguns controles intermodulares. Tabelas de decisão. Simples chamadas ou mensagens, incluindo <i>middleware</i> que suporte o processamento distribuído.	Utilização de padrões matemáticos e rotinas estatísticas. Operações básicas de matrizes e vetores.	Processamento de E/S em dispositivos selecionados, verificação de <i>status</i> e processamento de erros.	Entrada em múltiplos arquivos e saída em arquivo simples. Estrutura e edição simples. Consultas e atualizações complexas.	Utilização de um conjunto simples de componentes de interface.	1.00

Proposta para classificação de documentos de [Cla96].

	Operações de Controle	Operações Computacionais	Operações Dependentes de Dispositivos	Operações de Gerenciamento de Dados	Operações de Gerenciamento de Interfaces com o Usuário	Fatores de Ponderação
Alto	Operadores de programa estruturado altamente aninhados com muitos predicados. Controles de fila e pilha. Distribuição homogênea de processamento. Processamento simples e leve, controle em tempo real.	Análise numérica básica: multivariável, interpolação, equações diferenciais ordinárias. Preocupação com truncamento e arredondamento.	Operações de E/S em nível físico de (tradução de endereço físico de armazenamento; busca, leitura, etc.). E/S otimizada.	Gatilhos simples ativados por fluxos de dados. Reestruturação complexa de dados.	Desenvolvimento e extensão de um conjunto de componentes de interface. Simples E/S de voz e multimídia.	1.15
Muito Alto	Utilização de recursividade. Tratamento de interrupção de prioridade fixa. Sincronização de tarefas, chamadas complexas, processamento distribuído heterogêneo. Rigoroso controle de aplicação em tempo real.	Difícil análise numérica estruturada: equações matriciais, equação diferencial parcial. Paralelização simples.	Rotinas para diagnósticos para interrupção, manutenção e tratamento de exceções. Controle de vias de comunicação. Sistemas embarcados de alto desempenho.	Administração de bancos de dados distribuídos. Gatilhos complexos. Otimização de buscas.	Complexidade moderada em gráficos dinâmicos 2D e/ou 3D, multimídia.	1.30
Extra Alto	Agendamento de recursos múltiplos com troca dinâmica de prioridades. Controle em nível de micro-código. Rigoroso controle em aplicação de tempo real distribuída.	Análise numérica difícil e não-estruturada: análise de ruído altamente precisa, dados estocásticos. Paralelização complexa.	Código e operações micro-programadas dependente de tempo de dispositivo. Sistemas embarcados de desempenho crítico.	Alto acoplamento, relacionamentos dinâmicos e objetos estruturados. Linguagem natural de gerenciamento de dados.	Multimídia complexa, realidade virtual.	1.65

Continuação - Proposta para classificação de documentos de [Cla96].

C. Complexidade de Documentos

C.1 Regra de contagem para Documento de *Design* (Projeto)

Contagem para DD						
Função	Equipe				Granularidade	Impacta OBM
	Projeto	Servidor	Cliente	Testes		
Parâmetro do Serviço						
- Alteração / Inclusão / Exclusão	Muito Baixo	Muito Baixo	Muito Baixo	N/A	Contar 1 por documento.	X
Método de Negócio						
- Alteração / Inclusão / Exclusão	Muito Baixo	Muito Baixo	N/A	N/A	Para Projeto: Contar 1 para cada método de negócio. Para Servidor: Contar 1 por documento.	X
Log Único / Log Cheque						
- Alteração / Inclusão / Exclusão	Muito Baixo	N/A	N/A	N/A	Contar 1 por Log Único ou Log Cheque.	
Pacote de Comunicação						
- Inclusão	Muito Alto	Muito Alto	N/A	Extra Alto	Contar 1 por Pacote	
- Exclusão	Muito Baixo	Muito Baixo	N/A	Muito Baixo	Contar 1 por Documento	
- Alteração	Extra Baixo	Baixo	N/A	Muito Baixo	Contar 1 por Pacote	
Restart						
- Alteração	N/A	Muito Baixo	Muito Baixo	N/A	Contar por tipo de alteração no Restart.	
Interagência						
- Inclusão / Exclusão de tratamento	Extra Baixo	Muito Baixo	N/A	Alto	Contar 1 por Documento.	X
- Alteração de lógica	N/A	N/A	N/A	N/A	Contar conforme definido em Contagem Lógica.	
Autorização						
-Alteração / Inclusão / Exclusão	N/A	N/A	N/A	Nominal	Para Testes: Contar 1 por Documento. Para Projeto / Servidor: Contar conforme Contagem de Lógica.	
Acumulação (Método Summary)						
- Alteração / Inclusão / Exclusão	Extra Baixo	Muito Baixo	N/A	Muito Baixo	Contar 1 por Documento.	X
Estorno (Método CancelProcess)						
- Inclusão / Exclusão	Extra Baixo	Muito Baixo	N/A	Baixo		X
- Alteração de Lógica	N/A	N/A	N/A	N/A	Contar conforme definido em Contagem Lógica.	X

C.2 Regra de contagem para Relatórios

Contagem para Relatórios - Classe Persistente						
Função / Equipe	Projeto	Servidor	Cliente	Testes	Granularidade	Impacta OBM
Parâmetros						
- Alteração / Inclusão / Exclusão	Extra Baixo	Extra Baixo	N/A	N/A	Contar 1 por documento.	X
Alteração Lógica						
	Contar conforme Contagem de Lógica	Contar conforme Contagem de Lógica	N/A	N/A	Contar conforme Contagem de Lógica	
Tabela de Configuração do Relatório						
- Alteração	Extra Baixo	Muito Baixo	N/A	Nominal	Contar 1 por documento.	
Cabeçalho						
- Inclusão	Nominal	Alto	N/A	Nominal	Para Testes : Contar 1 por documento. Para Projeto / Servidor : Contar 1 por cabeçalho.	
- Alteração	Muito Baixo	Baixo	N/A	Baixo		
Detalhe						
- Inclusão	Alto	Muito Alto	N/A	Nominal	Para Testes : Contar 1 por documento. Para Projeto / Servidor : Contar 1 por linha detalhe.	
- Alteração	Nominal	Nominal	N/A	Baixo		
Relatórios com Leiute Personalizado						
- Criação	Nominal	Extra Alto	N/A	Nominal		
Especificação do Relatório					Aplicável a Projeto	
- Inclusão de Cabeçalho	Nominal	N/A	N/A	N/A	Contar 1 por Cabeçalho	
- Alteração de Cabeçalho	Muito Baixo	N/A	N/A	N/A		
- Inclusão de Detalhe	Alto	N/A	N/A	N/A	Contar 1 por linha de Detalhe	
- Alteração de Detalhe	Nominal	N/A	N/A	N/A		
Implementação do Relatório					Aplicável a Servidor	
- Elemento Texto	N/A	Extra Baixo	N/A	N/A	Contar 1 por Elemento criado ou alterado	
- Elemento Gráfico	N/A	Baixo	N/A	N/A		

C.3 Regra de contagem de Lógica

Contagem para Alteração de Lógica				
Função / Equipe	Projeto	Servidor	Cliente	Observação
Parâmetros / valor fixo / enumeração				
- Alteração de nome de	Extra Baixo	Extra Baixo	Extra Baixo	
Serviços / Métodos				
- Chamada	Muito Baixo	Muito Baixo	Baixo	
Manipulação de Arquivos				
- Alteração / Criação / Exclusão	Nominal	Nominal	N/A	
Manipulação de Pacotes de Comunicação				
- Montagem / Envio / Recepção / Desmontagem	Muito Baixo	Muito Baixo	N/A	
If / Cases / Loops				
- Simples	Extra Baixo	Extra Baixo	Extra Baixo	Sem existência de OR ou AND
- Aninhados Simples	Extra Baixo	Extra Baixo	Extra Baixo	De 2 a 5 aninhamentos. Sem existência de OR ou AND.
- Complexo	Muito Baixo	Muito Baixo	Muito Baixo	Com existência de OR ou AND
- Aninhados Complexo	Nominal	Nominal	Nominal	De 2 a 5 aninhamentos com a existência de OR e AND ou acima de 5 aninhamentos sem a existência de OR ou AND.
- Aninhados Muito Complexos	Alto	Alto	Alto	Acima de 5 aninhamentos com a existência de OR ou AND.
Banco de Dados				
- Acesso / Atualização em até 3 tabelas	Muito Baixo	Muito Baixo	N/A	Para Cliente: Considerar Muito Baixo para acesso a Banco de Dados local.
- Acesso / Atualização acima de 3 tabelas	Nominal	Nominal	N/A	Para Cliente: Considerar Nominal para acesso a Banco de Dados local.
SQL				
- Criação de query	Nominal	Nominal	N/A	
- Alteração de query	Extra Baixo	Muito Baixo	N/A	Para Cliente: Considerar Nominal para acesso a Banco de Dados local.

C.4 Regra de contagem de OBMaker

Contagem para Paradigm OBMaker					
Função / Equipe	Projeto	Servidor	Cliente	Testes	Granularidade
Regra Geral					
- Quando impacta em OBM	N/A	Muito Baixo	N/A	N/A	Contar 1 por documento. Para Servidor : sempre contar no máximo 8 horas.
Objection					
- Alteração	Extra Baixo	Baixo	N/A	N/A	Contar 1 por documento.
Classe Persistente					
- Criação	Extra Baixo	Nominal	N/A	N/A	Contar 1 por Tabela
- Alteração	Extra Baixo	Alto	N/A	N/A	
Enumeração					
- Alteração	Extra Baixo	Extra Baixo	N/A	N/A	Para Projeto : Contar 1 por Documento. Para Servidor : Contar 1 por classe.

C.5 Regra de contagem de MO

Contagem para MO						
Função / Equipe	Projeto	Servidor	Cliente	Testes	Granularidade	Impacta OBM
Log Único / Log Cheque						
- Inclusão de Serviço	Extra Baixo	Extra Baixo	N/A	Extra Alto	Contar 1 por Serviço.	
- Alteração do tamanho e versão do registro	Extra Baixo	Muito Baixo	N/A	Extra Alto	Contar 1 por Documento.	
- Inclusão de Campo	Extra Baixo	Muito Baixo	N/A	N/A	Para Projeto / Servidor : contar por campo. Quando envolver lógica contar conforme Lógica de Alteração.	
- Alteração / Exclusão de Campo	Extra Baixo	Extra Baixo	N/A	N/A		
Povoamento de Tabela						
- Inclusão / Exclusão de registros	Extra Baixo	Muito Baixo	N/A	Muito Baixo	Contar 1 por Tabela	
SQL / Stored Procedure						
- Alteração / Criação de Lógica	N/A	N/A	N/A	N/A	Contar conforme Lógica de Alteração.	
Genérica						
					Contar conforme Lógica de Alteração.	

C.6 Regra de contagem de DEIG

Contagem para DEIG						
Função / Equipe	Projeto	Servidor	Cliente	Testes	Granularidade	Impacta OBM
Tela						
- Inclusão	Extra Baixo	N/A	Nominal	N/A	Até 10 campos	X
	Nominal	N/A	Extra Alto	N/A	Acima de 10 campos	X
- Alteração de label	Extra Baixo	N/A	Extra Baixo	N/A	Contar 1 para cada tela	
- Alteração de atributos de campo	Extra Baixo	N/A	Muito Baixo	N/A	Contar 1 para cada tela	X
Descrição dos campos da tela						
- Alteração / Inclusão de Lógica campos e botões	Conforme definido em contagem lógica	N/A	Conforme definido em contagem lógica	N/A	Ver contagem lógica	
Navegação						
- Alteração	Extra Baixo	N/A	Nominal	N/A	Contar 1 para cada Navegação	
- Encadeamento	Extra Baixo	N/A	Extra Alto	N/A	Contar 1 para cada Navegação	
Método de Model/Serviço						X
- Alteração / Inclusão parâmetros	Extra Baixo	N/A	Extra Baixo	N/A	Contar 1 por documento que tenham parâmetros alterados	
- Exclusão serviço	Extra Baixo	N/A	Nominal	N/A	Contar 1 por serviço	
- Inclusão método da Model	Extra Baixo	N/A	Extra Baixo	N/A	Contar 1 por Método	
Autorização						
- Inclusão de Autenticação	Extra Baixo	N/A	N/A	N/A	Contar 1 por Documento	
Autenticação						X
- Inclusão de Autenticação	Extra Baixo	N/A	Nominal	N/A	Contar 1 por Autenticação	
- Exclusão de Autenticação	Extra Baixo	N/A	Extra Baixo	N/A	Contar 1 por Autenticação	
- Alteração de Autenticação	Extra Baixo	N/A	Muito Baixo	N/A	Contar 1 por conjunto de Alterações na Autenticação	
Descarga / Comprovante						X
- Alteração / Inclusão / Exclusão de Campo	Extra Baixo	N/A	Extra Baixo	N/A	De 1 a 3 Campos	
	Nominal	N/A	Nominal	N/A	De 4 a 10 Campos	
	Muito Alto	N/A	Extra Alto	N/A	Acima de 10 campos	
Periféricos						X
	Extra Baixo	N/A	Extra Baixo	N/A	Contar 1 por Documento	